

SIEMENS

SINUMERIK

SINUMERIK 840D sl NC プログラミング

プログラミングマニュアル

はじめに

1

基本的な安全に関する指示事項

2

基本編

3

作業の準備

4

テーブル

5

付録

A

適用

コントロールシステム
SINUMERIK 840D sl/840DE sl

CNC ソフトウェア バージョン 4.95

07/2021

6FC5398-2EP40-0TA2

法律上の注意

警告事項

本書には、ユーザーの安全性を確保し製品の損傷を防止するうえ守るべき注意事項が記載されています。ユーザーの安全性に関する注意事項は、安全警告サインで強調表示されています。このサインは、物的損傷に関する注意事項には表示されません。以下に表示された注意事項は、危険度によって等級分けされています。

 危険
回避しなければ、直接的な死または重傷に至る危険状態を示します。
 警告
回避しなければ、死または重傷に至るおそれのある危険な状況を示します。
 注意
回避しなければ、軽度または中度の人身傷害を引き起こすおそれのある危険な状況を示します。
通知
回避しなければ、物的損傷を引き起こすおそれのある危険な状況を示します。

複数の危険レベルに相当する場合は、通常、最も危険度の高い事項が表示されることになっています。安全警告サイン付きの人身傷害に関する注意事項があれば、物的損傷に関する警告が付加されます。

有資格者

本書が対象とする製品/システムは必ず有資格者が取り扱うものとし、各操作内容に関連するドキュメント、特に安全上の注意及び警告が遵守されなければなりません。有資格者とは、訓練内容及び経験に基づきながら当該製品/システムの取り扱いに伴う危険性を認識し、発生し得る危害を事前に回避できる者をいいます。

シーメンス製品を正しくお使いいただくために

以下の事項に注意してください。

 警告
シーメンス製品は、カタログおよび付属の技術説明書の指示に従ってお使いください。他社の製品または部品との併用は、弊社の推奨もしくは許可がある場合に限りです。製品を正しく安全にご使用いただくには、適切な運搬、保管、組み立て、据え付け、配線、始動、操作、保守を行ってください。ご使用になる場所は、許容された範囲を必ず守ってください。付属の技術説明書に記述されている指示を遵守してください。

商標

®マークのついた称号はすべて Siemens AG の商標です。本書に記載するその他の称号は商標であり、第三者が自己の目的において使用した場合、所有者の権利を侵害することになります。

免責事項

本書のハードウェアおよびソフトウェアに関する記述と、実際の製品内容との一致については検証済みです。しかしなお、本書の記述が実際の製品内容と異なる可能性もあり、完全な一致が保証されているわけではありません。記載内容については定期的に検証し、訂正が必要な場合は次の版で更新いたします。

目次

1	はじめに.....	21
1.1	SINUMERIK について	21
1.2	本書について	21
1.3	インターネット上の文書.....	25
1.3.1	SINUMERIK 840D sl マニュアルの一覧	25
1.3.2	取扱説明書概要 SINUMERIK オペレータコンポーネント	26
1.4	技術文書についてフィードバック	26
1.5	mySupport ドキュメンテーション.....	26
1.6	サービスとサポート	27
1.7	重要な製品情報.....	29
2	基本的な安全に関する指示事項	31
2.1	一般的な安全に関する指示事項	31
2.2	アプリケーション例に対する保証と責任.....	31
2.3	セキュリティ機能に関する情報	31
3	基本編	33
3.1	ジオメトリの基礎知識	33
3.1.1	ワークの位置	33
3.1.1.1	位置指定の基準系	33
3.1.1.2	直交座標系.....	33
3.1.1.3	極座標.....	36
3.1.1.4	アブソリュート指令	37
3.1.1.5	インクリメンタル指令	39
3.1.2	作業平面	40
3.1.3	原点、およびレファレンス点	42
3.1.4	座標系.....	43
3.1.4.1	機械座標系(MCS)	43
3.1.4.2	基本座標系(BCS)	46
3.1.4.3	基本ゼロオフセットシステム(BZS).....	47
3.1.4.4	設定可能ゼロオフセットシステム(SZS).....	48
3.1.4.5	ワーク座標系(WCS).....	48
3.1.4.6	さまざまな座標系間の関係	49
3.2	NC プログラミングの基礎知識	49
3.2.1	NC プログラム名称	50
3.2.2	NC プログラムの構成と内容.....	51

3.2.2.1	ブロックとブロック構成要素	51
3.2.2.2	ブロックの規則	54
3.2.2.3	値の割り当て	55
3.2.2.4	コメント	56
3.2.2.5	ブロックスキップ	57
3.3	NC プログラムの作成	58
3.3.1	基本手順	58
3.3.2	使用可能な文字	60
3.3.3	プログラムヘッダ	61
3.3.4	プログラム例	62
3.3.4.1	例 1: 最初のプログラミング手順	62
3.3.4.2	例 2: 旋削の NC プログラム	63
3.3.4.3	例 3: フライス加工の NC プログラム	65
3.4	工具交換	67
3.4.1	工具管理機能が有効時の工具交換	69
3.4.1.1	工具管理機能が有効時の機能 T=ロケーションでの工具交換	69
3.4.1.2	工具管理機能が有効時の M6 での工具交換	72
3.4.2	工具管理機能が解除時の工具交換	73
3.4.2.1	工具管理機能が解除時の T 番号による工具交換	73
3.4.2.2	工具管理機能が解除時の M6 での工具交換	74
3.4.3	T プログラミングにエラーがある場合の動作	76
3.5	工具補正	76
3.5.1	プログラム指令輪郭および工具軌跡	76
3.5.2	工具長補正	77
3.5.3	工具径補正	78
3.5.4	工具補正メモリ	78
3.5.5	工具タイプ	80
3.5.5.1	工具タイプ番号と工具グループ	80
3.5.5.2	フライス工具	80
3.5.5.3	ドリル工具	83
3.5.5.4	研削工具	84
3.5.5.5	旋削工具	86
3.5.5.6	特殊工具	88
3.5.6	工具オフセットの有効化/無効化(D、D0):	90
3.5.7	工具オフセットのマスク(SUPD)	93
3.5.8	プログラマブル工具オフセット(TOFFL、TOFF、TOFFR、TOFFLR)	94
3.6	主軸動作	101
3.6.1	主軸速度(S)、主軸回転方向(M3、M4、M5)	101
3.6.2	工具切削速度(SVC)	105
3.6.3	周速一定制御(G96/G961/G962、G97/G971/G972、G973、LIMS、SCC)	112
3.6.4	砥石周速度一定制御(GWPSO、GWPSOF)のスイッチオン/オフ:	117
3.6.5	プログラマブル主軸速度制限(G25、G26)	118
3.7	送り速度制御	119

3.7.1	送り速度(G93、G94、G95、F、FGROUP、FL、FGREF)	119
3.7.2	位置決め軸の移動(POS、POSA、POSP、FA、WAITP、WAITMC)	129
3.7.3	主軸位置制御モード(SPCON、SPCOF).....	132
3.7.4	主軸の位置決め(SPOS、SPOSA、M19、M70、WAITS).....	133
3.7.5	位置決め軸/主軸の送り速度(FA、FPR、FPRAON、FPRAOF)	139
3.7.6	プログラマブル送り速度オーバーライド(OVR、OVERRAP、OVRA)	143
3.7.7	プログラマブル加減速制御補正(ACC).....	144
3.7.8	ハンドルオーバーライドによる送り速度(FD、FDA).....	146
3.7.9	曲線軌跡区間の送り速度のオートチューニング(CFTCP、CFC、CFIN)	150
3.7.10	1 ブロックの複数送り速度値(F、ST、SR、FMA、STA、SRA)	152
3.7.11	ノンモーダル送り速度(FB)	156
3.7.12	1 刃当り送り速度(G95 FZ)	157
3.8	ジオメトリ設定.....	163
3.8.1	設定可能ゼロオフセット(G54～G57、G505～G599、G53、G500、SUPA、G153)....	163
3.8.2	設定可能ワークオフセット(G54～G57、G505～G599、G53、G500、SUPA、G153): 詳細情報	165
3.8.3	作業平面(G17/G18/G19)の選択	166
3.8.4	寸法	169
3.8.4.1	アブソリュート指令(G90、AC)	169
3.8.4.2	インクレメンタル指令(G91、IC)	171
3.8.4.3	旋削とフライス加工のアブソリュート指令、およびインクレメンタル指令(G90/G91)...	175
3.8.4.4	回転軸のアブソリュート指令(DC、ACP、ACN).....	176
3.8.4.5	メトリック/インチ寸法システム(G70/G71、G700/G710)	178
3.8.4.6	チャネル別の直径/半径指定(DIAMON、DIAM90、DIAMOF、DIAMCYCOF)	182
3.8.4.7	軸別の直径/半径指定(DIAMONA、DIAM90A、DIAMOF、DIAMCYCOF、 DIAMCHANA、DIAMCHAN、DAC、DIC、RAC、RIC)	184
3.8.5	旋削のためのワーク位置.....	189
3.9	動作命令	190
3.9.1	移動指令の概要.....	190
3.9.2	直交座標による移動指令(G0、G1、G2、G3、X...、Y...、Z...)	192
3.9.3	極座標による移動命令	194
3.9.3.1	極座標の基準点(G110、G111、G112)	194
3.9.3.2	極座標による移動指令(G0、G1、G2、G3、AP、RP).....	196
3.9.4	早送り移動.....	200
3.9.4.1	早送り(G0)の有効化	200
3.9.4.2	早送り移動の直線補間のオン/オフを切り替えます(RTLION、RTLIOF)	202
3.9.4.3	早送り移動(STOLF、CTOLG0、OTOLG0)の許容範囲を調整	204
3.9.5	直線補間(G1)	209
3.9.6	円弧補間	212
3.9.6.1	一覧	212
3.9.6.2	中心点と終点による円弧補間(G2/G3、X... Y... Z...、I... J... K...).....	213
3.9.6.3	半径と終点による円弧補間(G2/G3、X... Y... Z...、CR)	216
3.9.6.4	開口角度と終点/中心点による円弧補間(G2/G3、X...Y...Z... / I...J...K...、AR)	218
3.9.6.5	極座標による円弧補間(G2/G3、AP、RP).....	220

3.9.6.6	中間点と終点による円弧補間(CIP、X... Y... Z...、I1... J1... K1...)	222
3.9.6.7	接線方向の遷移による円弧補間(CT、X... Y... Z...)	225
3.9.7	ヘリカル補間(G2/G3、TURN)	229
3.9.8	インボリュート補間(INVCW、INVCCW)	232
3.9.9	輪郭定義	237
3.9.9.1	輪郭定義のプログラミング	237
3.9.9.2	輪郭定義: 1 つの直線	238
3.9.9.3	輪郭定義: 2 つの直線	240
3.9.9.4	輪郭定義: 3 つの直線	244
3.9.9.5	輪郭定義: 角度による終点プログラミング	247
3.9.10	ねじ切り	248
3.9.10.1	固定リードのねじ切り(G33、SF)	248
3.9.10.2	可変リードねじ切り(G34、G35)	255
3.9.10.3	G33、G34、および G35(DITS、DITE)でプログラムされた切り始めと切り上げの距離	256
3.9.10.4	ねじ切り時の高速リトラクト(LFON、LFOF、DILF、ALF、LFTXT、LFWP、LFPOS、POLF、POLFMASK、POLFMLIN)	259
3.9.10.5	凸型ねじ(G335、G336)	263
3.9.11	フローティングチャックのないタッピング	268
3.9.11.1	フローティングチャックおよび後退動作を使用しないタッピング(G331、G332)	268
3.9.11.2	例:G331/G332 のタッピング	270
3.9.11.3	例:現在のギヤ選択でプログラム指令穴あけ速度を出力	270
3.9.11.4	例:2 番目のギヤ選択データブロックの用途	271
3.9.11.5	例:速度はプログラムされておらず、ギヤボックス選択が監視されています	271
3.9.11.6	例:ギヤボックスの選択を変更できません、ギヤボックスの選択は監視されています	271
3.9.11.7	例:SPOS を使用しないプログラミング	272
3.9.12	フローティングチャックを使用したタッピング	273
3.9.12.1	補正チェックおよび後退動作(G63)によるタッピング	273
3.9.13	面取り、丸み付け(CHF、CHR、RND、RNDM、FRC、FRCM)	274
3.10	工具径補正	280
3.10.1	工具径補正(G40、G41、G42、OFFN)	280
3.10.2	輪郭へのアプローチと後退(NORM、KONT、KONTC、KONTT)	291
3.10.3	外側コーナの補正(G450、G451、DISC)	298
3.10.4	滑らかなアプローチと後退	302
3.10.4.1	アプローチと後退(G140～G143、G147、G148、G247、G248、G347、G348、G340、G341、DISR、DISCL、DISRP、FAD、PM、PR):	302
3.10.4.2	拡張された後退方法によるアプローチと後退(G460、G461、G462)	314
3.10.5	衝突検出(「ボトルネック検出」)の起動/解除(CDON、CDOF、CDOF2)	318
3.10.6	2 1/2 次元工具オフセット(CUT2D、CUT2DD、CUT2DF、CUT2DFD)	320
3.10.7	工具径補正の抑制(CUTCONON、CUTCONOF)	323
3.10.8	当該の工具の刃先位置を持つ工具	325
3.11	軌跡動作	327
3.11.1	イグザクトストップ(G60、G9、G601、G602、G603)	327
3.11.2	連続軌跡モード(G64、G641、G642、G643、G644、G645、ADIS、ADISPOS)	330

3.12	座標変換(フレーム).....	340
3.12.1	フレーム	340
3.12.2	フレーム命令	342
3.12.3	プログラマブルゼロオフセット(TRANS、ATRANS)	345
3.12.4	プログラマブルゼロオフセット(G58、G59).....	349
3.12.5	プログラマブル回転(ROT、AROT、RPL).....	352
3.12.6	立体角によるプログラマブルフレーム回転(ROTS、AROTS、CROTS).....	358
3.12.7	プログラマブルスケーリング係数(SCALE、ASCALE).....	363
3.12.8	プログラマブルミラーリング(MIRROR、AMIRROR).....	366
3.12.9	工具オリエンテーションに従ったフレーム生成(TOFRAME、TOROT、PAROT):.....	371
3.12.10	フレームの選択解除(G53、G153、SUPA、G500)	374
3.12.11	プログラミング軸別の重畳の解除(CORROF)	375
3.12.12	追加ゼロオフセットの選択解除(DRFROF).....	378
3.12.13	研削固有ゼロオフセット(GFRAME0、GFRAME1 ... GFRAME100).....	380
3.13	補助機能出力	381
3.13.1	M 機能	384
3.14	補助命令	388
3.14.1	メッセージ出力(MSG)	388
3.14.2	OPI 変数での文字列の書き込み(WRTPR).....	389
3.14.3	ワーキングエリアリミット	391
3.14.3.1	BCS のワーキングエリアリミット(G25/G26、WALIMON、WALIMOF)	391
3.14.3.2	WCS/SZS の作業領域リミット(WALCS0 ... WALCS10).....	394
3.14.4	リファレンス点復帰(G74).....	398
3.14.5	固定点へのアプローチ(G75)	399
3.14.6	突き当て点停止(FXS、FXST、FXSW)	404
3.14.7	ドウェル時間(G4)	408
3.14.8	内部先読み停止.....	411
3.15	その他の情報	411
3.15.1	軸.....	411
3.15.1.1	メイン軸/ジオメトリ軸	412
3.15.1.2	付加軸.....	413
3.15.1.3	主軸、メイン主軸	414
3.15.1.4	機械軸.....	414
3.15.1.5	チャネル軸.....	415
3.15.1.6	軌跡軸.....	415
3.15.1.7	位置決め軸.....	415
3.15.1.8	同期軸.....	416
3.15.1.9	コマンド軸.....	417
3.15.1.10	PLC 軸	417
3.15.1.11	リンク軸	418
3.15.1.12	マスタリンク軸.....	420
3.15.2	移動指令から機械移動まで	422
3.15.3	軌跡演算	422

3.15.4	アドレス	423
3.15.5	名称	425
3.15.6	定数	427
3.15.7	演算子および算術機能	429
4	作業の準備	433
4.1	フレキシブルな NC プログラミング	433
4.1.1	変数	433
4.1.1.1	システムデータ	434
4.1.1.2	予約ユーザー変数:チャンネル別算術変数(R)	437
4.1.1.3	予約ユーザー変数:グローバル算術変数(RG)	439
4.1.1.4	予約ユーザー変数リンク変数	441
4.1.1.5	ユーザー変数の定義(DEF)	443
4.1.1.6	システムデータ、ユーザーデータ、NC 命令(REDEF)の再定義	450
4.1.1.7	属性:初期値	454
4.1.1.8	属性:制限値(LLI、ULI)	457
4.1.1.9	属性:物理単位(PHU)	459
4.1.1.10	属性:アクセス権(APR、APW、APRP、APWP、APRB、APWB)	461
4.1.1.11	定義と再定義が可能な属性の概要	466
4.1.1.12	配列変数の定義と初期化(DEF、SET、REP)	468
4.1.1.13	データタイプ	474
4.1.1.14	変数の最小値、最大値、および範囲(MINVAL、MAXVAL、および BOUND)	475
4.1.1.15	変数の有効性確認(ISVAR)	476
4.1.1.16	属性値/データタイプ(GETVARPHU、GETVARAP、GETVARLIM、GETVARDIM、 GETVARDFT、GETVARTYP)の読み取り	478
4.1.1.17	実行可能なタイプ変換	485
4.1.2	間接プログラミング	487
4.1.2.1	アドレスの間接プログラミング	487
4.1.2.2	G 命令の間接プログラム	489
4.1.2.3	位置属性の間接プログラミング(GP)	490
4.1.2.4	パートプログラム行の間接プログラミング(EXECSTRING)	493
4.1.3	命令	494
4.1.3.1	算術機能	494
4.1.3.2	比較演算と論理演算	497
4.1.3.3	演算子の優先度	499
4.1.3.4	比較演算誤差の精度補正(TRUNC)	500
4.1.3.5	切り上げ(ROUNDUP)	502
4.1.4	文字列演算子	503
4.1.4.1	STRING へのタイプ変換(AXSTRING)	504
4.1.4.2	STRING からのタイプ変換(NUMBER、ISNUMBER、AXNAME)	505
4.1.4.3	文字列の結合(<<)	506
4.1.4.4	小文字/大文字への変換(TOLOWER、TOUPPER)	508
4.1.4.5	文字列の長さの特定(STRLEN)	508
4.1.4.6	文字列中の文字/文字列の検索(INDEX、RINDEX、MINDEX、MATCH)	509
4.1.4.7	抽出文字列の選択(SUBSTR)	510

4.1.4.8	個々の文字の読み取りと書き込み	511
4.1.4.9	文字列のフォーマット (SPRINT)	513
4.1.5	プログラムのジャンプと分岐	523
4.1.5.1	プログラムの先頭への復帰ジャンプ (GOTOS)	523
4.1.5.2	ジャンプマークへのプログラムのジャンプ (GOTOB、GOTOF、GOTO、GOTOC)	525
4.1.5.3	プログラム分岐 (CASE ... OF ... DEFAULT ...)	529
4.1.6	プログラム区間の繰り返し (REPEAT、REPEATB、ENDLABEL、P)	531
4.1.7	チェック命令	538
4.1.7.1	条件付き命令と分岐 (IF、ELSE、ENDIF)	540
4.1.7.2	連続プログラムループ (LOOP、ENDLOOP)	542
4.1.7.3	カウントループ (FOR ... TO ...、ENDFOR)	543
4.1.7.4	ループの先頭に条件があるプログラムループ (WHILE、ENDWHILE)	544
4.1.7.5	ループの終了に条件があるプログラムループ (REPEAT、UNTIL)	545
4.1.7.6	入れ子のチェック命令を含むプログラム例	546
4.1.8	チャンネル間プログラム協調 (INIT、START、WAITM、WAITMC、WAITE、SETM、 CLEARM)	546
4.1.9	マクロ機能 (DEFINE ... AS)	553
4.2	サブプログラム機能	556
4.2.1	概要	556
4.2.1.1	サブプログラム	556
4.2.1.2	サブプログラム名称	558
4.2.1.3	サブプログラムの入れ子	559
4.2.1.4	検索パス	561
4.2.1.5	仮パラメータと実パラメータ	561
4.2.1.6	パラメータ転送	562
4.2.2	サブプログラムの定義	563
4.2.2.1	パラメータ転送を含まないサブプログラム	563
4.2.2.2	値渡しパラメータ転送を含むサブプログラム (PROC)	564
4.2.2.3	参照渡しパラメータ転送を含むサブプログラム (PROC、VAR)	566
4.2.2.4	モータル G 機能の保存 (SAVE)	569
4.2.2.5	シングルブロック実行のマスク (SBLOF、SBLON)	570
4.2.2.6	実行中のブロック表示のマスク (DISPLOF、DISPLON、ACTBLOCNO)	575
4.2.2.7	解析によるサブプログラムの識別 (PREPRO)	579
4.2.2.8	サブプログラム戻り M17	580
4.2.2.9	RET サブプログラム戻り	581
4.2.2.10	設定可能なサブプログラム復帰ジャンプ (RET ...)	582
4.2.2.11	パラメータ設定可能なサブプログラム復帰ジャンプ (RETB ...)	589
4.2.3	サブプログラムの呼び出し	595
4.2.3.1	パラメータ転送を含まないサブプログラム呼び出し	595
4.2.3.2	パラメータ転送を含むサブプログラム呼び出し (EXTERN)	597
4.2.3.3	プログラム繰り返し回数 (P)	600
4.2.3.4	モータルサブプログラム呼び出し (MCALL)	601
4.2.3.5	間接サブプログラム呼び出し (CALL)	604

4.2.3.6	呼び出しプログラム要素を指定した間接サブプログラム呼び出し(CALL BLOCK ... TO ...)	605
4.2.3.7	ISO 言語で作成したプログラムの間接呼び出し(ISOCALL)	606
4.2.3.8	パス指定とパラメータによるサブプログラム呼び出し(PCALL)	607
4.2.3.9	サブプログラム呼び出しの拡張検索パス(CALLPATH)	608
4.2.3.10	外部サブプログラムの実行(EXTCALL)	610
4.3	割り込みルーチン(ASUB)	615
4.3.1	割り込みルーチンの機能	615
4.3.2	割り込みルーチンの作成	616
4.3.3	割り込みルーチンの割り当てと起動(SETINT、PRIO、BLSYNC)	617
4.3.4	割り込みルーチンの割り当ての解除/再起動(DISABLE、ENABLE)	619
4.3.5	割り込みルーチンの割り当ての解除(CLRINT)	620
4.3.6	輪郭からの高速リトラクト(SETINT LIFTFAST、ALF)	621
4.3.7	輪郭からの高速リトラクトの移動方向	623
4.3.8	割り込みルーチンの動作順序	626
4.4	ファイルとプログラムの管理	627
4.4.1	プログラムメモリ	627
4.4.1.1	NCK のプログラムメモリ	627
4.4.1.2	外部プログラムメモリ	630
4.4.1.3	プログラムメモリファイルのアドレス指定	632
4.4.1.4	サブプログラム呼び出しの検索パス	637
4.4.1.5	パスとファイル名の問い合わせ	639
4.4.2	作業メモリ(CHANDATA、COMPLETE、INITIAL)	641
4.5	ファイル処理	644
4.5.1	ファイルの書き込み(WRITE)	644
4.5.2	ファイルの削除(DELETE)	648
4.5.3	ファイルの行の読み取り(READ)	650
4.5.4	ファイルの存在の確認(ISFILE)	652
4.5.5	ファイル情報の読み出し(FILEDATE、FILETIME、FILESIZE、FILESTAT、FILEINFO)	653
4.6	保護領域	656
4.6.1	プロテクションゾーンの定義(CPROTDEF、NPROTDEF)	656
4.6.2	プロテクションゾーンの起動/解除(CPROT、NPROT)	660
4.6.3	プロテクションゾーン違反、作業領域リミット、およびソフトウェアリミットスイッチのチェック(CALCPOSI)	664
4.7	特殊動作命令	677
4.7.1	符号化位置へのアプローチ(CAC、CIC、CDC、CACP、CACN)	677
4.7.2	スプライン補間(ASPLINE、BSPLINE、CSPLINE、BAUTO、BNAT、BTAN、EAUTO、ENAT、ETAN、PW、SD、PL)	678
4.7.3	スプライングループ(SPLINEPATH)	690
4.7.4	NC ブロック圧縮(COMPON、COMPCURV、COMPCAD、COMPSURF、COMPOF)の有効化/無効化	691
4.7.5	多項式補間(POLY、POLYPATH、PO、PL)	692
4.7.6	設定可能な軌跡基準(SPATH、UPATH)	699

4.7.7	チャンネル別計測(MEAS、MEAW)	701
4.7.8	軸別計測(MEASA、MEAWA、MEAC) (オプション)	705
4.7.9	OEM ユーザー用の応用機能(OMA1 ... OMA5、OEMIPO1、OEMIPO2、G810 ... G829)...	715
4.7.10	コーナ減速による送り速度低下(FENDNORM、G62、G621)	716
4.7.11	プログラム指令可能な動作終了条件(FINEA、COARSEA、IPOENDA、IPOBRKA、 ADISPOSA)	717
4.8	座標変換(フレーム).....	720
4.8.1	フレーム変数による座標変換	720
4.8.1.1	予約フレーム変数(\$P_CHBFRAME、\$P_IFRAME、\$P_PFRAME、\$P_ACTFRAME)	722
4.8.2	フレームへの値の割り付け	726
4.8.2.1	値の直接割り当て(軸値、角度、スケール)	726
4.8.2.2	フレーム成分の読み取りと変更(TR、FI、RT、SC、MI)	728
4.8.2.3	フレームの接続.....	730
4.8.2.4	フレーム変数の定義(DEF FRAME)	731
4.8.3	荒削りオフセットと仕上げオフセット(CTTRANS、CFINE)	732
4.8.4	外部ゼロオフセット(\$AA_ETRANS)	734
4.8.5	原点確立状態をリセットして現在値を設定(PRESETON)	736
4.8.6	原点確立状態をリセットせずに現在値を設定(PRESETONS).....	737
4.8.7	空間の 3 つの計測点からのフレーム計算(MEAFRAME)	739
4.8.8	グローバルフレーム	743
4.8.8.1	チャンネル別フレーム(\$P_CHBFR、\$P_UBFR)	744
4.8.8.2	チャンネルの有効フレーム	745
4.9	座標変換	750
4.9.1	座標変換タイプの一般的なプログラミング	750
4.9.1.1	座標変換の旋回移動	753
4.9.1.2	方向座標変換 TRAORI の一覧	758
4.9.2	3 軸、4 軸、5 軸座標変換(TRAORI).....	761
4.9.2.1	ユニバーサル工具ヘッドの一般的な関係.....	761
4.9.2.2	3 軸、4 軸、5 軸座標変換(TRAORI).....	764
4.9.2.3	向きのプログラミングのタイプと初期設定(ORIRESET)	765
4.9.2.4	工具オリエンテーションのプログラミング(A...、B...、C...、LEAD、TILT):.....	767
4.9.2.5	正面削り(A4、B4、C4、A5、B5、C5)	773
4.9.2.6	旋回軸の基準(ORIWKS、ORIMKS):.....	774
4.9.2.7	旋回軸のプログラミング(ORIAXES、ORIVECT、ORIEULER、ORIRPY、ORIRPY2、 ORIVIRT1、ORIVIRT2).....	776
4.9.2.8	円錐面に沿った向きのプログラミング(ORIPLANE、ORICONCW、ORICONCCW、 ORICONTO、ORICONIO).....	779
4.9.2.9	2 つの接点の向きの指定(ORICURVE、PO[XH]=、PO[YH]=、PO[ZH]=)	782
4.9.3	旋回多項式(PO[角度]、PO[座標])	784
4.9.4	工具オリエンテーションの回転(ORIROTA、ORIROTR、ORIROTT、ORIROTC、THETA)...	786
4.9.5	軌跡に対する向き	789
4.9.5.1	軌跡に対する向きのタイプ	789
4.9.5.2	軌跡に対する工具オリエンテーションの回転(ORIPATH、ORIPATHS、回転角度).....	791
4.9.5.3	軌跡に対する工具回転の補間(ORIROTC、THETA)	792

4.9.5.4	旋回処理のスムージング(ORIPATHS A8=, B8=, C8=)	795
4.9.6	旋回の圧縮(COMPON、COMPCURV、COMPCAD、COMPOF、COMPSURF)	796
4.9.7	向きの特性の有効化と無効化(ORISON、ORISOF)	799
4.9.8	キネマティックトランスフォーメーション	800
4.9.8.1	面端座標変換(TRANSMIT)の起動	800
4.9.8.2	円筒補間(TRACYL)の起動	801
4.9.8.3	角度がプログラム指令可能な傾斜角座標変換(TRAANG)の有効化(TRAANG)	804
4.9.8.4	研削盤での傾斜プランジ切削(G5, G7)	805
4.9.9	座標変換重畳(TRACON)の起動	808
4.9.10	直交 PTP 移動	810
4.9.10.1	直交 PTP 移動(PTP、PTPGO、PTPWOC、CP)の有効化/無効化	810
4.9.10.2	ジョイント(STAT)の位置を指定します	811
4.9.10.3	軸角度の符号を指定します(TU)	816
4.9.10.4	例 1:ROBX 座標変換による 6 軸ロボットの PTP 移動	819
4.9.10.5	例 2:総合 5 軸座標変換の場合の PTP 移動	820
4.9.10.6	例 3:PTPGO および TRANSMIT	821
4.9.11	座標変換の選択時の制約事項	823
4.9.12	座標変換の解除(TRAFOOF)	824
4.10	キネマティック結合	824
4.10.1	要素の削除(DELOBJ)	824
4.10.2	名称を使用したインデックスの特定(NAMETOINT)	829
4.11	キネマティック結合付きの衝突防止	831
4.11.1	干渉ペアをチェックします(COLPAIR)	831
4.11.2	干渉回避のマシンモデルの再計算要求(PROTA)	831
4.11.3	プロテクションゾーン状態の設定(PROTS)	832
4.11.4	:2 つのプロテクションゾーンの間隔の決定(PROTD)	833
4.12	キネマティック結合による座標変換	836
4.12.1	座標変換の有効化(TRAFOON)	836
4.12.2	機械計測後の方向座標変換の変更(CORRTRAFO)	837
4.13	工具オフセット	846
4.13.1	オフセットメモリ	846
4.13.2	追加オフセット	849
4.13.2.1	追加オフセットの選択(DL)	849
4.13.2.2	摩耗値とセットアップ値の指定(\$TC_SCPxy[t,d]、\$TC_ECPxy[t,d])	851
4.13.2.3	追加オフセットの削除(DELDL)	851
4.13.3	工具補正の特殊処理	852
4.13.3.1	工具長のミラーリング	854
4.13.3.2	摩耗の符号評価	854
4.13.3.3	動作中の加工運転の座標系(TOWSTD、TOWMCS、TOWWCS、TOWBCS、TOWTCS、TOWKCS)	855
4.13.3.4	工具長と平面変更	858
4.13.4	オンライン工具補正	860
4.13.4.1	多項式機能(FCTDEF)の定義	860

4.13.4.2	オンライン工具補正の連続書き込み(PUTFTOCF)	862
4.13.4.3	オンライン工具補正の書き込み、個別(PUTFTOC)	863
4.13.4.4	オンライン工具オフセットの起動/解除(FTOCON/FTOCOF)	864
4.13.5	3次元工具径補正	865
4.13.5.1	3次元外周加工用の3次元工具径補正(CUT3DC、CUT3DCD、ISD)	865
4.13.5.2	3次元正面削りのための3次元工具径補正の選択(CUT3DF、CUT3DFS、CUT3DFF、CUT3DFD)	869
4.13.5.3	3次元外周加工、限界面の考慮(CUT3DCC、CUT3DCCD)	875
4.13.6	工具オリエンテーション(ORIC、ORID、OSOF、OSC、OSS、OSSE、ORIS、OSD、OST)	881
4.13.7	D番号の任意割り当て、刃先番号	887
4.13.7.1	D番号、刃先番号の任意割り当て(CEアドレス)	887
4.13.7.2	D番号の任意割り当て:D番号のチェック(CHKDNO)	887
4.13.7.3	D番号の任意割り当て:D番号の名称変更(GETDNO、SETDNO)	888
4.13.7.4	D番号の任意割り当て:指定したD番号に対してT番号の特定(GETACTTD)	889
4.13.7.5	D番号の任意割り当て:D番号の無効化(DZERO)	890
4.13.8	工具ホルダキネマティクス	890
4.13.9	旋回工具ホルダの工具長補正(TCARR、TCOABS、TCOFR、TCOFRX、TCOFRY、TCOFRZ)	897
4.13.10	機械計測に従って、旋回可能な工具ホルダを修正(CORRTC)	902
4.13.11	オンライン工具長補正(TOFFON、TOFFOF)	906
4.13.12	旋回工具のオフセットデータ変更	909
4.13.12.1	オリエンテーションの計算(ORISOLH)	909
4.13.12.2	旋回工具のオフセットデータの変更の起動(CUTMOD、CUTMODK)	923
4.13.13	工具環境の使用	933
4.13.13.1	工具環境の保存(TOOLENV)	933
4.13.13.2	工具環境の削除(DELTTOOLENV)	936
4.13.13.3	T番号、D番号、およびDL番号の読み出し(GETTENV)	937
4.13.13.4	保存した工具環境に関する情報の読み取り(\$P_TOOLENVN、\$P_TOOLENV)	939
4.13.13.5	工具長、工具長成分、またはその両方の読み取り(GETTCOR)	939
4.13.13.6	工具成分の変更(SETTCOR)	947
4.13.14	座標軸への工具長L1、L2、L3の割り当ての読み取り(LENTOAX)	962
4.14	軌跡の移動動作	966
4.14.1	送り速度特性(FNORM、FLIN、FCUB、FPO)	966
4.14.2	加減速動作	971
4.14.2.1	加減速モード(BRISK、BRISKA、SOFT、SOFTA、DRIVE、DRIVEA)	971
4.14.2.2	スレーブ軸に対する加減速の動作(VELOLIMA、ACCLIMA、JERKLIMA)	974
4.14.2.3	テクノロジ用のダイナミック応答値の起動(DYNNORM、DYNPOS、DYNROUGH、DYNSEMIFIN、DYNFINISH、DYNPREC)	976
4.14.3	フィードフォワード制御による移動(FFWON、FFWOF)	977
4.14.4	輪郭/旋回の許容範囲のプログラミング(CTOL、OTOL、ATOL)	978
4.14.5	プログラム指令可能な輪郭精度のオン/オフの切り替え(CPRECON、CPRECOF)	983
4.14.6	自動フィルタ切り替えの起動/解除(AFISON、AFISOF)	985
4.14.7	解析メモリによるプログラム手順(STOPFIFO、STARTFIFO、FIFOCTRL、STOPRE)	988

4.14.8	停止遅延範囲(DELAYFSTON、DELAYFSTOF)の定義	991
4.14.9	SERUPRO のプログラム位置のマスク(IPTRLOCK、IPTRUNLOCK)	993
4.14.10	輪郭への再位置決め(REPOSA、REPOSL、REPOSQ、REPOSQA、REPOSH、 REPOSHA、DISR、DISPR、RMIBL、RMBBL、RMEBL、RMNBL)	995
4.14.11	モーションコントロールへの影響	1004
4.14.11.1	最大軸速度または最大主軸速度の調整(VELOLIM)	1004
4.14.11.2	最大軸加々速度の調整(JERKLIM)	1006
4.14.11.3	最大軌跡速度の設定(FLIM)	1007
4.14.11.4	最大軌跡加速度の設定(PACCLIM)	1009
4.14.12	有効な連結によるブロック変更動作(CPBC)	1010
4.15	軸機能	1011
4.15.1	軸入れ替え、主軸入れ替え(RELEASE、GET、GETD)	1011
4.15.2	別のチャンネルへの軸の移行(AXTOCHAN)	1016
4.15.3	軸機能(AXNAME、AX、SPI、AXTOSPI、ISAXIS、AXSTRING、MODAXVAL)	1018
4.15.4	置換可能なジオメトリ軸(GEOAX)	1020
4.15.5	軸コンテナ(AXCTSWE、AXCTSWED、AXCTSWEC)	1025
4.15.6	有効な軸位置を待機します(WAITENC)	1027
4.15.7	プログラム指令可能なパラメータセット切り替え(SCPARA)	1029
4.15.8	調整の有効化/無効化(CADAPTON、CADAPTOF)	1030
4.15.9	FIR 加々速度フィルタのダイナミックモードへの設定(CALCFIR)	1033
4.15.10	パートプログラムにおけるドライブパラメータの読み書き(DRVPRD、DRVPCR)	1035
4.16	軸連結	1039
4.16.1	連結移動(TRAILON、TRAILOF)	1039
4.16.2	カーブテーブル(CTAB)	1043
4.16.2.1	カーブテーブルの定義(CTABDEF、CATBEND)	1043
4.16.2.2	カーブテーブルの有無の確認(CTABEXISTS)	1049
4.16.2.3	カーブテーブルの削除(CTABDEL)	1050
4.16.2.4	カーブテーブルの削除、および上書きを防止するロック(CTABLOCK、CTABUNLOCK)	1051
4.16.2.5	カーブテーブル: カーブテーブル機能の特定(CTABID、CTABISLOCK、CTABMEMTYP、 CTABPERIOD)	1053
4.16.2.6	カーブテーブル値の読み取り(CTABTSV、CTABTEV、CTABTSP、CTABTEP、 CTABSSV、CTABSEV、CTAB、CTABINV、CTABTMIN、CTABTMAX)	1055
4.16.2.7	カーブテーブル: メモリ使用の確認(CTABNO、CTABNOMEM、CTABFNO、 CTABSEGID、CTABSEG、CTABFSEG、CTABMSEG、CTABPOLID、CTABPOL、 CTABFPOL、CTABMPOL)	1060
4.16.3	軸間連動機能(LEADON、LEADOF)	1062
4.16.4	電子ギヤ(EG)	1068
4.16.4.1	電子ギヤの定義(EGDEF)	1069
4.16.4.2	電子ギヤの起動(EGON、EGONSYN、EGONSYNE)	1070
4.16.4.3	電子ギヤの起動(EGOFS、EGOFC)	1074
4.16.4.4	電子ギヤの定義の解除(EGDEL)	1075
4.16.4.5	毎回転送り速度(G95) / 電子ギヤ(FPR)	1075
4.16.5	主軸同期	1075

4.16.5.1	主軸同期: プログラミング(COUPDEF、COUPDEL、COUPON、COUPONC、COUPOF、COUPOFS、COUPRES、WAITC).....	1076
4.16.6	汎用連結(CP...)	1087
4.16.7	法線方向制御	1097
4.16.7.1	連結の定義(TANG).....	1097
4.16.7.2	挿入ブロックの生成の有効化(TLIFT)	1100
4.16.7.3	連結を起動(TANGON).....	1101
4.16.7.4	連結の無効化(TANGOF)	1103
4.16.7.5	連結の解除(TANGDEL)	1103
4.16.8	マスタ/スレーブ連結(MASLDEF、MASLDEL、MASLON、MASLOF、MASLOFS).....	1105
4.17	シンクロナイズドアクション	1108
4.17.1	シンクロナイズドアクションの定義	1108
4.18	揺動	1109
4.18.1	非同期揺動(OS、OSP1、OSP2、OST1、OST2、OSCTRL、OSNSC、OSE、OSB)	1109
4.18.2	シンクロナイズドアクションによる揺動制御(OSCILL).....	1115
4.19	パンチングとニブリング	1122
4.19.1	起動と解除.....	1122
4.19.1.1	パンチングとニブリングの起動または解除(SPOF、SON、PON、SONS、PONS、PDELAYON、PDELAYOF、PUNCHACC):.....	1122
4.19.2	自動軌跡分割	1127
4.19.2.1	軌跡軸の軌跡分割	1131
4.19.2.2	単独軸の軌跡分割	1132
4.20	研削加工	1133
4.20.1	研削用工具監視の有効化/無効化(TMON、TMOF).....	1133
4.21	停止延長と退避(ESR).....	1135
4.21.1	NC 制御 ESR	1136
4.21.1.1	NC 制御による退避(POLF、POLFA、POLFMASK、POLFMLIN).....	1136
4.21.1.2	NC 制御による停止	1141
4.21.2	ドライブ統合 ESR.....	1142
4.21.2.1	ドライブ内蔵の停止の設定(ESRS).....	1142
4.21.2.2	ドライブ内蔵の後退の設定(ESRS).....	1143
4.22	プログラム実行時間/ワークカウンタ	1144
4.22.1	プログラム実行時間	1144
4.22.2	ワークカウンタ	1149
4.23	プログラムシミュレーション	1151
4.23.1	機能	1151
4.23.2	コンパイルサイクルありのプログラムシミュレーション	1151
4.24	その他の機能	1152
4.24.1	マシンデータの起動(NEWCONF).....	1152
4.24.2	現在の NC 言語の適用範囲の確認(Stringis)	1153

4.24.3	対話形式によるパートプログラムからのウィンドウの呼び出し(MMC)	1158
4.24.4	Process DataShare - 外部機器/ファイルへの出力(EXTOPEN、WRITE、EXTCLOSE).....	1163
4.24.5	アラーム(SETAL).....	1168
4.24.6	Define blank (WORKPIECE)	1170
4.24.7	言語切り替えモード(G290, G291).....	1176
4.25	ユーザー荒削りプログラム	1178
4.25.1	荒削りの機能のサポート	1178
4.25.2	輪郭テーブルの生成(CONTPRON)	1178
4.25.3	コード化された輪郭テーブルの生成(CONTDCON)	1185
4.25.4	2 個の輪郭要素の交点の特定(INTERSEC).....	1189
4.25.5	テーブルの輪郭要素をブロック毎に実行(EXECTAB).....	1191
4.25.6	円弧データの計算(CALCDAT)	1191
4.25.7	輪郭解析の解除(EXECUTE)	1194
4.26	サイクルを外部的にプログラミング	1195
4.26.1	テクノロジサイクル.....	1195
4.26.1.1	はじめに	1195
4.26.1.2	テクノロジ別の概要.....	1197
4.26.1.3	HOLES1 - 行位置決めパターン	1199
4.26.1.4	HOLES2 - 円または円弧位置決めパターン.....	1200
4.26.1.5	POCKET3 - 長方形ポケット	1202
4.26.1.6	POCKET4 - 円形ポケット	1206
4.26.1.7	SLOT1- 直線溝	1210
4.26.1.8	SLOT2 - 円周溝	1214
4.26.1.9	LONGHOLE - 長穴	1217
4.26.1.10	CYCLE60 - 彫刻	1220
4.26.1.11	CYCLE61 - 正面削り	1223
4.26.1.12	CYCLE62 - 輪郭の呼び出し.....	1226
4.26.1.13	CYCLE63 - 輪郭ポケットのフライス削り/輪郭ポケットの削り残し/輪郭スピゴットの フライス削り/輪郭スピゴットの削り残し	1227
4.26.1.14	CYCLE64 - 輪郭ポケットの前加工	1231
4.26.1.15	CYCLE70 - ねじフライス削り	1234
4.26.1.16	CYCLE72 - 輪郭フライス削り	1236
4.26.1.17	CYCLE76 - 長方形スピゴット.....	1241
4.26.1.18	CYCLE77 - 円形スピゴット.....	1244
4.26.1.19	CYCLE78 - ドリルねじフライス削り	1247
4.26.1.20	CYCLE79 - 多角形	1251
4.26.1.21	CYCLE81 - 穴あけ、センタリング	1254
4.26.1.22	CYCLE82 - 穴あけ、座ぐり	1255
4.26.1.23	CYCLE83 - 深穴ドリル 1.....	1258
4.26.1.24	CYCLE84 - フローティングチャックなしのタッピング	1262
4.26.1.25	CYCLE85 - リーマ加工	1267
4.26.1.26	CYCLE86 - ボーリング	1268
4.26.1.27	CYCLE92 - 突切り	1270
4.26.1.28	CYCLE95 - 輪郭切削	1272

4.26.1.29	CYCLE98 - 連続ねじ	1275
4.26.1.30	CYCLE99 - ねじの旋削	1281
4.26.1.31	CYCLE435 - ドレッサー座標系の設定	1286
4.26.1.32	CYCLE495 - 成形研削	1287
4.26.1.33	CYCLE782 - 負荷に従った位置合わせ	1289
4.26.1.34	CYCLE800 - 旋回平面/旋回工具/工具の位置合わせ	1292
4.26.1.35	CYCLE801 - 格子またはフレーム位置決めパターン	1298
4.26.1.36	CYCLE802 - 任意の位置	1300
4.26.1.37	CYCLE806 - 補間旋削	1304
4.26.1.38	CYCLE830 - 深穴ドリル 2	1305
4.26.1.39	CYCLE832 - 高速設定	1313
4.26.1.40	CYCLE840 - フローティングチャックによるタッピング	1317
4.26.1.41	CYCLE899 - オープン溝	1321
4.26.1.42	CYCLE930 - 溝削り	1324
4.26.1.43	CYCLE940 - アンダーカット形状 E と形状 F/ねじのアンダーカット	1327
4.26.1.44	CYCLE951 - 荒削り	1331
4.26.1.45	CYCLE952 - 旋削/旋削削り残し/プランジ切削/プランジ切削削り残し/プランジ旋削/ プランジ旋削削り残し	1335
4.26.1.46	CYCLE4071 - 反転点での切り込みによる長手方向の研削	1342
4.26.1.47	CYCLE4072 - 反転点での切り込みによる長手方向の研削とキャンセル信号	1344
4.26.1.48	CYCLE4073 - 連続切り込みによる長手方向の研削	1348
4.26.1.49	CYCLE4074 - 連続切り込みによる長手方向の研削とキャンセル信号	1350
4.26.1.50	CYCLE4075 - 反転点での切り込みによる平面研削	1354
4.26.1.51	CYCLE4077 - 反転点での切り込みによる平面研削とキャンセル信号	1357
4.26.1.52	CYCLE4078 - 連続切り込みによる平面研削	1361
4.26.1.53	CYCLE4079 - 間欠切り込みによる平面研削	1363
4.26.1.54	GROUP_BEGIN - プログラムブロック開始	1366
4.26.1.55	GROUP_END - プログラムブロック終了	1367
4.26.1.56	GROUP_ADDEND - 試し切り追加終了	1367
4.26.1.57	必要条件	1368
4.26.2	計測サイクル	1370
4.26.2.1	計測サイクルパラメーター一覧	1370
5	テーブル	1433
5.1	命令	1433
5.2	アドレス	1485
5.2.1	アドレス文字	1485
5.2.2	固定アドレス	1486
5.2.3	設定可能アドレス	1492
5.3	G 命令	1500
5.3.1	G 命令	1500
5.3.2	G グループ 1: モーダルで有効な動作命令	1501
5.3.3	G グループ 2: ノンモーダルで有効な動作、ドウェル時間	1501

5.3.4	G グループ 3:プログラマブルフレーム、作業領域リミットおよび極座標プログラミング	1502
5.3.5	G グループ 4:FIFO	1503
5.3.6	G グループ 6:平面の選択	1504
5.3.7	G グループ 7:工具径補正	1504
5.3.8	G グループ 8:設定可能なゼロオフセット	1504
5.3.9	G グループ 9:フレームと工具オフセットのマスク	1505
5.3.10	G グループ 10:イグザクトストップ-連続軌跡モード	1506
5.3.11	G グループ 11:イグザクトストップ、ノンモーダル	1506
5.3.12	G グループ 12:イグザクトストップでのブロック切り替え条件(G60/G9)	1506
5.3.13	G グループ 13:インチ/メトリック単位のワーク計測	1507
5.3.14	G グループ 14:アブソリュート/インクレメンタルのワーク計測	1507
5.3.15	G グループ 15:送り速度タイプ	1507
5.3.16	G グループ 16:内側と外側の曲率での送り速度オーバーライド	1508
5.3.17	G グループ 17:アプローチと後退の動作、工具補正	1508
5.3.18	G グループ 18:コーナの動作、工具補正	1509
5.3.19	G グループ 19:スプライン開始時の曲線遷移	1509
5.3.20	G グループ 20:スプライン終了時の曲線遷移	1509
5.3.21	G グループ 21:加減速方法	1510
5.3.22	G グループ 22:工具補正タイプ	1510
5.3.23	G グループ 23:輪郭の内側の干渉監視	1511
5.3.24	G グループ 24:フィードフォワード	1511
5.3.25	G グループ 25:工具オリエンテーションの基準	1511
5.3.26	G グループ 26:REPOS のための再位置決め(モーダル)	1512
5.3.27	G グループ 27:外側コーナで向きを変更するための工具補正	1512
5.3.28	G グループ 28:作業領域リミット	1512
5.3.29	G グループ 29:半径指定/直径指定	1513
5.3.30	G グループ 30:NC ブロック圧縮	1513
5.3.31	G グループ 31:OEM G 命令	1514
5.3.32	G グループ 32:OEM G 命令	1514
5.3.33	G グループ 33:設定可能な精密工具補正	1515
5.3.34	G グループ 34:工具オリエンテーションのスムージング	1515
5.3.35	G グループ 35:パンチングとニブリング	1516
5.3.36	G グループ 36:遅延のあるパンチング	1516
5.3.37	G グループ 37:送り速度プロファイル	1516
5.3.38	G グループ 38:パンチング/ニブリングへの高速入力/出力の割り当て	1517
5.3.39	G グループ 39:プログラム指令可能な輪郭精度	1517
5.3.40	G グループ 40:工具径補正の抑制	1517
5.3.41	G グループ 41:中断可能なねじ切り	1517
5.3.42	G グループ 42:工具ホルダ	1518
5.3.43	G グループ 43:SAR アプローチ方向	1518
5.3.44	G グループ 44:SAR 軌跡分割	1518
5.3.45	G グループ 45:FGROUP 軸の軌跡基準	1519
5.3.46	G グループ 46:高速リトラクトのレベルの選択	1519
5.3.47	G グループ 47:外部 NC コードに対応したモード切り替え	1519

5.3.48	G グループ 48:工具径補正によるアプローチと後退の動作	1520
5.3.49	G グループ 49:ポイントツーポイント移動	1520
5.3.50	G グループ 50:旋回のプログラミング	1520
5.3.51	G グループ 51:旋回プログラミングの補間タイプ	1521
5.3.52	G グループ 52:ワークに対するフレーム回転	1521
5.3.53	G グループ 53:工具に対するフレーム回転	1521
5.3.54	G グループ 54:多項式プログラミングのベクトル回転	1522
5.3.55	G グループ 55:直線補間有り/無しの早送り	1522
5.3.56	G グループ 56:工具の摩耗を考慮	1523
5.3.57	G グループ 57:コーナ減速	1523
5.3.58	G グループ 59:軌跡補間のダイナミック応答モード	1523
5.3.59	G グループ 60:作業領域リミット	1524
5.3.60	G グループ 61:工具オリエンテーションのスミージング	1524
5.3.61	G グループ 62:REPOS のための再位置決め(ノンモーダル)	1525
5.3.62	G グループ 64:研削フレーム	1525
5.4	予約処理	1526
5.5	シンクロナイズドアクションでの予約処理	1560
5.6	予約機能	1562
5.7	HMI で現在設定されている言語	1582
A	付録	1585
A.1	略語の一覧	1585
	索引	1593

はじめに

1.1 SINUMERIK について

単純で、標準化された CNC 機械から、プレミアムなモジュール式機械構成まで。SINUMERIK CNC は、すべての機械コンセプトに適したソリューションを提供します。この部品でも、大量生産でも、単純なワークでも複雑なワークでも。SINUMERIK はきわめてダイナミックな自動化ソリューションで、すべての生産分野に統合できます。プロトタイプ作成から金型設計、金型製作、さらに大規模な量産まで、あらゆるニーズに応えます。

詳細情報は、当社の Web サイト SINUMERIK (<https://www.siemens.com/sinumerik>)をご覧ください。

1.2 本書について

この取扱説明書は、SINUMERIK プログラミングマニュアルの取扱説明書カテゴリの一部です。

『NC プログラミング』プログラミングマニュアル

『NC プログラミング』プログラミングマニュアルには、SINUMERIK コントローラの NC 機能のプログラミングに関連する、すべての情報が含まれています。

対象はプログラマーと設定エンジニアです。

上記の対象読者は、プログラミング説明書を使用して、プログラムとソフトウェアユーザーインターフェースの開発、プログラミング、テスト、デバッグをおこなうことができます。

概要

タイトルページの概要には、『NC プログラミング』プログラミングマニュアルが次の2つの主な章に分かれていることが示されています。

1. 基本編

主な章「基本編」は、穴あけ、フライス、旋盤加工の経験をお持ちの熟練オペレータを対象としています。本書は、簡単なプログラミング例を使用して、命令と命令文について解説します。これらの定義は DIN 66025 にも準拠します。

2. 作業の準備

主な章「作業の準備」は、詳細で包括的なプログラミング知識をお持ちの技術者を対象としています。SINUMERIK 制御装置は専用のプログラミング言語を使用するため、自由曲面やチャンネル協調などの複雑なワーク加工のプログラム指令が可能です。また、複雑な運転でも技術者が簡単にプログラム指令できるようにします。

適用

タイトルページには、この版のプログラミングマニュアルが有効な SINUMERIK 制御およびソフトウェアバージョンなど、マニュアルの有効性に関するすべての情報も含まれています。

章の構成

NC 言語エレメント(G コマンド、プロシージャ、機能など)については、統一された方法で構造的に説明し、その内容は、事前に定義した主な質問を考慮して作成しています。

次の図では、これについて、ある章を例として使用して説明します。

基本編

2.9 パスコマンド

2.9.早送りモーシオン

2.9.4.1 早送りの有効化(G0)

早送り速度による軌跡軸の移動は、G0コマンドで有効になります。

構文

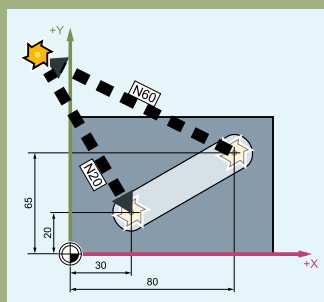
G0 X... Y... Z...
G0 RP=... AP=...

意味

G0:	早送り速度で軸を移動
有効性:	モーダル
X... Y... Z...:	直交座標で終点を指示
RP=... AP=... :	極座標で終点を指示

例

例1:フライス加工



プログラムコード	コメント
N10 G90 S400 M3	;アブソリュート指令入力、主軸右回転
N20 G0 X30 Y20 Z2	;開始位置にアプローチ
N30 G1 Z-5 F1000	;工具に供給
N40 X80 Y65	;直線上で移動
N50 G0 Z2	

主な質問:

コマンド/プロシージャ/機能は何を実行しますか。

構文全体はどのようなになっていますか。
このNCコマンドは、他のNCコマンドと共にプログラムできますか。
特定の順序に従う必要はありますか。
NCコマンドはプログラムブロックで単独にする必要がありますか。

リストされた言語工元素とはどういう意味ですか。
あるコマンドはどのGグループに割り当てられていますか。
このコマンドにはどのような効果がありますか。
パラメータ識別子とは?機能とは?ユニットとは?数値の範囲とは?

一般的なプログラミング例はどのようなものですか。

例で示した構文、意味、例の項目は常に存在しますが、それに加えて他の項目も存在する場合があります。

追加項目	使用/用途
前提条件	プログラム指令可能な機能の応用のために、特定の前提条件を満たす必要がある場合(オプションのライセンスなど)。
有効性	プログラム指令可能な機能が、特定条件下のみ(特定の操作モードのみなど)で有効である場合。

追加項目	使用/用途
制約事項	プログラミング中に、他の機能とのやり取りを考慮する場合。
詳細	機能のプログラミングを理解するために、さらに説明が必要な場合。

記述の範囲

本書には標準仕様の機能についてのみ記載しています。これは、実際に納品されるシステムの機能の範囲とは異なる場合があります。納品されたドライブシステムの機能については、注文書を参照してください。

本書で説明されていないシステムのその他の機能を実行することができる可能性もあります。ただし、これはそのような機能を新しい制御装置によって提供したり、サービス時に提供したりするということではありません。

明瞭化のために、本書ではすべての製品タイプの詳細を記載できているわけではありません。さらに、本書は据え付け、運転およびサービス/メンテナンスで想定されるあらゆるケースを考慮したものではありません。

工作機械メーカーは、自社製品に加えた追加や変更を文書に記載する必要があります。

他社の Web サイト

本書には、他社の Web サイトへのハイパーリンクが含まれる場合があります。当社は、そのような Web サイトとそれらの内容について責任を負わず、賠償を行いません。当社は、そのような Web サイトに表示される情報を管理することはできず、そこで提供される内容および情報に責任を負いません。それらを使用する場合は、ユーザーの責任で行ってください。

1.3 インターネット上の文書

1.3.1 SINUMERIK 840D sl マニュアルの一覧

バージョン 4.8 SP4 以上の SINUMERIK 840D sl の機能に関する広範囲なマニュアルが、840D sl のマニュアル一覧 (<https://support.industry.siemens.com/cs/ww/en/view/109766213>)にあります。



マニュアルは、PDF および HTML5 フォーマットで表示またはダウンロードできます。

マニュアルは、次のカテゴリに分かれています。

- ユーザー:操作
- ユーザー:プログラミング
- メーカー/サービス:機能
- メーカー/サービス:ハードウェア
- メーカー/サービス:コンフィグレーション/セットアップ
- メーカー/サービス:Safety Integrated
- メーカー/サービス:SINUMERIK Integrate / MindApp
- 情報とトレーニング
- メーカー/サービス:SINAMICS

1.3.2 取扱説明書概要 SINUMERIK オペレータコンポーネント

SINUMERIK オペレータコンポーネントに関する完全なドキュメントは、取扱説明書概要 SINUMERIK オペレータコンポーネント (<https://support.industry.siemens.com/cs/document/109783841/technische-dokumentation-zu-sinumerik-bedienkomponenten?dti=0&lc=en-WW>) で提供されています。

マニュアルは、PDF および HTML5 フォーマットで表示またはダウンロードできます。

マニュアルは、次のカテゴリに分かれています。

- 操作パネル
- 機械操作パネル
- 機械の押ボタンパネル
- ハンドヘルドユニット/ミニハンドヘルドデバイス
- その他のオペレータコンポーネント

もっとも重要なドキュメントの概要、SINUMERIK へのリンクとエントリは、SINUMERIK 概要 - トピックページ (<https://support.industry.siemens.com/cs/document/109766201/sinumerik-an-overview-of-the-most-important-documents-and-links?dti=0&lc=en-WW>) にあります。

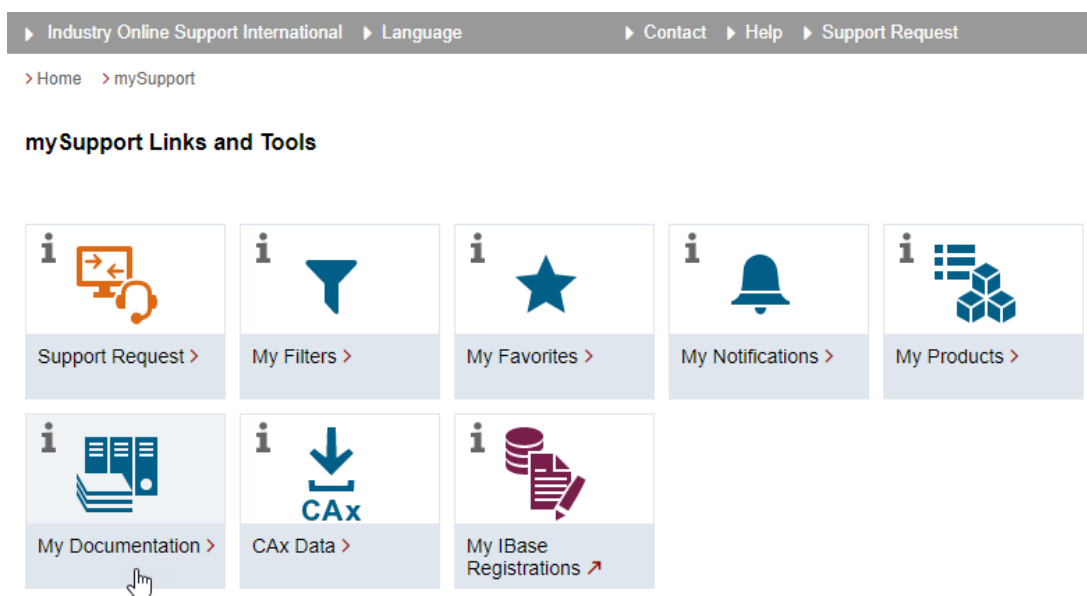
1.4 技術文書についてフィードバック

Siemens Industry Online Support で提供されている技術文書に関するご質問、ご提案や間違いのご指摘がある場合、エントリの最後に用意されている「フィードバックを送信」のリンクをご利用ください。

1.5 mySupport ドキュメンテーション

Web ベースの「mySupport ドキュメンテーション」を使って、当社の内容をベースにしてユーザー独自の文書を作成でき、ユーザーの機械の取扱説明書とすることができます。

このアプリケーションを起動するには、mySupport ホームページ (<https://support.industry.siemens.com/cs/ww/en/my>) の [My Documentation] のタイルをクリックします。



コンフィグレーションされたマニュアルは、RTF、PDF または XML 形式でエクスポートすることができます。

注記

当社の mySupport Documentation アプリケーションのサポート情報には、[設定]のリンクが表示されます。

1.6 サービスとサポート

製品サポート

製品情報については、以下のインターネットサイトを参照してください:

製品サポート (<https://support.industry.siemens.com/cs/ww/en/>)

このアドレスでは、以下が提供されています。

- 最新の製品情報(製品告知)
- FAQ (良くある質問)
- マニュアル
- ダウンロード
- 製品に関する最新情報が記載されたニュースレター

1.6 サービスとサポート

- ユーザーおよび専門家が情報および成功事例を共有するためのグローバルフォーラム
- Siemens データベース(→ [コンタクト先])の当社の現地連絡先情報
- 現場サービス、修理、スペアパーツに関する情報(→ [現場サービス])

テクニカルサポート

テクニカルサポートの国別電話番号については、インターネットの[コンタクト先]の下のアドレス (<https://support.industry.siemens.com/cs/ww/en/sc/4868>)を参照してください。

技術的な質問は、"サポート・リクエスト"エリアのオンラインフォームをご使用ください。

トレーニング

SITRAIN に関する情報は、以下のアドレス (<https://www.siemens.com/sitrain>)にあります。SITRAIN は、当社のオートメーションおよびドライブ装置製品、システムおよびソリューション向けのトレーニングコースを提供します。

Siemens 出張サポート



賞を受賞した「Siemens Industry Online Support」アプリを使用して、いつでもどこからでもシーメンス製品の 300,000 以上の文書にアクセスすることができます。このアプリでは、以下の分野のサポートを受けられます。

- プロジェクトを実行する場合の問題解決
- 不具合が発生した場合のトラブルシューティング
- システムの拡張または新しいシステムの計画

さらに、技術フォーラムとシーメンスの専門家が書いた文書にアクセスできます。

- FAQ
- 用途例

- マニュアル
- 認証
- 製品のお知らせ他

「Siemens Industry Online Support」アプリは、Apple iOS および Android で使用できます。

銘板のデータマトリックスコード

銘板のデータマトリックスコードには、装置固有のデータが含まれています。このコードをスマートフォンで読み取り、「Industry Online Support」モバイルアプリで装置の技術情報を確認することができます。

1.7 重要な製品情報

OpenSSL の利用

この製品には、以下のソフトウェアが含まれる場合があります。

- OpenSSL ツールキットで使用するために OpenSSL プロジェクトによって開発されたソフトウェア
- Eric Young 氏により作成された暗号化ソフトウェア。
- Eric Young 氏により開発されたソフトウェア

関連情報については、以下のインターネットサイトを参照してください：

- OpenSSL (<https://www.openssl.org>)
- Cryptsoft (<https://www.cryptsoft.com>)

一般データ保護規則 (GDPR) 遵守

シーメンスは、データ保護の原則、特にデータの最小化規定を尊重します(プライバシーバイデザイン)。

この製品では、以下を意味します。

この製品は、技術的な機能上のデータ(タイムスタンプ等)以外のいかなる個人データも処理、保存しません。ユーザーがこのデータを他のデータに結びつけている場合、または、同じデータメディア(例：ハードディスク)上に個人関連情報を保存し、このデータを個人的なものにしている場合、当該者は、適用されるデータ保護規定に準拠していることを保証する必要があります。

基本的な安全に関する指示事項

2.1 一般的な安全に関する指示事項

警告

安全に関する情報および残存危険性に注意しない場合の死亡の危険性

関連するハードウェアの資料/文書にある安全に関する情報の遵守や存在する危険性に対する注視がなされていない場合、重大な傷害または死亡事故が発生する可能性があります。

- ハードウェアドキュメントに記載された安全に関する指示事項を遵守してください。
- リスク評価では残存危険性を考慮してください。

警告

不正なまたは変更されたパラメータ設定による機械の誤作動

不正なまたは変更されたパラメータ設定により、傷害や死亡に至る機械の誤動作が発生する場合があります。

- 承認されないアクセスに対するパラメータ設定変更を保護してください。
- 適切な対策を講じることで、考えられる誤動作に対応します (例: 非常停止または非常電源遮断)。

2.2 アプリケーション例に対する保証と責任

アプリケーション例に拘束力はなく、設定、機器、または起こり得る不測の事態に関する完全性を主張するものではありません。アプリケーション例は、特定のカスタマソリューションを示したのではなく、代表的なタスクを支援することのみを目的にしています。

ユーザー自身が責任を持って本製品の適切な運用を確実なものとしてください。アプリケーション例は、機器の使用、取り付け、操作、および保守を行うときの安全な取扱いに対する責任からお客様を解放するものではありません。

2.3 セキュリティ機能に関する情報

シーメンスは、セキュアな環境下でのプラント、システム、機械およびネットワークの運転をサポートする産業用セキュリティ機能を有する製品およびソリューションを提供します。

2.3 セキュリティ機能に関する情報

プラント、システム、機械およびネットワークをサイバー脅威から守るためには、総合的かつ最新の産業用セキュリティコンセプトを実装し、それを継続的に維持することが必要です。シーメンスの製品とソリューションは、そのようなコンセプトの 1 要素を形成します。

お客様は、プラント、システム、機械およびネットワークへの不正アクセスを防止する責任があります。システム、機械およびコンポーネントは、企業内ネットワークのみに接続するか、必要な範囲内かつ適切なセキュリティ対策を講じている場合にのみ（例：ファイアウォールやネットワークセグメンテーションの使用など）インターネットに接続することとすべきとシーメンスは考えます。

産業用セキュリティ対策に関する詳細な情報は、<https://www.siemens.com/industrialsecurity> (<https://www.siemens.com/industrialsecurity>) をご覧下さい。

シーメンスの製品とソリューションは、セキュリティをさらに強化するために継続的に開発されています。シーメンスは、利用可能になったらすぐ製品の更新プログラムを適用し、常に最新の製品バージョンを使用することを強くお勧めします。サポートが終了した製品バージョンを使用すること、および最新の更新プログラムを適用しないことで、お客様はサイバー脅威にさらされる危険が増大する可能性があります。

製品の更新プログラムに関する最新情報を得るには、<https://www.siemens.com/industrialsecurity> (<https://new.siemens.com/global/en/products/services/cert.html#Subscriptions>) からシーメンス産業セキュリティ RSS フィードを購読してください。

関連情報はインターネットから入手できます。

産業セキュリティ設定マニュアル (<https://support.industry.siemens.com/cs/ww/en/view/108862708>)

警告

ソフトウェアの誤動作による安全でない運転状態

ソフトウェアの誤動作 (例: ウィルス、トロイの木馬、マルウェアまたはワーム) は、死亡事故、重大な傷害事故や物損事故に至る場合があるシステムにおける安全ではない運転状態の原因となる場合があります。

- 最新のソフトウェアを使用して下さい。
- オートメーションおよびドライブコンポーネントを、据えつけられた機器または機械装置に対する総合的で最先端の産業セキュリティコンセプトに組み込んでください。
- 据えつけられたすべての製品を総合的な産業セキュリティコンセプトに確実に組み込むようにしてください。
- 適切な保護対策で、例えば、ウィルススキャンで悪意のあるソフトウェアから交換可能な記憶媒体上に保存されたファイルを保護してください。
- セットアップが完了したら、すべての安全関連の設定をチェックします。

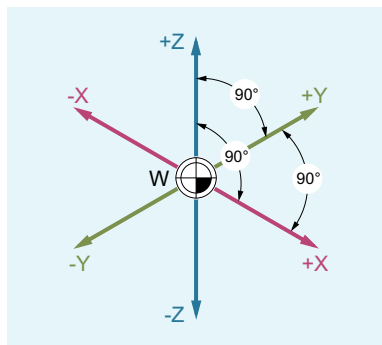
基本編

3.1 ジオメトリの基礎知識

3.1.1 ワークの位置

3.1.1.1 位置指定の基準系

機械または制御装置が、NC プログラムで指定された位置で動作するために、これらの位置指定は、機械軸の移動方向に移動可能な基準系で作成してください。工作機械のワーク座標系として、DIN 66217 に準拠した直交(つまり、右回りの互いに直交する)座標系が使用されます。



ワーク原点(W)はワーク座標系の原点です。

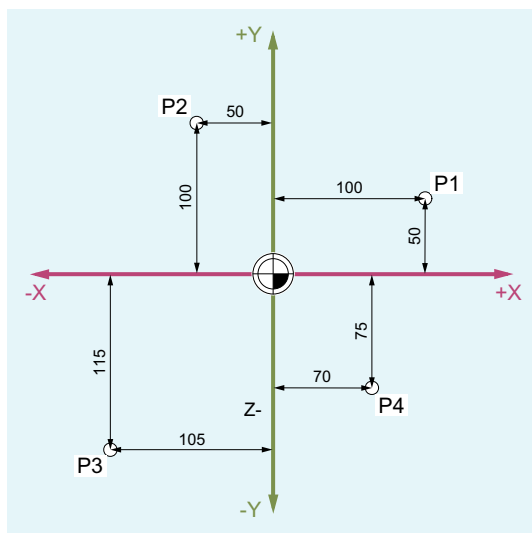
3.1.1.2 直交座標系

座標系の軸には、次元が割り当てられます。このため、座標系のすべての点を明確に記述でき、すべてのワーク位置を、(X、Y、および Z)方向と 3 つの数値に基づいて記述することができます。ワーク原点の座標は常に X0、Y0、Z0 です。

3.1 ジオメトリの基礎知識

直交座標系による位置指定

わかりやすいように、次の例では、座標系の 1 つの平面、X/Y 平面を使用して説明しています。

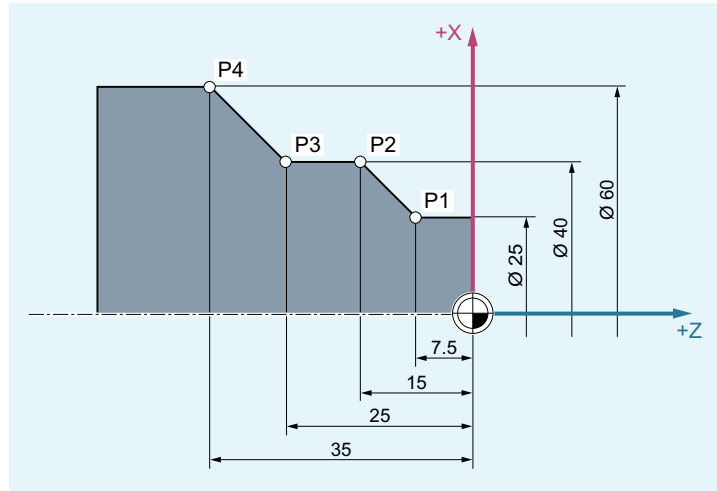


点 P1～P4 の座標は次のとおりです。

位置	座標
P1	X100 Y50
P2	X-50 Y100
P3	X-105 Y-115
P4	X70 Y-75

例:旋削のワークの位置

旋盤を使用する場合は、1つの平面で輪郭を記述できます。

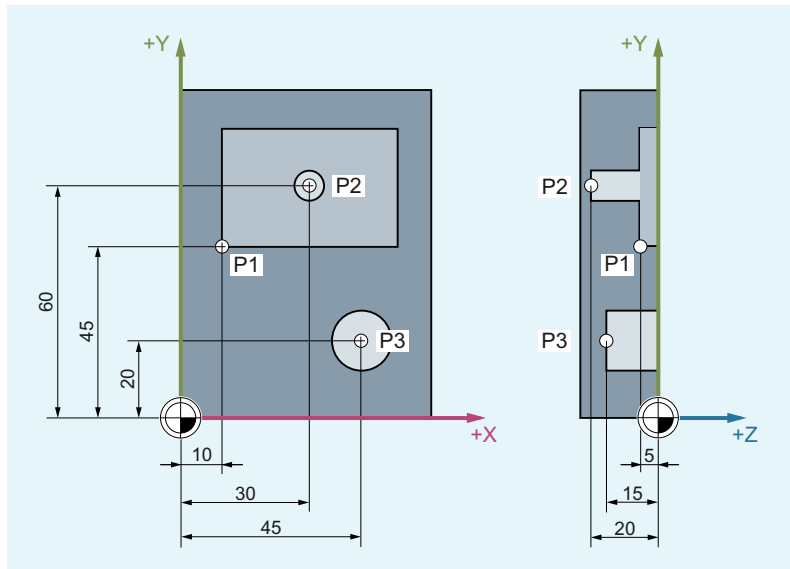


点 P1～P4 の座標は次のとおりです。

位置	座標
P1	X25 Z-7.5
P2	X40 Z-15
P3	X40 Z-25
P4	X60 Z-35

例:フライス加工のワークの位置

フライス加工の場合は、深さ方向の送り込みも記述してください。つまり、3番目の座標（この場合はZ）にも数値を割り当ててください。



点 P1～P3 の座標は次のとおりです。

位置	座標
P1	X10 Y45 Z-5
P2	X30 Y60 Z-20
P3	X45 Y20 Z-15

3.1.1.3 極座標

直交座標の代わりに極座標を使用して、ワークの位置を記述できます。これは、ワークまたはワークの一部が、半径と角度の寸法指定になっているときに便利です。寸法指定が開始される点を「極」と呼びます。

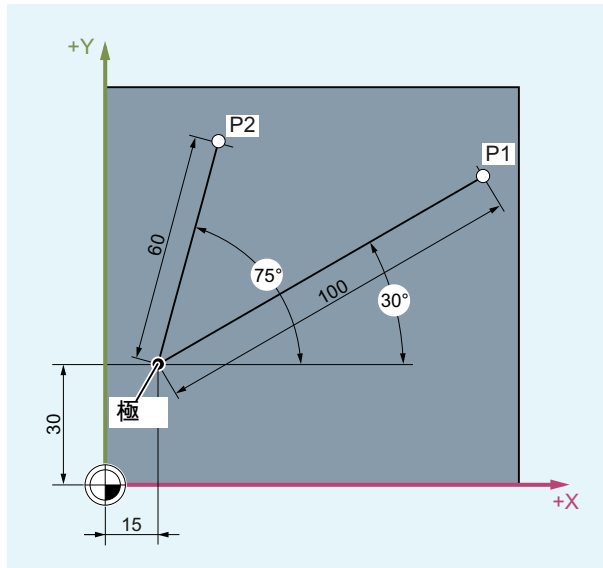
極座標形式による位置指定

極座標は、**極半径**と**極角度**で構成されます。

極半径は、極と位置の間の距離です。

極角度は、極半径と、作業平面の水平軸との間で成す角度です。負の極角度は右回り方向の、正の極角度は左回り方向の角度です。

例



点 P1 と P2 は、極を基準にして、次のように記述されます。

位置	極座標
P1	RP=100 AP=30
P2	RP=60 AP=75
RP:極半径	
AP:極角度	

3.1.1.4 アブソリュート指令

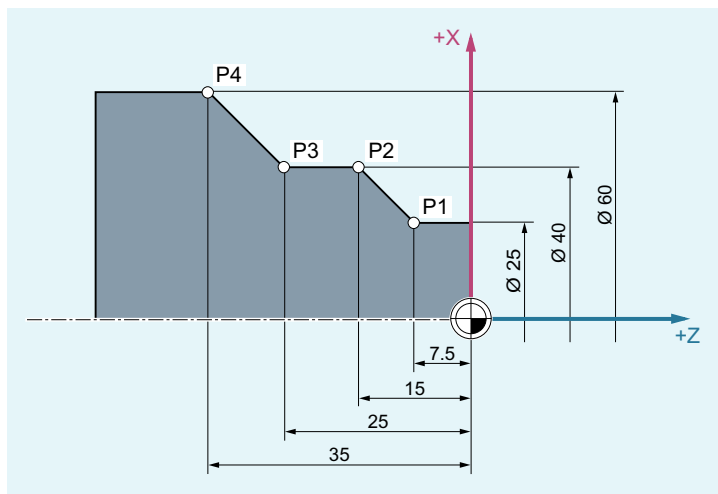
アブソリュート指令の位置指定

アブソリュート指令を使用すると、すべての位置指定は、現在有効な原点が基準点になります。

工具の移動に適用した場合は、次のようになります。

位置指定は、工具が移動する位置になります。

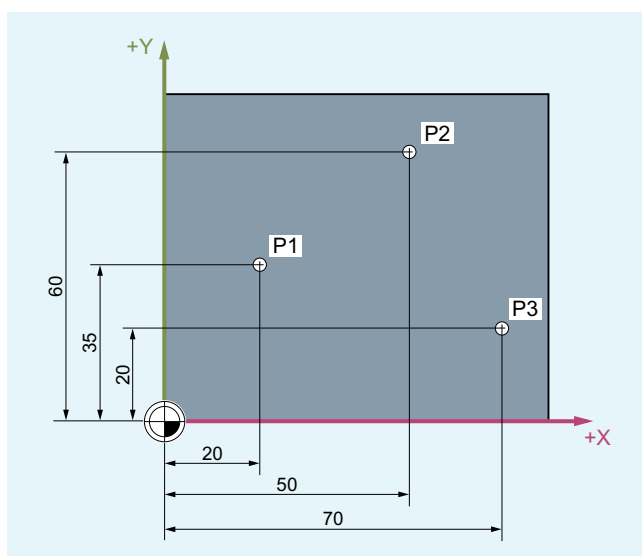
例:旋削



アブソリュート指令では、次の位置指定の結果は点 P1～P4 になります。

位置	アブソリュート指令の位置指定
P1	X25 Z-7.5
P2	X40 Z-15
P3	X40 Z-25
P4	X60 Z-35

例:フライス加工



アブソリュート指令では、次の位置指定の結果は点 P1～P3 になります。

位置	アブソリュート指令の位置指定
P1	X20 Y35
P2	X50 Y60
P3	X70 Y20

3.1.1.5 インクリメンタル指令

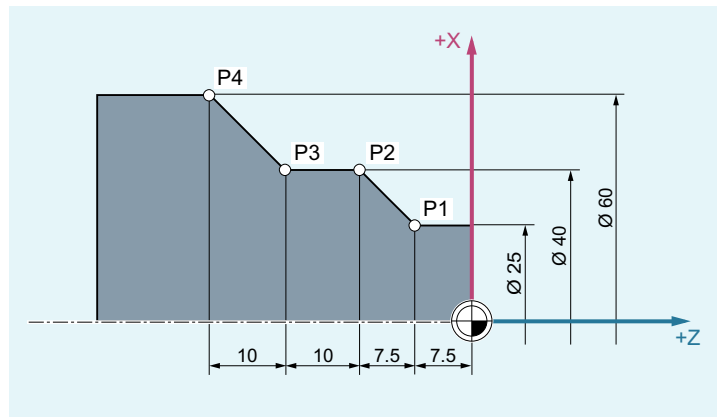
インクレメンタル指令の位置指定

加工図面では、寸法は、原点ではなく、別のワーク上の点を基準点にすることもよくあります。インクレメンタル指令で指定すると、このような指令を変換する必要がなくなります。この指令表記では、位置指定は以前の指令点を基準にします。

工具の移動に適用した場合は、次のようになります。

インクレメンタル指令は、工具が移動する距離を記述します。

例:旋削



インクレメンタル指令では、以下の位置指定が点 P2～P4 に取得されます。

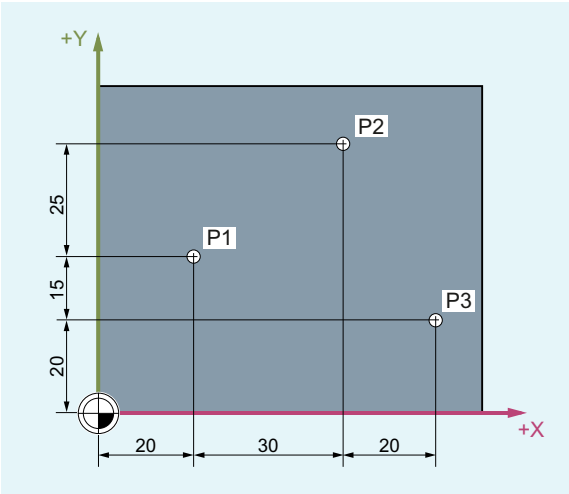
位置	インクレメンタル指令の位置指定	指定の基準点:
P2	X15 Z-7.5	P1
P3	Z-10	P2
P4	X20 Z-10	P3

注記

DIAMOF または DIAM90 (ページ 182)が有効な場合は、インクレメンタル指令(G91)で設定された距離が半径寸法としてプログラム指令されます。

例:フライス加工

インクレメンタル指令の点 P1～P3 の位置指定は次のとおりです。



インクレメンタル指令では、以下の位置指定が点 P1～P3 に取得されます。

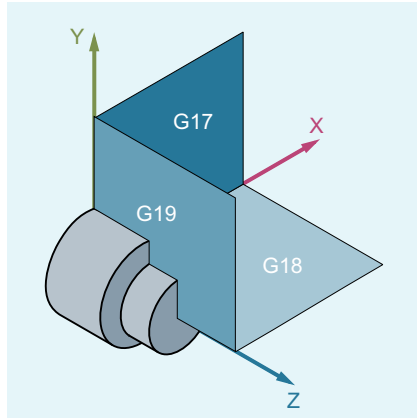
位置	インクレメンタル指令の位置指定	指定の基準点:
P1	X20 Y35	原点
P2	X30 Y20	P1
P3	X20 Y-35	P2

3.1.2 作業平面

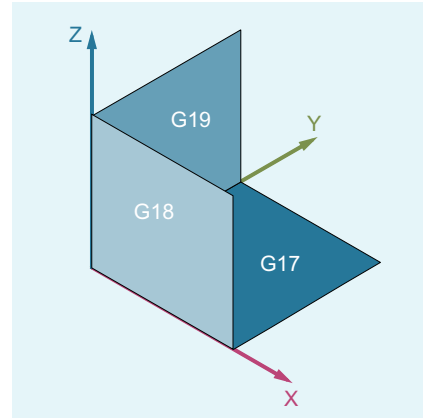
NC プログラムは加工平面に関する情報を必要とします。その後でのみ、制御装置は、たとえば工具補正値を正しく考慮に入れることができます。特定のタイプの円弧軌跡プログラミング、および極座標のプログラミングには、加工平面の指定も必要です。

加工平面は、2 つの座標軸を使用するベースとなる直交ワーク座標系で指定されます。3 番目の座標軸は、この加工平面に垂直に、工具の切り込み方向を特定します(2 次元加工など)。

旋削/フライス削りの加工平面



旋削の加工平面



フライス削りの加工平面

加工平面の有効化

作業平面は、NC プログラムで G 命令 G17、G18、G19 を使用して有効化されます。関係は以下のように定義されます。

G 命令	加工平面	横座標	縦座標	垂直△切り込み方向
G17	X/Y	X	Y	Z
G18	Z/X	Z	X	Y
G19	Y/Z	Y	Z	X

3.1 ジオメトリの基礎知識

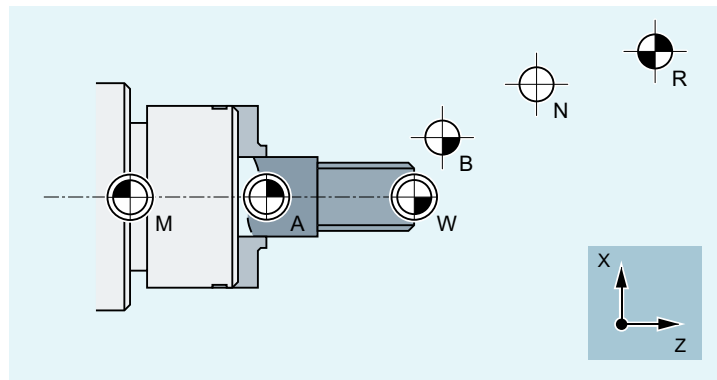
3.1.3 原点、およびレファレンス点

次のようなさまざまな原点および基準点が NC 工作機械で定義されます。

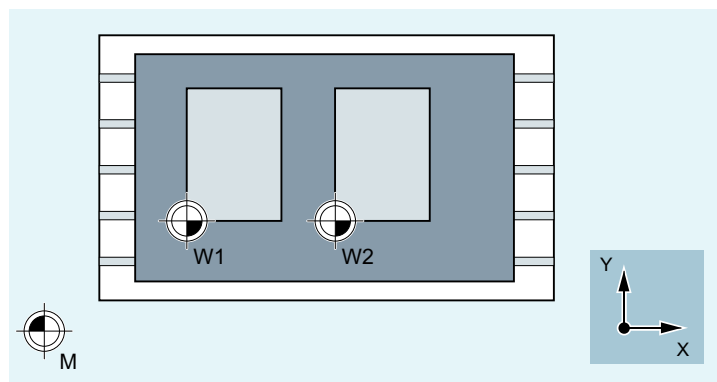
原点		
	M	機械原点 機械座標系(MCS)は機械原点により定義されます。他の基準点はすべて、機械原点を基準にします。
	W	ワーク原点 = プログラム原点 ワーク原点により、機械原点に対してワーク座標系が定義されます。
	A	停止点 ワーク原点と同じ場合があります(旋盤の場合のみ)。

基準点		
	R	基準点 出力カムと検出器で定義した位置この点の軸の位置を正確にレファレンス点に設定するために、機械原点 M までの距離を確認してください。
	B	起点 プログラムにより定義できます。1 番目の工具はここから加工を開始します。
	T	工具ホルダの基準点 工具ホルダ上にあります。工具長を入力すると、制御装置により、工具先端と工具ホルダの基準点との間の距離が計算されます。
	N	工具交換位置

旋削の原点とレファレンス点



フライス加工の原点



3.1.4 座標系

次の座標系に区別されます。

- (ページ 43)機械原点 **M** を基準とする機械座標系(MCS)
- 基本座標系(BCS) (ページ 46)
- 基本ゼロオフセットシステム(BZS) (ページ 47)
- 設定可能ゼロオフセットシステム(SZS) (ページ 48)
- (ページ 48)ワーク原点 **W** を基準とするワーク座標系(WCS)

3.1.4.1 機械座標系(MCS)

機械座標系は、物理的に存在するすべての機械軸で構成されます。

3.1 ジオメトリの基礎知識

機械座標系では、レファレンス点、工具、およびパレットの交換位置(機械の固定点)が定義されます。

プログラムを機械座標系で直接実行する場合(一部の G 命令では可能です)は、機械の物理軸が直接アドレス指定されます。存在しているワーククランプはいずれも、考慮されません。

注記

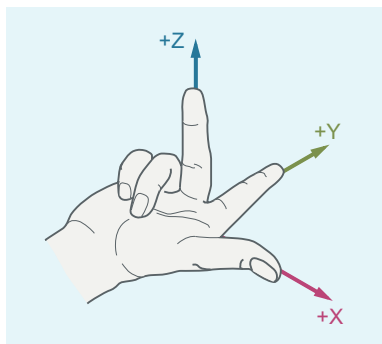
さまざまな機械座標系(5 軸座標変換など)が存在する場合は、内部座標変換を使用して、このプログラムが実行される座標系上の機械のキネマティクスにマッピングされます。

三本指の法則

機械に対しての座標軸の向きは、機械のタイプにより異なります。軸の方向は、右手の「三本指の法則」と呼ばれる法則に従います(DIN 66217 準拠)。

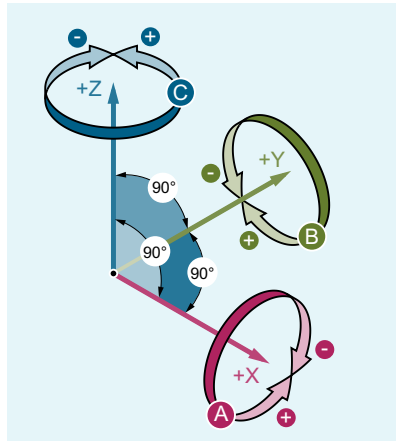
機械の前部から見ると、右手の中指は、機械の主軸の切り込み方向とは逆方向を指します。したがって、

- 親指は+X 方向を指します。
- 人差し指は+Y 方向を指します。
- 中指は+Z 方向を指します。



座標軸 X、Y、および Z を中心とする回転動作はそれぞれ、A、B、および C で指定されます。回転方向は、各座標軸を正方向に向かって見た場合の回転動作の方向から取得されます。

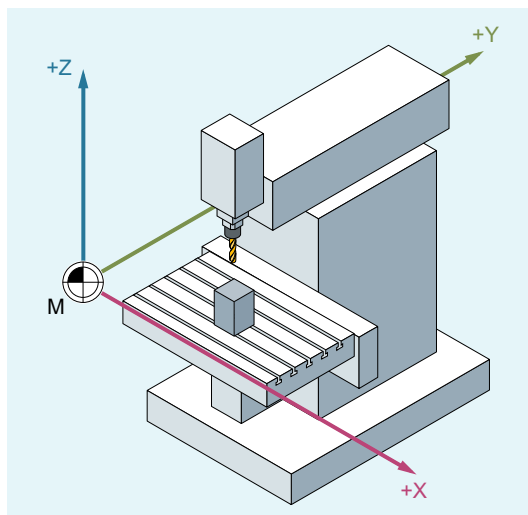
回転動作の方向	回転方向
右回り	正
反時計回り	負



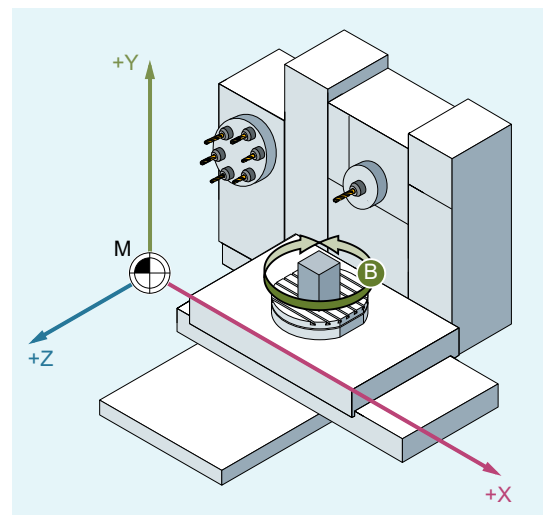
X、Y、Z 互いの上に配置された垂直座標軸
 Z
 A、B、C X、Y、Z を中心とした回転軸
 C

さまざまな機械のタイプの座標系の位置

「三本指の法則」で決められる座標系の位置の向きは、機械のタイプにより異なる場合があります。以下に2つの例を示します。



立3軸フライス盤



横4軸フライス盤

3.1 ジオメトリの基礎知識

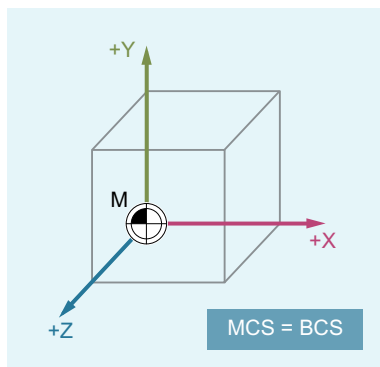
3.1.4.2 基本座標系(BCS)

基本座標系(BCS)は、互いに垂直な 3 つの軸(ジオメトリ軸)と、幾何学的な相互関係を持たない他の付加軸で構成されます。

キネマティック座標変換のない工作機械の場合

キネマティック座標変換(5 軸座標変換、TRANSMIT/TRACYL/TRAANG など)を使用しないで、BCS を MCS にマッピングできるときは、BCS と MCS は常に一致します。

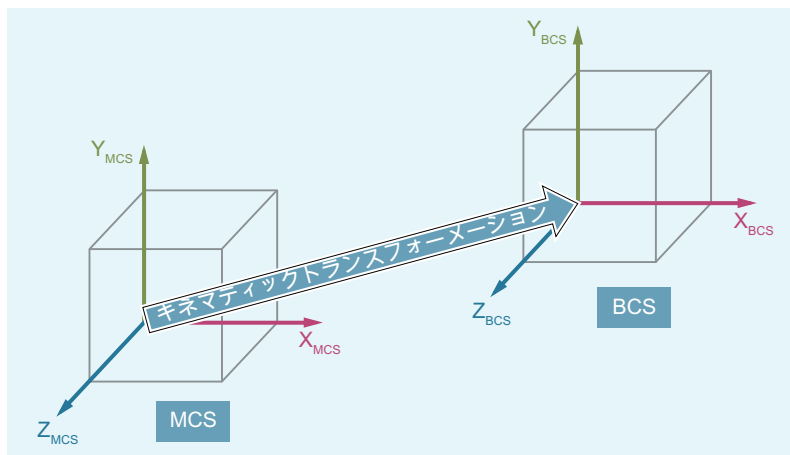
そのような機械上では、機械軸とジオメトリ軸は同じ名称にできます。



キネマティック座標変換のある工作機械の場合

キネマティック座標変換(5 軸座標変換、TRANSMIT/TRACYL/TRAANG など)を使用して BCS が MCS にマッピングされる場合は BCS と MCS は一致しません。

そのような機械上では、機械軸とジオメトリ軸は異なる名称にしなければなりません。

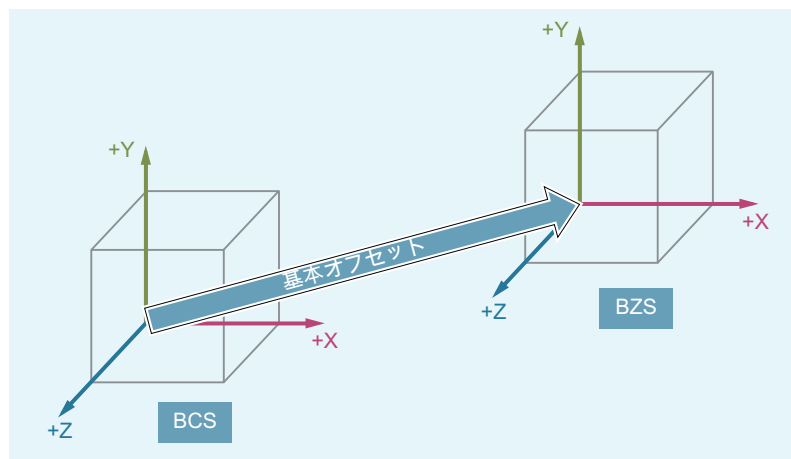


機械のキネマティクス

ワークは常に、2次元または3次元の直交座標系(WCS)でプログラム指令されます。しかし、そのようなワークは、互いに垂直でない回転軸または直線軸を装備した工作機械上でこれまで以上に頻繁にプログラム指令されています。キネマティック座標変換は、ワーク座標系(直交)でプログラム指令された座標を実際の機械軸動作で表すために使用されます。

3.1.4.3 基本ゼロオフセットシステム(BZS)

基本ゼロオフセットシステム(BZS)は、基本オフセットを使用して基本座標系から取得されます。



基本オフセット

基本オフセットは、BCS と BZS の間の座標変換を記述します。たとえば、パレット原点の範囲を定義するために使用されます。

基本オフセットは、次の要素で構成されます。

- 外部ゼロオフセット
- DRF オフセット
- 重畳移動
- システムフレーム結合
- 基本フレーム結合

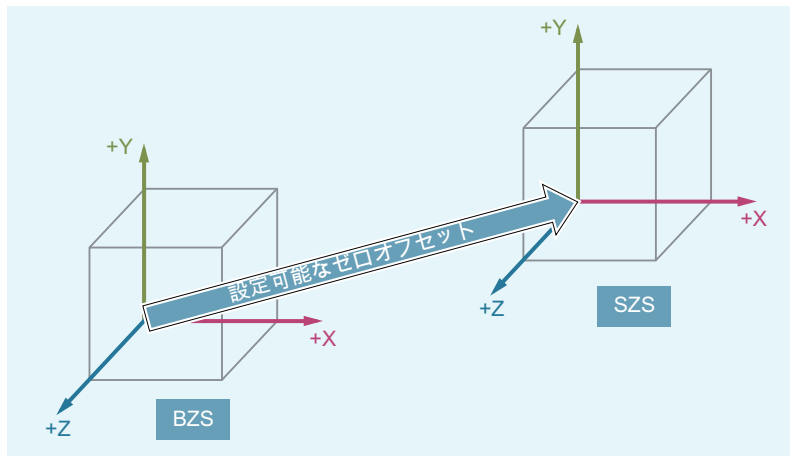
3.1 ジオメトリの基礎知識

3.1.4.4 設定可能ゼロオフセットシステム(SZS)

設定可能なゼロオフセット

「設定可能ゼロオフセットシステム」(SZS)は、設定可能ワークオフセットの結果として、基本ゼロオフセットシステム(BZS)から得られます。

設定可能ワークオフセットは、NC プログラムの G 命令 G54～G57、および G505～G599 により有効になります。



プログラマブル座標変換(フレーム)が無効の場合、「設定可能ゼロオフセットシステム」はワーク座標系(WCS)になります。

プログラマブル座標変換(フレーム)

NC プログラム内で、最初に選択されたワーク座標系(または「設定可能ゼロオフセットシステム」)を別の位置に移動したり、必要に応じて、回転、反転、拡大/縮小をおこなったりすることが便利な場合、または必要な場合があります。これは、プログラマブル座標変換(フレーム) (ページ 340)を使用しておこなわれます。

注記

プログラマブル座標変換(フレーム)は常に「設定可能ゼロオフセットシステム」を基準にします。

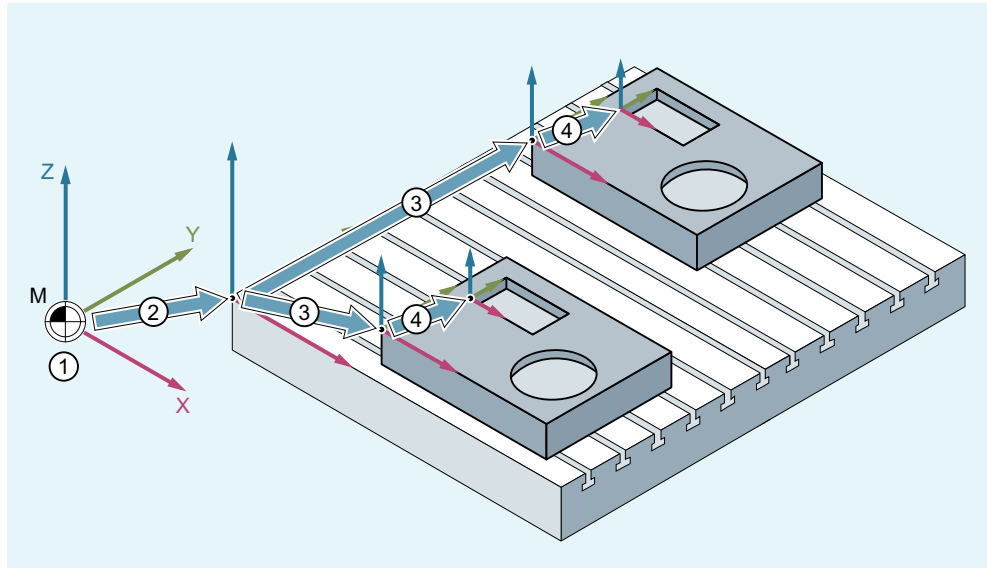
3.1.4.5 ワーク座標系(WCS)

ワークのジオメトリはワーク座標系(WCS)で記述されます。つまり、NC プログラムのデータはワーク座標系を基準にします。

ワーク座標系は常に直交座標系であり、特定のワークに割り当てられます。

3.1.4.6 さまざまな座標系間の関係

次の例は、さまざまな座標系間の関係をわかりやすく表わしています。



- ① キネマティック座標変換が有効ではありません。つまり、機械座標系と基本座標系は一致しています。
- ② パレット原点を含む基本ゼロオフセットシステム(BZS)は、基本オフセットから得られます。
- ③ ワーク 1 またはワーク 2 の「設定可能ゼロオフセットシステム」(SZS)は、設定可能ワークオフセット G54 または G55 で指定されます。
- ④ ワーク座標系(WCS)は、プログラマブル座標変換から得られます。

3.2 NC プログラミングの基礎知識

注記

DIN 66025 は NC プログラミングのガイドラインです。

3.2 NC プログラミングの基礎知識

3.2.1 NC プログラム名称

規則

各 NC プログラムには、作成時にプログラム名称(識別子)を割り当ててください。プログラム名称は、以下の規則に従って自由に選択できます。

- 使用可能な文字:
 - 英字:A ... Z、a ... z
 - 数字:0 ... 9
 - アンダースコア:_
- 先頭の 2 文字は、英字 2 文字またはアンダースコアと英字 1 文字にしてください。

注記

この条件を満たす場合は、プログラム名称を指定するだけで、別のプログラムからサブプログラムとして NC プログラムを呼び出すことができます。ただし、プログラム名称の先頭が数字である場合、サブプログラム呼び出しは CALL 命令でのみ呼び出すことができます。

- 最大長さ:24 文字

注記

大文字/小文字

SINUMERIK NC 言語では、大文字と小文字は **区別されません**。

注記

使用できないプログラム名称

Windows アプリケーションで問題が生じることのないように、次のプログラム名の使用は避けてください。

- CON、PRN、AUX、NUL
- COM1、COM2、COM3、COM4、COM5、COM6、COM7、COM8、COM9
- LPT1、LPT2、LPT3、LPT4、LPT5、LPT6、LPT7、LPT8、LPT9

その他の制限事項については、「名称 (ページ 425)」を参照してください。

制御装置内部の拡張

プログラムの作成時に割り当てたプログラム名称には、制御装置内で接頭語と接尾語が追加され、拡張されます。

- 接頭語: `_N_`
- 接尾語:
 - メインプログラム: `_MPF`
 - サブプログラム: `_SPF`

せん孔テープフォーマットのファイル

V.24 インタフェース経由で読み込まれる外部で作成したプログラムファイルは、せん孔テープフォーマットで作成してください。

せん孔テープフォーマットのファイルのプログラム名称には、次の規則が追加されて適用されます。

- 最初の文字: `%`
- 次に 4 文字のファイル拡張子: `_xxx`

例:

- `%_N_SHAFT123_MPF`
- `%Flange3_MPF`

3.2.2 NC プログラムの構成と内容

3.2.2.1 ブロックとブロック構成要素

ブロック

NC プログラムは、一連の NC ブロックで構成されています。各ブロックには、ワーク加工手順を実行するためのデータが含まれます。

ブロック構成要素

NC ブロックは、次の構成要素から成ります。

- DIN 66025 準拠の命令(ステートメント)
- NC 高機能言語の要素

DIN 66025 準拠の命令

DIN 66025 準拠の命令は、アドレス文字と 1 つの数字、または算術値を表わす数字列から構成されます。

アドレス文字(アドレス)

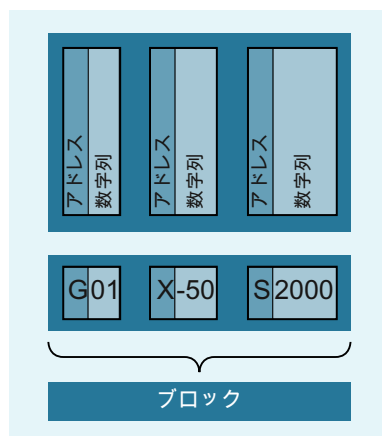
アドレス文字(通常は英字)で、命令の意味が定義されます。

例:

アドレス文字	意味
G	G 命令(準備機能)
X	X 軸の位置データ
S	主軸速度

数字列

数字列は、アドレス文字に割り当てられた数値です。これらの数字列には、符号と小数点を含めることができます。符号は常に、アドレス文字と数字列の間にあります。正符号(+)と先頭のゼロ(0)は指定する必要はありません。



NC 高機能言語の要素

DIN 66025 準拠の命令セットは、最近の工作機械の複雑な加工処理のプログラミングには不十分であるため、NC 高機能言語の要素により拡張されています。

これらには、以下の例があります。

- NC 高機能言語の命令
DIN 66025 準拠の命令とは対照的に、NC 高機能言語の命令は、次のような複数のアドレス文字から成ります。
 - OVR(速度オーバーライド)
 - SPOS(主軸の位置決め)
- 識別子(定義名称)は次のとおりです。
 - システム変数
 - ユーザー定義変数
 - サブプログラム
 - キーワード
 - ジャンプマーク
 - マクロ

注記

識別子は一義的にしてください。また、複数の異なる目的には使用できません。

- 比較演算子
- 論理演算子
- 算術機能
- 制御構造

命令の効果

命令は、モーダル、またはノンモーダルのいずれかです。

- モーダル
モーダル命令は、次に示す項目が指令されるまで、プログラム指令値(以降のすべてのブロックで)の有効性を保持します。
 - 同じ命令で新しい値がプログラム指令された。
 - 以前に有効であった命令の働きを無効にする命令がプログラム指令された。
- ノンモーダル
ノンモーダル命令は、この命令がプログラム指令されたブロックにのみ適用されます。

3.2 NC プログラミングの基礎知識

エンドオブプログラム

実行処理の最後のブロックには、プログラム終了を表わす特別な命令の M2、M17、または M30 が含まれます。

3.2.2.2 ブロックの規則

ブロックの先頭

NC ブロックは、ブロックの先頭のブロック番号により識別されます。この番号は、次のように「N」と正の整数から成ります。

N40 ...

ブロック番号の順序は任意ですが、昇順にすることを推奨します。

注記

ブロック番号は、検索時に一義的な結果が得られるように、プログラム内で一義的にしてください。

ブロック終了

ブロックは文字 LF (LINE FEED = 新しい行)で終了します。

注記

LF 文字を記述する必要はありません。改行により、自動的に生成されます。

ブロック長

ブロックには最大 **512 文字**(コメントとブロック終了文字の LF を含む)を入れることができます。

注記

通常は、それぞれ 66 文字までの 3 つのブロックが、実行中のブロック表示画面に表示されます。コメントも表示されます。メッセージは、別のメッセージウィンドウに表示されます。

命令の順序

ブロック構成をできるだけ明確にするために、ブロック内の命令は次の順序で並べてください。

N... G... X... Y... Z... F... S... T... D... M... H...

アドレス	意味
N	ブロック番号のアドレス
G	準備機能
X、Y、Z	位置データ
F	送り速度
S	主軸速度
T	工具
D	工具オフセット番号
M	追加機能
H	補助機能

注記

次のような一部のアドレスは、1 ブロック内で繰り返し使用できます。

G...、M...、H...

3.2.2.3 値の割り当て

アドレスには値を割り当てることができます。これには次の規則が適用されます。

- 次の場合は、アドレスと値の間に「=」記号を入れてください。
 - アドレスに複数の文字が含まれる。
 - 値に複数の定数が含まれる。
 アドレスが 1 文字の英字で、値が 1 つの定数のみから成る場合は、「=」記号を省略できます。
- 複数の記号を使用できます。
- アドレス文字の後にはセパレータを使用できます。

3.2 NC プログラミングの基礎知識

例:

X10	アドレス(X)に割り当てられた値(10 です)。「=」は不要です。
X1 = 10	数値拡張子(1)を含むアドレス(X)に割り当てられた値(10)です。「=」が必要です。
X = 10 * (5 + SIN(37.5))	数値式を使用して割り当てられた値です。「=」が必要です。

注記

数値拡張子の後には、特殊文字「=」、「(」、「[」、「)」、「]」、「,」のいずれかまたは演算子を付けてください。これは、数値拡張子を含むアドレスと、値が割り当てられたアドレスを区別するためです。

3.2.2.4 コメント

NC プログラムをわかりやすくするために、NC ブロックにコメントを追加できます。

コメントはブロックの末尾に配置され、NC ブロックのプログラム区間からはセミコロン(「;」)で区切られます。

例 1:

プログラムコード	コメント
N10 G1 F100 X10 Y20	; NC ブロックの説明コメント

例 2:

プログラムコード	コメント
N10	; 会社名 G&S、注文番号 12A71
N20	; プログラム作成者 H. スミス、TV 4 部; 1994 年 11 月 21 日
N50	; 章番号 12、水中ポンプ TP23A タイプのハウジング

注記

コメントは保存されて、プログラム運転時に実行中のブロック表示画面に表示されます。

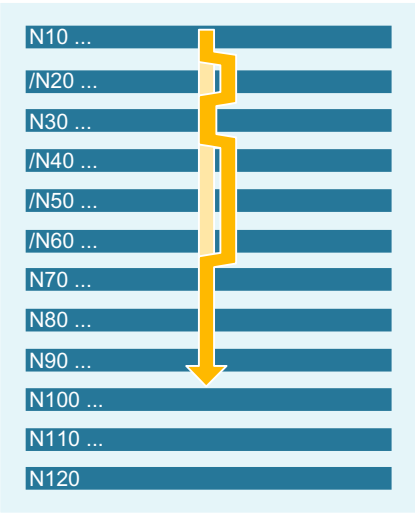
3.2.2.5 ブロックスキップ

プログラムを実行する毎に実行しなくてもよい NC ブロックは、スキップして加工を止めることができます。この機能は、たとえば、新しいプログラムを試験および/または試運転する場合に使用します。

ブロックスキップ

スキップされるブロックは、パートプログラム内でブロック番号の前に記号「/」を付けて示されます。複数の連続するブロックもスキップできます。スキップされたブロックの命令は実行されません。プログラムは、スキップされない次のブロックで再開されます。

例:



プログラムコード	コメント
N10 ...	; 実行されます
/N20 ...	; スキップされます
N30 ...	; 実行されます
/N40 ...	; スキップされます
/N50 ...	; スキップされます
/N60 ...	; スキップされます
N70 ...	; 実行されます
...	

スキップレベル

ブロックにはスキップレベル(最大 10)を割り当てることができます。このレベルは、ユーザーインターフェースまたは PLC ユーザープログラムで有効にすることができます。

3.3 NC プログラムの作成

割り当ては、NC プログラムで行われます(スラッシュと、その後にスキップレベルの数が続く)。各ブロックに指定できるスキップレベルは 1 つだけです。

例:

プログラムコード	コメント
/ ...	; ブロックをスキップします(スキップレベル 1)
/0 ...	; ブロックをスキップします(スキップレベル 1)
/1 N010...	; ブロックをスキップします(スキップレベル 2)
/2 N020...	; ブロックをスキップします(スキップレベル 3)
...	
/7 N100...	; ブロックをスキップします(スキップレベル 8)
/8 N080...	; ブロックをスキップします(スキップレベル 9)
/9 N090...	; ブロックをスキップします(スキップレベル 10)

注記

スキップレベルを変更できるのは、コントロールシステムが停止/リセット状態にある場合のみです。

注記

使用可能なスキップレベルの数は、表示マシンデータにより異なります。

注記

ブロックのスキップは、ブロック検索中も有効なままです。

注記

システム変数とユーザー変数を使用して、条件付きジャンプで、プログラム実行も制御できます。

3.3 NC プログラムの作成

3.3.1 基本手順

通常は、NC 言語による個々の運転手順のプログラミングが NC プログラムの開発作業に占める割合はわずかです。

実際の命令をプログラミングする前に、運転手順の計画と準備をおこなってください。事前に計画する NC プログラムの構成と実行が、正確であるほど、分かりやすくエラーの

ない完全なプログラムを、短期間に、簡単に作成できます。明確な構成を持つプログラムは、後で変更が必要になった場合に、特に便利です。

すべての部分が同じというわけではないため、すべてのプログラムを同じ方法で作成することは無意味です。ただし、次の手順は、ほとんどの場合に使用できることがわかっています。

手順

1. ワーク図面を準備する。

- ワーク原点を定義する。
- 座標系を描く。
- 抜けている全ての座標を計算する。

2. 加工処理を定義する。

- どの工具が、いつ、どの輪郭の加工に使用されるか？
- ワークの個々の要素がどんな順序で加工されるか？
- 個々のどの要素が繰り返されるか(場合によっては、さらに回転されるか)、および個々のどの要素をサブプログラムに設定するか？
- 他のパートプログラムまたはサブプログラムに、現在のワークで使用可能な輪郭区間がないか？
- ゼロオフセット、回転、ミラーリング、およびスケーリングが便利な、または必要である場所はどこか(フレーム仕様)?

3. 加工計画を作成する。

次の例のように、すべての加工運転を手順毎に定義する。

- 位置決め用の早送り移動
- 工具交換
- 加工平面の定義
- 点検のための後退
- 主軸、冷却液のオン/オフの切り替え
- 工具データの呼び出し
- 送り
- 軌跡補正
- 輪郭へのアプローチ
- 輪郭からの後退
- その他

4. プログラミング言語で加工手順を編集する。

- 個々の手順を 1 つまたは複数の NC ブロックとして書く

5. 個々の手順をプログラムにまとめる。

3.3 NC プログラムの作成

3.3.2 使用可能な文字

NC プログラムは、次の文字を使用して書くことができます。

- 大文字:
A、B、C、D、E、F、G、H、I、J、K、L、M、N、(O)、P、Q、R、S、T、U、V、W、X、Y、Z
- 小文字:
a、b、c、d、e、f、g、h、i、j、k、l、m、n、o、p、q、r、s、t、u、v、w、x、y、z
- 数字:
0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- 特殊文字:
以下の表を参照してください。

特殊文字	意味
%	プログラム開始文字(外部 PC でプログラムを作成する場合にのみ使用)
(	パラメータまたは式を一括してまとめるため
)	パラメータまたは式を一括してまとめるため
[	アドレスまたはインデックスを一括してまとめるため
]	アドレスまたはインデックスを一括してまとめるため
<	より小さい
>	より大きい
:	メインブロック、ラベルの終了、連鎖演算子
=	割り当て、式の等号
/	除算、ブロックスキップ
*	乗算
+	加算
-	減算、負符号
"	二重引用符、文字列の識別子
'	一重引用符、次の特別な数値の識別子:16 進数、2 進数
\$	システム変数の識別子
s_	アンダースコア、英字に属する
?	予備
!	予備

特殊文字	意味
.	小数点
,	コンマ、パラメータのセパレータ
;	コメントの開始
&	書式文字、スペース文字と同じ働き
LF	ブロック終了
タブ文字	セパレータ
素材	セパレータ(ブランク)

注記

英字「O」と数字「0」を区別できるように、配慮してください。

注記

アルファベットの大文字と小文字は区別されません(例外:工具呼び出し)。

注記

印字ができない特殊文字はブランクとして扱われます。

3.3.3 プログラムヘッダ

ワーク輪郭の加工をおこなう実際の動作ブロックの前に配置された NC ブロックは、プログラムヘッダーと呼ばれます。

プログラムヘッダーには、下記の関連情報/命令が含まれます。

- 工具交換
- 工具オフセット
- 主軸動作
- 送り速度制御
- ジオメトリ設定(ゼロオフセット、作業平面の選択)

3.3 NC プログラムの作成

旋削のプログラムヘッダー

次の例は、旋削の NC プログラムヘッダーの一般的な構成を示します。

プログラムコード	コメント
N10 G0 G153 X200 Z500 T0 D0	; 工具タレットが旋回する前に工具ホルダを後退させます。
N20 T5	; 工具 5 で旋回します。
N30 D1	; 工具刃先データセットを有効にします。
N40 G96 S300 LIMS=3000 M4 M8	; 周速一定制御 (Vc) = 300 m/min.、速度制限 = 3000 rpm、回転方向は左回り、冷却液オン。
N50 DIAMON	; X 軸が直径でプログラム指令されます。
N60 G54 G18 G0 X82 Z0.2	; ゼロオフセットと作業平面を呼び出し、開始位置へアプローチします。
...	

フライス加工のプログラムヘッダー

次の例は、フライス加工の NC プログラムヘッダーの一般的な構成を示します。

プログラムコード	コメント
N10 T="SF12"	; 代替方法:T123
N20 M6	; 工具交換を起動します
N30 D1	; 工具刃先データセットを有効にします。
N40 G54 G17	; ゼロオフセットと作業平面
N50 G0 X0 Y0 Z2 S2000 M3 M8	; ワークへアプローチ、主軸と冷却液をオン
...	

工具オリエンテーション/座標変換を使用している場合は、プログラムの先頭で、まだ有効な座標変換をすべて解除してください。

プログラムコード	コメント
N10 CYCLE800()	; 旋回平面のリセット
N20 TRAFOOF	; TRAORI、TRANSMIT、TRACYL、...のリセット
...	

3.3.4 プログラム例

3.3.4.1 例 1: 最初のプログラミング手順

プログラム例 1 を使用して、最初のプログラミング手順の実行とテストをおこないます。

処理

1. 新しいパートプログラム(名称)の作成
2. パートプログラムの編集
3. パートプログラムの選択
4. シングルブロック起動
5. パートプログラムを起動します。

注記

プログラムが機械で実行されるように、マシンデータを適切に設定してください(→ 工作機械メーカーのデータを参照してください)。

注記

プログラムのテスト時にアラームが発生した場合は、リセットしてください。

NC プログラム

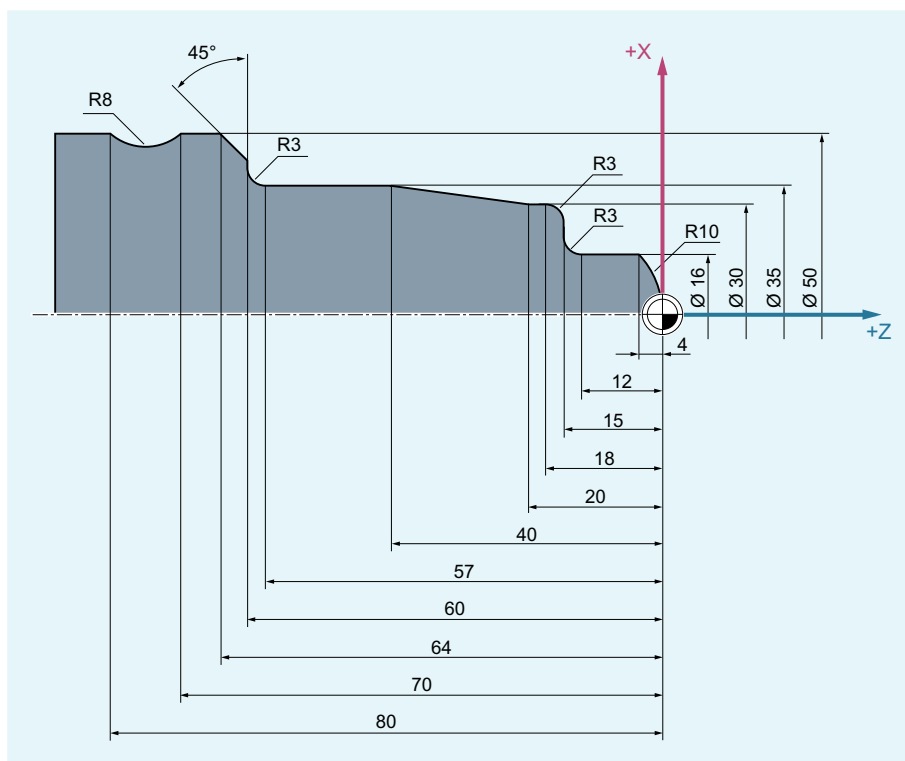
プログラムコード	コメント
N10 MSG("THIS IS MY NC プログラム")	; メッセージ「THIS IS MY NC プログラム」がアラーム行に表示されます。
N20 F200 S900 T1 D2 M3	; 送り速度、主軸、工具、工具オフセット、主軸は右回り。
N30 G0 X100 Y100	; 早送りでアプローチ位置へ移動。
N40 G1 X150	; 送り速度を使用した長方形指令、X の直線指令。
N50 Y120	; Y の直線指令。
N60 X100	; X の直線指令。
N70 Y100	; Y の直線指令。
N80 G0 X0 Y0	; 早送りで後退。
N100 M30	; ブロック終了。

3.3.4.2 例 2: 旋削の NC プログラム

プログラム例 2 は、旋盤のワークの加工用に作成されています。これには、半径指定と工具径補正を含みます。

3.3 NC プログラムの作成

ワークの外形寸法図



NC プログラム

プログラムコード	コメント
N5 G0 G53 X280 Z380 D0	; 起点。
N10 TRANS X0 Z250	; ゼロオフセット。
N15 LIMS=4000	; 速度制限 (G96)。
N20 G96 S250 M3	; 周速一定制御を選択します。
N25 G90 T1 D1 M8	; 工具とオフセットを選択します。
N30 G0 G42 X-1.5 Z1	; 工具にノーズ R 補正を設定します。
N35 G1 X0 Z0 F0.25	
N40 G3 X16 Z-4 I0 K-10	; 半径 10 の円弧を旋削します。
N45 G1 Z-12	
N50 G2 X22 Z-15 CR=3	; 半径 3 の円弧を旋削します。
N55 G1 X24	
N60 G3 X30 Z-18 I0 K-3	; 半径 3 の円弧を旋削します。
N65 G1 Z-20	
N70 X35 Z-40	
N75 Z-57	
N80 G2 X41 Z-60 CR=3	; 半径 3 の円弧を旋削します。
N85 G1 X46	

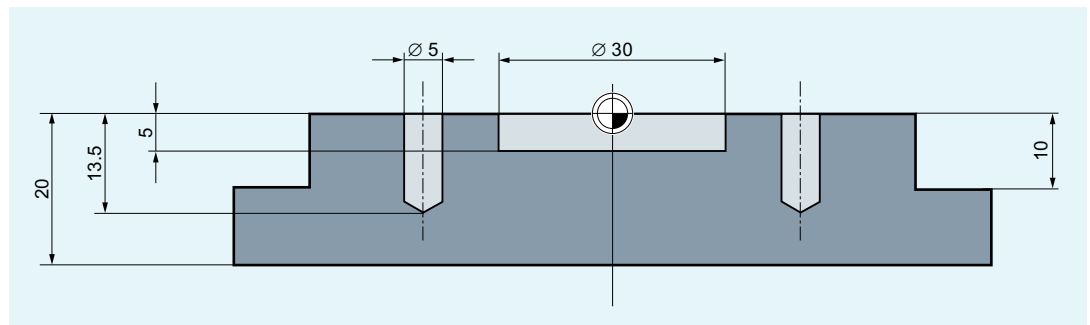
プログラムコード	コメント
N90 X52 Z-63	
N95 G0 G40 G97 X100 Z50 M9	; ノーズ R 補正を選択解除し、工具交換ロケーションへアプローチします。
N100 T2 D2	; 工具を呼び出してオフセットを選択します。
N105 G96 S210 M3	; 周速一定制御を選択します。
N110 G0 G42 X50 Z-60 M8	; 工具にノーズ R 補正を設定します。
N115 G1 Z-70 F0.12	; 直径 50 で旋削します。
N120 G2 X50 Z-80 I6.245 K-5	; 半径 8 の円弧を旋削します。
N125 G0 G40 X100 Z50 M9	; 工具を後退させてノーズ R 補正を選択解除します。
N130 G0 G53 X280 Z380 D0 M5	; 工具交換ロケーションへアプローチします。
N135 M30	; プログラム終了

3.3.4.3 例 3: フライス加工の NC プログラム

プログラム例 3 は、立フライス盤のワークの加工用に作成されています。これには、平面と側面のフライス加工の他に穴あけも含まれます。

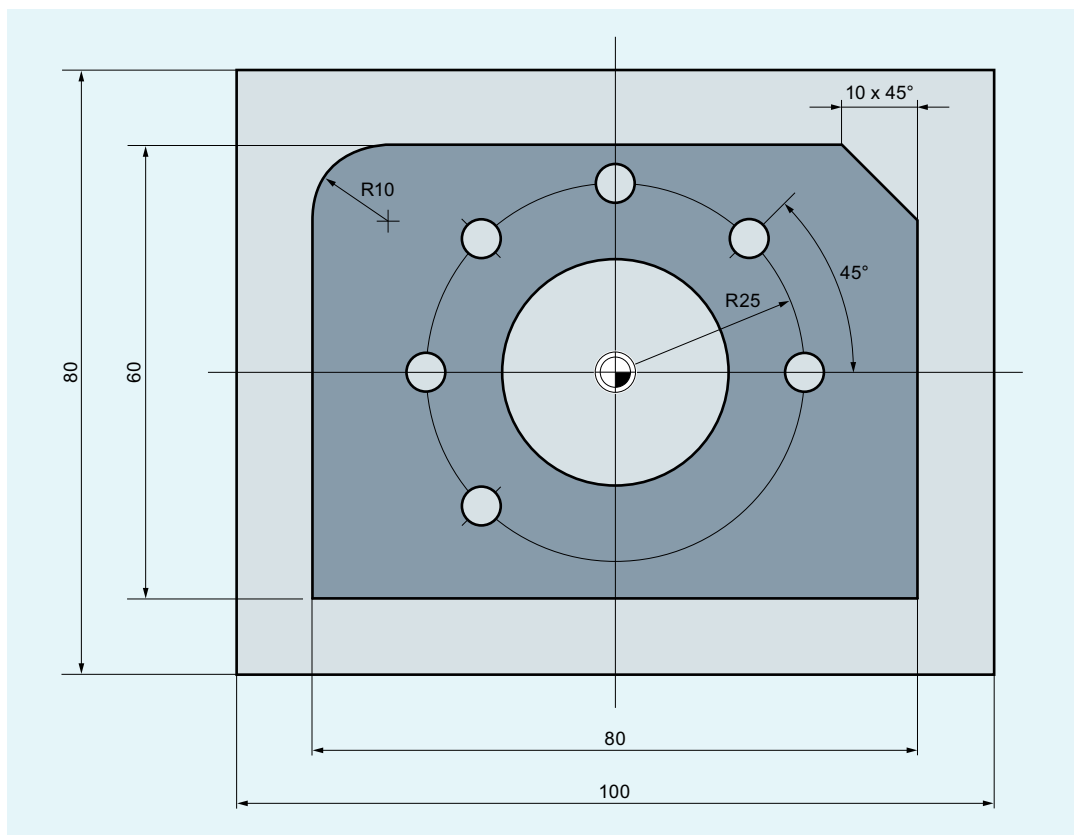
ワークの外形寸法図

側面図



3.3 NC プログラムの作成

平面図



NC プログラム

プログラムコード	コメント
N10 T="PF60"	; PF60 という名称の工具の事前選択。
N20 M6	; 工具を主軸に装着。
N30 S2000 M3 M8	; 回転速度、回転方向、冷却液オン。
N40 G90 G64 G54 G17 G0 X-72 Y-72	; ジオメトリの初期設定と起点へアプローチ。
N50 G0 Z2	; Z 軸との安全間隔
N60 G450 CFTCP	; 動作中の G41/G42 の働き。
N70 G1 Z-10 F3000	; 送り速度 = 3000 mm/min のフライス工具で深さ方向を加工。
N80 G1 G41 X-40	; フライス工具の工具径補正の適用。
N90 G1 X-40 Y30 RND=10 F1200	; 送り速度 = 1200 mm/min で輪郭へ移動。
N100 G1 X40 Y30 CHR=10	
N110 G1 X40 Y-30	
N120 G1 X-41 Y-30	
N130 G1 G40 Y-72 F3000	; フライス工具の工具径補正を選択解除。
N140 G0 Z200 M5 M9	; フライス工具の後退、主軸と冷却液をオフ。
N150 T="SF10"	; SF10 という名称の工具の事前選択。

プログラムコード	コメント
N160 M6	; 工具を主軸に装着。
N170 S2800 M3 M8	; 回転速度、回転方向、冷却液オン。
N180 G90 G64 G54 G17 G0 X0 Y0	; ジオメトリの初期設定と起点へアプローチ。
N190 G0 Z2	
N200 POCKET4(2,0,1,-5,15,0,0,0,0,800,1300,0,21,5,,2,0.5)	; ポケットミリングサイクルの呼び出し。
N210 G0 Z200 M5 M9	; フライス工具の後退、主軸と冷却液をオフ。
N220 T="ZB6"	; 6 mm のセンタ穴ドリルの呼び出し。
N230 M6	
N240 S5000 M3 M8	
N250 G90 G60 G54 G17 X25 Y0	; 正確な位置決め用のイグザクトストップ指令 G60。
N260 G0 Z2	
N270 MCALL CYCLE82(2,0,1,-2.6,,0)	; 穴あけサイクルのモーダル呼び出し。
N280 POSITION:	; 繰り返しのジャンプマーク。
N290 HOLES2(0,0,25,0,45,6)	; 穴あけの位置決めパターン。
N300 ENDLABEL:	; 繰り返しの終了区切り文字。
N310 MCALL	; モデル呼び出しのリセット。
N320 G0 Z200 M5 M9	
N330 T="SPB5"	; 5 mm 径のドリルの呼び出し。
N340 M6	
N350 S2600 M3 M8	
N360 G90 G60 G54 G17 X25 Y0	
N370 MCALL CYCLE82(2,0,1,-13.5,,0)	; 穴あけサイクルのモーダル呼び出し。
N380 REPEAT POSITION	; センタリングから位置記述の繰り返し。
N390 MCALL	; 穴あけサイクルのリセット。
N400 G0 Z200 M5 M9	
N410 M30	; プログラム終了。

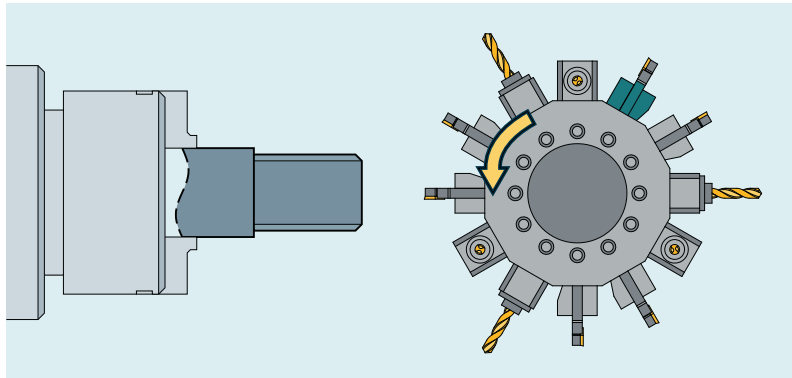
3.4 工具交換

工具交換方法

注記

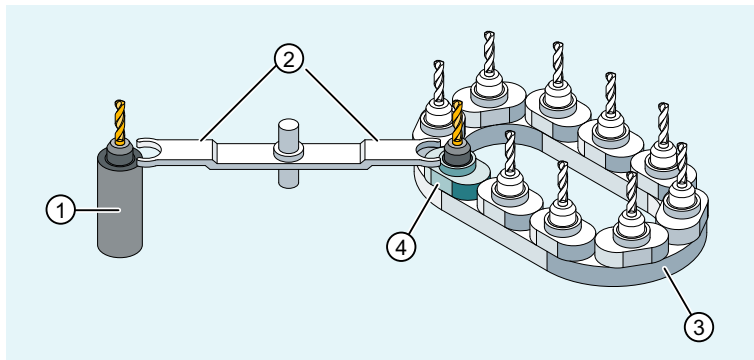
工具交換メカニズムのタイプは、コミッショニング時にマシン OEM によって指定されます。

工具タレット付き旋盤の工具交換



旋盤のタレットマガジンでは、工具交換、つまり、工具の検索と装着はT命令によってのみ呼び出されます。

チェーン方式、ロータリーテーブル方式、プレーン方式マガジンの工作機械の工具変換



- ① 主軸
- ② グリッパ
- ③ マガジン(ここでは:チェーンマガジン)
- ④ 主軸の交換位置

チェーン方式、ロータリーテーブル方式、プレーン方式の各マガジンでは通常、工具交換を2段階でおこないます。

1. マガジンの工具はT命令で探します。
2. その後、M命令で工具を主軸に装着します。

工具管理が有効/無効時の工具交換のプログラミング

工具管理が有効/無効時の工具交換のプログラミングは、工具管理が解除時の工具交換のプログラミングとは異なります。そのため、両方のタイプが説明されています。

→ 工具管理機能が有効時の工具交換 (ページ 69)

→ 工具管理機能が解除時の工具交換 (ページ 73)

作業平面のプログラミング

工具交換に適切な加工平面 (ページ 40) がプログラムされている必要があります (初期状態: G18)。これにより、工具長補正が正しい軸に割り当てられます。

工具オフセットの適用

工具交換により、D 番号 (ページ 90) に保存されている工具オフセットの値が適用されます。

3.4.1 工具管理機能が有効時の工具交換

工具管理機能

「工具管理機能」により、工具がいつでも正しいロケーションにあり、工具に割り当てられたデータが最新であることが保証されます。また、迅速な工具交換を可能にし、工具の寿命監視によって工具破損を防いだり、予備工具の考慮によって機械のダウンタイムの発生を防ぎます。

工具名称

有効な工具管理機能がある工作機械では、工具を明確に識別するために、名称と番号を割り当ててください (「Drill」、「3」など)。

こうすると、次のように工具名称を使用して工具を呼び出せます。

```
T="Drill"
```

注記

工具名称には、どの特殊文字も含めることはできません。

3.4.1.1 工具管理機能が有効時の機能 T=ロケーションでの工具交換

機能 T=ロケーションがプログラムされている場合、工具交換が直接行われます。

3.4 工具交換

用途

円形マガジン搭載の旋盤です。

構文

工具の選択

T=<番号>

T=<名称>

T<n>=<番号>

T<n>=<名称>

工具の選択解除

T0

意味

T=:	工具交換がある工具選択と工具オフセットを適用するためのアドレス以下の値を割り当てることができます。	
	<番号>:	マガジンロケーション番号
	<名称>:	工具名称 注: 工具名称のプログラム指令時には、正しい表記(大文字/小文字)を使用してください。
<n>:	アドレス拡張機能としての主軸番号 注: 機械の構成によっては、アドレス拡張子として主軸番号をプログラム指令することがあります(工作機械メーカーの仕様書を参照してください)。	
T0:	工具の選択解除(マガジンロケーションが空いている)	

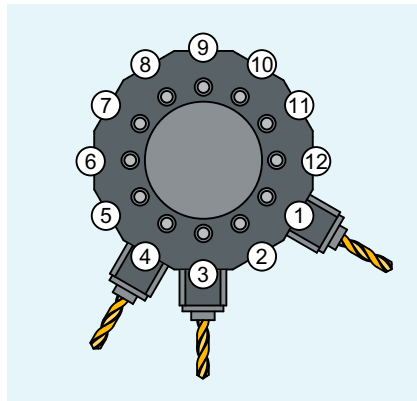
注記

工具マガジンの選択したマガジンロケーションが空いている場合、命令は T0 として機能します。空いているマガジンロケーションを選択することで、空きロケーションを割り出すことができます。

例

円形マガジンには、次のように工具が割り当てられた 1～12 のロケーションがあります。

ロケーション	工具	内部 T 番号	ステータス
1	Drill、予備工具番号= 1	T15	無効
2	空き		
3	Drill、予備工具番号= 2	T10	有効
4	Drill、予備工具番号= 3	T1	有効
5 ... 12	空き		



- ① ... マガジン/ロケーション番号
⑫

NC プログラムでは、次の工具呼び出しがプログラム指令されています。

N10 T=1

呼び出しは次のように処理されます。

1. マガジンロケーション 1 が評価されて、工具識別子が特定されます。
2. 工具管理機能により、この工具がブロックされているために使用できないことが認識されます。
3. T="drill"の工具検索が、次の検索方式指令に従って開始されます。
「動作中の工具を見つける。見つからない場合は、次に大きい予備工具番号を選択する。」
4. 使用可能な次の工具が見つかります。
「Drill」、予備工具番号 3 (マガジンロケーション 4)
これにより、工具選択処理が完了し、工具交換が開始されます。

注記

「グループで最初に使用可能な工具を選択する」という検索方式が用いられている場合は、グループの順序は定義されません。この場合、**T10 または T1** が選択されます。

「グループで「動作中」状態の最初の工具を使用」という検索方式を適用した場合は、**T1** が装着されます。

3.4.1.2 工具管理機能が有効時の M6 での工具交換

工具は、T 命令がプログラム指令されると、選択されます。工具は、M6 によってのみ有効になります(工具オフセットを含みます)。

用途

チェーン方式、ロータリーテーブル方式、プレーン方式マガジンの工作機械。

構文

工具の選択
T=<番号>
T=<名称>
T<n>=<番号>
T<n>=<名称>

工具交換
M6

工具の選択解除
T0

意味

T=:	工具選択アドレス	
	以下の値を割り当てることができます。	
	<番号>:	マガジンロケーション番号
	<名称>:	工具名称
		注: 工具名称のプログラム指令時には、正しい表記(大文字/小文字)を使用してください。

<n>:	アドレス拡張機能としての主軸番号 注: 機械の構成によっては、アドレス拡張子として主軸番号をプログラム指令することがあります(工作機械メーカーの仕様書を参照してください)。
M6:	工具交換のための M 機能(DIN 66025 準拠) M6 により、選択された工具(T...)と工具オフセット(D...)が有効になります。
T0:	工具の選択解除(マガジンロケーションが空いている)

例

プログラムコード	コメント
N10 T=1 M6	; マガジンロケーション 1 からの工具の装着。
N20 D1	; 工具長補正の選択。
N30 G1 X10 ...	; 工具 T=1 による加工。
...	
N70 T="Drill"	; 「Drill」という名称の工具の事前選択。
N80 ...	; 工具 T=1 による加工。
...	
N100 M6	; 工具 T="Drill"の装着
N140 D1 G1 X10 ...	; 工具 T="Drill"による加工。
...	

3.4.2 工具管理機能が解除時の工具交換

3.4.2.1 工具管理機能が解除時の T 番号による工具交換

T 番号がプログラムされている場合、工具交換が直接行われます。

用途

円形マガジン搭載の旋盤です。

構文

工具の選択
T<番号>
T=<番号>
T<n>=<番号>

3.4 工具交換

工具の選択解除

T0
T0=<番号>

意味

T:	工具交換がある工具選択と工具オフセットを適用するためのアドレス		
<番号>:	工具番号		
	値の範囲:	0 ... 32000	
<n>:	アドレス拡張機能としての主軸番号 注: 機械の構成によっては、アドレス拡張子として主軸番号をプログラム指令することがあります(工作機械メーカーの仕様書を参照してください)。		
T0:	有効な工具の選択解除		

例

プログラムコード	コメント
N10, T1, D1	; 工具 T1 の装着と工具オフセット D1 の適用。
...	
N70 T0	; 工具 T1 を選択解除します。
...	

3.4.2.2 工具管理機能が解除時の M6 での工具交換

工具は、T 命令がプログラム指令されると、選択されます。工具は、M6 によってのみ有効になります(工具オフセットを含みます)。

用途

チェーン方式、ロータリーテーブル方式、プレーン方式マガジンの工作機械。

構文

工具の選択
T<番号>
T=<番号>
T<n>=<番号>

工具交換

M6

工具の選択解除

T0

T0=<番号>

意味

T:	工具選択アドレス	
<番号>:	工具番号	
	値の範囲:	0 ... 32000
<n>:	アドレス拡張機能としての主軸番号 注: 機械の構成によっては、アドレス拡張子として主軸番号をプログラム指令することがあります(工作機械メーカーの仕様書を参照してください)。	
M6:	工具交換のための M 機能(DIN 66025 準拠) M6 により、選択された工具(T...)と工具オフセット(D...)が有効になります。	
T0:	有効な工具の選択解除	

例

プログラムコード	コメント
N10 T1 M6	; 工具 T1 の装着。
N20 D1	; 工具長補正の選択。
N30 G1 X10 ...	; T1 による加工。
...	
N70 T5	; 工具 T5 の事前選択。
N80 ...	; T1 による加工。
...	
N100 M6	; 工具 T5 の装着。
N110 D1 G1 X10 ...	; 工具 T5 による加工
...	

3.4.3 Tプログラミングにエラーがある場合の動作

Tプログラミングにエラーがある場合の動作は、次のように、機械構成によって異なります。

MD22562 TOOL_CHANGE_ERROR_MODE		
ビット	値	意味
7	0	初期設定 Tプログラムにより、NCがT番号を認識するかどうか直ちにチェックされます。認識しない場合は、アラームが発生します。
	1	プログラム指令したT番号は、Dの選択後にのみチェックされます。NCが工具番号を認識しない場合は、Dの選択時にアラームが発生します。 たとえば、工具のプログラム指令が位置決めを目的としていて、工具データは必ずしも存在しない場合(円形マガジン)などは、これが望ましい動作です。

3.5 工具補正

3.5.1 プログラム指令輪郭および工具軌跡

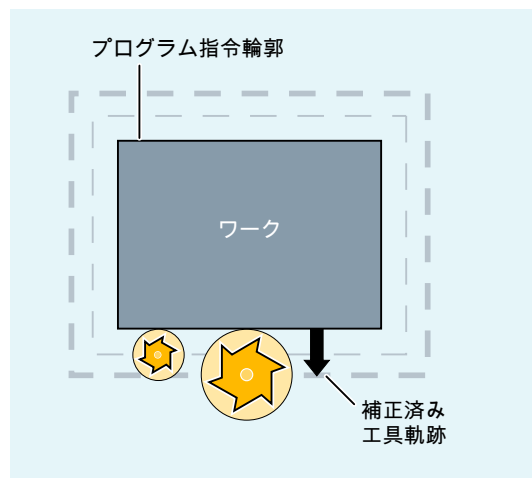
ワークの寸法は直接プログラム指令されます(加工図面に従う等)。したがって、フライス工具径、旋削工具(左回り/右回りの旋削工具)の刃先位置、および工具長などの工具データは、プログラムの作成時に考慮する必要はありません。

制御装置による移動軌跡の補正

ワークの加工中は、プログラム指令輪郭をどの工具でも作成できるように、工具形状に従って工具軌跡が制御されます。

制御装置が工具軌跡を計算できるよう、制御装置の工具補正メモリに工具データを入力してください。必要工具(T...)と必要オフセットデータ(D...)のみがNCプログラムで呼び出されます。

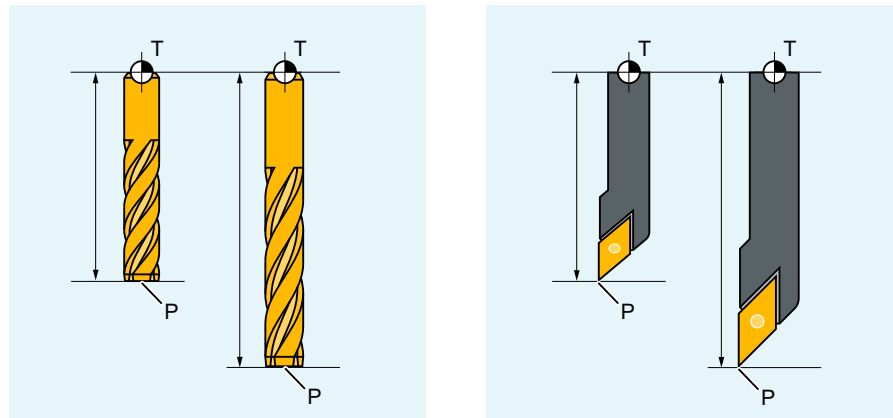
プログラム処理中に、制御装置が必要オフセットデータを工具オフセットメモリから読み出して、各工具の工具軌跡を補正します。



3.5.2 工具長補正

工具長補正により、使用される工具間の長さの差が補正されます。

工具長とは、工具ホルダの基準点から工具先端までの距離です。



T 工具ホルダレファレンス点

P 工具先端

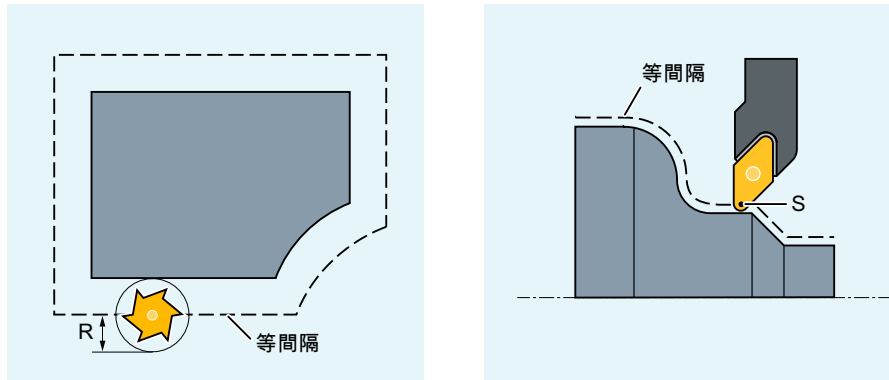
この長さが計測され、定義可能な摩耗値とともに、制御装置の工具補正メモリに入力されます。制御装置はこのデータから、切り込み方向への移動動作を計算します。

注記

工具長の補正值は、空間の工具の向きに応じて変わります。

3.5.3 工具径補正

輪郭と工具軌跡は同じではありません。フライス工具または刃先中心点は、輪郭から等距離の工具の半径に対応した軌跡に沿って移動します(工具中心点軌跡)。これを行うために、プログラムを実行中に、制御装置はプログラムされた工具中心点軌跡を－有効な工具の工具半径に基づいて(工具オフセットメモリ)－工具の刃先がプログラムされた輪郭に沿って正確に移動できるようシフトします。



- R 工具半径
S 刃先の中心点

工具径補正は章「工具径補正 (ページ 280)」で詳しく説明されています。

下記も参照

2 1/2 次元工具オフセット(CUT2D、CUT2DD、CUT2DF、CUT2DFD) (ページ 320)

3.5.4 工具補正メモリ

次のデータが各工具刃先について、制御装置の工具オフセットメモリで利用できるようにしてください。

- 工具タイプ
- 刃先位置
- 工具形状変数(長さ、半径)

これらのデータは、工具パラメータとして入力されます(最大 25)。工具に必要なパラメータは、工具タイプによって違います。不必要な工具パラメータはすべて「0」に設定してください(システムの初期設定に対応します)。

注記

値がオフセットメモリに入力されると、その値は各工具呼び出し処理に含まれます。

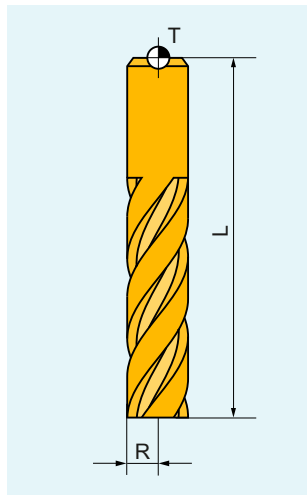
工具タイプ

工具タイプ(ドリル、フライス削り、または旋削工具)によって、必要な形状データとその計算方法が特定されます。

刃先位置

刃先位置は、刃先中心点を基準にした工具先端の位置を記述します。旋盤工具(工具タイプ 5xx) (ページ 86)の工具径補正の計算には、刃先位置と共に刃先半径が必要です。

工具形状変数(長さ、半径)



T 工具ホルダレファレンス点

R 工具半径

L 工具長

工具形状変数は、複数の成分(形状、摩耗)から成ります。制御装置は、この成分の結果として得られる寸法(長さ 1 の全長、半径の合計など)を計算します。オフセットメモリが有効になると、関連する全部の寸法が有効になります。

3.5 工具補正

各軸のこれらの値の計算方法は、工具タイプと現在の平面(G17/G18/G19)により特定されます。

3.5.5 工具タイプ

3.5.5.1 工具タイプ番号と工具グループ

各工具タイプには、3桁の固有の番号が割り当てられています。最初の桁(100の位)を使用して、次の用途または工具グループの1つへの工具割り当てを行います。

工具タイプ	工具グループ
1xy	フライス工具 (ページ 80)
2xy	ドリル工具 (ページ 83)
3xy	予備
4xy	研削工具 (ページ 84)
5xy	旋削工具 (ページ 86)
6xy	予備
7xy	特殊工具 (ページ 88)

3.5.5.2 フライス工具

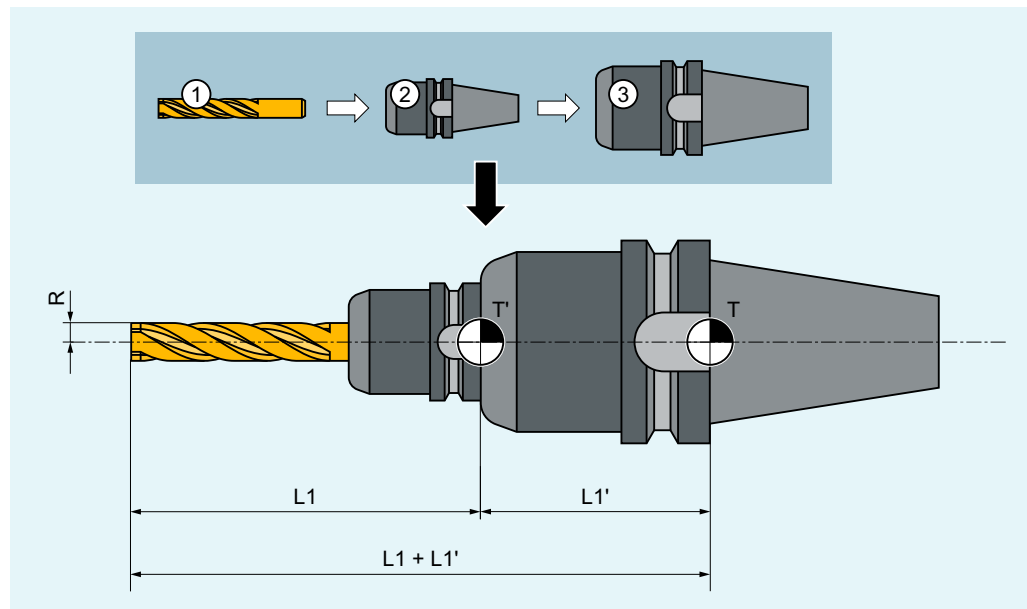
「フライス工具」グループには、次の工具タイプがあります。

100	CLDATA (カッターロケーションデータ)に準拠したフライス工具
110	ボールエンドミル(テーパ)
111	ボールエンドミル(ストレート)
120	丸コーナなしのエンドミルカッタ
121	ラジラスエンドミル
130	丸コーナなしのアングルヘッドカッタ
131	丸コーナ角度付きヘッドフライス
140	正面フライス
145	ねじ切り工具
150	サイドカッタ
151	のこぎり

155	テーパ刃エンドミル(丸コーナなし)
156	テーパラジアスエンドミル
157	テーパボールエンドミル
160	ドリル付きタップ

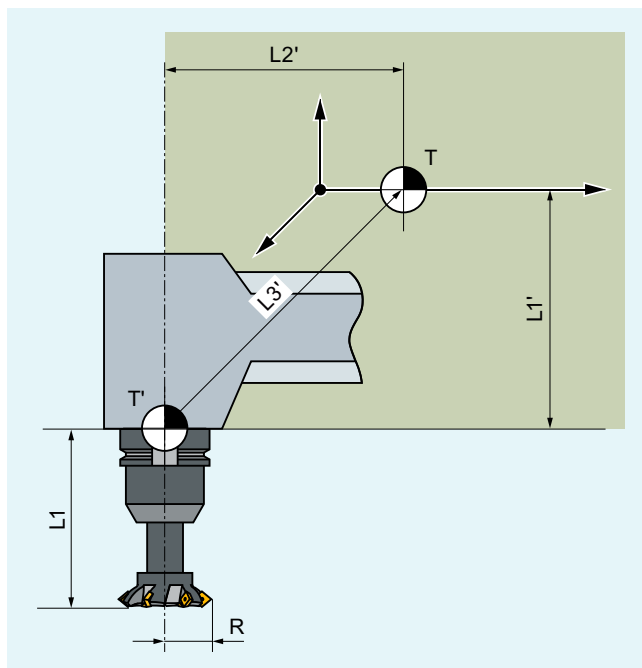
工具パラメータ

以下の図に、補正メモリに入力するフライス工具パラメータの概要を示します。



- ① 工具
- ② 工具ホルダ
- ③ 工具アダプタ
- T アダプタレファレンス点(実装された工具の場合 = 工具ホルダレファレンス点)
- T' 工具ホルダレファレンス点
- L1 ジオメトリ - 長さ 1
- L1' アダプタ寸法 - 長さ 1
- L1 + L1' 全体長 L1
- R 半径

工具パラメータ	意味
\$TC_DP1	工具タイプ 1xy
\$TC_DP3	ジオメトリ - 長さ 1
\$TC_DP6	ジオメトリ - 半径
\$TC_DP21	アダプタ寸法 - 長さ 1
- 要件に対応した摩耗値。 - その他の値は 0 に設定してください。	



T 工具ホルダレファレンス点

T' 工具ホルダレファレンス点

L1 ジオメトリ - 長さ 1

R 工具半径

L1' 基本寸法 - 長さ 1

L2' 基本寸法 - 長さ 2

L3' 基本寸法 - 長さ 3

工具パラメータ	意味
\$TC_DP1	工具タイプ
\$TC_DP3	ジオメトリ - 長さ 1

工具パラメータ	意味
\$TC_DP6	ジオメトリ - 半径
\$TC_DP21	基本寸法 - 長さ 1
\$TC_DP22	基本寸法 - 長さ 2
\$TC_DP23	基本寸法 - 長さ 3
- 要件に対応した摩耗値。 - その他の値は 0 に設定してください。	

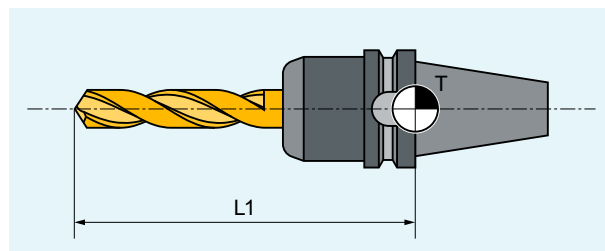
3.5.5.3 ドリル工具

「ドリル工具」グループには、次の工具タイプがあります。

No.	工具タイプ
200	ドリル
205	ドリル
210	ボーリングバー
220	センタ穴ドリル
230	皿取りドリル
231	座ぐりフライス
240	標準ねじタップ
241	精密ねじタップ
242	ウィットねじタップ
250	リーマ

工具パラメータ

以下の図に、補正メモリに入力するドリル工具パラメータの概要を示します。



T 工具ホルダレファレンス点

L1 長さ 1

工具パラメータ	意味
\$TC_DP1	工具タイプ
\$TC_DP3	ジオメトリ - 長さ 1
- 要件に対応した摩耗値。 - その他の値は 0 に設定してください。	

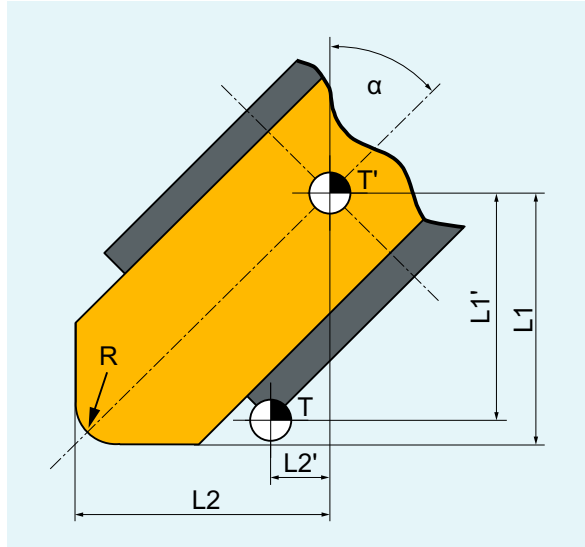
3.5.5.4 研削工具

「研削工具」グループには、次の工具タイプがあります。

400	平面研削砥石
401	平面研削砥石(監視機能あり)
402	平面研削砥石(監視機能なし、基本寸法なし)(TOOLMAN)
403	研削砥石周速度 GWPS 用平面研削砥石(監視機能あり、基本寸法なし)
410	正面研削砥石
411	正面研削砥石(TOOLMAN) (監視機能あり)
412	正面研削砥石(TOOLMAN) (監視機能なし)
413	研削砥石周速度 GWPS 用正面研削砥石(監視機能あり、基本寸法なし)
490	ドレッサ

工具パラメータ

以下の図に、補正メモリに入力する研削工具パラメータの概要を示します。



- T 工具ホルダレファレンス点
- T' 工具ホルダレファレンス点
- L1 ジオメトリ - 長さ 1
- L1' 基本寸法 - 長さ 1
- L2 ジオメトリ - 長さ 2
- L2' 基本寸法 - 長さ 2
- R 半径
- α 傾斜といしの角度

刃先用パラメータ	意味
\$TC_DP1	工具タイプ 4xy
\$TC_DP2	刃先位置
\$TC_DP3	ジオメトリ長さ 1
\$TC_DP4	ジオメトリ長さ 2
\$TC_DP6	半径
\$TC_DP21	基本寸法長さ 1
\$TC_DP22	基本寸法長さ 2
- 要件に対応した摩耗値。 - その他の値は 0 に設定。	

工具用パラメータ	意味
\$TC_TPG1	主軸番号
\$TC_TPG2	連動規則 ¹⁾
\$TC_TPG3	最小といし半径
\$TC_TPG4	最小といし幅
\$TC_TPG5	現在のといし幅
\$TC_TPG6	最高速度
\$TC_TPG7	最大外周速度
\$TC_TPG8	傾斜といしの角度
\$TC_TPG9	半径計算用パラメータ番号
\$TC_TPG_DRSPATH	目立てプログラムへのディレクトリパス
\$TC_TPG_DRSPROG	目立てプログラム名称

- ¹⁾ 工具形状長補正、摩耗、および基本寸法では、左右両方のノーズ R 補正を連動させることができます。つまり、左刃先の工具長補正が変更されると、右刃先の値が自動的に入力されます。逆の場合も同様です。

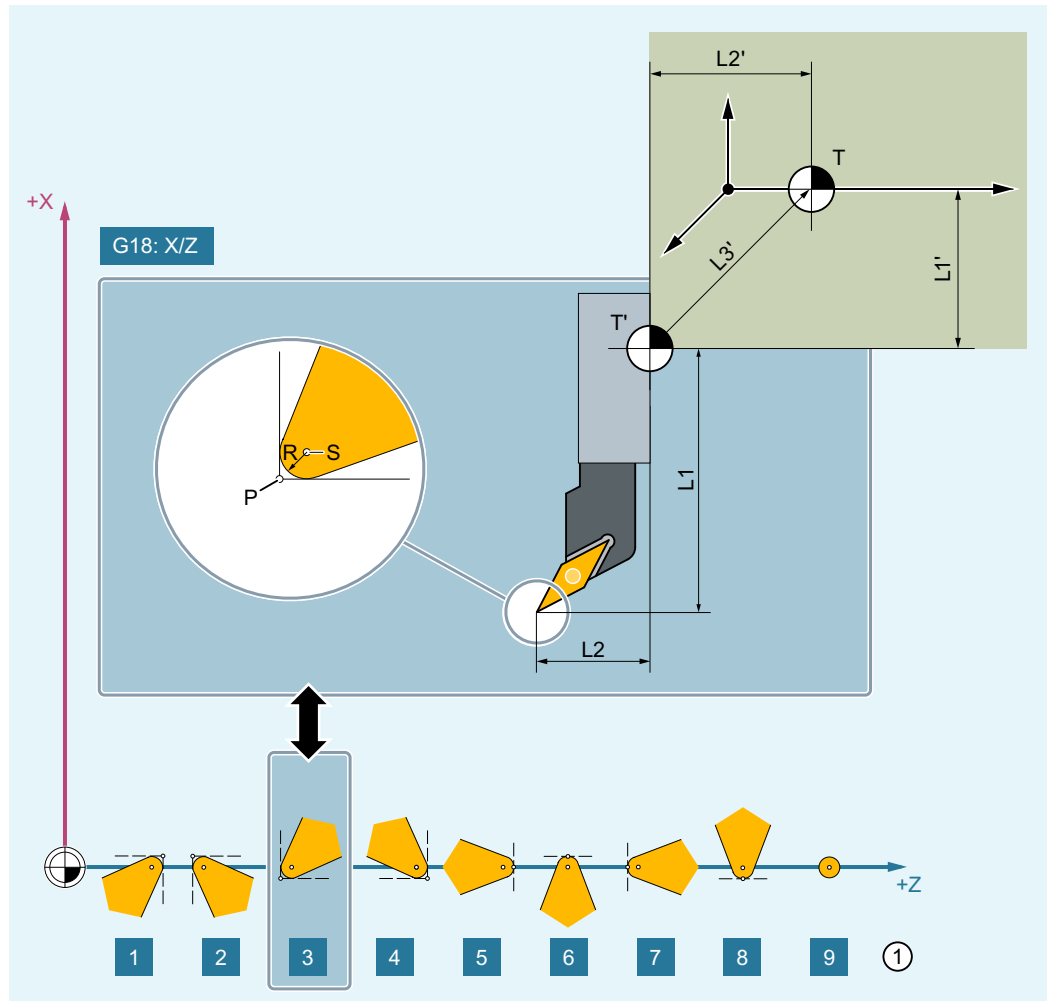
3.5.5.5 旋削工具

「旋削工具」グループには、次の工具タイプがあります。

500	荒削り工具
510	仕上げ工具
520	プランジカッタ
530	突っ切りバイト
540	ねじ切りバイト
550	丸こまバイト/総形バイト(TOOLMAN)
560	回転ドリル(ECOCUT)
580	プローブ(刃先位置パラメータ付き)

工具パラメータ

以下の図に、補正メモリに入力する旋削工具パラメータの概要を示します。



① ターニングセンタの背面での加工の刃先位置(1 ... 9)

P 工具先端

S 刃先の中心点

R 刃先半径

T 工具ホルダレファレンス点

T' 工具ホルダレファレンス点

L1 ジオメトリ - 長さ 1

L2 ジオメトリ - 長さ 2

L1' 基本寸法 - 長さ 1

3.5 工具補正

L2' 基本寸法 - 長さ 2

L3' 基本寸法 - 長さ 3

工具パラメータ	意味
\$TC_DP1	工具タイプ
\$TC_DP2	刃先位置
\$TC_DP3	ジオメトリ - 長さ 1
\$TC_DP4	ジオメトリ - 長さ 2
\$TC_DP6	ジオメトリ - 半径(刃先半径)
\$TC_DP21	基本寸法 - 長さ 1
\$TC_DP22	基本寸法 - 長さ 2
\$TC_DP23	基本寸法 - 長さ 3
- 要件に対応した摩耗値。 - その他の値は 0 に設定。	

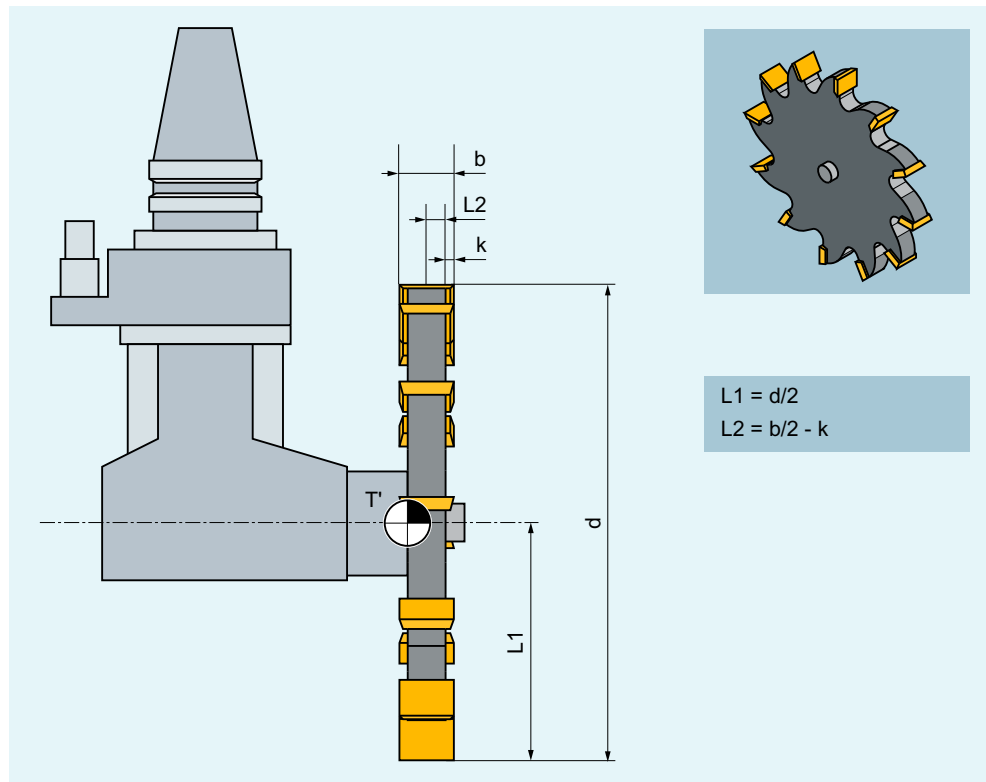
3.5.5.6 特殊工具

「特殊工具」グループには、次の工具タイプがあります。

700	メタルソー
710	3次元プローブ
711	エッジプローブ
712	一方向プローブ
713	L型プローブ
714	スター型プローブ
725	校正工具
730	停止
731	主軸スリーブ
732	エンドサポート

工具パラメータ

以下の図に、補正メモリに入力する「メタルソー」工具タイプの工具パラメータの概要を示します。



T' 工具ホルダレファレンス点

L1 ジオメトリ - 長さ 1

L2 ジオメトリ - 長さ 2

d 直径

b 溝幅

k 突起

工具パラメータ	意味
\$TC_DP1	工具タイプ
\$TC_DP3	ジオメトリ - 長さ 1
\$TC_DP4	ジオメトリ - 長さ 2
\$TC_DP6	直径
\$TC_DP7	溝幅

工具パラメータ	意味
\$TC_DP8	突起
\$TC_DP21	基本寸法長さ 1
\$TC_DP22	基本寸法長さ 2
\$TC_DP23	基本寸法長さ 3
- 要件に対応した摩耗値。 - その他の値は 0 に設定してください。	

3.5.6 工具オフセットの有効化/無効化(D、D0):

(有効な工具管理 12 が動作中の)工具刃先 1～8 には、異なる工具オフセットデータブロックを割り当てることができます(溝切り工具の左右の刃先でオフセット値が異なる場合など)。

特殊刃先のオフセットデータ(工具長補正のデータを含む)を適用するには、D 番号を呼び出します。D0 をプログラム指令すると、工具オフセットが無効になります。

工具径補正を有効にするには、G41/G42 も使用してください。

注記

D 番号をプログラム指令すると、工具長補正が直ちに有効になります。D 番号をプログラム指令しない場合、工具交換時にマシンデータで定義した初期設定が有効になります(→ 工作機械メーカーの仕様書を参照してください)。

構文

```

D<番号>
X...Y...Z...
G41/G42 X...Y...Z...
G40
D0

```

意味

D:	動作中の工具のオフセットデータブロックを適用するアドレス 工具長補正は、関連する工具長補正軸の最初のプログラム指令移動で、適用されます。 補足: 工具交換で工具刃先の自動適用が設定されている場合は、D をプログラミングせずに工具長補正を適用することもできます(→工作機械メーカーの仕様書を参照してください)。	
<番号>:	有効にする工具のオフセットデータブロックは、D 番号で指定します。 D プログラミングのタイプは、機械の設定により異なります(→工作機械メーカーからの情報を参照してください)。 以下の選択肢があります。 • D 番号 = 刃先番号 各工具 T<番号>(TOOLMAN なし)または T=「名称」(TOOLMAN あり)に、1 から最大 12 までの D 番号を使用できます。これらの D 番号は、工具刃先に直接割り当てられます。各 D 番号(= 刃先番号)には、1 つのオフセットデータセット(\$TC_DPx[<t>,<d>])が割り当てられています。 • D 番号の自由選択 D 番号は、工具刃先番号に自由に割り当てることができます。使用可能な D 番号の上限は、マシンデータで指定されます。	
	値の範囲:	0 ... 32000
D0:	動作中の工具のオフセットデータブロックの無効化	
G41:	輪郭の加工方向の 左側 の工具径補正を適用する命令	
G42:	輪郭の加工方向の 右側 の工具径補正を適用する命令	
G40:	工具半径補正を無効にする命令	

G40/G41/G42 の追加情報は「工具径補正 (ページ 280)」の章に記載されています。

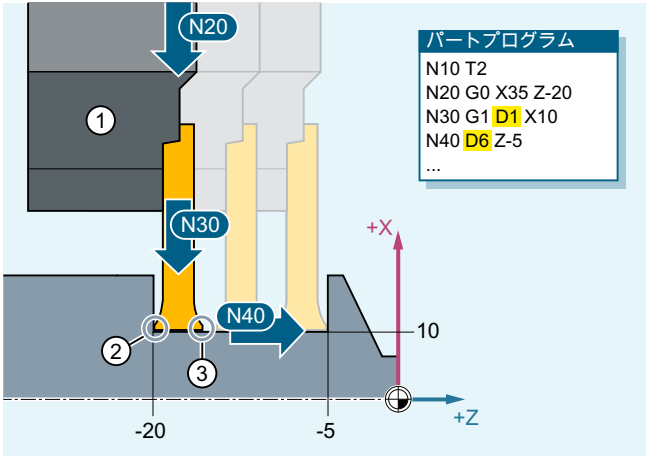
例

例 1:T 命令による工具交換(旋削)

プログラムコード	コメント
N10, T1, D1	; 工具 T1 を装着して、T1 の工具オフセットデータブロック D1 を有効にします。
N11 G0 X...Z...	; 工具長補正が適用されます。
N50, T4, D2	; 工具 T4 を装着して、T4 の工具オフセットデータブロック D2 を有効にします。
...	

プログラムコード	コメント
N70 G0 Z...D1	; 工具 T4 に対して別の刃先 D1 を有効にします。

例 2:溝切り工具の左右の刃先で補正値が異なる場合



- ① プランジ旋削工具(T2)
- ② 刃先 D1
- ③ 刃先 D6

下記も参照

工具径補正 (ページ 78)

詳細情報

工具オフセットデータの変更

標準設定では、工具オフセットデータの変更は、次回の T または D 番号のプログラム指令時にはじめて有効になります。

次のマシンデータを使用すると、入力した工具オフセットデータが直ちに有効になるよう指定できます。

MD9440 \$MM_ACTIVATE_SEL_USER

**警告**

衝突の可能性
MD9440 を設定した場合、パートプログラムの一時停止中に工具オフセットデータの変更された工具オフセットは、パートプログラムの再開時に適用されます。

3.5.7 工具オフセットのマスク(SUPD)

SUPD 命令を使用すると、工具オフセットをブロック単位でマスクできます。

D0 で工具オフセットを解除すると、工具オフセットデータが SUPD で保持されるという利点があります。この結果、D 番号をプログラム指令して工具オフセットデータを再度有効にする必要がありません。

SUPD は、工具長の代わりに原点を演算に含めるパートプログラムブロックなどで、工具長補正をマスクするためにのみ使用します。工具径補正の追加マスクを行うために有効な G41/G42 と使用することはお勧めしません。

構文

```
D<番号>
X...Y...Z...
X...Y...Z...SUPD
...
D0
```

意味

SUPD:	有効なブロックで動作中の工具のオフセットデータブロックを無効にする G 命令	
	有効性:	ノンモーダル

制約事項

- SUPD は直線ブロックでのみ使用されます。
- シンクロナイズドアクションでは使用できません。
- SUPD は、以下の機能と組み合わせて使用できません。
 - 3次元正面削りのための3次元工具径補正(CUT3DFxx)
 - カーブテーブル(CTAB)
- 工具径補正(G41/G42)を有効にした状態でも SUPD を使用することはできますが、お勧めはしません。
それでもこの機能を有効にする場合は、後述のセッティングデータに「0」を設定してください。
SD42480 \$SC_STOP_CUTCOM_STOPRE = 0
こうすることで、有効な G41/G42 によるプログラムの中断を避けることができます。

例

移動時、サブプログラム SUB_SUP の工具長はマスクされます。

パートプログラム

プログラムコード	コメント
...	
N300 \$P_UIFR[1]=CTTRANS(X,1000,Y,400,Z,-120)	
N310 T="BALL_D3"	
N320 M6	
N330 TRAFOOF	
N340 G54 G0 Z49 D1	
N350 G0 X1100 Y500 C0 A0	
N360 SUB_SUP	; Calling the subroutine.
N370 G0 X1200	
N380 M30	

D0 を含むサブプログラム

プログラムコード	コメント
N10 PROC SUB_SUP	
N20 DEF INT NUMBER	
N30 NUMBER=\$P_TOOL	
N40 G0 Z49 D0	; 工具オフセットの選択解除
N50 D=NUMBER	; 工具オフセットの再選択。
N60 RET	

SUPD を含むサブプログラム

プログラムコード	コメント
N10 PROC SUB_SUP	
N40 G0 Z49 SUPD	; 有効なブロックでの工具オフセットのマスク。
N60 RET	

3.5.8 プログラマブル工具オフセット(TOFFL、TOFF、TOFFR、TOFFLR)

TOFFx アドレスを基に、補正メモリに格納された工具オフセットデータを変更せずに、NC プログラムで有効工具長および有効工具半径を変更できます。

これらのプログラム指令オフセットは、プログラムの終了時に再度、解除されます。

構文

工具長オフセット

```
TOFFL=<値>  
TOFFL[1]=<値> TOFFL[2]=<値> TOFFL[3]=<値>  
TOFF[<GeoAx>]=<値>
```

工具長は、3つの工具長成分の全部で同時に変更できます。ただし、TOFFL/TOFFL[1..3]グループの命令、およびTOFF[<ジオメトリ軸>]グループの命令は、1ブロックに同時に指令できません。同様に、TOFFLとTOFFL[1]は、1ブロックで同時に書き込むことができません。

1ブロックに、3つの工具長成分のいずれかがプログラム指令されていない場合は、プログラム指令されていない成分は変更されません。これにより、複数の成分のオフセットをブロックごとに構築することができます。ただし、これは、工具成分がTOFFLのみ、またはTOFFのみを使用して変更されている場合にかぎり、適用されます。プログラミングバージョンをTOFFLからTOFFへ、またはその逆の変更をおこなうと、以前にプログラム指令した工具長オフセットがすべて解除されます(例3を参照してください)。

工具径補正

```
TOFFR=<値>
```

同時の工具長補正と工具半径補正

```
TOFFLR=<値>
```

意味

TOFFL:	有効工具長の補正 TOFFL は、インデックスを使用しても、使用しなくてもプログラム指令できます。	
	TOFFL=...	プログラム指令オフセット値が、補正メモリに格納されている工具長成分 L1 と同じ方向に適用されます。 TOFFL と TOFFL[1]命令は、同じ動作です。
	TOFFL[1]=.. .. TOFFL[2]=.. .. TOFFL[3]=.. ..	プログラム指令オフセット値が、オフセットメモリに格納されている工具長成分 L1 、 L2 、または L3 と同じ方向に適用されます。
	注: 各軸におけるこれらの工具長補正值の計算方法は、工具タイプと現在の作業平面(G17/G18/G19)によって特定されます。	
TOFF:	指定されたジオメトリ軸に平行な成分で工具長を補正 TOFF は工具長成分の向きに適用されます。この成分は、インデックスで指定されたジオメトリ軸に平行な非回転工具(旋回工具ホルダまたは方向座標変換)に対して有効です。 注: フレームは、工具長成分へのプログラム指令値の割り当てには影響しません。つまり、ワーク座標系(WCS)は、工具長成分ではなく、基本工具位置にある工具座標系のジオメトリ軸への割り当てに使用されます。	
<GeoAx>:	ジオメトリ軸の識別子	
TOFFR:	有効工具半径の補正 TOFFR は、プログラム指令オフセット値により 動作中の工具径補正 の有効工具半径を変更します。	
TOFFLR:	成分 L1 の有効工具長 および 有効工具半径の補正 注: 丸コーナありの工具の場合、(タイプ 111、121、131、および 156)、TOFFLR はコーナ半径も補正します。	
<値>:	オフセット値	
	タイプ:	REAL

例

例 1: 正方向の工具長オフセット

動作中の工具は、長さ $L1 = 100 \text{ mm}$ のドリルです。

有効平面は **G17** です。つまり、**Z** 方向のドリル点です。

有効ドリル長は 1 mm だけ長くなります。この工具長オフセットのプログラミングには、次のタイプを使用できます。

- `TOFFL=1`
- `TOFFL[1]=1`
- `TOFF[Z]=1`

例 2: 負方向の工具長オフセット

動作中の工具は、長さ $L1 = 100 \text{ mm}$ のドリルです。

有効平面は **G18** です。つまり、**Y** 方向のドリル点です。

有効ドリル長は 1 mm だけ短縮されます。この工具長オフセットのプログラミングには、次のタイプを使用できます。

- `TOFFL=-1`
- `TOFFL[1]=-1`
- `TOFF[Y]=1`

例 3: プログラミングバージョンを TOFFL から TOFF に変更

動作中の工具はフライス工具です。有効平面は **G17** です。

プログラムコード	コメント
N10 TOFFL[1]=3 TOFFL[3]=5	; 有効オフセット:L1=3、L2=0、L3=5
N20 TOFFL[2]=4	; 有効オフセット:L1=3、L2=4、L3=5
N30 TOFF[Z]=1.3	; 有効オフセット:L1=0、L2=0、L3=1.3

例 4: 平面の変更後のオフセット値の割り当て

プログラムコード	コメント
N10 \$TC_DP1[1,1]=120	
N20 \$TC_DP3[1,1]= 100	; 工具長 L1=100 mm。
N30 T1 D1 G17	
N40 TOFF[Z]=1.0	; Z 方向へのオフセット (G17 の L1 に対応します)。
N50 G0 X0 Y0 Z0	; 機械軸の位置 X0 Y0 Z101
N60 G18 G0 X0 Y0 Z0	; 機械軸の位置 X0 Y100 Z1。
N70 G17	

3.5 工具補正

プログラムコード	コメント
N80 TOFFL=1.0	; L1 方向へのオフセット (G17 の Z に対応します)。
N90 G0 X0 Y0 Z0	; 機械軸の位置 X0 Y0 Z101。
N100 G18 G0 X0 Y0 Z0	; 機械軸の位置 X0 Y101 Z0。

この例では、N60 ブロックの G18 への変更時に、Z 軸の 1 mm のオフセットは保持されます。Y 軸の有効工具長は変更されず、100 mm のままです。

ただし、N100 ブロックでは、G18 への変更時にオフセットは Y 軸で有効です。これは、オフセットがプログラミングで工具長 L1 に割り当てられているため、この工具長成分が、G18 により Y 軸で有効になるためです。

例 5:同時の工具長補正と工具半径補正

a、丸コーナなしのエンドミルカッタ(工具タイプ 120):

プログラムコード	コメント
...	
TOFFLR=0.1	; 有効オフセット: ; 工具長補正 (L1) = 5 工具径補正 = 5
...	

b、ラジアスエンドミル(工具タイプ 121):

プログラムコード	コメント
...	
TOFFLR=0.1	; 有効オフセット: ; 工具長補正 (L1) = 0.1 工具径補正 = 0.1 ; オフセットコーナ半径 = 0.1
...	

詳細情報

工具長オフセット

プログラム指令工具長オフセットは、プログラミングのタイプに応じて、補正メモリに格納された工具長成分 L1、L2、および L3 (TOFFL)またはジオメトリ軸(TOFF)のいずれかに

割り当てられます。プログラム指令オフセットは、平面指定の変更(G17/G18/G19 ↔ G17/G18/G19)に従って処理されます。

- オフセット値が工具長成分に割り当てられている場合は、プログラム指令オフセットが適用される方向が、その割り当てに従って入れ替えられます。
- オフセット値がジオメトリ軸に割り当てられている場合は、平面指定の変更は、座標軸の割り当てには影響しません。

プログラム指令オフセット値を工具長成分に割り当てるときは、次のセッティングデータが使用されます。

SD42940 \$SC_TOOL_LENGTH_CONST (平面の変更による工具長成分の変更)

SD42950 \$SC_TOOL_LENGTH_TYPE (工具タイプに依存しない工具長補正の割り当て)

このセッティングデータが、0 以外の有効値である場合は、それが、G グループ 6 (平面の選択 G17/G18/G19)、または工具データに含まれる工具タイプ(\$TC_DP1[<T 番号>, <D 番号>])より優先されます。つまり、このセッティングデータは、工具長成分 L1～L3 と同様に、オフセットの適用に影響します。

工具径補正

TOFFR アドレスは OFFN (ページ 280)アドレスとほとんど同じ動作です。動作中の円筒補間 (TRACYL)と動作中の溝壁補正に関してのみ、相違点があります。溝壁補正の場合、工具半径は、負符号付きの OFFN ではなく、正符号付きの TOFFR に対応します。

OFFN と TOFFR は、同時に有効にすることができます。通常は、この 2 種類の動作が加算されます(溝壁補正は除きます)。

工具交換

すべてのオフセット値は工具交換(刃先交換)のときに、記憶されます。つまり、新しい工具(新しい刃先)に対しても有効です。

現在のオフセット値を読み出すためのシステム変数

現在有効なオフセットは、次のシステム変数で読み出すことができます。

システム変数	意味
\$P_TOFFL [<n>] 0 ≤ n ≤ 3	NC プログラムの先読み処理で、TOFFL (n = 0 の場合)または TOFFL[1...3] (n = 1、2、3 の場合)の現在のオフセット値を読み込みます。
\$P_TOFF [<GeoAx>]	NC プログラムの先読み処理で、TOFF[GeoAx]の現在のオフセット値を読み込みます。

システム変数	意味
\$P_TOFFR	NC プログラムの先読み処理で、TOFFR の現在のオフセット値を読み込みます。
\$P_TOFFCR	NC プログラムの先読み処理で、コーナ半径の現在のオフセット値を読み込みます。
\$AC_TOFFL[<n>] $0 \leq n \leq 3$	メインラン処理(シンクロナイズドアクション)で、TOFFL (n=0 の場合)または TOFFL[1...3] (n=1、2、3 の場合)の現在のオフセット値を読み込みます。
\$AC_TOFF[<GeoAx>]	メインラン処理(シンクロナイズドアクション)で、TOFF[GeoAx]の現在のオフセット値を読み込みます。
\$AC_TOFFR	メインラン処理(シンクロナイズドアクション)で、TOFFR の現在のオフセット値を読み込みます。
\$AC_TOFFCR	メインラン処理(シンクロナイズドアクション)で、コーナ半径の現在のオフセット値を読み込みます。

注記

システム変数\$AC_TOFFL、\$AC_TOFF、AC_TOFFR、および AAC_TOFFCR は、先読み処理 (NC プログラム)から読み込みをおこなうと、自動的に先読み停止をおこないます。

用途

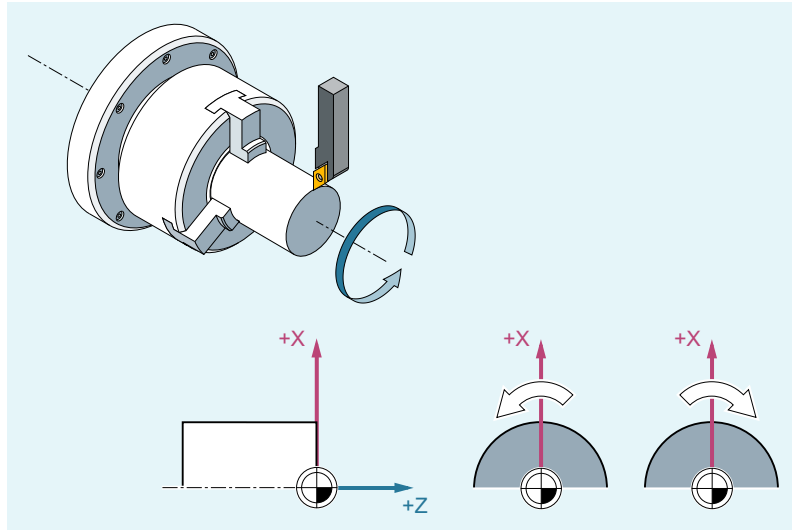
「プログラマブル工具オフセット」機能は、コーナに丸みのあるボールエンドミルとフライス工具で特に重要です。CAM システムではコーナの丸みが、ボールの先端ではなく、ボールの中心に対して計算されることが多いためです。ただし、通常は、工具計測時に工具先端が計測され、工具長として補正メモリに格納されます。

ボールエンドミルの 3 次元工具径補正では、工具長と工具半径を同じ値で同時に補正するのが有効です。このために、TOFFLR アドレスが提供されています。

3.6 主軸動作

3.6.1 主軸速度(S)、主軸回転方向(M3, M4, M5)

主軸速度と回転方向の値が、回転動作を行う主軸に設定され、切り屑処理がおこなわれます。



主軸の他にも軸が存在する場合があります(対向主軸、または旋盤の回転工具など)。一般に、主軸は、マシンデータによって、メイン主軸として宣言されます。この割り当ては、NC 命令を使用して変更できます。

構文

S... / S<n>=...

M3 / M<n>=3

M4 / M<n>=4

M5 / M<n>=5

SETMS (<n>)	
...	
SETMS	

3.6 主軸動作

意味

S...:	メイン主軸の主軸速度(rpm 単位)
S<n>=...:	主軸の主軸速度(rpm 単位) <n>
	注: S0=...で指定された速度はメイン主軸に適用されます。
M3:	メイン主軸の主軸回転方向は右回り
M<n>=3:	主軸<n>の主軸回転方向は右回り
M4:	メイン主軸の主軸回転方向は左回り
M<n>=4:	主軸<n>の主軸回転方向は左回り
M5:	メイン主軸の停止
M<n>=5:	主軸<n>の停止
SETMS (<n>):	主軸<n>をメイン主軸として設定
SETMS:	SETMS が主軸名称なしでプログラム指令されている場合は、設定されているメイン主軸がその代わりに使用されます。

注記

次のように、NC ブロックごとに 3 つまでの S 値をプログラム指令できます。

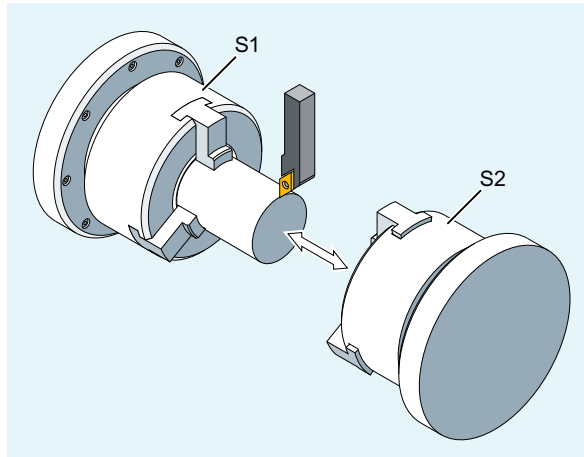
S...S2=...S3=...

注記

SETMS は、個別のブロックに指令してください。

例

S1 はメイン主軸、S2 は第 2 主軸です。部品は、両側から加工します。そのためには、運転を複数の手順に分けてください。切断後は、同期主軸(S2)が、切断後のワークの加工を引き継ぎます。これをおこなうために、この主軸 S2 がメイン主軸として定義され、S2 に対して G95 が適用されます。



プログラムコード	コメント
N10 S300 M3	; ドライブ主軸の速度と回転方向 = メイン主軸へ設定
...	; ワークの右側面の加工
N100 SETMS(2)	; S2 がメイン主軸になります
N110 S400 G95 F...	; 新しいメイン主軸の速度
...	; ワークの左側面の加工
N160 SETMS	; 元のメイン主軸 S1 へ切り替え

詳細情報

メイン主軸の S 値の解釈

G331 機能または G332 機能が G グループ 1 (モーダルに有効な動作命令)で有効な場合、プログラム指令の S 値は常に、rpm 単位で速度と解釈されます。それ以外の場合は、S 値の解釈が G グループ 15 (送り速度タイプ)により、次のように異なります。G96、G961、または G962 が有効な場合、S 値は m/min 単位の周速一定制御と解釈されます。それ以外の場合は、rpm 単位の速度と解釈されます。

G96/G961/G962 から G331/G332 に変更すると、周速一定制御の値がゼロに設定されます。G331/G332 から、G グループ 1 内の G331/G332 以外の機能に変更すると、速度値がゼロに設定されます。対応する S 値は、必要に応じて、再度プログラム指令をおこなってください。

M 命令 M3、M4、M5 の設定

M 命令に軸命令を含むブロックでは、M3、M4、M5 機能が、軸移動の開始前に有効になります(制御装置の初期設定)。

例:

プログラムコード	コメント
N10 G1 F500 X70 Y20 S270 M3	; 主軸速度が 270 rpm へ上昇し、その後に X と Y の移動を実行します。
N100 G0 Z150 M5	; Z の後退移動以前に主軸停止します。

注記

マシンデータを使用して、軸移動の実行タイミングを、主軸が一度指令速度まで加速後、またはプログラム指令の主軸指令切り替え動作がおこなわれた直後の、いずれかに設定できます。

複数主軸の加工

1 つのチャンネルで、5 つの主軸(メイン主軸とつの追加主軸)を同時に使用できます。

主軸の 1 つはマシンデータで、**メイン主軸**として定義されます。ねじ切り、タッピング、毎回転送り速度、およびドウェル時間などの応用機能が、この主軸に適用されます。その他の主軸(第 2 主軸と回転工具など)については、回転速度と主軸回転/主軸停止の方向に、対応する数字を指定してください。

例:

プログラムコード	コメント
N10 S300 M3 S2=780 M2=4	; メイン主軸:300 rpm、右回転 第 2 主軸:780 rpm、左回転

プログラム指令可能なメイン主軸の切り替え

NC プログラムで SETMS (<n>) 命令を使用して、任意の主軸をメイン主軸として定義できます。SETMS は、個別のブロックに指令してください。

例:

プログラムコード	コメント
N10 SETMS (2)	; 主軸 2 がメイン主軸になります。

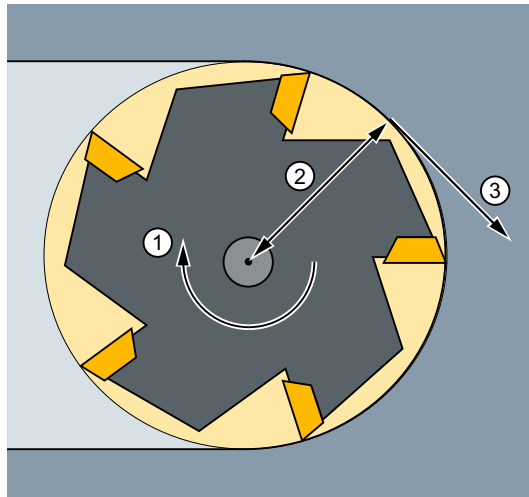
注記

これで、S... で指定された速度、および M3、M4、M5 でプログラム指令した機能が、新しく宣言されたメイン主軸に適用されます。

SETMS が主軸名称なしでプログラム指令されている場合は、マシンデータに指令したメイン主軸が代わりに使用されます。

3.6.2 工具切削速度(SVC)

主軸速度の代わりに、実際にはより一般的に使用されている工具の切削速度を、フライス加工運転用にプログラム指令できます。



- ① 主軸速度
- ② 工具半径
- ③ 工具切削速度

制御装置は動作中の工具半径を使用して、プログラム指令された工具切削速度から、有効な主軸速度を計算します。

$$S = (SVC * 1000) / (R_r * 2\pi)$$

- 意味: S: 主軸速度(rpm 単位)
 SVC: 工具切削速度(m/min または ft/min 単位)
 R_r: 動作中の工具の半径(mm 単位)

動作中の工具の工具タイプ(\$TC_DP1)は考慮されません。

プログラム指令された工具切削速度は、軌跡送り速度 F と G 機能グループ 15 (送り速度タイプ)には依存しません。回転方向と主軸起動は、M3 と M4 それぞれを使用して、また、主軸停止は M5 を使用して実装されます。

オフセットメモリの工具半径データの変更は、次の工具オフセット選択時、または次の動作中のオフセットデータ指令の更新時に適用されます。

3.6 主軸動作

工具交換または工具オフセットデータセットの選択/選択解除をおこなうと、有効な主軸速度が再計算がおこなわれます。

必要条件

工具切削速度のプログラミングには、次の要素が必要です。

- 回転工具の幾何比(フライス工具またはドリル工具)
- 動作中の工具オフセットデータセット

構文

```
T...D...SVC[<n>]=<値>  
...  
S...M3/M4
```

意味

SVC:	工具切削速度のプログラミング時のキーワード	
[<n>]:	主軸番号 このアドレス拡張子により、プログラム指令切削速度が適用される主軸が指定されます。アドレス拡張子がない場合は、速度は常にメイン主軸に適用されます。 注: 各主軸には、用途別の切削速度を設定しておくことができます。 注: アドレス拡張子なしで SVC をプログラム指令するには、メイン主軸に動作中の工具が存在することが必要です。メイン主軸が変更された場合は、ユーザーが、変更に応じて工具を選択してください。	
<値>:	工具切削速度の値	
	単位:	m/min (G71/G710 の場合)または ft/min (G70/G700 の場合)

T...D...:	工具半径は SVC を使用してブロックで決めてください。そのため、工具オフセットデータブロックを含む対応する工具が、ブロック内で動作しているか選択されている必要があります。同一ブロックにプログラミングしたときの SVC、および T/D の指令には、決められた順序はありません。
S...M3/M4:	主軸速度のプログラミングは、工具切削速度の選択解除に影響します。 注: SVC プログラミングと S プログラミングの切り替えは、主軸の回転中であろうといつでも可能です。いずれの場合も、無効な値は解除されます。

注記

SVC プログラミングは、スレーブ主軸送り速度の動作が有効な場合にはできません。

- 周速一定制御:G96/G961/G962 S... (ページ 112)
- 砥石周速度一定:SUG (ページ 117)
- 主軸の位置決め SPOS/SPOSA/M19 (ページ 133)
- ;メイン主軸を軸モードに切り替え:M70 (ページ 133)

逆に、これらの機能のいずれかをプログラミングすると、SVC (工具切削速度)の選択解除に影響します。

注記**最大工具速度**

システム変数\$TC_TP_MAX_VELO[<工具番号>]を使用して、最大工具速度(主軸速度)を設定しておくことができます。

速度制限が定義されていない場合、速度は監視されません。

注記

すでに工具半径が考慮され、ノーズ R の標準工具からの誤差のみを含むような CAD システムを使用して作成された「標準工具」の工具軌跡は、SVC プログラミングとの組み合わせでは、サポートされません。

例

すべての例に次の関係が適用されます。工具ホルダ = 主軸(標準フライス削りの場合)

3.6 主軸動作

例 1:半径 6 mm のフライス工具

プログラムコード	コメント
N10 G0 X10 T1 D1	; \$TC_DP6[1,1] = 6 (tool radius = 6 mm)などのフライス工具の選択
N20 SVC=100 M3	; 切削速度 = 100 m/min ⇒ 切削速度から得られた主軸速度: $S = (100 \text{ m/min.} * 1000) / (6.0 \text{ mm} * 2 * 3.14) = 2653.93 \text{ rpm}$
N30 G1 X50 G95 FZ=0.03	; SVC と 1 刃当り送り速度
...	

例 2:工具選択と SVC は同一ブロック

プログラムコード	コメント
N10 G0 X20	
N20 T1 D1 SVC=100	; ブロックに SVC と工具とオフセットデータセットの選択 (指定の順序はありません)
N30 X30 M3	; 右回転方向の主軸起動、切削速度 100m/min
N40 G1 X20 F0.3 G95	; SVC と毎回転送り速度

例 3:2 つの主軸の切削速度の定義

プログラムコード	コメント
N10 SVC[3]=100 M6 T1 D1	
N20 SVC[5]=200	; 動作中の工具オフセットの工具半径は、両方の主軸で同じです。有効な速度は、主軸 3 と主軸 5 では異なります。

例 4:

前提条件:

次の設定では、メイン主軸の変更または工具交換は工具ホルダで特定されます。

MD20124 \$MC_TOOL_MANAGEMENT_TOOL CARRIER > 1

次の設定では、工具交換をおこなったとき、旧工具オフセットは保持されます。新しい工具の工具オフセットは、D がプログラム指令されたタイミングでのみ有効になります。

MD20270 \$MC_CUTTING_EDGE_DEFAULT = - 2

プログラムコード	コメント
N10 \$TC_MPP1[9998,1]=2	; マガジンロケーションは工具ホルダです
N11 \$TC_MPP5[9998,1]=1	; マガジンロケーションは工具ホルダ 1 です
N12 \$TC_MPP_SP[9998,1]=3	; 工具ホルダ 1 は主軸 3 に割り当てられます
N20 \$TC_MPP1[9998,2]=2	; マガジンロケーションは工具ホルダです
N21 \$TC_MPP5[9998,2]=4	; マガジンロケーションは工具ホルダ 4 です
N22 \$TC_MPP_SP[9998,2]=6	; 工具ホルダ 4 は主軸 6 に割り当てられます
N30 \$TC_TP2[2]="WZ2"	

プログラムコード	コメント
N31 \$TC_DP6[2,1]=5.0	; T2 の半径= 5.0 mm、オフセット D1
N40 \$TC_TP2[8]="WZ8"	
N41 \$TC_DP6[8,1]=9.0	; T8 の半径= 9.0 mm、オフセット D1
N42 \$TC_DP6[8,4]=7.0	; T8 の半径= 7.0 mm、オフセット D4
...	
N100 SETMTH(1)	; メイン工具ホルダ番号を設定します
N110 T="WZ2" M6 D1	; 工具 T2 が装着され、オフセット D1 が有効になります
N120 G1 G94 F1000 M3=3 SVC=100	; S3 = (100 m/min * 1000) / (5.0 mm * 2 * 3.14) = 3184.71 rpm
N130 SETMTH(4)	; メイン工具ホルダ番号を設定します
N140 T="WZ8"	; T8="WZ8"に対応します
N150 M6	; M4=6 に対応します 工具「WZ8」はメイン工具ホルダにありますが、MD20270=-2 のため、旧工具オフセットが有効です。
N160 SVC=50	; S3 = (50 m/min * 1000) / (5.0 mm * 2 * 3.14) = 1592.36 rpm 工具ホルダ 1 に適用されたオフセットはまだ有効であり、主軸 3 に割り当てられます。
N170 D4	; 新しい工具「WZ8」のオフセット D4 が (工具ホルダ 4 で) 有効になります。
N180 SVC=300	; S6 = (300 m/min * 1000) / (7.0 mm * 2 * 3.14) = 6824.39 rpm ; 主軸 6 は工具ホルダ 4 に割り当てられます。

例 5:

前提条件:

次の設定では、主軸が同時に工具ホルダになります。

MD20124 \$MC_TOOL_MANAGEMENT_TOOL CARRIER = 0

次の設定では、工具交換をおこなったとき、工具オフセットデータセット D4 が自動的に選択されます。

MD20270 \$MC_CUTTING_EDGE_DEFAULT = 4

プログラムコード	コメント
N10 \$TC_MPP1[9998,1]=2	; マガジンロケーションは工具ホルダです
N11 \$TC_MPP5[9998,1]=1	; マガジンロケーションは工具ホルダ 1 =主軸 1 です
N20 \$TC_MPP1[9998,2]=2	; マガジンロケーションは工具ホルダです
N21 \$TC_MPP5[9998,2]=3	; マガジンロケーションは工具ホルダ 3 =主軸 3 です
N30 \$TC_TP2[2]="WZ2"	
N31 \$TC_DP6[2,1]=5.0	; T2 の半径= 5.0 mm、オフセット D1
N40 \$TC_TP2[8]="WZ8"	
N41 \$TC_DP6[8,1]=9.0	; T8 の半径= 9.0 mm、オフセット D1
N42 \$TC_DP6[8,4]=7.0	; T8 の半径= 7.0 mm、オフセット D4
...	
N100 SETMS(1)	; 主軸 1 =メイン主軸
N110 T="WZ2" M6 D1	; 工具 T2 が装着され、オフセット D1 が有効になります
N120 G1 G94 F1000 M3 SVC=100	; S1 = (100 m/min * 1000) / (5.0 mm * 2 * 3.14) = 3184.71 rpm

3.6 主軸動作

プログラムコード	コメント
N200 SETMS(3)	; 主軸 3 =メイン主軸
N210 M4 SVC=150	; S3 = (150 m/min * 1000) / (5.0 mm * 2 * 3.14) = 4777.07 rpm T="WZ2"の工具オフセット D1 を参照し、S1 は以前の速度で回転し続けます。
N220 T="WZ8"	; T8="WZ8"に対応します
N230 M4 SVC=200	; S3 = (200 m/min * 1000) / (5.0 mm * 2 * 3.14) = 6369.43 rpm T="WZ2"の工具オフセット D1 を参照します。
N240 M6	; M3=6 に対応します 工具「WZ8」はメイン主軸にあり、新しい工具の工具オフセット D4 が有効になります。
N250 SVC=50	; S3 = (50 m/min * 1000) / (7.0 mm * 2 * 3.14) = 1137.40 rpm メイン主軸のオフセット D4 が有効です。
N260 D1	; 新しい工具「WZ8」のオフセット D1 が有効です。
N270 SVC[1]=300	; S1 = (300 m/min * 1000) / (9.0 mm * 2 * 3.14) = 5307.86 rpm S3 = (50 m/min. * 1000) / (9.0 mm * 2 * 3.14) = 884.64 rpm
...	

詳細情報

工具半径

(動作中の工具に関連した) 次の工具オフセットデータは、工具半径に影響します。

- \$TC_DP6 (半径 - 形状)
- \$TC_DP15(半径 - 摩耗)
- \$TC_SCPx6(\$TC_DP6 のオフセット)
- \$TC_ECPx6(\$TC_DP6 のオフセット)

次の項目は考慮されません。

- オンライン工具径補正
- プログラム指令輪郭の仕上げ代(OFFN)

工具径補正(G41/G42)

工具径補正(G41/G42)、および SVC は両方とも工具半径に関係しますが、機能に関しては相互に関連性はなく、独立しています。

フローティングチャックを使用しないタッピング(G331、G332)

SVC プログラミングは、G331 または G332 と組み合わせることができます。

シンクロナイズドアクション

SVC は、シンクロナイズドアクションからはプログラム指令できません。

切削速度と主軸速度設定タイプの読み出し

主軸の切削速度と速度プログラミングタイプ(主軸速度 S、または切削速度 SVC)は、次のシステム変数を使用して読み出すことができます。

- 次のシステム変数は、パートプログラムで先読み停止をおこないます。

\$AC_SVC[<n>] 番号<n>の主軸に対する現在のメインランブロックを先読みしたときに有効な切削速度です。

\$AC_S_TYPE[<n>] 番号<n>の主軸に対する現在のメインランブロックを先読みしたときに有効な主軸速度プログラミングタイプです。

値: **意味:**

1 主軸速度 S (rpm 単位)

2 工具切削速度 SVC (m/min または ft/min 単位)

- 次のシステム変数は、パートプログラムでは先読み停止をおこないません。

\$P_SVC[<n>] 主軸<n>のプログラム指令切削速度です。

\$P_S_TYPE[<n>] 主軸<n>のプログラム指令主軸速度プログラミングタイプです。

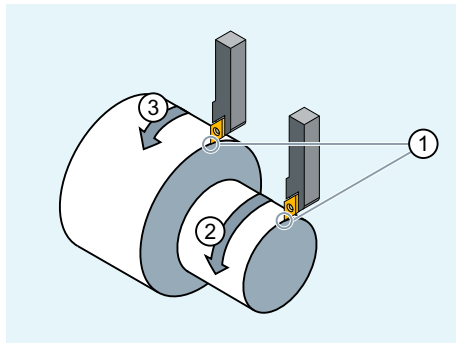
値: **意味:**

1 主軸速度 S (rpm 単位)

2 工具切削速度 SVC (m/min または ft/min 単位)

3.6.3 周速一定制御(G96/G961/G962、G97/G971/G972、G973、LIMS、SCC)

「周速一定制御」機能が有効になると、主軸速度がそれぞれのワーク直径に応じて処理されて、工具刃先の切削速度 S (m/min または ft/min) が一定に保たれます。



- ① 定切削速度
- ② 主軸速度の加速
- ③ 主軸速度の減速

この結果、次の長所が生まれます。

- 回転部品の加工面品質が均一になるため、品質が向上します。
- 加工時の工具摩耗が低減されます。

構文

メイン主軸の周速一定制御の起動/解除:

```
G96/G961/G962 S...  
...  
G97/G971/G972/G973
```

メイン主軸の速度制限:

```
LIMS=<値>  
LIMS [<主軸>] = <値>
```

G96/G961/G962 のその他の基準軸:

```
SCC [<軸>]
```

注記

SCC [<軸>] は、G96/G961/G962 と一緒にプログラム指令することも、単独でプログラム指令することもできます。

意味

G96:	毎回転送り速度(G95 (ページ 119)に対して)と周速一定制御 G95 は、G96 で自動的に有効になります。これまでに G95 が無効だった場合は、G96 の呼び出し時に新しい送り速度値 F...を指定してください。	
G961:	毎分送り速度(G94 (ページ 119)に対して)と周速一定制御	
G962:	毎分送り速度と毎回転送り速度、および周速一定制御	
S...:	G96、G961、または G962 と組み合わせた場合は、S...が主軸速度ではなく、切削速度と解釈されます。切削速度は常に、メイン主軸に適用されます。	
	単位:	m/min (G71/G710 の場合)または ft/min (G70/G700 の場合)
	値の範囲:	0.1 m/min～9999 9999.9 m/min
G97:	毎回転送り速度および一定主軸速度(周速一定制御オフ)	
G971:	毎分送り速度および一定主軸速度(周速一定制御オフ)	
G972:	毎分送りまたは毎回転送り速度および一定主軸速度(周速一定制御オフ)	
G973:	主軸速度制限のない毎回転送り速度、および一定主軸速度(ISO モードでは、LIMS なしの G97)	
	注: G97(または G971 ... G973)の後は、S...が再び、主軸速度(rpm 単位)として解釈されます。新しい主軸速度が指定されない場合は、G96 で設定された最後の速度(それぞれ G961 または G962)が保持されます。	
LIMS:	メイン主軸の速度制限(G96/G961/G97 が有効な場合にのみ適用されます) メイン主軸を選択できる機械の場合は、1 ブロック内で 4 つまでの主軸に、異なる複数の制限値をプログラム指令できます。	
	<主軸>:	主軸番号
	<値>:	主軸速度の上限(rpm 単位)
SCC:	G96/G961/G962 機能のいずれかが有効である場合、SCC[<軸>]を使用して、任意のジオメトリ軸を基準軸として割り当てることができます。	

注記

G96/G961/G962 の初回の選択時には、周速一定制御 S...を入力してください。G96/G961/G962 を再度選択するときは、入力値は省略できます。

注記

LIMS でプログラムされた速度制限は、G26 でプログラムされた速度制限、およびセッティングデータで定義した速度制限の、いずれも超えないようにしてください。

注記

G96/G961/G962 基準軸は、SCC[<軸>]のプログラム指令時は、チャンネルに割り当てられているジオメトリ軸にしてください。SCC[<軸>]は、G96/G961/G962 のどの機能が有効なときにもプログラム指令できます。

例**例 1:速度制限がある周速一定制御の起動**

プログラムコード	コメント
N10 SETMS (3)	
N20 G96 S100 LIMS=2500	; 周速一定制御 = 100 m/min、最高速度 2500 rpm
...	
N60 G96 G90 X0 Z10 F8 S100 LIMS=444	; 最高速度 = 444 rpm

例 2:4 つの主軸の速度制限の定義

速度制限は、次のように、主軸 1 (メイン主軸)と主軸 2、3、および 4 に対して定義されます。

プログラムコード
N10 LIMS=300 LIMS[2]=450 LIMS[3]=800
LIMS[4]=1500
...

例 3:X 軸の正面切削のための Y 軸割り付け

プログラムコード	コメント
N10 G18 LIMS=3000 T1 D1	; 3000 rpm での速度制限
N20 G0 X100 Z200	
N30 Z100	
N40 G96 S20 M3	; 周速一定制御 (= 20 m/min) は Y 軸に応じて変わります。
N50 G0 X80	
N60 G1 F1.2 X34	; X 方向の 1.2 mm/rev の正面切削。
N70 G0 G94 X100	
N80 Z80	
N100 T2 D1	
N110 G96 S40 SCC[Y]	; Y 軸が G96 に割り当てられ、G96 が有効になります (単独のブロックで実行可能です)。周速一定制御 (= 40 m/min) は Y 軸に応じて変わります。

プログラムコード	コメント
...	
N140 Y30	
N150 G01 F1.2 Y=27	; Y のプランジ切削、送り速度 F = 1.2 mm/rev
N160 G97	; 周速一定制御オフ
N170 G0 Y100	

詳細情報

主軸速度の計算

正面軸の SZS 位置(半径)を使用して、プログラム指令切削速度から主軸速度を計算します。

注記

主軸速度の計算では WCS と SZS 間のフレーム(たとえば、SCALE、TRANS、ROT 等のプログラマブルフレーム)が考慮されます。また、このフレームは速度変化をもたらすことがあります(たとえば、SCALE 使用時の有効径に変更があった場合)。

速度制限 LIMS

加工が必要なワークの直径が大幅に異なっている場合は、LIMS (最大主軸速度)で主軸の制限速度を指定することをお勧めします。こうすると、直径が小さい場合の速度の上がり過ぎが防止されます。LIMS は、G96、G961、および G97 が有効なときにのみ適用されます。LIMS は、G971 が選択されているときは適用されません。このブロックがメインランにロードされると、すべてのプログラム指令値がセッティングデータに転送されます。

注記

パートプログラムで LIMS により変更された速度制限値はセッティングデータに設定されるため、プログラム終了後もそのまま保存されます。

ただし、LIMS で変更された速度制限値がプログラム終了後に適用されなくなる場合、以下の定義を工作機械メーカーの GUD ブロックに挿入してください。

```
REDEF $SA_SPIND_MAX_VELO_LIMS PRLOC
```

周速一定制御の解除(G97/G971/G972/G973)

G97 (または G971 ... G973)の後は、S...が再び、主軸速度(rpm 単位)として解釈されます。新しい主軸速度が指定されない場合は、G96 で設定された最後の速度(それぞれ G961 または G962)が保持されます。

G96/G961 機能は、G94 または G95 でも無効にすることができます。この場合、最後のプログラム指令速度 S...を使用して、以降の加工運転が実行されます。

G97 は、事前に G96 を使用せずにプログラム指令できます。このときの、この機能の効果は、G95 と同じです。LIMS もプログラム指令できます。

G973 を使用して、主軸速度の制限を有効にせずに周速一定制御を無効にすることができます。

注記

マシンデータで径方向軸を設定してください。

早送り G0

早送り G0 を使用した場合、速度は変更されません。

例外:

輪郭へ早送りでアプローチし、次の NC ブロックに G1/G2/G3 などの軌跡命令が含まれる場合は、次の軌跡命令に対する G0 アプローチブロックで速度が調整されます。

G96/G961/G962 のその他の基準軸

G96/G961/G962 機能のいずれかが有効である場合、SCC[<軸>]を使用して、任意のジオメトリ軸を基準軸として割り当てることができます。基準軸が変更された場合は、それは周速一定制御の TCP(工具中心点)基準位置が変更されます。その結果、主軸速度は設定された減速カーブまたは加速カーブで、最終の目標速度に達します。

割り当てられたチャネル軸の軸入れ替え

ジオメトリ軸には常に、G96/G961/G962 の基準軸機能が割り当てられます。割り当てられたチャネル軸が軸入れ替えに使用された場合は、G96/G961/G962 の基準軸機能は、旧チャネルに保持されます。

ジオメトリ軸入れ替えは、周速一定制御へのジオメトリ軸の割り当て方法には影響しません。G96/G961/G962 の TCP 基準位置がジオメトリ軸入れ替えに影響する場合は、主軸がカーブを描いて新しい速度に達します。

ジオメトリ軸入れ替え(GEOAX(0,X)など)の結果、チャネル軸がまったく割り当てられない場合は、主軸速度は G97 に従って決められます。

基準軸の割り当てによるジオメトリ軸入れ替えの例:

プログラムコード	コメント
N05 G95 F0.1	
N10 GEOAX(1, X1)	; チャネル軸 X1 が 1 番目のジオメトリ軸になります。
N20 SCC[X]	; 1 番目のジオメトリ軸 (X) が ; G96/G961/G962 の基準軸になります。
N30 GEOAX(1, X2)	; チャネル軸 X2 が 1 番目のジオメトリ軸になります。
N40 G96 M3 S20	; G96 の基準軸はチャネル軸 X2 です。

プログラムコード	コメント
N05 G95 F0.1	
N10 GEOAX(1, X1)	; チャネル軸 X1 が 1 番目のジオメトリ軸になります。
N20 SCC[X1]	; X1 および自動的に 1 番目のジオメトリ軸 (X) が G96/G961/G962 の基準軸になります。
N30 GEOAX(1, X2)	; チャネル軸 X2 が 1 番目のジオメトリ軸になります。
N40 G96 M3 S20	; G96 の基準軸は X2 または X で、アラームは発生しません。

プログラムコード	コメント
N05 G95 F0.1	
N10 GEOAX(1, X2)	; チャネル軸 X2 が 1 番目のジオメトリ軸になります。
N20 SCC[X1]	; X1 はジオメトリ軸ではなく、アラームが発生します。

プログラムコード	コメント
N05 G0 Z50	
N10 X35 Y30	
N15 SCC[X]	; G96/G961/G962 の基準軸は X です。
N20 G96 M3 S20	; 周速一定制御が 10 mm/min でオン。
N25 G1 F1.5 X20	; X 方向の 1.5 mm/rev の正面切削。
N30 G0 Z51	
N35 SCC[Y]	; G96 の基準軸は Y で、 ; 主軸速度 (Y30) が減速されます。
N40 G1 F1.2 Y25	; Y 方向の 1.2 mm/rev の正面切削。

3.6.4 砥石周速度一定制御(GWPSON、GWPSOF)のスイッチオン/オフ:

事前提起された手順 GWPSON(...)および GWPSOF(...)により、研削工具(工具タイプ:400～499)の砥石周速度一定(制御) (GWPS)がスイッチオン/オフになります。

構文

```
GWPSON (<TNo>)
S<n>=... :
...
GWPSOF (<TNo>)
```

3.6 主軸動作

意味

GWPSON (...):	砥石周速度一定(制御)のスイッチオン
GWPSOF (...):	砥石周速度一定(制御)のスイッチオフ
<TNo>:	T 番号 注記: 砥石周速度一定(制御)を現在使用されている有効なといしではなく、無効なといしに対してオンまたはオフにする場合にのみ必要です。
S<n>=...:	主軸<n>のといし周速度(m/s または ft/s 単位)
S0=...または S...:	メイン主軸のといし周速度

状態確認

このシステム変数を使用すると、特定の主軸で砥石周速度一定(制御)が有効であるかどうかをパートプログラムから確認できます。

\$P_GWPS[<n>]; ここで、<n> = 主軸番号

値	意味
0 (= FALSE)	GWPS が無効です。
1 (= TRUE)	GWPS が有効です。

3.6.5 プログラマブル主軸速度制限(G25、G26)

マシンデータとセッティングデータで定義した主軸速度の最小値と最大値は、パートプログラム命令を使用して変更できます。

主軸速度制限は、チャンネルのすべての主軸に対してプログラム指令できます。

構文

```
G25 S... S1=... S2=...
G26 S... S1=... S2=...
```

意味

G25: 主軸速度の下限
 G26: 主軸速度の上限
 S...S1=... S2=... : 主軸速度の最小値または最大値

注:

各ブロックに最大 3 つの主軸速度制限値をプログラム指令できます。

値の範囲: 0.1～9999 9999.9 rpm

注記

G25 または G26 でプログラム指令された主軸速度制限は、セッティングデータの速度制限値を上書きします、このため、プログラム終了後も、そのまま保存されます。

ただし、G25/G26 で変更された速度制限値がプログラム終了後に適用されなくなる場合、以下の定義を工作機械メーカーの GUD ブロックに挿入してください。

```
REDEF $SA_SPIND_MIN_VELO_G25 PRLOC
```

```
REDEF $SA_SPIND_MAX_VELO_G26 PRLOC
```

例

プログラムコード	コメント
N10 G26 S1400 S2=350 S3=600	; メイン主軸、主軸 2、および主軸 3 の上限速度

3.7 送り速度制御

3.7.1 送り速度(G93、G94、G95、F、FGROUP、FL、FGREF)

これらの命令を NC プログラムで使用して、加工処理に関連するすべての軸の送り速度を設定します。

構文

```
G93
G94
G95
F<値>
FGROUP (<軸_1>,<軸_2>,...)
```

3.7 送り速度制御

FGREF [<回転軸>] = <基準半径>
 FL [<軸>] = <値>

意味

G93:	軌跡速度タイプ:インバースタイム送り[rpm]
G94:	軌跡速度タイプ:毎分送り速度[mm/min]、[inch/min]または[°/min]
G95:	軌跡速度タイプ:毎回転送り速度[mm/rev]または[inch/rev] 毎回転送り速度は、メイン主軸、他の主軸または回転軸から算出できます。
F<値>	すべてまたは FGROUP で選択した軌跡軸の軌跡送り速度
FGROUP:	F でプログラム指令した軌跡送り速度の対象となる軌跡軸の定義
FGREF:	FGREF を使用して、有効半径(<基準半径>)を、FGROUP で指定された各回転軸に対してプログラム指令します。
FL:	同期軸/軌跡軸の制限速度 G94 で設定された単位が適用されます。 軸(チャンネル軸、ジオメトリ軸、または旋回軸)毎に 1 つの FL 値をプログラム指令できます。
<軸>:	チャンネル軸名称、タイプ:AXIS

例

例 1:FGROUP の動作モード

次の例は、軌跡と軌跡送り速度に対する FGROUP の働きを示すものです。変数\$AC_TIME には、秒単位のブロック起動時間が含まれます。これは、シンクロナイズドアクションのみで使用できます。

プログラムコード	コメント
N100 G0 X0 A0	
N110 FGROUP (X,A)	
N120 G91 G1 G710 F100	; 送り速度= 100 mm/min または 100°/min
N130 DO \$R1=\$AC_TIME	
N140 X10	; 送り速度= 100 mm/min、軌跡= 10 mm、R1 =約 6 秒
N150 DO \$R2=\$AC_TIME	
N160 X10 A10	; 送り速度= 100 mm/min、軌跡= 14.14 mm、R2 =約 8 秒
N170 DO \$R3=\$AC_TIME	
N180 A10	; 送り速度= 100°/min、軌跡= 10°、R3 =約 6 秒
N190 DO \$R4=\$AC_TIME	

プログラムコード	コメント
N200 X0.001 A10	; 送り速度= 100 mm/min、軌跡= 10 mm、R4 =約 6 秒
N210 G700 F100	; 送り速度= 2540 mm/min または 100°/min
N220 DO \$R5=\$AC_TIME	
N230 X10	; 送り速度= 2540 mm/min、軌跡= 254 mm、R5 =約 6 秒
N240 DO \$R6=\$AC_TIME	
N250 X10 A10	; 送り速度= 2540 mm/min、軌跡= 254.2 mm、R6 =約 6 秒
N260 DO \$R7=\$AC_TIME	
N270 A10	; 送り速度= 100°/min、軌跡= 10°、R7 =約 6 秒
N280 DO \$R8=\$AC_TIME	
N290 X0.001 A10	; 送り速度= 2540 mm/min、軌跡= 10 mm、R8 =約 0.288 秒
N300 FGREF[A]=360/(2*\$PI)	; 有効半径を使用して 1 度= 1 インチを設定します
N310 DO \$R9=\$AC_TIME	
N320 X0.001 A10	; 送り速度= 2540 mm/min、軌跡= 254 mm、R9 =約 6 秒
N330 M30	

例 2 :制限速度 FL の同期軸の移動

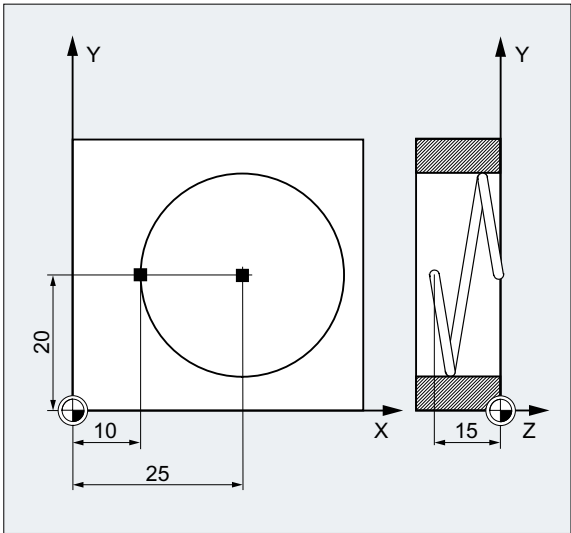
同期軸 Z が制限速度に達すると、軌跡軸の軌跡速度は減速します。

プログラムコード
N10 G0 X0 Y0
N20 FGROUP(X)
N30 G1 X1000 Y1000 G94 F1000 FL[Y]=500
N40 Z-50

例 3 :へリカル補間

軌跡軸 X と Y は、プログラム指令送り速度で移動します。切り込み軸 Z は同期軸です。

3.7 送り速度制御

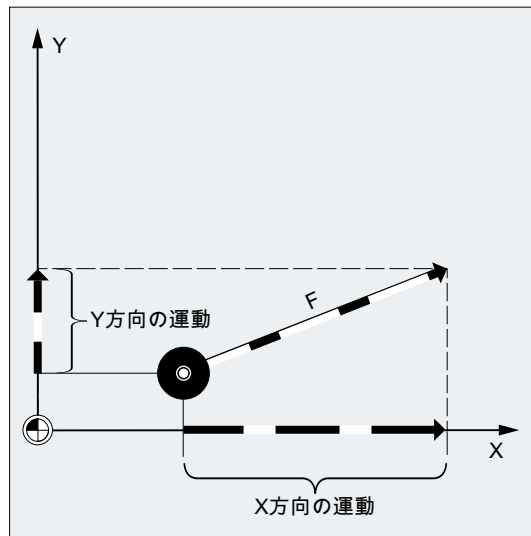


プログラムコード	コメント
N10 G17 G94 G1 Z0 F500	; 工具の送り速度。
N20 X10 Y20	; 開始位置へアプローチします。
N25 FGROUP (X,Y)	; 軸 X/Y は軌跡軸で、Z は同期軸です。
N30 G2 X10 Y20 Z-15 I15 J0 F1000 FL[Z]=200	; 円弧軌跡では送り速度が 1,000 mm/min、Z 方向の移動は同期します。
...	
N100 FL[Z]=\$MA_AX_VELO_LIMIT[0,Z]	; MD からの速度の読み取りにより、制限速度が選択解除されます。MD から値を読み取ります。
N110 M30	; プログラム終了

詳細情報

軌跡軸の送り速度(F)

軌跡送り速度は通常、移動に使用されるすべてのジオメトリ軸の個々の速度成分で構成され、切削工具または旋削工具先端の中心点を基準にします。



送り速度は、アドレス F で指定されます。そして新しい送り速度が指令されるまで有効です。マシンデータの初期設定に応じて、G 命令で指定された単位系は、mm と inch のいずれかとなります。

NC ブロック毎に 1 つの F 値をプログラム指令できます。送り速度の単位は、G 命令 G93/G94/G95 のいずれかで定義されます。送り速度 F は軌跡軸に対してのみ機能し、新しい送り速度がプログラム指令されるまで有効です。アドレス F の後には区切り文字を使用できます。

例:

F100 または F 100

F.5

F=2*FEED

送り速度のタイプ(G93/G94/G95)

G 命令 G93、G94、および G95 はモーダルです。G93、G94、および G95 のいずれかに切り替えるときは、軌跡送り速度値を再度プログラム指令してください。回転軸を使用して加工する場合は、送り速度を°/min 単位でも指定できます。

インバースタイム送り (G93)

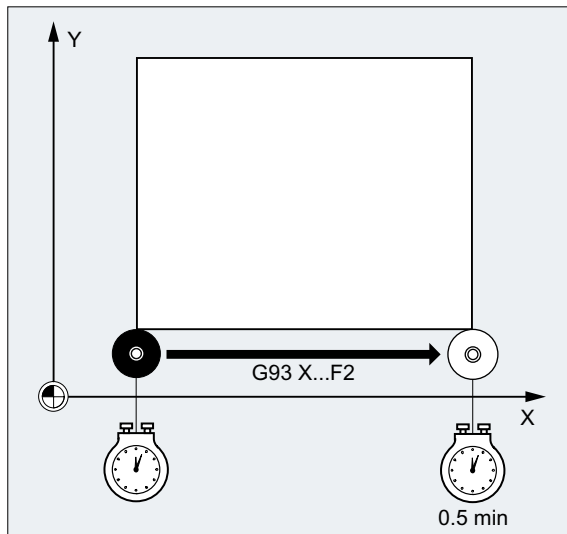
インバースタイム送りは、1 つのブロックの動作命令を実行するために必要な時間を指定します。

単位: 1/min

例:

N10 G93 G01 X100 F2

意味:プログラム指令軌跡を 0.5 分で移動します。



注記

軌跡長がブロック毎に大きく異なる場合は、各ブロックで新しい F 値を、G93 を使用して指定してください。回転軸を使用して加工する場合は、送り速度を°/min 単位でも指定できます。

同期軸の送り速度

アドレス F でプログラム指令された送り速度は、一つのブロックにプログラム指令されたすべての軌跡軸に適用されますが、同期軸には適用されません。同期軸は、その軌跡が軌跡軸と同じ時間を必要とし、すべての軸が同時にその終点に到達するよう制御されます。

同期軸の制限速度(FL)

FL 命令を使用して、同期軸の制限速度をプログラム指令できます。FL がプログラム指令されていない場合は、早送り速度が適用されます。FL は、MD (MD36200 \$MA_AX_VELO_LIMIT)に割り当て値で、解除されます。

同期軸としての軌跡軸の移動(FGROUP)

FGROUP を使用して、軌跡軸を軌跡送り速度で移動するか、同期軸として移動するかを定義します。たとえば、ヘリカル補間で、2つのジオメトリ軸、XとYのみが、プログラム指令送り速度で移動するよう定義できます。この場合は、切り込み軸 Z が同期軸となります。

例: FGROUP (X, Y)

FGROUP の変更

FGROUP による設定は、次の方法で変更できます。

1. FGROUP の再プログラミング: FGROUP (X, Y, Z) など
2. 軸を指定しない FGROUP のプログラミング: FGROUP ()
FGROUP () に従って、マシンデータの初期設定が適用されます。これで、ジオメトリ軸が再度、軌跡軸グループで移動します。

注記

FGROUP を使用する場合は、軸識別子はチャンネル軸の名称にしてください。

送り速度 F の単位系

形状設定に加えて、G700 と G710 の G 命令も、送り速度 F の単位系の定義に使用されます。言い換えると、

- G700 の場合: [inch/min]
- G710 の場合: [mm/min]

注記

G70/G71 は、送り速度の設定には影響しません。

制限速度 FL の同期軸の単位系

G 命令 G700/G710 を使用した F の単位設定は FL にも適用されます。

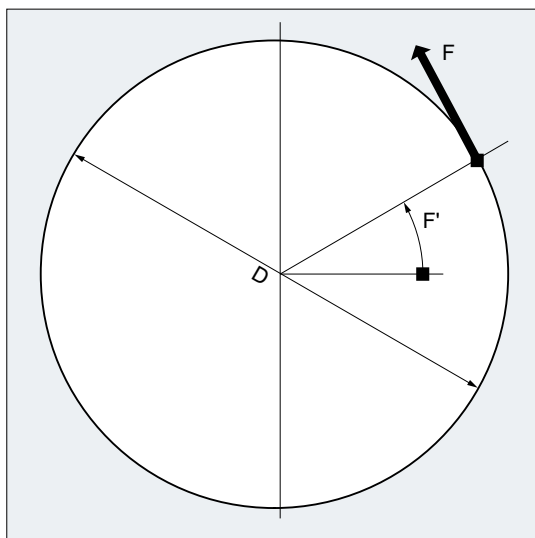
回転軸と直線軸の単位

FGROUP で結合され、共に軌跡を移動する直線軸と回転軸の場合は、送り速度の解釈に直線軸の単位が使用されます(G94/G95 の mm/min または inch/min、および mm/rev または inch/rev の指令によります)。

回転軸の接線速度(mm/min または inch/min)は、次の式に従って計算されます。

$$F[\text{mm/min}] = F'[^{\circ}/\text{min}] * \pi * D[\text{mm}]/360[^{\circ}]$$

各要素 F: 接線速度
 の意味 F': 角速度
 は次の
 とおり π : 円周率
 です。 D: 直径



軌跡速度 F による回転軸の移動(FGREF)

工具またはワークまたはその両方が回転軸で移動する加工運転の場合、有効な加工送り速度は、F 値を基準とした通常の方法で、軌跡速度として解釈されます。このために、使用される各回転軸に対し、有効半径(基準半径)を指定する必要があります。

基準半径の単位は、G70/G71/G700/G710 の設定によって異なります。

使用される全ての軸が軌跡送り速度の計算で考慮されるためには、全ての軸を FGROUP 命令に含めてください。

FGREF プログラム指令なしの動作との互換性を保証するために、システムの電源投入時と RESET 時は係数 $1^\circ = 1 \text{ mm}$ が有効になります。これは、 $\text{FGREF} = 360 \text{ mm} / (2\pi) = 57.296 \text{ mm}$ という基準半径に対応します。

注記

この初期設定は、有効単位系(MD10240 \$MN_SCALING_SYSTEM_IS_METRIC)にも、現在動作中の G70/G71/G700/G710 設定にも依存しません。

特別な事例:

プログラムコード

```
N100 FGROUP (X,Y,Z,A)
N110 G1 G91 A10 F100
N120 G1 G91 A10 X0.0001 F100
```

このタイプのプログラミングの場合、N110 にプログラム指令された F 値は回転軸の送り速度(°/min 単位)で使用され、N120 で使用される送り速度は、現在動作中の G70/G71/G700/G710 設定に応じて、100inch/min または 100 mm/min となります。

通知

送り速度差分

FGREF による計算は、ブロックにプログラム指令されている軸が回転軸の場合にも機能します。この場合、°/min 単位の通常の F 値の解釈は、基準半径が FGREF の初期設定に対応する場合のみ適用されます。

- G71/G710 の場合: FGREF[A]=57.296
- G70/G700 の場合: FGREF[A]=57.296/25.4

基準半径の読み込み

回転軸の基準半径値は、次のシステム変数を使用して読み込むことができます。

- 次のシステム変数は、シンクロナイズドアクション、またはパートプログラムで先読み停止をおこなって読み込みます。

\$AA_FGREF[<軸>] 現在のメインラン値

- 次のシステム変数は、パートプログラムでは先読み停止をおこないません。

\$PA_FGREF[<軸>] プログラム指令値

値がまったくプログラム指令されていない場合は、既定の $360 \text{ mm}/(2\pi) = 57.296 \text{ mm}$ (1°あたり 1 mm に対応します)が両方のシステム変数で読み込まれます。

直線軸の場合、両方のシステム変数の値は常に 1 mm です。

速度に影響する軌跡軸の読み込み

軌跡補間に関係する軸を、次のように、システム変数を使用して読み込むことができます。

- 次のシステム変数は、シンクロナイズドアクション、またはパートプログラムで先読み停止をおこなって読み込みます。

\$AA_FGROUP[<軸>] 指定軸が、初期設定または FGROUP プログラミングにより、現在のメインラン処理の軌跡速度に影響する場合は、値「1」を返します。それ以外の場合は、値「0」を返します。

\$AC_FGROUP_MASK FGROUP でプログラム指令され、軌跡速度に影響するチャネル軸のビットキーを返します。

3.7 送り速度制御

- 次のシステム変数は、パートプログラムでは先読み停止をおこないません。

<code>\$PA_FGROUP[<軸>]</code>	指定軸が、初期設定または FGROUP プログラミングにより軌跡速度に影響する場合は、値「1」を返します。それ以外の場合は、値「0」を返します。
<code>\$P_FGROUP_MASK</code>	FGROUP でプログラム指令され、軌跡速度に影響するチャネル軸のビットキーを返します。

FGREF による旋回軸の軌跡基準係数

旋回軸では、FGREF[] 係数の動作モードは、回転軸補間とベクトル補間のどちらを使用し、工具の向きが変更されるかに応じて異なります。

回転軸では**回転軸補間**が使用され、この場合は、旋回軸の当該の FGREF 係数が、軸軌跡の基準半径として個々に計算されます。

ベクトル補間の場合は、個々の FGREF 係数の相乗平均値として動作中の FGREF 係数が計算され、適用されます。

$FGREF[\text{有効値}] = [(FGREF[A] * FGREF[B]...)] \text{の } n \text{ 乗根}$

各要素 A: 1 番目の旋回軸の軸識別子
 の意味 B: 2 番目の旋回軸の軸識別子
 は次の C: 3 番目の旋回軸の軸識別子
 とおり n: 旋回軸数
 です。

例:

標準の 5 軸座標変換には 2 つの旋回軸があるため、有効な係数は、2 つの軸係数の積の平方根です。

$FGREF[\text{effective}] = [(FGREF[A] * FGREF[B])] \text{の平方根}$

注記

したがって、旋回軸 FGREF の有効係数を使用して、プログラム指令軌跡送り速度の基準となる、工具の基準点を定義できます。

3.7.2 位置決め軸の移動(POS、POSA、POSP、FA、WAITP、WAITMC)

位置決め軸は、用途別の軸別送り速度で、軌跡軸とは無関係に移動します。補間命令はありません。POS/POSA/POSP 命令を使用すれば、位置決め軸の移動とモーションのシーケンスの協調を同時に行えます。

位置決め軸の代表例は次のとおりです。

- パレット送り機器
- 検査ステーション

WAITP を使用すると、これ以前の NC ブロックで POSA を使用してプログラム指令された軸が、終了位置に到達するまでプログラムが待機する NC プログラムの位置を、指定できます。

WAITMC は、指定した待機マークを受信すると、直ちに次の NC ブロックをロードします。

構文

```

POS [<軸>] =<位置>
POSA [<軸>] =<位置>
POSP [<軸>] = (<終了位置>, <区間長>, <モード>)
FA [<軸>] =<値>
WAITP (<軸>) ; 個別の NC ブロックでプログラミング
WAITMC (<待機マーク>)

```

意味

POS/POSA:	指定した位置に位置決め軸を移動します		
	POS と POSA の機能は同じですが、ブロック切り替えの動作は異なります。		
	• POS は、位置に達するまで、NC ブロックが有効になるのを遅らせます。 • POSA は、位置に達していなくても、NC ブロックを有効にします。		
	<軸>:	移動軸の名称(チャネルまたはジオメトリ軸識別子)	
	<位置>:	アプローチする軸の位置	
		タイプ:	REAL

3.7 送り速度制御

POSP:	複数区間で、指定した終了位置に位置決め軸を移動します。			
	<終了位置>:		アプローチする軸の終了位置	
	<区間長>:		区間長	
	<モード>:		アプローチモード	
			= 0:	最後の 2 つの区間は、終了位置までの残りの軌跡が、同一の大きさ (設定済み) の 2 つの残り区間に分割されます。
			= 1:	計算したすべての区間長の合計が終了位置までの軌跡に正確に対応するように、区間長が調整されます。
注: POSP を使用すると、揺動動作を明確にプログラム (ページ 1109) できます。				
FA:	指定した位置決め軸の送り速度			
	<軸>:		移動軸の名称(チャネルまたはジオメトリ軸識別子)	
	<値>:		送り速度	
			単位:	mm/min または inch/min または° /min
	注: 5 つまでの FA 値を、NC ブロックごとにプログラム指令できます。			
WAITP:	位置決め軸が移動するまで待機します。 以降のブロックは、これ以前の NC ブロックで POSA を使用してプログラム指令した指定位置決め軸が(精密イグザクトストップによる)終了位置に到達するまで処理されません。			
	<軸>:		WAITP 命令が適用される軸の名称です(チャネル軸またはジオメトリ軸の識別子)。	
	注: WAITP を使用すると、1 つの軸を、揺動軸として、または (PLC を介した)同時位置決め軸の移動用に使用できます。			
WAITMC:	指定した待機マークの受信を待機します。 指定した待機マークを受信すると、直ちに次の NC ブロックをロードします。			
	<待機マーク>:		待機マークの番号	

**注意****POSA による移動**

先読み停止となる命令が、自動的に後続のブロックで読み取られた場合、すでに先読みがおこなわれて保存されている他のすべてのブロックが実行完了するまでは、この POSA ブロックは実行されません。これ以前の POSA ブロックはイグザクトストップ(G9 として)で停止します。

例**例 1: POSA による移動と機械状態データへのアクセス**

制御装置は、機械状態データ(\$A...)へのアクセス時に内部の先読み停止をおこないます。加工は、先読みがおこなわれて保存されたすべてのブロックが完全に実行されるまで停止します。

プログラムコード	コメント
N40 POSA[X]=100	
N50 IF \$AA_IM[X]==R100 GOTOF LABEL1	; 機械状態データへのアクセス。
N60 G0 Y100	
N70 WAITP(X)	
N80 LABEL1:	
N...	

例 2: WAITP で移動の終了を待機

パレット送り機器

軸 U: パレット格納

作業領域へのワークパレットの搬送

軸 V: 処理を支援するための抜き取り検査が実行される検査ステーションへの搬送装置

プログラムコード	コメント
N10 FA[U]=100 FA[V]=100	; 個々の位置決め軸 U と V の軸別送り速度の指定。
N20 POSA[V]=90 POSA[U]=100 G0 X50 Y70	; 位置決め軸と軌跡軸で移動します。
N50 WAITP(U)	; プログラムの実行は、N20 でプログラム指令した終点に軸 U が到達するまでは再開されません。

3.7 送り速度制御

プログラムコード	コメント
...	

詳細情報

POSA による移動

ブロック切り替えの有効化とプログラムの実行のいずれも、POSA には影響されません。終了位置への移動は、以降の NC ブロックの実行中におこなうことができます。

POS による移動

次のブロックは、POS でプログラム指令されたすべての軸がその終了位置に到達するまで実行されません。

WAITP で移動の終了を待機

WAITP の後は、NC プログラムへの軸の割り当てが無効になります。この状態は再度、軸をプログラム指令するまで適用されます。このとき、この軸は、PLC による位置決め軸として、または NC プログラム/PLC、または HMI からの揺動軸として、動作することができます。

IPOBRKA と WAITMC による減速カーブのブロックの切り替え

軸が待機マークに到達していない場合、または別のブロック終了条件によりブロックを切り替えられない場合にのみ減速します。WAITMC の後、ブロックの切り替えを妨げる他のブロック終了条件がない場合は、軸が直ちに起動します。

3.7.3 主軸位置制御モード(SPCON、SPCOF)

SPCON または SPCOF 命令を使用すると、主軸の位置制御動作が明示的に有効または無効になります。

注記

SPCON を使用して位置制御モードをオンにするには、最大で 3 つの位置制御サイクルが必要です。

構文

SPCON

SPCON (<n>)

SPCON (<n>, <m>, ...)

SPCOF

SPCOF (<n>)

SPCOF (<n>,<m>,...)

意味

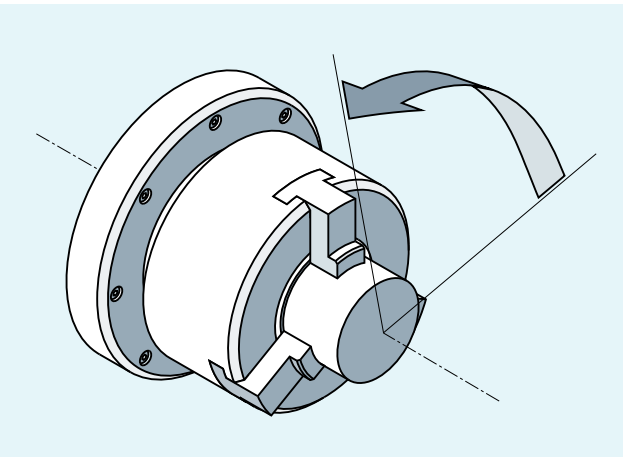
SPCON	位置制御モードを有効にします。 指定した主軸が、速度制御から位置制御へ切り替わります。 SPCON はモーダルで、SPCOF まで保持されます。
SPCOF	位置制御モードを解除します 指定した主軸が、位置制御から速度制御へ切り替わります。
<n>,<m>,... ...:	主軸番号 主軸番号の指定なし:チャンネルのメイン主軸

注記

指令値連結を行う主軸同期の場合、SPCOF を使用してマスタ主軸を速度制御モードに切り替えることはできません。

3.7.4 主軸の位置決め(SPOS、SPOSA、M19、M70、WAITS)

SPOS、SPOSA、または M19 を使用して、工具交換のときなどに、主軸を特定の回転位置に設定できます。



SPOS、SPOSA、および M19 により、次の M3/M4/M5/M41～M45 実行まで、一時的に位置制御モードに切り替わります。

軸モードの位置決め

主軸は、マシンデータで定義したアドレスで軌跡軸、同期軸、または位置決め軸としても動作できます。軸識別子を指定すると、主軸は軸モードとなります。M70 の場合は、主軸が直接、軸モードに切り替わります。

位置決めの終了

FINEA、CORSEA、IPOENDA、または IPOBRKA を使用して、主軸位置決めの動作終了条件をプログラム指令できます。

実行中のブロックでプログラム指令したすべての主軸または軸の動作終了条件、および軌跡補間のブロック切り替え条件が満たされた場合は、プログラムが次のブロックへ進みます。

同期制御

主軸移動を同期させるには、WAITS を使用すると、主軸位置に到達するまで待機できます。

必要条件

位置決めされる主軸は、位置制御モードの動作を可能にしてください。

構文

主軸の位置決め:

SPOS=<値> / SPOS [<n>]=<値>

SPOSA=<値> / SPOSA [<n>]=<値>

M19 / M<n>=19

主軸を軸モードに切り替え:

M70 / M<n>=70

動作終了条件を定義:

FINEA/FINEA[S<n>]

COARSEA/COARSEA[S<n>]

IPOENDA/IPOENDA[S<n>]

IPOBRKA/IPOBRKA (<軸>[, <タイミング>]) ; 個別の NC ブロックでプログラミング

主軸移動を同期化:

WAITS/WAITS (<n>, <m>) ; 個別の NC ブロックでプログラミング

意味

SPOS/SPOSA:	主軸を指定角度に設定 SPOS と SPOSA の機能は同じですが、ブロック切り替えの動作は異なります。 • SPOS を使用する場合、設定位置に達してはじめて NC ブロックが有効になります。 • SPOSA を使用する場合は、この位置に達していなくても、NC ブロックが有効になります。		
<n>:	位置決めを行う主軸番号。 主軸番号を指定していない場合、または主軸番号を「0」に設定している場合は、SPOS または SPOSA はメイン主軸に適用されます。		
<値>:	主軸回転位置を設定します。		
	単位:	°	
	タイプ:	REAL	
	次のオプションを、位置アプローチモードのプログラム指令に使用できます。		
	=AC (<値>):	アブソリュート指令	
		値の範囲:	0 ... 359.9999
	=IC (<値>):	インクレメンタル指令	
		値の範囲:	0 ... ±99 999.999
	=DC (<値>):	絶対値へ直接にアプローチ	
	=ACN (<値>):	アブソリュート指令、負方向へアプローチ	
	=ACP (<値>):	アブソリュート指令、正方向へアプローチ	
	=<値>:	DC(<値>)と同じ	
M<n>=19:	メイン主軸(M19 または M0=19)または番号<n> (M<n>=19)の主軸へ、SD43250 \$SA_M19_SPOSMODE に設定された位置アプローチモードで、SD43240 \$SA_M19_SPOS に設定された回転位置を指令します。 NC ブロックは、設定位置に達するまでは有効になりません。		

3.7 送り速度制御

M<n>=70:	メイン主軸(M70 または M0=70)または番号<n> (M<n>=70)の主軸を軸モードに切り替えます。 定義位置へはアプローチしません。NC ブロックは、切り替えがおこなわれた後に有効になります。			
FINEA:	「精密イグザクトストップ」に達すると動作終了			
COARSEA:	「汎用イグザクトストップ」に達すると動作終了			
IPOENDA:	「補間停止」に達すると動作終了			
S<n>:	プログラム指令動作終了条件が有効となる主軸			
	<n>:	主軸番号		
	主軸が[S<n>]で指定されていない場合、または「0」の主軸番号が指定されている場合は、プログラム指令動作終了条件はメイン主軸に適用されます。			
IPOBRKA:	減速カーブでブロックの切り替えが可能です。			
	<軸>:	チャンネル軸識別子		
	<タイミング>:	減速カーブによるブロック切り替えのタイミング		
		単位:	パーセント	
		値の範囲:	100 (減速カーブの適用点)～0 (減速カーブの終了)	
		値を<タイミング>パラメータに割り当てない場合は、現在の次のセッティングデータ値が適用されます。 SD43600 \$SA_IPOBRAKE_BLOCK_EXCHANGE 注: タイミングが「0」の IBOBRKA は IPOENDA と同じです。		

WAITS:	指定主軸の同期制御命令 以降のブロックは、以前の NC ブロックで SPOSA を使用してプログラム指令した主軸が、その (精密イグザクトストップの) 終了位置に達するまで処理されません。	
	M5 の後の WAITS:	指定した主軸が停止状態になるまで待機します。
	M3/M4 の後の WAITS:	指定した主軸が指令速度に達するまで待機します。
	<n>, <m>:	同期制御命令が適用される主軸番号です。 主軸番号を指定していない場合、または主軸番号を「0」に設定している場合は、WAITS はメイン主軸に適用されます。

注記

それぞれの NC ブロックに対して 3 種類の主軸位置を設定できます。

注記

インクレメンタル指令 IC(<値>)では、主軸の位置決めは複数回転にわたって行うことができます。

注記

SPOS の前に、位置制御が SPCON で有効になっている場合は、位置制御は SPCOF が発行されるまで有効です。

注記

制御装置は、プログラム処理から、軸モードへの移行を自動的に検出します。したがって、パートプログラムで別途 M70 のプログラミングをおこなう必要はありません。ただし、パートプログラムを読みやすくするなどの目的で、M70 のプログラム指令を行うことができます。

詳細情報**SPOSA による位置決め**

ブロック切り替えの有効化とプログラムの実行のいずれも、SPOSA には影響されません。主軸の位置決めは、以降の NC ブロックの実行中におこなうことができます。プログラムは、実行中のブロックでプログラム指令したすべての機能(主軸を除く)がそのブロック終

了条件に到達すると、次のブロックに移動します。主軸位置決め動作を、複数のブロックにわたってプログラム指令できます(WAITS を参照)。

注記

自動的に先読み停止となる命令が後続のブロックで読み込まれた場合、後続のブロックは、位置決めされるすべての主軸が停止状態になった後に実行されます。

SPOS/M19 による位置決め

ブロック切り替え有効化条件は、ブロックにプログラム指令されたすべての機能がそのブロック終了条件に達し(すべての補助機能が PLC で応答された、すべての軸が終点に達した、など)、主軸がプログラム指令位置に到達したときに満たされます。

移動速度:

位置決めの速度と遅延動作は、マシンデータに格納されます。設定値は、プログラミングまたはシンクロナイズドアクションで変更できます。以下を参照してください。

- 位置決め軸/主軸の送り速度(FA、FPR、FPRAON、FPRAOF) (ページ 139)
- プログラマブル加減速制御補正(ACC) (ページ 144)

主軸位置の指定:

G90/G91 命令はここでは有効でないため、AC、IC、DC、ACN、ACP などの、対応する指令が別途適用されます。何も指定しない場合、移動は自動的に DC として実行されます。

WAITS による主軸移動の同期化

WAITS を使用して、これ以前の NC ブロックで SPOSA を使用してプログラム指令された複数の主軸がその位置に到達するまで NC プログラムが待機している位置を、指定できます。

例:

プログラムコード	コメント
N10 SPOSA[2]=180 SPOSA[3]=0	
...	
N40 WAITS (2,3)	; ブロックは、主軸 2 と主軸 3 が、N10 ブロックで指定された位置に到達するまで待機します。

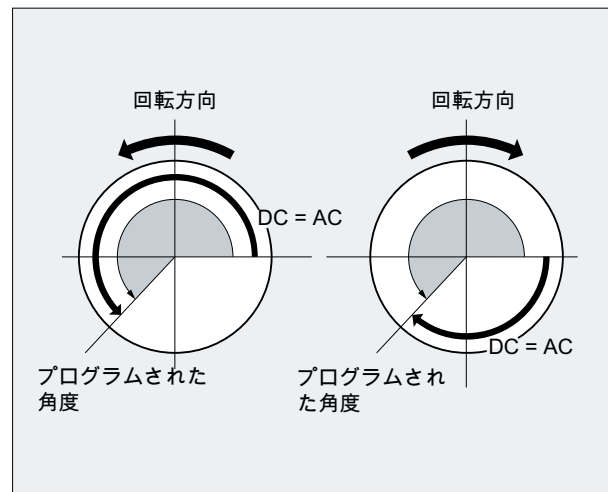
WAITS を M5 の後に使用すると、主軸が停止するまで待機できます。WAITS を M3/M4 の後に使用すると、指定した回転速度/回転方向に主軸が到達するまで待機できます。

注記

主軸が同期制御マークで原点同期済みでない場合は、マシンデータ(出荷時の状態)から正の回転方向が取得されます。

回転中(M3/M4)からの主軸の位置決め

M3 または M4 が動作中のときは、主軸がプログラム指令値で停止状態になります。



DC と AC の間で位置指令方法に違いはありません。両方の場合とも、M3/M4 で選択した方向に、アブソリュート終了位置に到達するまで回転が続行されます。ACN と ACP の場合は、必要に応じて減速し、当該のアプローチ方向になります。IC の場合は、主軸が現在の主軸位置からさらに、指定値だけ回転します。

停止状態(M5)からの主軸の位置決め

停止状態 (M5) からは、プログラム指令距離だけ正確に移動します(M5)。

3.7.5 位置決め軸/主軸の送り速度(FA、FPR、FPRAON、FPRAOF)

軌跡軸と同期軸の毎回転送り速度、または別の回転軸または主軸から、個々の位置決め軸/主軸の毎回転送り速度を、算出することもできます。

ワーク搬送システム、工具タレット、および支持台などの位置決め軸は、軌跡軸、および同期軸とは無関係に移動します。したがって、用途別送り速度が各位置決め軸に定義されます。

用途別軸送り速度は、主軸にもプログラム指令できます。

3.7 送り速度制御

構文

位置決め軸の送り速度

FA[<軸>]=...

主軸の軸送り速度:

FA[SPI(<n>)] =...

FA[S<n>] =...

軌跡/同期軸の毎回転送り速度の算出:

FPR (<回転軸>)

FPR (SPI (<n>))

FPR (S<n>)

位置決め軸/主軸の毎回転送り速度の算出:

FPRAON (<軸>, <回転軸>)

FPRAON (<軸>, SPI (<n>))

FPRAON (<軸>, S<n>)

FPRAON (SPI (<n>), <回転軸>)

FPRAON (S<n>, <回転軸>)

FPRAON (SPI (<n>), SPI (<n>))

FPRAON (S<n>, S<n>)

FPRAOF (<軸>, SPI (<n>), ...)

FPRAOF (<軸>, S<n>, ...)

意味

FA[...]=...:	指定した位置決め軸の送り速度、または指定した主軸の位置決め速度(軸送り速度)	
	単位:	mm/min または inch/min または °/min
	値の範囲:	... 999,999.999 mm/min、°/min ... 39 999.9999 inch/min
FPR(...):	FPR を使用して、G95 でプログラム指令された軌跡軸と同期軸の毎回転送り速度の算出に使用される回転軸(<回転軸>)または主軸(SPI (<n>)/S<n>)を指定します。	

FPRAON (...):	位置決め軸と主軸の毎回転送り速度の算出: 1 番目のパラメータ(<軸>/SPI (<n>)/S<n>)は、毎回転送り速度で移動する位置決め軸/主軸を指定します。 2 番目のパラメータ(<回転軸>/SPI (<n>)/S<n>)は、毎回転送り速度の算出に使用される回転軸/主軸を指定します。 注: 2 番目のパラメータは省略できますが、その場合は、送り速度はメイン主軸から算出されます。	
FPRAOF (...):	FPRAOF を使用して、指定した軸または主軸の毎回転送り速度の算出を、選択解除します。	
<軸>:	軸識別子(位置決め軸またはジオメトリ軸)	
SPI (<n>)/S<n>:	主軸識別子 SPI (<n>) と S<n>の機能は同じです。	
	<n>:	主軸番号
	注: SPI は、主軸番号を軸識別子に変換します。転送パラメータ (<n>) には、有効な主軸番号を入れてください。	

注記

プログラム指令送り速度 FA[...]はモーダルです。

位置決め軸または主軸の送り速度は、各 NC ブロックに 5 つまでプログラム指令できます。

注記

送り速度の算出は次の公式に従って計算されます。

送り速度の算出値 = プログラム指令送り速度 * マスタのアブソリュート送り速度

例**例 1:主軸同期連結**

主軸同期連結の場合、スレーブ主軸の位置決め速度を、メイン主軸とは無関係にプログラム指令して、位置決め動作などを実行できます。

プログラムコード	コメント
...	
FA[S2]=100	; スレーブ主軸(主軸 2)の位置決め速度= 100°/min
...	

例 2 :軌跡軸の毎回転送り速度の算出

軌跡軸 X、Y は、回転軸 A から算出される毎回転送り速度で移動します。

プログラムコード

```
...
N40 FPR (A)
N50 G95 X50 Y50 F500
...
```

例 3 :メイン主軸の毎回転送り速度の算出

プログラムコード	コメント
N30 FPRAON (S1, S2)	; メイン主軸 (S1) の毎回転送り速度は、主軸 2 から算出します。
N40 SPOS=150	; メイン主軸を位置決めします。
N50 FPRAOF (S1)	; メイン主軸の毎回転送り速度を選択解除します。

例 4 :位置決め軸の毎回転送り速度の算出

プログラムコード	コメント
N30 FPRAON (X)	; 位置決め軸 X の毎回転送り速度は、メイン主軸から算出します。
N40 POS[X]=50 FA[X]=500	; 位置決め軸は、メイン主軸の 500 mm/rev で移動しています。
N50 FPRAOF (X)	

詳細情報**FA[...]**

送り速度タイプは常に G94 です。G70/G71 が有効なときは、マシンデータの初期設定に従い、単位がメトリック/インチとなります。G700/G710 を使用して、プログラムで使われる単位を変更できます。

注記

FA をプログラム指令しない場合は、マシンデータで定義した値が適用されます。

FPR(...)

G95 命令(メイン主軸を基準とする毎回転送り速度)の拡張機能である FPR を使用すると、選択した任意の主軸または回転軸からも毎回転送り速度を算出できます。G95 FPR (...) は、軌跡軸と同期軸に適用されます。

FPR 命令で指定した回転軸/主軸が位置制御で動作している場合は、指令値連結が有効となります。それ以外の場合は、現在値連結が有効です。

FPRAON(...)

FPRAON を使用して、位置決め軸と主軸の毎回転送り速度を、別の回転軸または主軸の現在の送り速度から算出します。

FPRAOF(...)

毎回転送り速度は、FPRAOF 命令で、複数の軸/主軸に対して同時に無効にすることができます。

3.7.6 プログラマブル送り速度オーバーライド(OVR、OVERRAP、OVRA)

軌跡軸/位置決め軸、および主軸の速度は、NC プログラムで変更できます。

構文

```
OVR=<値>
OVERRAP=<値>
OVRA [<軸>]=<値>
OVRA [SPI (<n>)] =<値>
OVRA [S<n>]=<値>
```

意味

OVR:	軌跡送り速度 F の送り速度変更	
OVERRAP:	早送り速度の送り速度変更	
OVRA:	位置決め送り速度 FA または主軸速度 S の送り速度変更	
<軸>:	軸識別子(位置決め軸またはジオメトリ軸)	
SPI (<n>)/S<n>:	主軸識別子	
	SPI (<n>) と S<n>の機能は同じです。	
	<n>:	主軸番号
注:		
SPI は、主軸番号を軸識別子に変換します。転送パラメータ (<n>) には、有効な主軸番号を入れてください。		

3.7 送り速度制御

<値>:	送り速度変更(% 単位)	
	この値は、この送り速度変更を基準にするか、機械操作パネルで設定された送り速度オーバライドとの組み合わせになります。	
	値の範囲:	0 ... 200%、積分
注: 跡オーバライドと早送りオーバライドによって、マシンデータで設定された最大速度を超えることはありません。		

3.7.7 プログラマブル加減速制御補正(ACC)

重要なプログラム区間では、機械的振動の発生防止などのために、加減速を最大値より低く制限することが必要な場合があります。

プログラマブル加減速制御オーバライドを使用して、各軌跡軸または各主軸の加減速を、NC プログラムの命令で変更できます。制限は、すべてのタイプの補間に有効です。マシンデータで定義した値は、100%の加減速として適用されます。

構文

```
ACC [<軸>]=<値>
ACC [SPI (<n>)] =<値>
ACC (S<n>) =<値>
```

解除:

```
ACC [...]=100
```

構文

ACC:	指定した軌跡軸の加減速の変更、または指定した主軸速度の変更。	
<軸>:	軌跡軸のチャネル軸名称	
SPI (<n>) / S<n>:	主軸識別子	
	SPI (<n>) と S<n>の機能は同じです。	
	<n>:	主軸番号
注: SPI は、主軸番号を軸識別子に変換します。転送パラメータ (<n>) には、有効な主軸番号を入れてください。		

<値>:	加減速の変更(% 単位)	
	この値は、この加減速の変更を基準にするか、機械操作パネルで設定された送り速度オーバーライドとの組み合わせになります。	
	値の範囲:	1～200%、整数

注記

加速度が大きい場合は、工作機械メーカーの許容値を超える可能性があります。

例

プログラムコード	コメント
N50 ACC[X]=80	; X 方向の軸送り台は、80%の加減速で移動します。
N60 ACC[SPI(1)]=50	; 主軸 1 は、加減速能力の 50%で、加速または減速をおこないます。

詳細情報**ACC でプログラム指令された加減速制御オーバーライド**

ACC[...]でプログラム指令された加減速制御オーバーライドは、常時、システム変数\$AA_ACC へ出力処理がおこなわれます。パートプログラムとシンクロナイズドアクションでの読み出しは、NC 実行処理とは別のタイミングでおこなわれます。

パートプログラムで

書き込み動作中に ACC がシンクロナイズドアクションで変更されていない場合にのみ、システム変数\$AA_ACC が、パートプログラムでの書き込みとして処理されます。

シンクロナイズドアクションの場合

したがって、下記が適用されます。書き込み動作中に ACC がパートプログラムにより変更されていない場合にのみ、システム変数\$AA_ACC が、シンクロナイズドアクションでの書き込みとして処理されます。

シンクロナイズドアクションを使用して、指定された加減速を変更することもできます。

例:

プログラムコード
...
N100 EVERY \$A_IN[1] DO POS[X]=50 FA[X]=2000 ACC[X]=140

実際の加減速値をシステム変数\$AA_ACC[<軸>]で呼び出すことができます。マシンデータを使用して、最後に設定された ACC 値をリセット/パートプログラムの終了時に適用するか、100%に設定するかを特定できます。

3.7.8 ハンドルオーバーライドによる送り速度(FD、FDA)

FD と FDA 命令を使用して、パートプログラムの実行中に手動パルス発生器で軸を移動できます。軸移動用のプログラム指令の設定に、軌跡入力または速度入力として読み込まれた手動パルス発生器のパルスが重畳されます。

軌跡軸

軌跡軸の場合は、プログラム指令軌跡送り速度を重畳できます。手動パルス発生器は、チャンネルの 1 番目のジオメトリ軸として使用されます。回転方向に応じて補間サイクルごとに読み込まれた手動パルス発生器のパルスは、重畳する軌跡速度に対応します。ハンドルオーバーライドを使用して得られる軌跡速度の制限値は、次のとおりです。

- 最小値 : 0
- 最大値 : 移動に関連する軌跡軸のマシンデータ制限値

注記

軌跡送り速度

軌跡送り速度 F とハンドル送り速度 FD は、同じ NC ブロックではプログラム指令できません。

位置決め軸

位置決め軸の場合は、移動軌跡または移動速度を、軸の値として重畳できます。軸に割り当てられた手動パルス発生器が使用されます。

- 軌跡オーバーライド
回転方向に応じて読み込まれた手動パルス発生器のパルスが、移動する軸軌跡に対応します。プログラム指令位置方向の手動パルス発生器のパルスのみ読み込まれます。
- 速度オーバーライド
回転方向に応じて補間サイクルごとに読み込まれた手動パルス発生器のパルスが、重畳する軸速度に対応します。ハンドルオーバーライドを使用して得られる軌跡速度の制限値は、次のとおりです。
 - 最小値 : 0
 - 最大値 : 位置決め軸のマシンデータ制限値

構文

FD=<速度>

FDA [<軸>] =<速度>

意味

FD=<速度>:

軌跡送り速度、および手動パルス発生器による速度オーバライドの有効化

<速度>:

- 値= 0:許可されません。
- 値≠ 0:軌跡速度

FDA [<軸>] =<速度>:

軸の送り速度

<速度>:

- 値= 0:手動パルス発生器による軌跡指令
- 値≠ 0:軸の速度

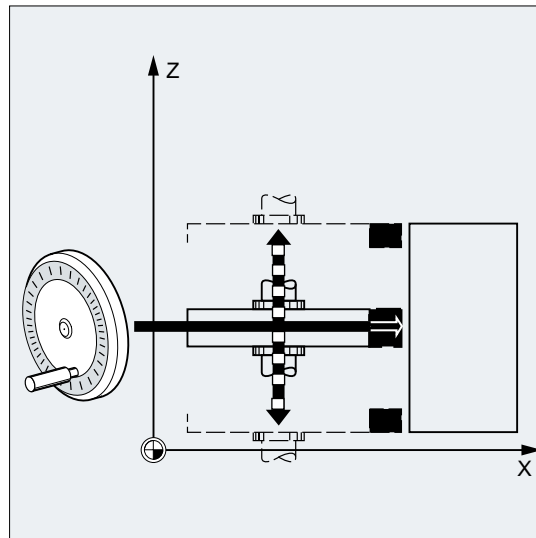
<軸>:

位置決め軸の軸識別子

注記

FD と FDA はノンモーダルです。

例



軌跡の定義:Z 方向に揺動する砥石は、手動パルス発生器で X 方向のワークへと移動します。

3.7 送り速度制御

オペレータは、スパークが均一に飛ぶまで、手動で送り込むことができます。「残移動距離削除」を有効にすると、次の NC ブロックに切り替わり、加工は AUTOMATIC モードで続行されます。

詳細情報

速度オーバーライド(FD=<速度>)による軌跡軸の移動

軌跡速度オーバーライドがプログラム指令されているパートプログラムのブロックでは、次の条件を満たしてください。

- 軌跡命令 G1、G2、または G3 が有効
- イグザクトストップ G60 が有効
- 毎分送り G94 が有効

送り速度オーバーライド

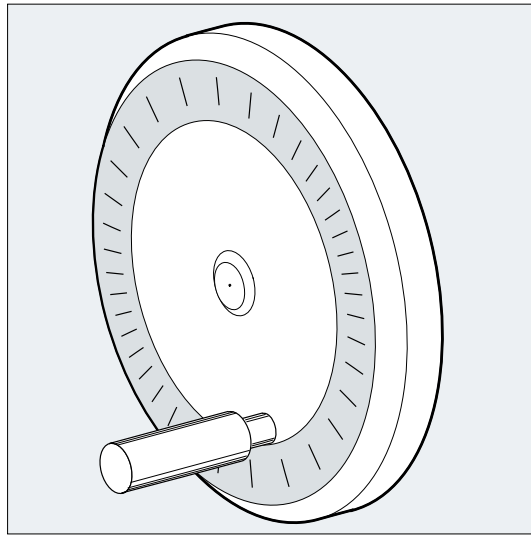
送り速度オーバーライドは、プログラム指令軌跡速度のみに影響し、手動パルス発生器により生成される速度成分には影響しません(例外: 送り速度オーバーライド = 0)。

例:

プログラムコード	内容/説明
N10 X... Y... F500	; 送り速度= 500 mm/min
N20 X... Y... FD=700	; 送り速度= 700 mm/min、および手動パルス発生器による 速度オーバーライド ; N20 で 500 mm/min から 700 mm/min への加速。手動パルス発生器を 使用して、回転方向に応じて、0 〜最大値 (マシンデータ) の範囲で速度を変更できます。

軌跡指令(FDA[<軸>]=0)による位置決め軸の移動

FDA[<軸>]=0 がプログラム指令された NC ブロックでは、プログラムによってどのような移動動作も生成されないように、送り速度がゼロに設定されます。これで、プログラム指令目標位置への移動動作は、手動パルス発生器を回転させるオペレータによってのみ制御されます。



例:

プログラムコード	内容/説明
...	
N20 POS[V]=90 FDA[V]=0	; 目標位置= 90 mm、軸の送り速度= 0 mm/min、および 手動パルス発生器による軌跡オーバーライド。 ; ブロックの先頭の軸 V の速度= 0 mm/min。 ; 軌跡指令と速度指令は手動パルス発生器のパルスを使用して設定されます。

動作の方向、移動速度

軸は、手動パルス発生器の符号方向によって設定された軌跡に追従します。回転方向に応じて、前後への移動できます。手動パルス発生器が高速で回転するほど、移動速度が増加します。

移動範囲:

移動範囲は、開始位置とプログラム指令終点によって制限されます。

速度オーバーライドによる位置決め軸の移動(FDA[<軸>]=<速度>)

FDA[...]=...がプログラム指令された NC ブロックでは、最後のプログラム指令 FA 値による送り速度が、FDA でプログラム指令された値へ加速または減速されます。手動パルス発生器を回転して、現在の送り速度 FDA から、プログラム指令動作を目標位置に加速するか、ゼロに減速することができます。マシンデータのパラメータとして設定した値は、最大速度として機能します。

例:

プログラムコード	内容/説明
N10 POS[V]=... FA[V]=100	; 軸の送り速度= 100 mm/min

3.7 送り速度制御

プログラムコード	内容/説明
N20 POS[V]=100 FAD[V]=200	; 軸の目標位置= 100、軸の送り速度= 200 mm/min ; および手動パルス発生器による速度オーバーライド。 ; N20 で 100 mm/min から 200 mm/min への加速。 ; 手動パルス発生器を使用して、回転方向に応じて ; 0 ～最大値 (マシンデータ) の範囲で速度を変更できます。

移動範囲:

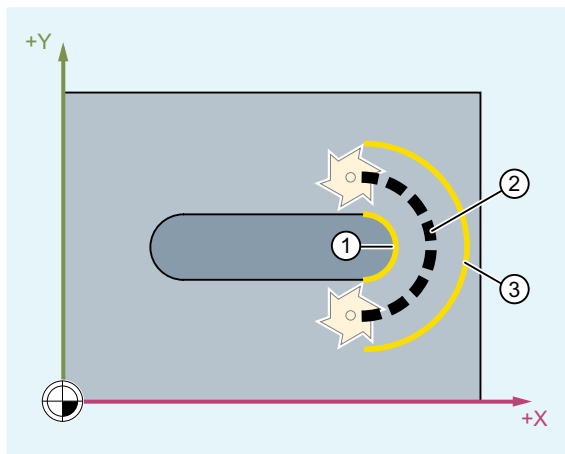
移動範囲は、開始位置とプログラム指令終点によって制限されます。

3.7.9 曲線軌跡区間の送り速度のオートチューニング(CFTCP、CFC、CFIN)

補正モード G41/G42 が有効な場合、プログラムしたフライス工具半径の送り速度は、最初はフライス工具中心軌跡が基準となります(「座標変換(フレーム) (ページ 340)」の章を参照してください)。

円弧のフライス加工(多項式補間とスプライン補間の場合も同様です)時には、特定の環境では、送り速度が刃先でどのくらい変化するかが、非常に重要です。これは、それによって加工部品の品質が損なわれる場合があるためです。

例:大径工具による小さい外側半径のフライス加工。フライス工具の外側が、輪郭の周囲よりかなり長い軌跡を移動します。



このため、輪郭の加工は非常に低い送り速度で行われます。その弊害を防ぐため、輪郭曲線に応じた送り速度の制御が必要です。

構文

CFTCP
CFC

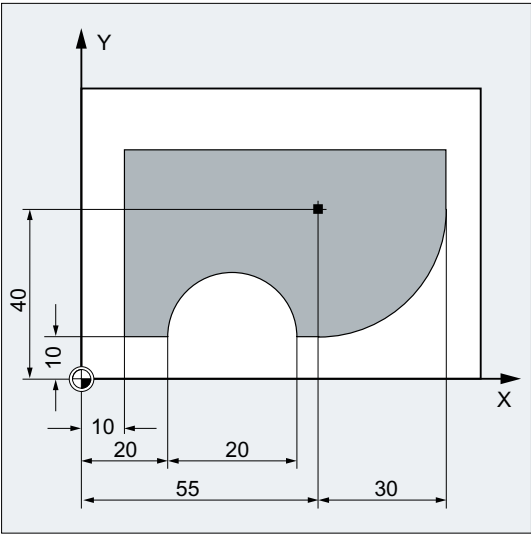
CFIN

意味

CFTCP:	フライス工具中心軌跡の送り速度が一定 コントローラによって送り速度が一定に保たれ、送り速度オーバライドは無効になります。
CFC:	輪郭(工具刃先)で送り速度が一定。 この機能が初期設定として設定されています。
CFIN:	凹型輪郭のときのみ工具刃先の送り速度が一定。それ以外の場合は、フライス工具中心軌跡で送り速度が一定。 送り速度は、内側半径では減速されます。

例

この例では、輪郭がまず、CFC 補正速度で加工されます。仕上切削のときは、切削加工ベースも CFIN により加工されます。これにより、送り速度が速すぎるために、外側半径で切削加工ベースが損傷することを防止します。



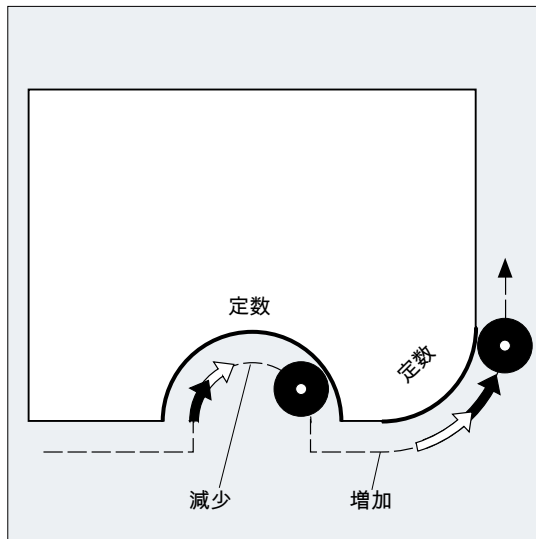
プログラムコード	コメント
N10 G17 G54 G64 T1 M6	
N20 S3000 M3 CFC F500 G41	
N30 G0 X-10	
N40 Y0 Z-10	; 1 番目の切削深さまで送り込みます
N50 CONTOUR1	; サブプログラム呼び出し

3.7 送り速度制御

プログラムコード	コメント
N40 CFIN Z-25	; 2 番目の切削深さまで送り込みます
N50 CONTOUR1	; サブプログラム呼び出し
N60 Y120	
N70 X200 M30	

詳細情報

CFC によって輪郭の送り速度が一定



送り速度は内側半径では減速され、外側半径では加速されます。この指令により、工具刃先の速度が確実に一定に保たれ、したがって、輪郭でも一定に保たれます。

3.7.10 1 ブロックの複数送り速度値(F、ST、SR、FMA、STA、SRA)

「1 ブロックの複数送り速度値」機能を使用すると、外部デジタルかアナログ入力、または両方に応じて、1 つの NC ブロック、ドウェル時間、または後退移動に同期した異なる送り速度値を有効にすることができます。

構文

軌跡移動

F=...F7=...F6=...F5=...F4=...F3=...F2=...ST=...SR=...

軸移動:

FA[<軸>]=...FMA[7,<軸>]=...FMA[6,<軸>]=...FMA[5,<軸>]=...FMA[4,<軸>]=...FMA[3,<軸>]=...FMA[2,<軸>]=...STA[<軸>]=...SRA[<軸>]=...

意味

F=... :	軌跡送り速度はアドレス F でプログラム指令され、入力信号がないときは有効です。	
	効果:	モーダル
F2=...~F7=... :	軌跡送り速度に加えて、さらに 6 つまでの送り速度をこのブロックにプログラム指令できます。数字の部分は入力ビット番号を示し、これにより、変更時に送り速度が有効になります。	
	効果:	ノンモーダル
ST=...	秒(s) 単位のドウェル時間(研削加工の場合: スパークアウト時間)	
	入力ビット番号:	1
	効果:	ノンモーダル
SR=...	後退軌跡 後退軌跡の単位は、現在有効な単位系 (mm または inch) を基準にします。	
	入力ビット番号:	0
	効果:	ノンモーダル
FA [<Ax>]=... :	軸送り速度がアドレス FA にプログラム指令され、入力信号がないときは、軸送り速度がそのまま有効です。	
	効果:	モーダル
FMA [2, <Ax>]=... ~ FMA [7, <Ax>]=... :	軸送り速度 FA に加えて、軸毎にさらに 6 つまでの送り速度を、FMA で、このブロックにプログラム指令できます。1 番目のパラメータは入力ビット番号を示し、2 番目のパラメータは送り速度が適用される軸を示します。	
	効果:	ノンモーダル

3.7 送り速度制御

STA [<Ax>] = . . . :	秒(s) 単位の軸ドウェル時間(研削加工の場合:スパークアウト時間)	
	入力ビット番号:	1
	効果:	ノンモーダル
SRA [<Ax>] = . . . :	軸の後退軌跡	
	入力ビット番号:	0
	効果:	ノンモーダル
<Ax>:	送り速度を適用する軸	

注記

信号の優先度

信号は、入力ビット 0 (I0) を先頭に昇順でスキャンされます。したがって、後退移動の優先度は最高、送り速度 F7 の優先度は最低になります。ドウェル時間と後退移動は、F2～F7 によって有効にされた送り速度移動を終了します。

優先度が最高の信号により、現在の送り速度が決定します。

注記

残移動距離削除

入力ビット番号 1 がドウェル時間に対して有効な場合、または入力ビット番号 0 が戻り軌跡に対して有効な場合は、軌跡軸または当該の単独軸の残移動距離が削除され、ドウェル時間または戻り軌跡が開始されます。

注記

後退軌跡

後退軌跡の単位は、現在有効な単位系(mm または inch)を基準にします。

常に逆のストロークが、現在の移動に対して反対の方向で発生します。SR/SRA は、常に逆のストロークの値をプログラム指令します。符号はプログラム指令しません。

注記

POSA ではなく POS

外部入力により軸の送り速度、ドウェル時間、または戻り軌跡をプログラム指令する場合は、このブロックでは、この軸を POSA 軸(複数ブロックにわたる位置決め軸)としてプログラム指令しないでください。

注記**状態確認**

さまざまな軸の同期命令に対する入力状態をポーリングすることもできます。

注記**先読み**

1 ブロックの複数送り速度には、先読みも有効となります。これにより、現在の送り速度を先読み値によって制限できます。

例**例 1: 軌跡移動**

プログラムコード	コメント
G1 X48 F1000 F7=200 F6=50 F5=25 F4=5 ST=1.5 SR=0.5	; 軌跡送り速度 = 1000 ; 追加の軌跡送り速度値: ; 200 (入力ビット 7) ; 50 (入力ビット 6) ; 25 (入力ビット 5) ; 5 (入力ビット 4) ; ドウエル時間 1.5 s ; 後退 0.5 mm

例 2: 軸移動

プログラムコード	コメント
POS[A]=300 FA[A]=800 FMA[7,A]=720 FMA[6,A]=640 FMA[5,A]=560 STA[A]=1.5 SRA[A]=0.5	; 軸 A の送り速度 = 800 ; 軸 A の追加の送り速度値: 720 (入力ビット 7) ; 640 (入力ビット 6) ; 560 (入力ビット 5) ; 軸ドウエル時間: 1.5 秒 ; 軸後退: 0.5 mm

例 3: 1 ブロックの複数命令

プログラムコード	コメント
N20 T1 D1 F500 G0 X100	初期設定

3.7 送り速度制御

プログラムコード	コメント
N25 G1 X105 F=20 F7=5 F3=2.5 F2=0.5 ST=1.5 SR=0.5	; Fによる通常の送り速度、 ; F7による荒削り、 ; F3による仕上げ、 ; F2による滑らかな仕上げ、 ; ドウェル時間 1.5 s、 ; 後退距離 0.5 mm
...	

3.7.11 ノンモーダル送り速度(FB)

「ノンモーダル送り速度」機能を使用して、1つのブロックに用途別送り速度を定義できます。このブロックの後は、その前のモーダル送り速度が再び有効になります。

構文

FB=<値>

意味

FB:	実行中のブロックのみの送り速度
<値>:	プログラム指令値はゼロより大きい値にしてください。 値は、動作中の送り速度タイプに基づいて解釈されます。 • G94:送り速度(mm/min または °/min 単位) • G95:送り速度(mm/rev または inch/rev 単位) • G96:定切削速度

注記

ブロックに移動動作をプログラム指令していない場合(計算ブロックなど)は、FBは無効となります。

面取り/丸み付けの送り速度を別途プログラム指令していない場合、FBの値は、このブロックの輪郭要素の面取り/丸み付けのいずれにも適用されます。

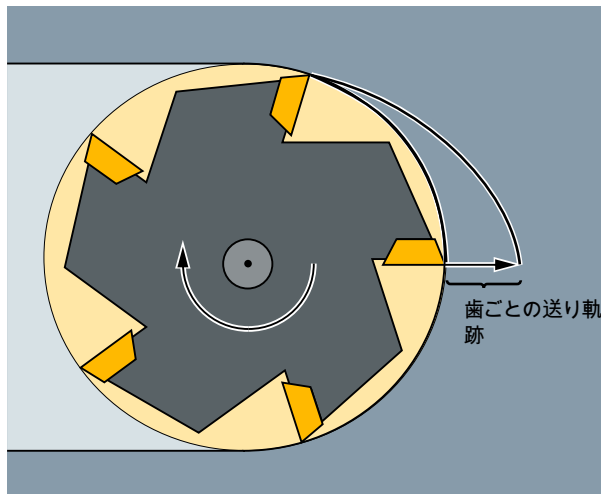
送り速度補間 FLIN、FCUB など、制限なく使用できます。

FB と、FD(送り速度オーバーライドによる手動パルス発生器の移動)または F(モーダル軌跡送り速度)は、同時にプログラミングできません。

例

プログラムコード	コメント
N10 G0 X0 Y0 G17 F100 G94	; 初期設定
N20 G1 X10	; 送り速度 100 mm/min
N30 X20 FB=80	; 送り速度 80 mm/min
N40 X30	; 送り速度は再び 100 mm/min になります。
...	

3.7.12 1 刃当り送り速度(G95 FZ)



コントローラは、動作中の工具オフセットデータブロックに関連した\$TC_DPNT (刃数)工具パラメータを使用して、プログラム指令した 1 刃当り送り速度から、各移動ブロックの有効毎回転送り速度を計算します。

$$F = FZ * \$TC_DPNT$$

意味: F: 毎回転送り速度(mm/rev または inch/rev)

FZ: 1 刃当り送り速度(mm/tooth または inch/tooth)

\$TC_DPNT: システム変数工具パラメータ:刃数/rev

動作中の工具の工具タイプ(\$TC_DP1)は考慮されません。

プログラム指令した 1 刃当り送り速度は、工具交換にも工具オフセットデータブロックの選択/選択解除にも依存せずに、モーダル形式で保持されます。

動作中の工具刃先に関連した\$TC_DPNT 工具パラメータの変更は、次回の工具オフセットの選択時、または次回の有効なオフセットデータ指令の更新時に適用されます。

3.7 送り速度制御

工具交換または工具オフセットデータブロックの選択/選択解除をおこなうと、有効毎回転送り速度の再計算がおこなわれます。

注記

1 刃当り送り速度は軌跡のみを対象とします(軸別のプログラミングはできません)。

構文

G95 FZ...

意味

G95:	送り速度タイプ:mm/rev または inch/rev 単位の毎回転送り速度(G700/G710 に依存します) G95 については「送り速度(G93、G94、G95、F、FGROUP、FL、FGREF) (ページ 119)」を参照してください。	
FZ:	1 刃当り送り速度	
	適用タイミング:	G95 を使用
	有効性:	モーダル
	単位:	mm/tooth または inch/tooth(G700/G710 に依存します)

通知

工具交換/メイン主軸の交換

ユーザーでは、以降におこなわれる工具交換またはメイン主軸の交換では、FZ の再プログラミングなどのような、対応プログラミングを考慮してください。

通知

工具動作未定義

軌跡形状(直線、円弧など)と同様に、下向きフライス削りまたは上向きフライス削り、正面フライス削りまたは側面フライス削りなどの加工種別関連事項は、自動的に考慮されません。したがって、1 刃当り送り速度のプログラム指令時には、これらの要素を考慮してください。

注記

G95 F...と G95 FZ...の切り替え

G95 F... (回転送り速度)と G95 FZ... (1 刃当り送り速度)の切り替えにより、無効な送り速度値がそのたびに削除されます。

注記

FPR で送り速度を算出

毎回転送り速度の場合と同様に、FPR を使用して、任意の回転軸または主軸の 1 刃当り送り速度を算出することもできます(「位置決め軸/主軸の送り速度(FA、FPR、FPRAON、FPRAOF) (ページ 139)」を参照してください)。

例

例 1:5 刃のフライスカッター(\$TC_DPNT = 5)

プログラムコード	コメント
N10 G0 X100 Y50	
N20 G1 G95 FZ=0.02	; 1 刃当り送り速度 0.02 mm/tooth
N30 T3 D1	; 工具を装着して、工具オフセットデータブロックを有効にします。
M40 M3 S200	; 主軸速度 200 rpm
N50 X20	; フライス削りの 1 刃当り送り速度:
	FZ = 0.02 mm/tooth
	有効毎回転送り速度:
	$F = 0.02 \text{ mm/tooth} * 5 \text{ 刃/rev} = 0.1 \text{ mm/rev}$
	または
	$F = 0.1 \text{ mm/rev} * 200 \text{ rpm} = 20 \text{ mm/min}$
...	

例 2:G95 F...と G95 FZ...の切り替え

プログラムコード	コメント
N10 G0 X100 Y50	
N20 G1 G95 F0.1	; 毎回転送り速度 0.1 mm/rev
N30 T1 M6	
N35 M3 S100 D1	
N40 X20	
N50 G0 X100 M5	
N60 M6 T3 D1	; 例: 5 刃の工具を装着します (\$TC_DPNT = 5)。
N70 X22 M3 S300	
N80 G1 X3 G95 FZ=0.02	; G95 F... を G95 FZ...に変更、1 刃当り送り速度が 0.02 mm/tooth で有効になります。
...	

3.7 送り速度制御

例 3:主軸の 1 刃当り送り速度を算出(FBR)

プログラムコード	コメント
...	
N41 FPR (S4)	; 主軸 4 (メイン主軸ではありません) の工具。
N51 G95 X51 FZ=0.5	; 主軸 S4 に応じた 1 刃当り送り速度 0.5 mm/tooth。
...	

例 4:工具交換がある場合

プログラムコード	コメント
N10 G0 X50 Y5	
N20 G1 G95 FZ=0.03	; 1 刃当り送り速度 0.03 mm/tooth
N30 M6 T11 D1	; 例: 7 刃の工具を装着します (\$TC_DPNT = 7)。
N30 M3 S100	
N40 X30	; 有効毎回転送り速度 0.21 mm/rev
N50 G0 X100 M5	
N60 M6 T33 D1	; 例: 5 刃の工具を装着します (\$TC_DPNT = 5)。
N70 X22 M3 S300	
N80 G1 X3	; モーダル 1 刃当たり送り速度 0.03 mm/tooth、 有効毎回転送り速度 0.15 mm/rev
...	

例 5:メイン主軸の交換

プログラムコード	コメント
N10 SETMS (1)	; 主軸 1 がメイン主軸です。
N20 T3 D3 M6	; 工具 3 が主軸 1 に交換されます。
N30 S400 M3	; 主軸 1 の速度は S400 です (そして、これは T3 の速度です)。
N40 G95 G1 FZ0.03	; 1 刃当り送り速度 0.03 mm/tooth
N50 X50	; 軌跡移動、有効送り速度は次の項目に応じて異なります。 - 1 刃当り送り速度 FZ - 主軸 1 の速度 - 動作中の工具 T3 の刃数
N60 G0 X60	
...	
N100 SETMS (2)	; 主軸 2 がメイン主軸になります。
N110 T1 D1 M6	; 工具 1 が主軸 2 に交換されます。
N120 S500 M3	; 主軸 2 の速度は S500 です (そして、これは T1 の速度です)。
N130 G95 G1 FZ0.03 X20	; 軌跡移動、有効送り速度は次の項目に応じて異なります。 - 1 刃当り送り速度 FZ - 主軸 2 の速度 - 動作中の工具 T1 の刃数

注記

メイン主軸の交換(N100)後、主軸 2 による回転工具を交換してください(N110)。

詳細情報**G93、G94、および G95 の切り替え**

G95 が有効でないときにも、FZ はプログラム指令できます。ただし、効果はまったくなく、G95 を選択すると、解除されます。言い換えると、G93、G94、および G95 のいずれかに切り替えると、F と同様に、FZ 値も解除されます。

G95 の再選択

G95 がすでに有効なときに G95 を再選択しても、効果はありません(ただし、F と FZ の切り替えをプログラム指令している場合を除きます)。

ノンモーダル送り速度(FB)

G95 FZ... (モーダル)が有効なときは、ノンモーダル送り速度 FB...は 1 刃当り送り速度として解釈されます。

SAVE のメカニズム

SAVE 属性を使用するサブプログラムでは、サブプログラムの開始前に FZ がセーブ値に書き込まれます(F と同様に)。

1 ブロックの複数送り速度値

「1 ブロックの複数送り速度値」機能は、1 刃当り送り速度には使用できません。

シンクロナイズドアクション

FZ は、シンクロナイズドアクションからはプログラム指令できません。

1 刃当り送り速度と軌跡送り速度タイプの読み取り

1 刃当り送り速度と軌跡送り速度タイプは、システム変数を使用して読み取ることができます。

- 次のシステム変数は、パートプログラムで先読み停止をおこないます。

3.7 送り速度制御

	\$AC_FZ	現在のメインランブロックが先読みされたときの、有効1刃 送り速度	
	\$AC_F_TYPE	現在のメインランブロックが先読みされたときの、有効軌跡 送り速度タイプ	
		値:	意味
		0	mm/min.
		1	mm/rev
		2	inch/min
		3	inch/rev
		11	mm/刃
		33	inch/刃

- 次のシステム変数は、パートプログラムでは先読み停止をおこないません。

	\$P_FZ	プログラム指令した1刃送り速度	
	\$P_F_TYPE	プログラム指令した軌跡送り速度タイプ	
		値:	意味
		0	mm/min
		1	mm/rev
		2	inch/min
		3	inch/rev
		11	mm/tooth
		33	inch/tooth

注記

G95 が有効でない場合は、\$P_FZ 変数と\$AC_FZ 変数は常にゼロの値を返します。

3.8 ジオメトリ設定

3.8.1 設定可能ゼロオフセット(G54～G57、G505～G599、G53、G500、SUPA、G153)

G54～G57 および G505～G599 命令は、操作画面から設定された基本座標系と比較して、ワーク座標系のオフセットのための設定可能ゼロオフセットを有効にします。

構文

電源投入:

G54
...
G57
G505
...
G599

電源切断またはマスク:

G500 / G53 / G153 / SUPA

意味

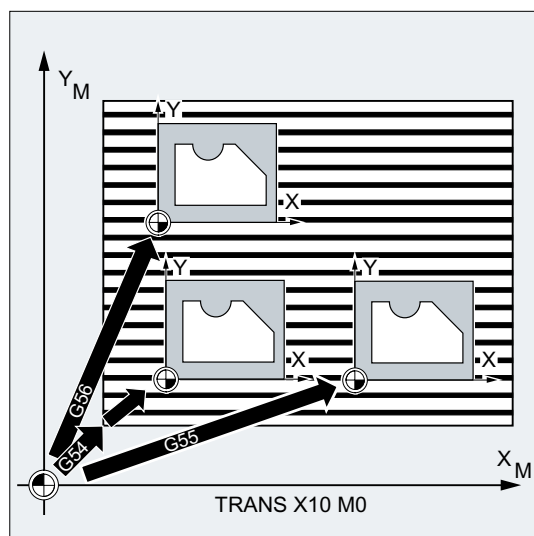
G54～G57:	1 番目から 4 番目までの設定可能ゼロオフセットを呼び出します(WO)。	
G505～G599:	5 番目から 99 番目までの設定可能ゼロオフセットを呼び出します。	
G500:	現在の設定可能ゼロオフセットを解除します。	
	G500=原点フレーム: (初期設定; オフセット、回転、ミラーリング、およびスケーリングのいずれも含みません)	設定可能ゼロオフセットを次の呼び出しまで解除して、基本フレーム全体を適用します(\$P_ACTBFRAME)。
	G500 が原点フレーム以外:	1 番目の設定可能ゼロオフセットの適用(\$P_UIFR[0])と基本フレーム全体の適用(\$P_ACTBFRAME)をおこないます。場合によっては、変更された基本フレームが有効になります。

3.8 ジオメトリ設定

G53:	G53 は、設定可能ワークオフセットとプログラマブルワークオフセットをノンモーダルにマスクします。
G153:	G153 には、G53 と同じ効果があり、基本フレーム全体もマスクします。
SUPA:	SUPA には、G153 と同じ働きがあり、同様に、次のものもマスクします。 • ハンドルオフセット(DRF) • 重畳移動 • 外部ゼロオフセット • PRESET オフセット

例

パレット上に、ゼロオフセット値 G54～G56 に従って配置された 3 つのワークが、連続して加工されます。加工処理はサブプログラム L47 にプログラム指令されています。



プログラムコード	コメント
N10 G0 G90 X10 Y10 F500 T1	;アプローチ
N20 G54 S1000 M3	; 1 番目の WO の呼び出し、主軸は右回り
N30 L47	; プログラムをサブプログラムとして実行
N40 G55 G0 Z200	; 2 番目の WO の呼び出し、障害物を避けるための z 位置
N50 L47	; プログラムをサブプログラムとして実行
N60 G56	; 3 番目の WO の呼び出し
N70 L47	; プログラムをサブプログラムとして実行
N80 G53 X200 Y300 M30	; ゼロオフセットのマスク、プログラム終了

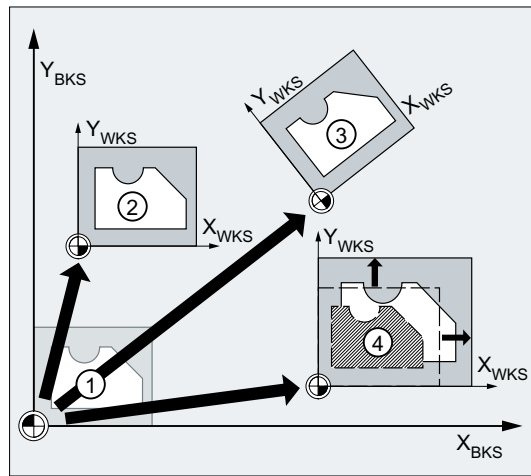
3.8.2 設定可能ワークオフセット(G54～G57、G505～G599、G53、G500、SUPA、G153):詳細情報

詳細情報

設定可能なゼロオフセット

設定可能ゼロオフセットは、基本的には設定フレーム (ページ340)です。したがって、次の成分とフレーム値は、設定可能ゼロオフセットにも使用できます。

- オフセット
- 座標回転
- スケーリング
- スケール



- ① BCS での初期位置
- ② オフセット
- ③ オフセット + 回転
- ④ オフセット + スケーリング

設定可能ゼロオフセットのフレーム値は、次の操作画面から入力します。

SINUMERIK Operate:[パラメータ|ゼロオフセット|詳細]操作エリア

パラメータ設定可能なフレーム(G505～G599)のパラメータ設定された数

特定の各チャネルに対してユーザー固有の設定可能ゼロオフセット(G505～G599)の数を設定できます

MD28080 \$MC_MM_NUM_USER_FRAMES = <番号>

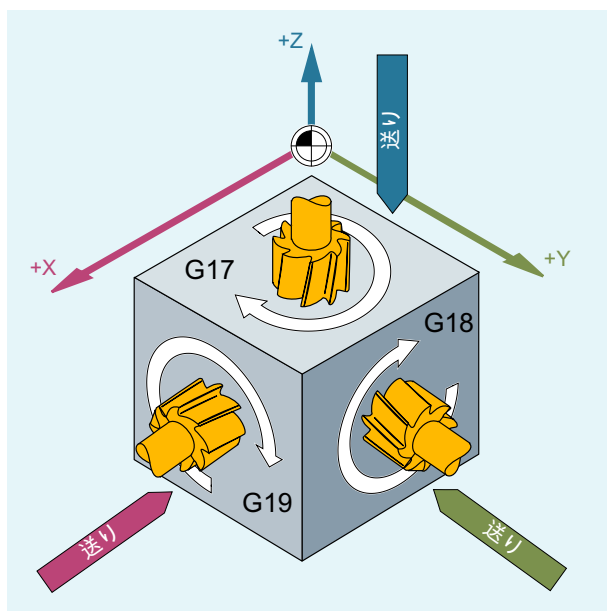
下記も参照

プログラマブルゼロオフセット(G58、G59) (ページ 349)

3.8.3 作業平面(G17/G18/G19)の選択

目的の輪郭が加工される作業平面を指定すると、次の機能も定義されます。

- 工具径補正平面
- 工具タイプに応じた工具長補正の切り込み方向
- 円弧補間平面



構文

G17/G18/G19、など

意味

G17:	作業平面 X/Y 切り込み方向 Z、平面選択、1 番目、2 番目のジオメトリ軸
G18:	作業平面 Z/X 切り込み方向 Y、平面選択、3 番目、1 番目のジオメトリ軸
G19:	作業平面 Y/Z 切り込み方向 X、平面選択、2 番目、3 番目のジオメトリ軸

注記

初期設定では、G17 (X/Y 平面)がフライス加工用に、G18 (Z/X 平面)が旋削用に定義されています。

工具軌跡補正 G41/G42 (「工具径補正 (ページ 280)」の章を参照してください)を呼び出すときは、制御装置が工具の長さと半径を補正できるように、作業平面を定義してください。

例

フライス加工の「通常の」アプローチは次のとおりです。

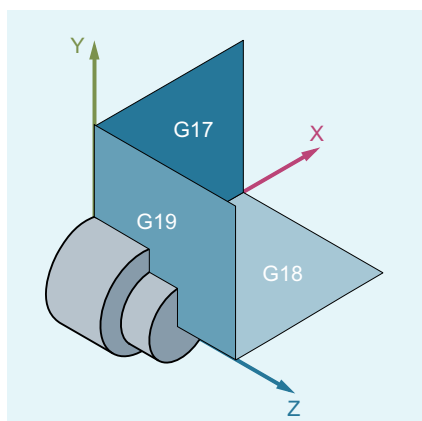
1. 作業平面(G17 フライス加工用の初期設定)を定義します。
2. 工具タイプ(T)と工具オフセット値(D)を選択します。
3. 軌跡補正(G41)を起動します。
4. 移動動作をプログラム指令します。

プログラムコード	コメント
N10 G17 T5 D8	; 作業平面 X/Y の呼び出し、工具呼び出し。工具長補正が Z 方向におこなわれます。
N20 G1 G41 X10 Y30 Z-5 F500	; X/Y 平面で工具径補正をおこないます。
N30 G2 X22.5 Y40 I50 J40	; X/Y 平面の円弧補間/工具径補正です。

詳細情報

一般事項

プログラムの先頭で作業平面 G17～G19 を選択することを推奨します。初期設定では、旋削 G18 用に Z/X 平面が設定されています。

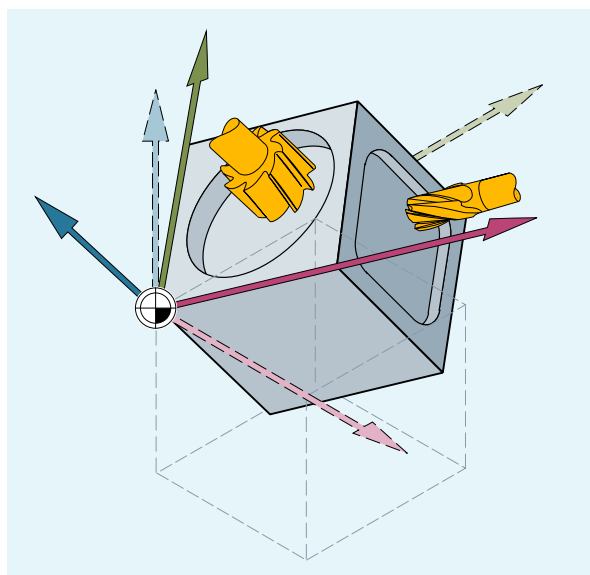


コントローラが回転方向を計算するためには、作業平面を指定する必要があります。関連情報は、「円弧補間 (ページ 212)」章を参照してください。

傾斜面の加工

座標系を ROT (ページ 352) で回転し、傾斜面上に座標軸を位置決めします。作業平面は、それに従って回転します。

傾斜面上の工具長補正



一般的な規則として、工具長補正は常に、固定された回転なしの作業平面を基準にします。

注記

工具長成分は、回転した作業平面に従って、「旋回工具の工具長補正」機能を使用して計算できます。

補正平面は CUT2D、CUT2DF で選択します。存在する演算方法の追加情報と説明は、「工具径補正 (ページ 280)」の章に記載されています。

コントローラには、作業平面の空間定義に便利な座標変換機能があります。関連情報は、「座標変換(フレーム) (ページ 340)」章を参照してください。

3.8.4 寸法

ほとんどの NC プログラムの基本は、具体的な寸法が記載されたワーク図面です。

これらの寸法では、次のものが使用されます。

- アブソリュート指令またはインクレメンタル指令
- ミリメートルまたはインチ
- 半径または直径(旋削の場合)

外形寸法図からのデータが (変換なしで) 直接、NC プログラムに移せるように、さまざまな寸法のオプションに対して特定のプログラミング命令が使用できます。

3.8.4.1 アブソリュート指令(G90、AC)

アブソリュート指令では、位置指定は、常に現在有効な座標系の原点を基準にします。つまり、工具が移動するアブソリュート位置がプログラム指令されます。

モーダルアブソリュート指令

モーダルアブソリュート指令は、G90 命令で有効にします。この指令は一般的には、指令以降の NC ブロックの、すべてのプログラム指令軸に適用されます。

ノンモーダルアブソリュート指令

既にインクレメンタル指令(G91)が設定されている場合、AC 命令を使用して、個々の軸にノンモーダルアブソリュート指令を設定できます。

注記

ノンモーダルアブソリュート指令(AC)は、主軸の位置決め(SPOS、SPOSA)、および補間パラメータ(I、J、K)にも使用できます。

構文

G90
<軸>=AC (<値>)

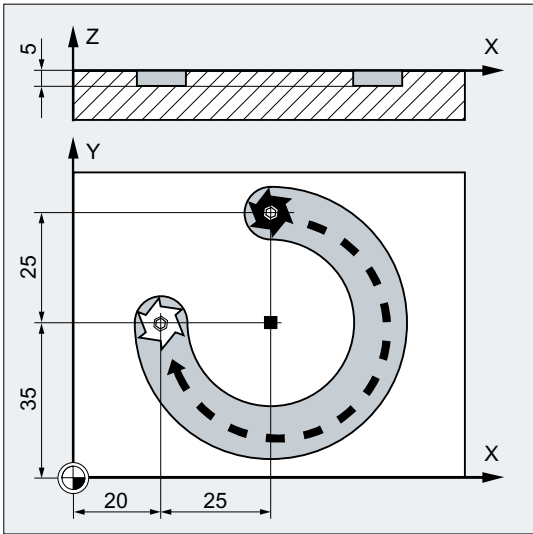
3.8 ジオメトリ設定

意味

G90:	モーダルアブソリュート指令を適用する命令
AC:	ノンモーダルアブソリュート指令を適用する命令
<軸>:	移動軸の軸識別子
<値>:	アブソリュート指令の移動軸の位置指令値

例

例 1: フライス加工

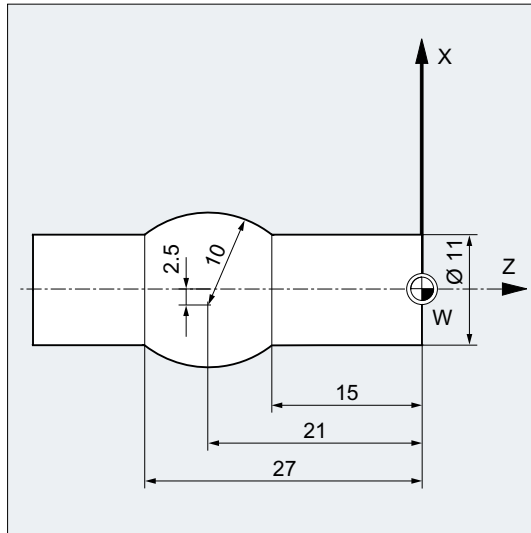


プログラムコード	コメント
N10 G90 G0 X45 Y60 Z2 T1 S2000 M3	; アブソリュート指令入力、XYZ の位置へ早送り、工具選択、主軸が右回転方向でオンします。
N20 G1 Z-5 F500	; 直線補間、工具の送り速度です。
N30 G2 X20 Y35 I=AC(45) J=AC(35)	; 右回りの円弧補間、アブソリュート指令の円弧終点、および円弧中心点です。
N40 G0 Z2	; 移動
N50 M30	; ブロック終了。

注記

円弧中心点座標 I と J の入力情報については、「円弧補間」の章を参照してください。

例 2: 旋削



プログラムコード	コメント
N5 T1 D1 S2000 M3	; 工具 T1 を装着して、主軸が右回転方向でオンします。
N10 G0 G90 X11 Z1	; アブソリュート指令入力、XZ の位置へ早送りです。
N20 G1 Z-15 F0.2	; 直線補間、工具の送り速度です。
N30 G3 X11 Z-27 I=AC(-5) K=AC(-21)	; 左回りの円弧補間、アブソリュート指令の円弧終点、および円弧中心点です。
N40 G1 Z-40	; 移動
N50 M30	; ブロック終了。

注記

円弧中心点座標 I と J の入力情報については、「円弧補間」の章を参照してください。

下記も参照

旋削とフライス加工のアブソリュート指令、およびインクレメンタル指令(G90/G91) (ページ 175)

3.8.4.2 インクレメンタル指令(G91、IC)

インクレメンタル指令では、最後のアプローチ点を基準点にして位置が指定されます。つまり、インクレメンタル指令のプログラミングでは、工具の移動距離を記述します。

モーダルインクレメンタル指令

モーダルインクレメンタル指令は、G91 命令で有効になります。この指令は一般的には、指令以降の NC ブロックの、すべてのプログラム指令軸に適用されます。

3.8 ジオメトリ設定

ノンモーダルインクレメンタル指令

既にアブソリュート指令(G90)が設定されている場合、IC 命令を使用して、個々の軸にノンモーダルインクレメンタル指令を指令できます。

注記

ノンモーダルインクレメンタル指令(IC)は、主軸の位置決め(SPOS、SPOSA)、および補間パラメータ(I、J、K)にも使用できます。

構文

G91
<軸>=IC (<値>)

意味

G91:	モーダルインクレメンタル指令を適用する命令
IC:	ノンモーダルインクレメンタル指令を適用する命令
<軸>:	移動軸の軸識別子
<値>:	インクレメンタル指令の移動軸の位置指令値

G91 の拡張

接触計測などの特定の用途では、インクレメンタル指令で、プログラム指令距離のみを移動することが必要な場合があります。動作中のゼロオフセットと工具長補正は移動しません。

この動作は、次のセッティングデータを使用して、動作中のゼロオフセットと工具長補正に対して個別に設定できます。

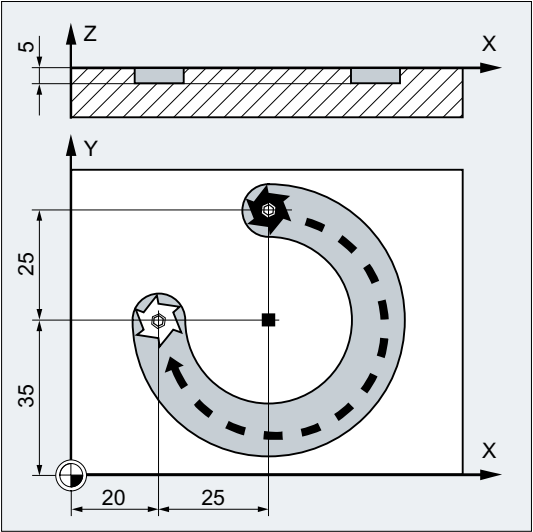
SD42440 \$SC_FRAME_OFFSET_INCR_PROG (フレームのゼロオフセット)

SD42442 \$SC_TOOL_OFFSET_INCR_PROG (工具長補正)

値	意味
0	軸のインクレメンタルプログラミング(インクレメンタル指令)で、ゼロオフセット、または工具長補正は 移動しません 。
1	軸のインクレメンタルプログラミング(インクレメンタル指令)で、ゼロオフセット、または工具長補正が移動します。

例

例 1:フライス加工

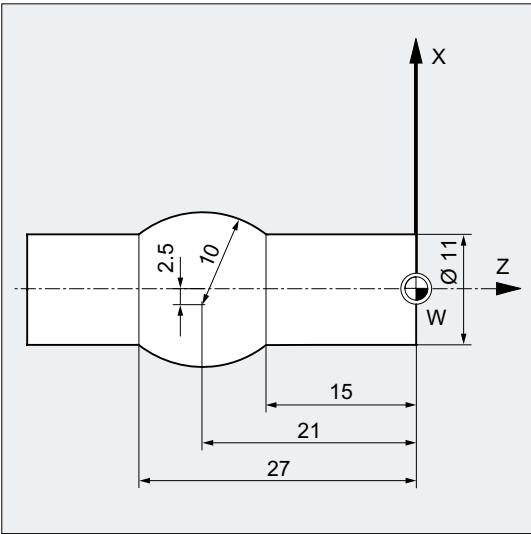


プログラムコード	コメント
N10 G90 G0 X45 Y60 Z2 T1 S2000 M3	; アブソリュート指令入力、XYZ の位置へ早送り、工具選択、主軸が右回転方向でオンします。
N20 G1 Z-5 F500	; 直線補間、工具の送り速度です。
N30 G2 X20 Y35 I0 J-25	; 右回りの円弧補間、アブソリュート指令の円弧終点、インクレメンタル指令の円弧中心点です。
N40 G0 Z2	; 移動
N50 M30	; ブロック終了。

注記

円弧中心点座標 I と J の入力情報については、「円弧補間」の章を参照してください。

例 2 :旋削



プログラムコード	コメント
N5 T1 D1 S2000 M3	; 工具 T1 を装着して、主軸が右回転方向でオンします。
N10 G0 G90 X11 Z1	; アブソリュート指令入力、 XZ の位置へ早送りです。
N20 G1 Z-15 F0.2	; 直線補間、工具の送り速度です。
N30 G3 X11 Z-27 I-8 K-6	; 左回りの円弧補間、アブソリュート指令の円弧終点、インクレメンタル 指令の円弧中心点です。
N40 G1 Z-40	; 移動
N50 M30	; ブロック終了。

注記

円弧中心点座標 I と J の入力情報については、「円弧補間」の章を参照してください。

例 3 :動作中のゼロオフセットの移動なしのインクレメンタル指令

設定内容:

- G54 には、25 の X オフセットが含まれます。
- SD42440 \$SC_FRAME_OFFSET_INCR_PROG = 0

プログラムコード	コメント
N10 G90 G0 G54 X100	
N20 G1 G91 X10	; インクレメンタル指令が有効、10 mm の X の移動 (ゼロオフセットは移動しません)。
N30 G90 X50	; アブソリュート指令が有効、X75 の位置へ移動 (ゼロオフセットが移動します)。

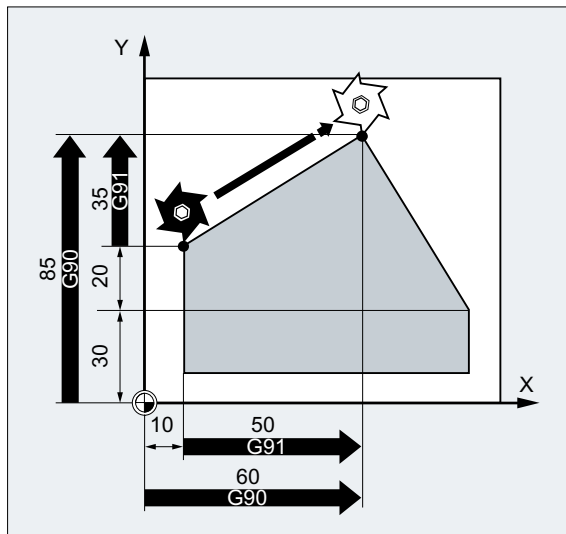
下記も参照

旋削とフライス加工の絶対指令、およびインクリメンタル指令(G90/G91) (ページ 175)

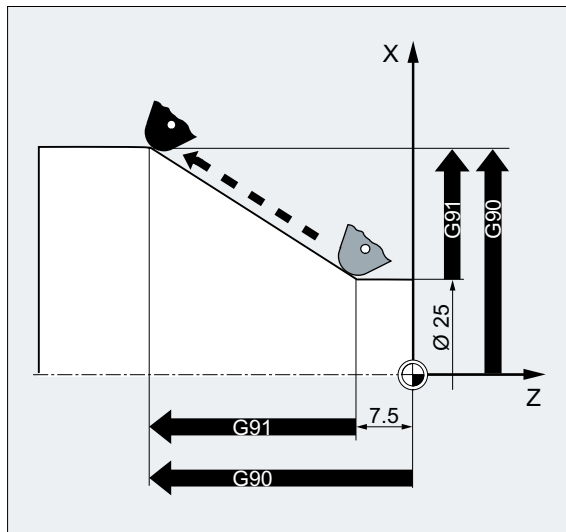
3.8.4.3 旋削とフライス加工の絶対指令、およびインクリメンタル指令(G90/G91)

次の2つの図は、旋削とフライス加工の加工例を使用して、絶対指令(G90)、またはインクリメンタル指令(G91)のプログラミングを示します。

フライス削り:



旋削:



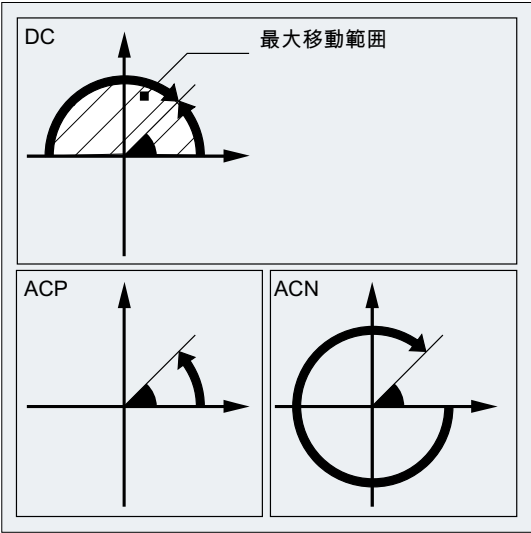
注記

汎用旋盤では、直径指定が基準寸法に適用された場合でも、径方向軸のインクレメンタル移動ブロックは半径値と見なされます。G90 のためのこの変換は、命令 DIAMON、DIAMOF、または DIAM90 を使用しておこなわれます。

3.8.4.4 回転軸のアブソリュート指令(DC、ACP、ACN)

アブソリュート指令で回転軸の位置決めをおこなうためには、ノンモーダルで G90/G91 に依存しない DC、ACP、および ACN の命令を使用できます。

DC、ACP、および ACN は、基本的なアプローチ方法が異なります。



構文

- <回転軸>=DC (<値>)
- <回転軸>=ACP (<値>)
- <回転軸>=ACN (<値>)

意味

<回転軸>:	移動する回転軸の識別子(A、B、またはC など)
DC:	位置に 直接 アプローチする命令 回転軸は、プログラム指令位置に直接、最短軌跡でアプローチします。回転軸は、最大 180°の範囲を移動します。

ACP:	正方向で指定位置へアプローチする命令 回転軸は、正の軸回転方向(左回り)で、プログラム指令位置へアプローチします。	
ACN:	負方向で指定位置へアプローチする命令 回転軸は、負の軸回転方向(右回り)で、プログラム指令位置へアプローチします。	
<値>:	アブソリュート指令でアプローチする回転軸の位置	
	値の範囲:	0～360°

注記

正の回転方向(右回りまたは左回り)はマシンデータで設定されます。

注記

方向を指定して位置決めするためには、マシンデータに 0°～360°の移動範囲を設定してください(モジュロ動作)(ACP、ACN)。モジュロ回転軸を、360°を超えて移動するには、1つのブロックに G91 または IC をプログラム指令してください。

注記

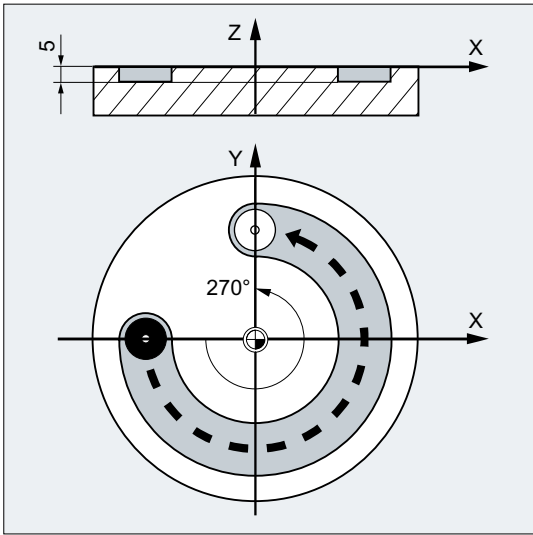
停止状態からの主軸の位置決め(SPOS、SPOSA)にも、DC、ACP、および ACN 命令が使用できます。

例:SPOS=DC(45)

例**回転テーブルのフライス加工**

工具は停止して、テーブルは**右回り方向に 270°**回転します。円弧の溝/スロットが加工されます。

3.8 ジオメトリ設定



プログラムコード	コメント
N10 SPOS=0	; 位置制御の主軸
N20 G90 G0 X-20 Y0 Z2 T1	; アブソリュート指令、工具 T1 を早送りで移動します。
N30 G1 Z-5 F500	; 工具を送り速度で下降します。
N40 C=ACP(270)	; テーブルが右回りに 270° (正) 回転して、工具が円弧の溝を切削します。
N50 G0 Z2 M30	; 後退、プログラム終了

3.8.4.5 メトリック/インチ寸法システム(G70/G71、G700/G710)

パートプログラムで G グループ 13(インチ/メトリック単位系)の命令を使用することで、メトリック単位系とインチ単位系を切り替えることができます。

起動

命令 G700 および G710 が使用可能であるために、拡張された単位系の機能をオンにする必要があります(MD10260 \$MN_CONVERT_SCALING_SYSTEM = 1)。

構文

G70
G71
G700
G710

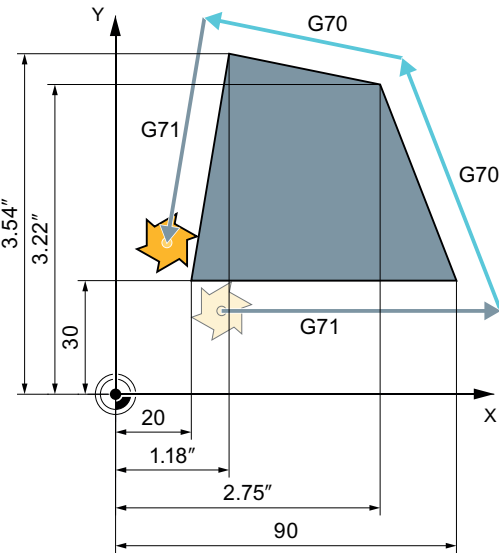
意味

G70:	インチ単位系の機動 インチ単位系を使用すると、インチ系の長さの単位で、ジオメトリデータの読み取りと書き込みができます。 長さの単位の加工条件(たとえば、送り速度、工具オフセット、設定可能ゼロオフセット、マシンデータ、システム変数など)は、パラメータ設定された基本単位系を使用して読み取ることができます。	
	G グループ:	13
	初期設定:	MD20150 \$MC_GCODE_RESET_VALUES を使用して設定可能
	有効性:	モーダル
G71:	メトリック単位系の機動 メトリック単位系を使用すると、メトリック系の長さの単位で、ジオメトリデータの読み取りと書き込みができます。 長さの単位の加工条件(たとえば、送り速度、工具オフセット、設定可能ゼロオフセット、マシンデータ、システム変数など)は、パラメータ設定された基本単位系を使用して読み取ることができます。	
	G グループ:	13
	初期設定:	MD20150 \$MC_GCODE_RESET_VALUES を使用して設定可能
	有効性:	モーダル
G700:	インチ単位系の機動 すべてのジオメトリデータと加工データの長さの単位は、インチ単位系を使用して読み取りと書き込みが行われます。	
	G グループ:	13
	初期設定:	MD20150 \$MC_GCODE_RESET_VALUES を使用して設定可能
	有効性:	モーダル
G710:	メトリック単位系の機動 すべてのジオメトリデータと加工データの長さの単位は、メトリック単位系を使用して読み取りと書き込みが行われます。	
	G グループ:	13
	初期設定:	MD20150 \$MC_GCODE_RESET_VALUES を使用して設定可能
	有効性:	モーダル

通知
回転軸の軸別のデータ 回転軸の軸別のデータは、パラメータ設定された基本単位系を使用して読み取りと書き込みが行われます。

例

基本単位系はメトリックです(MD10240 \$MN_SCALING_SYSTEM_IS_METRIC = 1)。ただし、ワーク図の寸法はインチで表されます。このことが、パートプログラムでインチ単位系を選択した理由です。インチ寸法の処理後、メトリック単位系が再度選択されます。



プログラムコード	コメント
N10 G0 G90 X20 Y30 Z2 S2000 M3 T1	; X=20 mm、Y=30 mm、Z=2 mm、F=早送り mm/min
N20 G1 Z-5 F500	; Z=-5 mm、F=500 mm/min
N30 X90	; X=90 mm
N40 G70 X2.75 Y3.22	; プログラム指令単位系:inch ; X=2.75 インチ、Y=3.22 インチ、F=500 mm/min
N50 X1.18 Y3.54	; X=1.18 inch、Y=3.54 inch、F=500 mm/min
N60 G71 X20 Y30	; プログラム指令単位系:メトリック ; X=20 mm、Y=30 mm、F=500 mm/min
N70 G0 Z2	; Z=2 mm、F=早送り mm/min
N80 M30	; プログラム終了

詳細情報

G70/G71 と G700/G710 の場合のデータの読み取りと書き込み

データ領域	G70/G71		G700/G710	
	読み取り	書き込み	読み取り	書き込み
表示、小数位(WCS)	P	P	P	P
表示、小数位(MCS)	G	G	G	G
送り速度	G	G	P	P
位置データ(X、Y、Z、...)	P	P	P	P
補間パラメータ I、J、K	P	P	P	P
円弧半径(CR)	P	P	P	P
極半径(RP)	P	P	P	P
ねじピッチ	P	P	P	P
プログラマブルフレーム	P	P	P	P
設定可能フレーム	G	G	P	P
基本フレーム	G	G	P	P
外部ゼロオフセット	G	G	P	P
軸プリセットオフセット	G	G	P	P
作業領域リミット(G25/G26)	G	G	P	P
プロテクションゾーン	P	P	P	P
工具オフセット	G	G	P	P
長さ関連のマシンデータ	G	G	P	P
長さ関連のセッティングデータ	G	G	P	P
長さ関連のシステム変数	G	G	P	P
GUDs	G	G	G	G
LUD	G	G	G	G
PUD	G	G	G	G
R 変数	G	G	G	G
Siemens サイクル	P	P	P	P
ジョグ/ハンドル分割係数	G	G	G	G
P:読み取り/書き込みがプログラム指令単位系で実行されます。 G:読み取り/書き込みが、設定した基本単位系で実行されます。				

シンクロナイズドアクション

注記

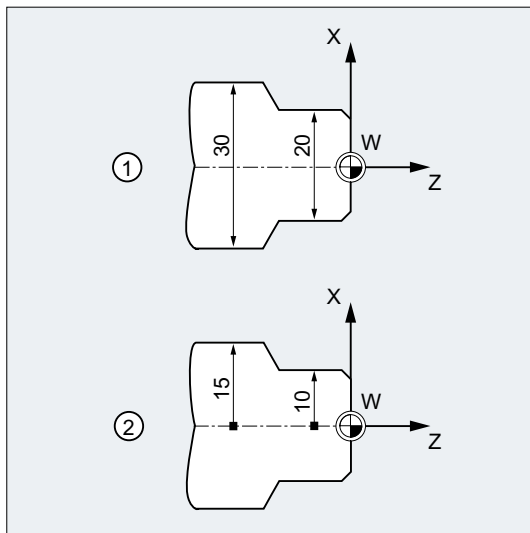
シンクロナイズドアクションでの位置データの読み取り

単位系をシンクロナイズドアクションで明示的にプログラム指令していない(条件要素と動作要素のいずれかまたは両方)場合は、シンクロナイズドアクション内の長さに関連する位置データを常に、パラメータで設定した基本単位系で読み取ります。

詳細情報:『機能マニュアル、シンクロナイズドアクション』

3.8.4.6 チャネル別の直径/半径指定(DIAMON、DIAM90、DIAMOF、DIAMCYCOF)

旋削中に、次のように径方向軸の寸法を、直径(①)または半径(②)で指定できます。



加工図面の寸法を NC プログラムに直接(変換なしで)移せるように、チャネル別の直径指定または半径指定が、モーダル命令 DIAMON、DIAM90、DIAMOF、および DIAMCYCOF で有効になります。

注記

チャネル別の直径/半径指定は、MD20100 \$MC_DIAMETER_AX_DEF で径方向軸として定義したジオメトリ軸を基準とします(→工作機械メーカーの仕様書を参照してください)。

チャネル毎に 1 つの径方向軸のみ MD20100 で定義できます。

構文

DIAMON
DIAM90

DIAMOF

意味

DIAMON:	指令モードに依存しないチャネル別直径指定を適用する命令です。 DIAMON の動作は、プログラム指令寸法モード(アブソリュート指令 G90 またはインクレメンタル指令 G91)には影響されません。	
	• G90 の場合:	直径寸法
	• G91 の場合:	直径寸法
DIAM90:	指令モードに依存するチャネル別直径指定を適用する命令です。 DIAM90 の動作は、プログラム指令寸法モードにより異なります。	
	• G90 の場合:	直径寸法
	• G91 の場合:	半径寸法
DIAMOF:	チャネル別直径指定を解除する命令です。 直径指定解除されると、チャネル別半径指定が有効になります。DIAMOF の動作は、プログラム指令寸法モードには依存しません。	
	• G90 の場合:	半径寸法
	• G91 の場合:	半径寸法
DIAMCYCOF:	サイクル処理中に、チャネル別直径指定を解除する命令。 この指令では、サイクルの計算を常に半径指定で実行できます。このグループで動作中の最後の G 命令は、位置表示と基本ブロック表示で有効になります。	

注記

DIAMON または DIAM90 を使用すると、径方向軸の現在位置が常に直径として表示されます。これは、MEAS、MEAW、\$P_EP[x]、および\$AA_IW[x]によるワーク座標系の現在位置の読み取りにも適用されます。

3.8 ジオメトリ設定

例

プログラムコード	コメント
N10 G0 X0 Z0	; 起点にアプローチします。
N20 DIAMOF	; 直径指定がオフです。
N30 G1 X30 S2000 M03 F0.7	; X 軸=径方向軸、半径指定が有効; X30 の半径位置へ移動します。
N40 DIAMON	; 直径指定が径方向軸で有効です。
N50 G1 X70 Z-20	; X70 の直径位置と Z-20 への移動。
N60 Z-30	
N70 DIAM90	; アブソリュート指令の直径指定とインクレメンタル指令の半径指定。
N80 G91 X10 Z-20	; インクレメンタル指令が有効。
N90 G90 X10	; アブソリュート指令が有効。
N100 M30	; プログラム終了

他の情報

直径値(DIAMON/DIAM90)

直径値は、次のデータに適用されます。

- ワーク座標系の径方向軸の現在位置表示
- JOG モード:インクレメンタル指令と手動ハンドル移動量
- 終了位置のプログラム:
G2/G3 の補間パラメータ I、J、K (AC のアブソリュート指令でプログラム指令している場合)
I、J、K をインクリメンタル指令(IC)でプログラム指令した場合は、常に半径値で計算します。
- ワーク座標系の以下の現在位置の読み取り:
MEAS、MEAW、\$P_EP[X]、\$AA_IW[X]

3.8.4.7 軸別の直径/半径指定(DIAMONA、DIAM90A、DIAMOF A、DIACYCOFA、DIAMCHANA、DIAMCHAN、DAC、DIC、RAC、RIC)

チャネル別の直径指定に加えて、軸別の直径指定機能を使用すると、モーダルまたはノンモーダル指令、および複数の軸の直径表示が有効になります。

注記

軸別の直径指定は、MD30460 \$MA_BASE_FUNCTION_MASK による軸別の直径指定のために、追加の径方向軸として使用される軸に対してのみ可能です(→ 工作機械メーカーの仕様書を参照してください)。

構文

チャネルの複数の径方向軸用のモーダルの軸別直径指定:

DIAMONA [<軸>]
 DIAM90A [<軸>]
 DIAMOFA [<軸>]
 DIACYCOFA [<軸>]

チャネル別の直径/半径指定の反映:

DIAMCHANA [<軸>]
 DIAMCHAN

ノンモーダルの軸別の直径/半径指定:

<軸>=DAC (<値>)
 <軸>=DIC (<値>)
 <軸>=RAC (<値>)
 <軸>=RIC (<値>)

意味

モーダルの軸別直径指定		
DIAMONA:	指令モードに依存しない軸別直径指定を適用する命令 DIAMONA の動作は、プログラム指令寸法モード(G90/G91 または AC/IC)には依存しません。	
	• G90、AC の場合:	直径寸法
	• G91、IC の場合:	直径寸法
DIAM90A:	指令モードに依存する軸別直径指定を適用する命令 DIAM90A の動作は、プログラム指令寸法モードにより異なります。	
	• G90、AC の場合:	直径寸法
	• G91、IC の場合:	半径寸法
DIAMOFA:	軸別直径指定を解除する命令 直径指定が解除されると、軸別半径指定が有効になります。DIAMOFA の動作は、プログラム指令寸法モードには依存しません。	
	• G90、AC の場合:	半径寸法
	• G91、IC の場合:	半径寸法

3.8 ジオメトリ設定

DIACYCOFA:	サイクル処理中の軸別直径指定を解除する命令。 この方法では、サイクルの計算を常に半径指定で実行できます。このグループで動作中の最後の G 命令は、位置表示と基本ブロック表示で有効になります。	
<軸>:	軸別の直径指定が有効になる軸の軸識別子。 使用できる軸識別子は以下のとおりです。 - ジオメトリ/チャンネル軸名称 または - 機械軸名称	
	値の範囲:	指定軸は、チャンネルの有効軸にしてください。 その他の条件: - 軸は、MD30460 \$MA_BASE_FUNCTION_MASK により、軸別直径指定が可能な軸を使用してください。 - 回転軸は、径方向軸としては使用できません。
チャンネル別の直径/半径指定の反映		
DIAMCHANA:	DIAMCHANA [<軸>] 命令を使用すると、 指定軸 は直径/半径指定のチャンネル状態を受け取って、チャンネル別の直径/半径指定に割り当てられます。	
DIAMCHAN:	DIAMCHAN 命令を使用すると、軸別の直径指定が使用された すべての軸 は、直径/半径指定のチャンネル状態を受け取って、チャンネル別の直径/半径指定に割り当てられます。	
ノンモーダルの軸別の直径/半径指定		
ノンモーダルの軸別の直径/半径指定により、パートプログラムとシンクロナイズドアクションで、寸法タイプを直径値または半径値として指定します。直径/半径指定のモーダル状態は変更されません。		
DAC:	DAC 命令により、指定軸に対して、以下の指令がノンモーダルに設定されます。 アブソリュート指令の直径	
DIC:	DIC 命令により、指定軸に対して、以下の指令がノンモーダルに設定されます。 インクレメンタル指令の直径	

RAC:	RAC 命令により、指定軸に対して、以下の指令がノンモーダルに設定されます。 アブソリュート指令の半径
RIC:	RIC 命令により、指定軸に対して、以下の指令がノンモーダルに設定されます。 インクレメンタル指令の半径

注記

DIAMONA [<軸>] または DIAM90A [<軸>] を使用すると、径方向軸の現在位置が常に直径として表示されます。これは、MEAS、MEAW、\$P_EP[x]、および\$AA_IW[x] によるワーク座標系の現在位置の読み取りにも適用されます。

注記

GET 要求による追加の径方向軸の入れ替えのときに、別のチャネルの直径/半径指定の状態を、RELEASE [<軸>] により受け取ります。

例**例 1: モーダルの軸別の直径/半径指定**

X はチャネルの径方向軸で、軸別の直径指定が Y に対して使用されています。

プログラムコード	コメント
N10 G0 X0 Z0 DIAMON	; チャネル別の直径指定が X に対して有効です。
N15 DIAMOF	; チャネル別の直径指定がオフです。
N20 DIAMONA[Y]	; モーダルの軸別直径指定が Y に対して有効です。
N25 X200 Y100	; 半径指定が X に対して有効です。
N30 DIAMCHANA[Y]	; Y は、チャネル別の直径/半径指定の状態を受け取り、これに割り当てられます。
N35 X50 Y100	; 半径指定が X と Y に対して有効です。
N40 DIAMON	; チャネル別の直径指定がオンです。
N45 X50 Y100	; 直径指定が X と Y に対して有効です。

例 2: ノンモーダルの軸別の直径/半径指定

X はチャネルの径方向軸で、軸別の直径指定が Y に対して使用されています。

プログラムコード	コメント
N10 DIAMON	; チャネル別の直径指定がオンです。
N15 G0 G90 X20 Y40 DIAMONA[Y]	; モーダルの軸別直径指定が Y に対して有効です。

3.8 ジオメトリ設定

プログラムコード	コメント
N20 G01 X=RIC(5)	; このブロックでXに対して有効な寸法:インクレメンタル指令の半径。
N25 X=RAC(80)	; このブロックでXに対して有効な寸法:アブソリュート指令の半径。
N30 WHEN \$SAA_IM[Y]> 50 DO POS[X]=RIC(1)	; Xはコマンド軸です。 このブロックでXに対して有効な寸法:インクレメンタル指令の半径。
N40 WHEN \$SAA_IM[Y]> 60 DO POS[X]=DAC(10)	; Xはコマンド軸です。 このブロックでXに対して有効な寸法:アブソリュート指令の半径。
N50 G4 F3	

詳細情報

直径値(DIAMONA/DIAM90A)

直径値は、次のデータに適用されます。

- ワーク座標系の径方向軸の現在位置表示
- JOG モード:インクレメンタル指令と手動ハンドル移動量
- 終了位置のプログラム:
G2/G3 の補間パラメータ I、J、K (AC のアブソリュート指令でプログラム指令している場合)
I、J、K をインクリメンタル指令(IC)でプログラム指令した場合は、常に半径値で計算します。
- ワーク座標系の以下の現在位置の読み取り:
MEAS、MEAW、\$P_EP[X]、\$AA_IW[X]

ノンモーダルの軸別直径指定(DAC、DIC、RAC、RIC)

命令 DAC、DIC、RAC、RIC は、チャンネル別の直径指定に関連する下記のすべての命令で使用できます。

- 軸の位置:X...、POS、POSA
- オシレーション:OSP1、OSP2、OSS、OSE、POSP
- 補間パラメータ:I、J、K
- 輪郭定義:角度を指定した直線
- 高速リトラクト:POLF[AX]

- 工具方向の移動:MOV T
- 滑らかなアプローチと後退:
G140～G143、G147、**G148**、G247、G248、G347、G348、G340、G341

3.8.5 旋削のためのワーク位置

軸識別子

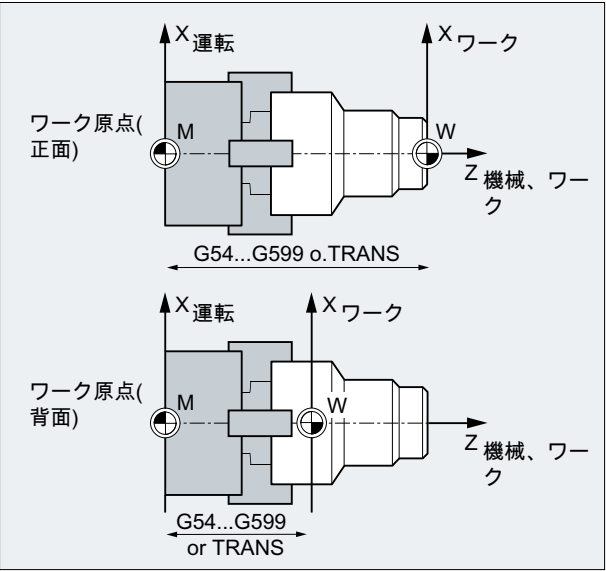
互いに直交する 2 つのジオメトリ軸は通常、次のように呼ばれます。

長手軸	= Z 軸(横軸)
径方向軸	= X 軸(縦軸)

ワーク原点

機械原点が固定して定義されているのに対し、ワーク原点は、長手軸で自由に選択できます。一般に、ワーク原点はワークの前側または後ろ側にあります。

機械原点とワーク原点はいずれも、旋削の中心にあります。したがって、X 軸の設定可能オフセットはゼロです。



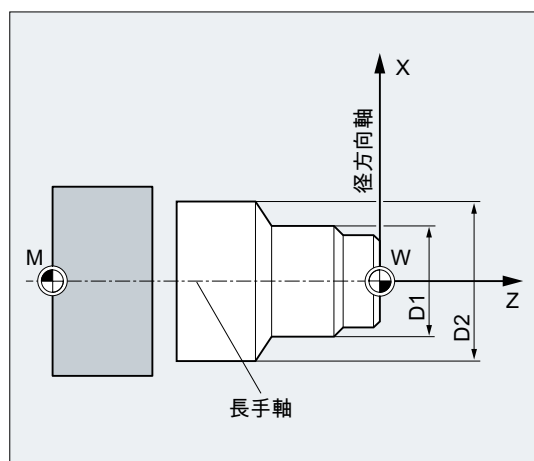
M	機械原点
W	ワーク原点

3.9 動作命令

Z	長手軸
X	径方向軸
G54~G599 または TRANS	ワーク原点の位置の呼び出し

径方向軸

一般に、径方向軸の寸法は直径指定です(他の軸に対して2倍の軌跡寸法です)。



径方向軸となるジオメトリ軸は、マシンデータで定義されます(→工作機械メーカー)。

3.9 動作命令

3.9.1 移動指令の概要

輪郭要素

プログラム指令のワーク輪郭は、以下の輪郭要素から構成されます。

- 直線
- 円弧
- ヘリカル曲線(直線と円弧の重畳をおこないます)

移動指令

これらの輪郭要素は、以下の移動指令を使用して作成できます。

- 早送り移動(G0)
- 直線補間(G1)
- 右回りの円弧補間(G2)
- 左回りの円弧補間(G3)

移動指令はモーダルです。

目標位置

移動ブロックには、移動する軸(軌跡軸、同期軸、位置決め軸)の目標位置が含まれます。

目標位置は、直交座標または極座標でプログラム指令できます。

注記

軸アドレスは、ブロック毎に 1 回だけプログラム指令できます。

起点-目標点

移動動作は常に、プログラム指令目標位置が、到達する最終位置になります。その後、この目標位置が、次の移動指令の開始位置となります。

ワーク輪郭

通知
工具動作未定義 加工前には、工具またはワークを損傷しないように、ワークを配置してください。

移動ブロックが連続して実行されると、次のようなワーク輪郭を加工します。

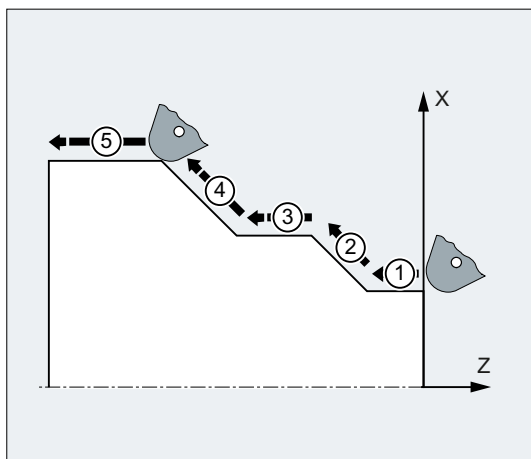


図 3-1 旋削の移動ブロック

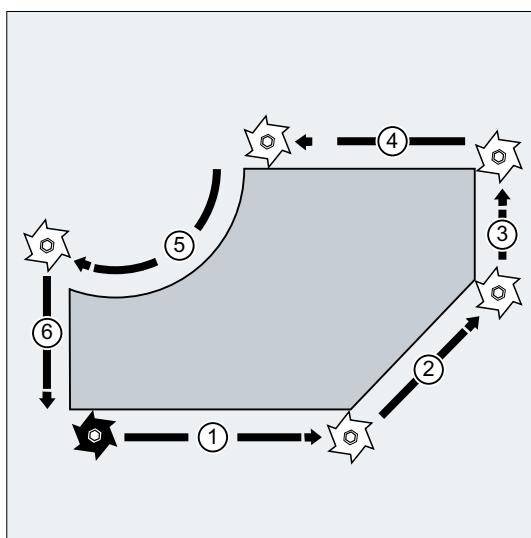


図 3-2 フライス削りの移動ブロック

3.9.2 直交座標による移動指令(G0、G1、G2、G3、X...、Y...、Z...)

NC ブロックに直交座標で指定された位置へ、早送り移動 G0、直線補間 G1、または円弧補間 G2 / G3 でアプローチできます。

構文

```
G0 X...Y...Z...
G1 X...Y...Z...
G2 X...Y...Z... ...
G3 X...Y...Z... ...
```

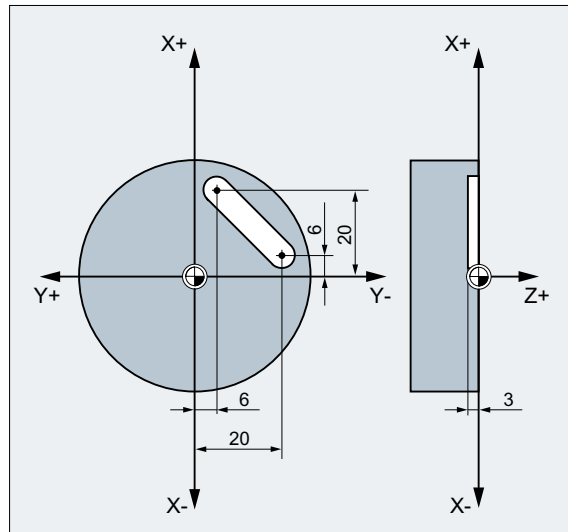
意味

G0:	早送り移動を適用する命令
G1:	直線補間を適用する命令
G2:	右回りの円弧補間を適用する命令
G3:	左回りの円弧補間を適用する命令
X...:	X 方向の目標位置の直交座標
Y...:	Y 方向の目標位置の直交座標
Z...:	Z 方向の目標位置の直交座標

注記

目標位置の座標 X...、Y...、Z...の他に、円弧補間 G2 / G3 には、さらにデータ(円弧中心点座標など。「一覧 (ページ 212)」を参照してください)が必要です。

例



プログラムコード	コメント
N10 G17 S400 M3	; 作業平面の選択、主軸は右回り
N20 G0 X40 Y-6 Z2	; 直交座標で指定した開始位置へ早送りでアプローチ
N30 G1 Z-3 F40	; 直線補間の適用、工具の送り速度
N40 X12 Y-20	; 直交座標で指定した終了位置への斜線上を移動
N50 G0 Z100 M30	; 工具交換のための早送り後退

3.9.3 極座標による移動命令

3.9.3.1 極座標の基準点(G110、G111、G112)

寸法指令を開始する点を「極」と呼びます。

極は、直交座標または極座標で指定できます。

極座標の基準点は、G110～G112 命令により明確に定義されます。したがって、アブソリュート指令もインクレメンタル指令入力は影響しません。

構文

```
G110/G111/G112 X... Y... Z...
G110/G111/G112 AP=... RP=...
```

意味

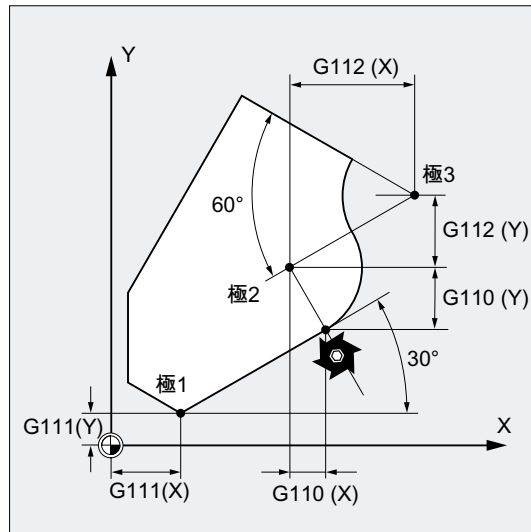
G110 ...:	G110 命令を使用すると、以降の極座標は、最後の到達位置を基準とします。		
G111 ...:	G111 命令を使用すると、以降の極座標は、現在のワーク座標系の原点を基準とします。		
G112 ...:	G112 命令を使用すると、以降の極座標は、最後の有効な極を基準とします。		
	注: G110～G112 の命令は、個別の NC ブロックでプログラム指令してください。		
X... Y... Z...:	直交座標の極の指定		
AP=... RP=...:	極座標の極の指定		
	AP=...:	極角度 極半径と、作業平面の水平軸(G17 の X 軸など)が成す角度です。回転の正方向は左回りです。	
		値の範囲:	± 0...360°
	RP=...:	極半径 正の絶対値を必ず、[mm]または[inch]単位で指定します。	

注記

NC プログラムのブロック毎に、極座標と直交座標を切り替えることができます。直交座標識別子(X...、Y...、Z...)を使用すると、直交座標系に直接戻ることができます。また、定義した極は、プログラムの終了まで保持されます。

注記

極を指定していない場合は、現在のワーク座標系の原点が適用されます。

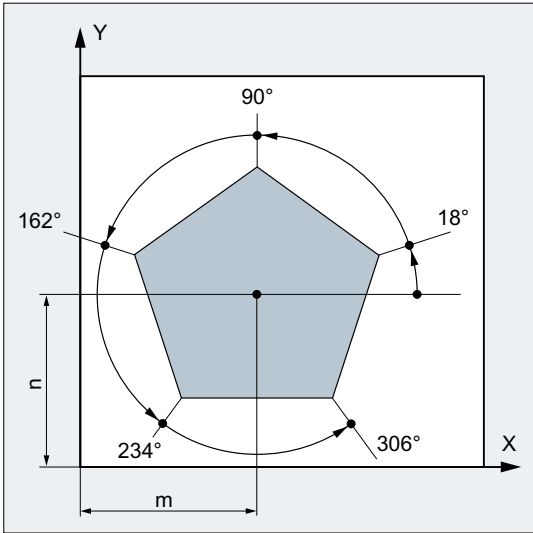
例

極 1～3 は、次のように定義されます。

- 極 1 は G111 X... Y...
- 極 2 は G110 X... Y...
- 極 3 は G112 X... Y...

3.9.3.2 極座標による移動指令(G0、G1、G2、G3、AP、RP)

極座標による移動指令は、ワークまたはワークの一部の寸法が中心点を基準に表わされて、寸法が角度と半径で指定される場合(穴あけパターンの場合など)に便利です。



構文

G0/G1/G2/G3 AP=... RP=...

意味

G0:	早送り移動を適用する命令
G1:	直線補間を適用する命令
G2:	右回りの円弧補間を適用する命令
G3:	左回りの円弧補間を適用する命令

AP:	極角度	
	極半径と、作業平面の水平軸(G17 の X 軸など)が成す角度です。回転の正方向は左回りです。	
	値の範囲:	$\pm 0 \dots 360^\circ$
	角度は、インクレメンタル指令またはアブソリュート指令で指定できます。	
	AP=AC (...):	アブソリュート指令の入力
	AP=IC (...):	インクレメンタル指令の入力 インクレメンタル指令の入力では、最後のプログラム指令角度が基準として適用されます。
極角度は、新しい極を定義するか、作業平面を変更するまで、そのまま保持されます。		
RP:	極半径	
	正の絶対値を必ず、[mm]または[inch]単位で指定します。	
	極半径は、新しい値が入力されるまで、そのまま保持されます。	

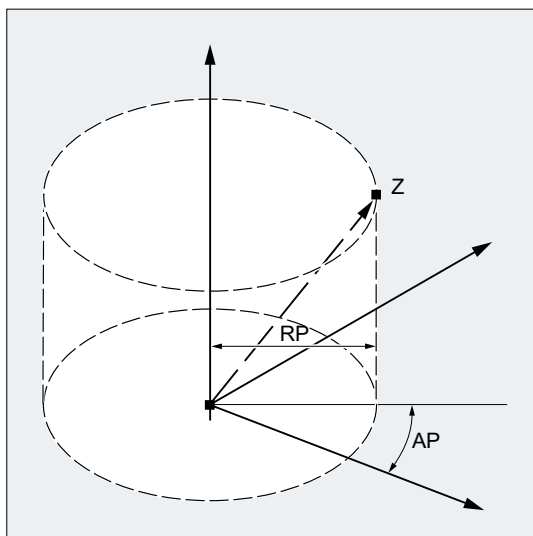
注記

極座標は、G110～G112 で指定した極を基準にして、G17～G19 で選択した作業平面で適用されます。

注記

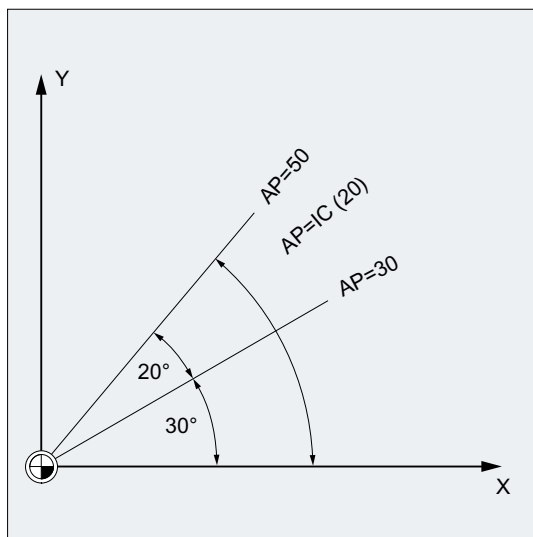
作業平面に直交する第 3 のジオメトリ軸は、直交座標でも指定できます(次の図を参照してください)。これにより、円筒座標による空間指定のプログラム指令が可能になります。

例:G17 G0 AP... RP... Z...



必要条件

- 補間パラメータ、軸アドレスなどの直交座標はいずれも、NC ブロックで選択した作業平面に対して、極の終点座標でプログラム指令することはできません。
- 極を G110～G112 で定義していない場合は、自動的に現在のワーク座標系の原点が極と見なされます。



- 極半径 $RP = 0$ のとき

極半径は、極平面の起点ベクトルと動作中の極ベクトルとの間の距離から計算されます。その後、計算された極半径はモードルとして保持されます。

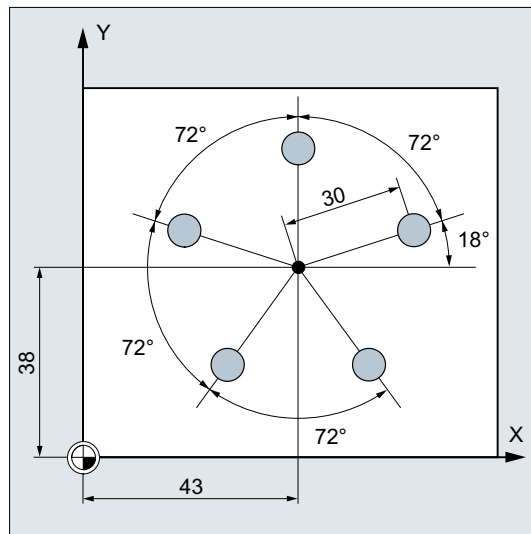
これは、選択した極定義(G110～G112)に関係なく適用されます。両方の点を同時にプログラム指令している場合は、この半径は=0 となって、アラーム 14095 が発生します。

- 極角度 AP だけのプログラム指令のとき

実行中のブロックで極半径 RP をプログラム指令せずに、極角度 AP のみをプログラム指令している場合は、現在の位置とワーク座標の極に差があると、この差が極半径として使用され、モードルとして保持されます。差=0 の場合は、極座標が再度指定され、モードルの極半径は、ゼロのままになります。

例

穴あけパターンの作成



穴の位置は極座標で指定されます。

それぞれの穴は、同じ加工順序です。

(下穴あけ、指令寸法の穴あけ、リーマ仕上げ...)で加工されます。

加工処理はサブプログラムに格納されています。

プログラムコード	コメント
N10 G17 G54	; 作業平面 X/Y、ワーク原点。
N20 G111 X43 Y38	; 極の指定。
N30 G0 RP=30 AP=18 Z5	; 起点へアプローチ、円筒座標による指定。
N40 L10	; サブプログラム呼び出し。
N50 G91 AP=72	; 早送りで次の位置へアプローチします、インクリメンタル指令の極角度です、N30 ブロックからの極半径は、そのまま保持され、指定する必要はありません。
N60 L10	; サブプログラム呼び出し。
N70 AP=IC (72)	.

3.9 動作命令

プログラムコード	コメント
N80 L10	...
N90 AP=IC(72)	
N100 L10	...
N110 AP=IC(72)	
N120 L10	...
N130 G0 X300 Y200 Z100 M30	; 工具の後退、プログラム終了。

下記も参照

一覧 (ページ 212)

3.9.4 早送り移動

3.9.4.1 早送り(G0)の有効化

早送り速度での軌跡軸の移動は、G 命令 G0 により有効になります。

構文

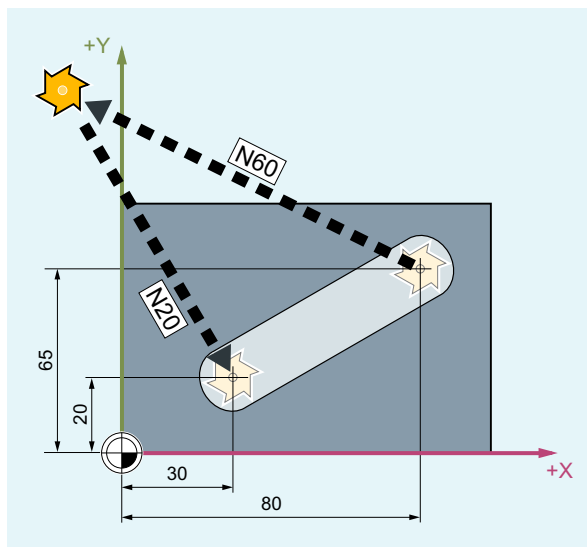
G0 X... Y... Z...
G0 RP=... AP=...

意味

G0:	早送り速度での軸の移動	
	有効:	モーダル
X...Y...Z...:	直交座標系での終点の指定	
RP=...AP=... :	極座標系での終点の指定	

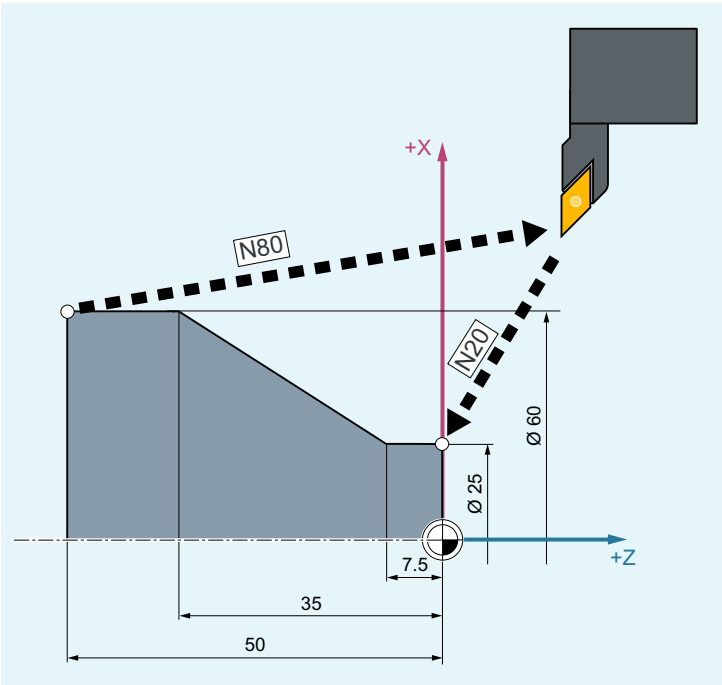
例

例 1:フライス加工



プログラムコード	コメント
N10 G90 S400 M3	; アブソリュート指令の入力、主軸は右回り
N20 G0 X30 Y20 Z2	; 開始位置へアプローチします。
N30 G1 Z-5 F1000	; 工具の切り込み。
N40 X80 Y65	; 直線上を移動
N50 G0 Z2	
N60 G0 X-20 Y100 Z100 M30	; 工具の後退、プログラム終了

例 2 :旋削



プログラムコード	コメント
N10 G90 S400 M3	; アブソリュート指令の入力、主軸は右回り
N20 G0 X25 Z5	; 開始位置へアプローチします
N30 G1 G94 Z0 F1000	; 工具の切り込み
N40 G95 Z-7.5 F0.2	
N50 X60 Z-35	; 直線上を移動
N60 Z-50	
N70 G0 X62	
N80 G0 X80 Z20 M30	; 工具の後退、プログラム終了

3.9.4.2 早送り移動の直線補間のオン/オフを切り替えます(RTLION、RTLIOF)

初期設定(MD20730 \$MC_G0_LINEAR_MODE)とは無関係に、早送り移動の補間動作は、Gグループ 55 の命令を使用してパートプログラムで設定することもできます。

構文

```
RTLIOF
...
RTLION
```

意味

RTLIOF:	直線補間をオフに切替えるための G 命令 ⇒ 早送りモード(G0)では、 非直線 補間が有効です。すべての軌跡軸は、相互に無関係に終点に到達します。	
	有効:	モーダル
RTLION:	直線補間をオンに切替えるための G 命令 ⇒ 早送りモード(G0)では、 直線 補間が有効です。すべての軌跡軸は、同時に終点に到達します。	
	有効:	モーダル

注記

RTLIOF の必要条件

RTLIOF **非直線**補間が確実に行われるよう、次の条件を満たしてください。

- 座標変換(TRAORI、TRANSMIT など)が有効でないこと。
- G60 が有効であること(ブロック終点で停止)。
- 圧縮(COMPOF)が有効でないこと。
- 工具半径補正(G40)が有効でないこと。
- 輪郭ハンドルを選択していないこと。
- ニブリングが有効でないこと。

これらの条件のいずれかを満たしていない場合、RTLION による直線補間が行われます。

例

プログラムコード	コメント
...	; 初期設定は直線補間です:
	; MD20730 \$MC_GO_LINEAR_MODE == TRUE
N30 RTLIOF	; 直線補間をオフに切替えます。
N40 G0 X0 Y10	; G0 ブロックは、非直線補間を使用して移動します。
N50 G41 X20 Y20	; TRC 有効 ⇒ G0 ブロックは、直線補間を使用して移動します。
N60 G40 X30 Y30	; TRC 無効 ⇒ G0 ブロックは、非直線補間を使用して移動します。
N70 RTLION	; 直線補間をオンに切替えます。
...	

詳細情報

現在の補間動作の読み取り

現在の補間動作は、システム変数\$AA_GOMODE で読み取ることができます。

3.9.4.3 早送り移動(STOLF、CTOLG0、OTOLG0)の許容範囲を調整

マシンデータで設定した早送り移動の許容範囲(G0 許容範囲)を、パートプログラムで一時的に調整することができます。このとき、マシンデータの設定は変更されません。チャネルまたはエンドオブプログラムをリセットすると、設定した許容範囲が再び有効になります。

必要条件

G0 許容範囲は、以下の条件に適合する場合のみ効果的です。

- 次の機能のいずれかが動作中である。
 - コンプレッサ機能 COMP...
 - スムージング機能(G642 または G645)
 - 旋回スムージング OST
 - ORISON 旋回スムージング
 - 軌跡に関連する旋回のスムージング(ORIPATH)
- パートプログラム内の複数の(≥ 2)連続する G0 ブロック
単一の G0 ブロックの場合は、G0 許容範囲が無効です。これは、G0 以外の移動から G0 移動へ(および G0 移動から G0 以外の移動)の遷移時には常に、「小さい方の許容範囲」が適用されるためです(ワーク加工許容範囲)。

構文

相対 G0 許容範囲の調整

STOLF=<値>

絶対 G0 許容範囲の調整

CTOLG0=<値>

OTOLG0=<値>

意味

STOLF:	早送り移動の一時的に有効な許容範囲係数をプログラム指令するアドレス			
	<Value >:	G0 許容範囲係数		
		タイプ:	REAL	
		値:	≥ 0:	G0 許容範囲係数には、1.0 より大きい、または小さい値を指定できます。この係数が 1.0(初期値)である場合は、非早送り移動と同じ許容範囲が早送り移動で有効になります。通常、許容範囲係数は 1.0 より大きい値に設定されます。 プログラム指令された G0 許容範囲係数は、更新された STOLF プログラミングで上書きされるか、CTOLG0/OTOLG0 プログラミングで置換されるか、チャンネルまたはエンドオブプログラムのリセットで削除されるまで、有効になります。
		< 0:	プログラムされた許容範囲係数が削除されます ⇒ マシンデータで事前に設定された許容範囲値が再度有効になります。	

CTOLG0:	早送り移動の一時的に有効な輪郭許容範囲係数をプログラム指令するアドレス			
	<値>:	輪郭許容範囲の絶対値		
		タイプ:	REAL	
		値:	≥ 0:	プログラム指令された輪郭許容範囲の絶対値は、更新された CTOLG0 プログラミングで上書きされるか、STOLF プログラミングで置換されるか、チャンネルまたはエンドオブプログラムのリセットで削除されるまで、有効になります。
		< 0:	プログラムされた許容範囲値が削除されます ⇒ マシンデータで事前に設定された許容範囲値が再度有効になります。	

3.9 動作命令

OTOLG0:	早送り移動の一時的に有効な方向許容範囲係数をプログラム指令するアドレス			
	<値>:	方向許容範囲の絶対値		
		タイプ:	REAL	
		値:	≥ 0: プログラム指令された方向許容範囲の絶対値は、更新された OTOLG0 プログラミングで上書きされるか、STOLF プログラミングで置換されるか、チャネルまたはエンドオブプログラムのリセットで削除されるまで、有効になります。 < 0: プログラムされた許容範囲値が削除されます ⇒ マシンデータで事前に設定された許容範囲値が再度有効になります。	

注記

次の例に示すように、最後にプログラムされたアドレスが常に優先されます。

- CTOLG0 が有効な STOLF でプログラムされている場合、CTOLG0 でプログラムされた許容範囲値が輪郭の平滑に適用されます。
- OTOLG0 が有効な STOLF でプログラムされている場合、OTOLG0 でプログラムされた許容範囲値が方向の平滑に適用されます。
- STO を再度プログラムすると、輪郭許容範囲および方向許容範囲の許容範囲係数が適用されます。

例

例 1: 相対 G0 許容範囲の調整

プログラムコード	コメント
COMPCAD G645 G1 F10000	; コンプレッサ機能 COMPCAD
X...Y...Z...	; ここで、マシンデータとセッティングデータが適用されます。
X...Y...Z...	
X...Y...Z...	
G0 X...Y...Z...	
G0 X...Y...Z...	; マシンデータ \$MC_G0_TOLERANCE_FACTOR (=3 など) がここで適用されます。つまり、スムージング許容範囲は次の通りです。
	\$MC_G0_TOLERANCE_FACTOR * \$MA_COMPRESS_POS_TOL
CTOL=0.02	
STOLF=4	
G1 X...Y...Z...	; これ以降、0.02 mm の輪郭の許容範囲の適用が開始されます。
X...Y...Z...	
X...Y...Z...	

プログラムコード	コメント
G0 X...Y...Z...	
X...Y...Z...	; ここから、G0 許容範囲係数 4 が適用されます、つまり輪郭許容範囲は 0.08 mm です。
...	

例 2:絶対 G0 許容範囲の調整

次の絶対 G0 許容範囲をマシンデータで事前に設定してください。

- G0 輪郭許容範囲:0.1
- G0 方向許容範囲:1.0

これらの許容範囲は一時的にパートプログラムで調整されます:

プログラムコード	コメント
COMPCAD G645 G1 F10000	; コンプレッサ機能 COMPCAD
X...Y...Z...	; 設定したワーク加工許容範囲がここから適用されます。
X...Y...Z...	
X...Y...Z...	
G0 X...Y...Z...	
G0 X...Y...Z...	; 設定した絶対 G0 許容範囲がここで適用されます。
CTOLG0=0.2 OTOLG0=2.0	; 絶対 G0 許容範囲のプログラミング。
G1 X...Y...Z...	
X...Y...Z...	
X...Y...Z...	
G0 X...Y...Z...	
X...Y...Z...	; プログラムした G0 許容範囲がここから適用されます。
...	

詳細情報

G0 許容範囲係数の読み取り

早送り移動の現在有効な許容範囲係数を、システム変数を使用して読み取ることができます。

- 次のシステム変数は、シンクロナイズドアクション、またはパートプログラムで先読み停止をおこなって読み取ります。

\$AC_STOLF	有効な G0 許容範囲係数
	現在のメインランブロックの処理時に有効であった G0 許容範囲係数。

3.9 動作命令

- 次のシステム変数は、パートプログラムでは先読み停止をおこないません。

\$P_STOLF プログラム指令の G0 許容範囲係数

動作中のパートプログラムで STOLF に値を指定していない場合は、これらの 2 つのシステム変数は、マシンデータで設定した値を示します。

ブロックで早送り (G0) が動作中でない場合は、これらのシステム変数が常に値 1 を示します。

絶対 G0 許容範囲の読み取り

早送り移動の現在有効な絶対許容範囲を、システム変数を使用して読み取ることができます。

- 次のシステム変数は、シンクロナイズドアクション、またはパートプログラムで先読み停止をおこなって読み込みます。

\$AC_CTOL_G0_ABS G0 移動の有効な輪郭許容範囲
現在のメインランブロックが先読みされたときの有効な G0 輪郭許容範囲。

\$AC_OTOL_G0_ABS G0 移動の有効な方向許容範囲
現在のメインランブロックが先読みされたときの有効な G0 方向許容範囲。

- 次のシステム変数は、パートプログラムでは先読み停止をおこないません。

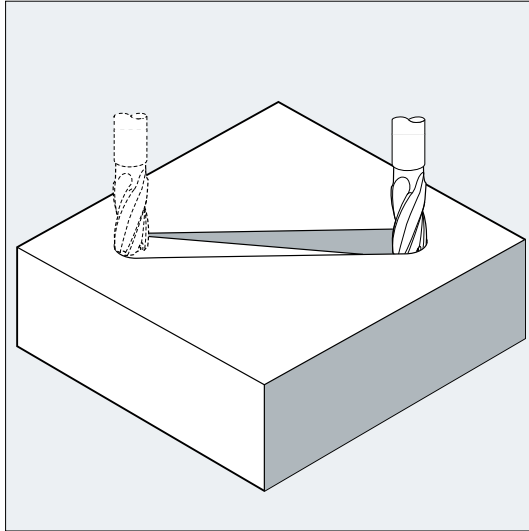
\$P_CTOL_G0_ABS G0 移動のプログラムされた輪郭許容範囲

\$P_OTOL_G0_ABS G0 移動のプログラムされた方向許容範囲

動作中のパートプログラムで CTOLG0 および OTOLG0 に値を指定していない場合は、これらの 2 つのシステム変数は、マシンデータで設定した値を示します。

3.9.5 直線補間(G1)

G1 を使用すると、工具は空間に任意に配置された斜線、または直線を、その線に平行に移動します。直線補間では、3 次元表面、溝などの加工ができます。



構文

```
G1 X... Y... Z ... F...  
G1 AP=... RP=... F...
```

意味

G1:	送り速度で直線補間(直線補間)
X...Y...Z...:	直交座標の終点
AP=...:	極座標の終点、この場合は極角度
RP=...:	極座標の終点、この場合は極半径
F...:	mm/min 単位の送り速度。工具は、現在の起点からプログラム指令終点まで、直線上を送り速度 F で移動します。終点は、直交座標または極座標で入力できます。ワークは、この軌跡に沿って加工されます。 例:G1 G94 X100 Y20 Z30 A40 F100 X、Y、Z の終点へ、100 mm/min の送り速度でアプローチします。回転軸 A は同期軸として移動するため、4 つの移動はすべて同時に完了します。

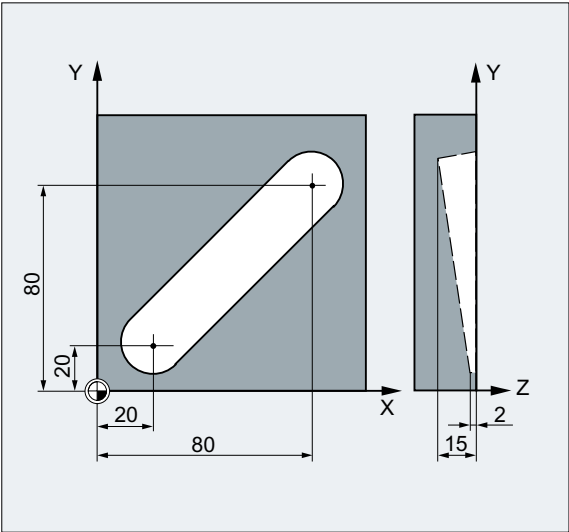
注記

G1 はモーダルです。
主軸速度 S と主軸回転方向 M3/M4 を、加工のために指定してください。
軌跡送り速度 F が適用される軸グループは、FGROUP で定義できます。詳細については、「軌跡動作」の章を参照してください。

例

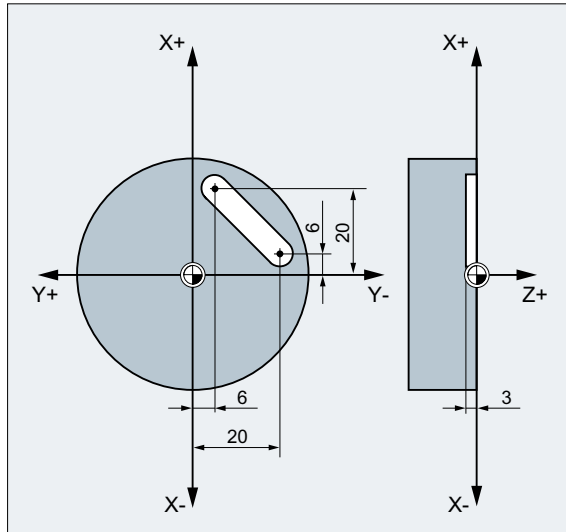
例 1:溝の加工(フライス加工)

工具は X/Y 方向に、起点から終点まで移動します。切り込みは、同時に Z 方向に実行されます。



プログラムコード	コメント
N10 G17 S400 M3	; 作業平面の選択、主軸は右回り
N20 G0 X20 Y20 Z2	; 開始位置へアプローチします。
N30 G1 Z-2 F40	; 工具の切り込み。
N40 X80 Y80 Z-15	; 斜線上の移動
N50 G0 Z100 M30	; 工具交換のための後退

例 2 :溝の加工(旋削)



プログラムコード	コメント
N10 G17 S400 M3	; 作業平面の選択、主軸は右回り
N20 G0 X40 Y-6 Z2	; 開始位置へアプローチします。
N30 G1 Z-3 F40	; 工具の切り込み。
N40 X12 Y-20	; 斜線上の移動
N50 G0 Z100 M30	; 工具交換のための後退

3.9.6 円弧補間

3.9.6.1 一覧

円弧補間により、一周円または円弧を加工できます。

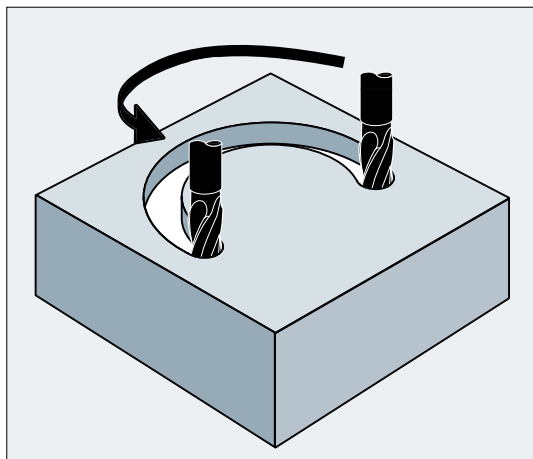


図 3-3 用途例:円弧のフライス削り

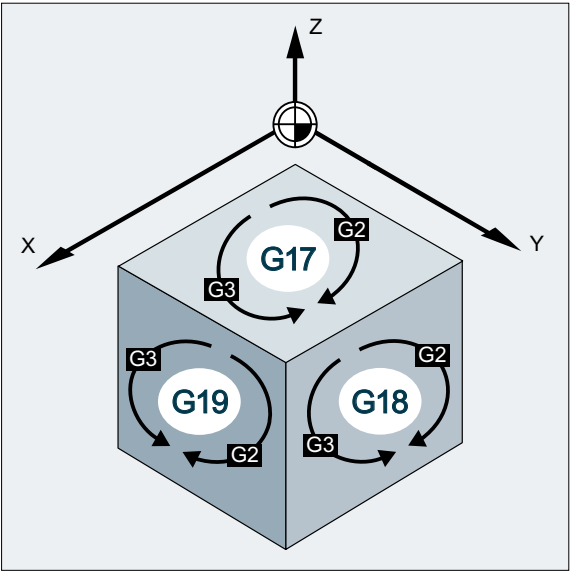
プログラムオプション

このコントロールシステムの円弧移動のプログラムには 2 つの選択肢があります。これにより、ほとんどのタイプの図面の寸法を直接、実行できます。

- 中心点と終点による円弧補間(G2/G3、X... Y... Z...、I... J... K...) (ページ 213)
- 半径と終点による円弧補間(G2/G3、X... Y... Z...、CR) (ページ 216)
- 開口角度と終点/中心点による円弧補間(G2/G3、X...Y...Z.../I...J...K...、AR) (ページ 218)
- 極座標による円弧補間(G2/G3、AP、RP) (ページ 220)
- 中間点と終点による円弧補間(CIP、X... Y... Z...、I1... J1... K1...) (ページ 222)
- 接線方向の遷移による円弧補間(CT、X... Y... Z...) (ページ 225)

円弧補間平面

制御装置が円弧回転方向(G2 で右回り、または G3 で左回り)を計算するには、作業平面設定 (ページ 166)が必要です。



例外:

選択した作業平面外で円弧を作成することもできます(この作業平面に開口角度が指定されていない場合)。この場合は、円弧終点として指定する軸識別子で、円弧平面が特定されます。

3.9.6.2 中心点と終点による円弧補間(G2/G3、X... Y... Z...、I... J... K...)

補間の円弧軌跡要素の中心点と終点を使用する円弧補間のバージョンです。

終点を使用せずに円弧をプログラムした場合、結果は一周円となります。

構文

```
G2/G3 X... Y... Z... I... J... K...
G2/G3 X... Y... Z... I=AC (...) J=AC (...) K=(AC...)
```

意味

G2:	右回りの円弧補間	
	効果:	モーダル

3.9 動作命令

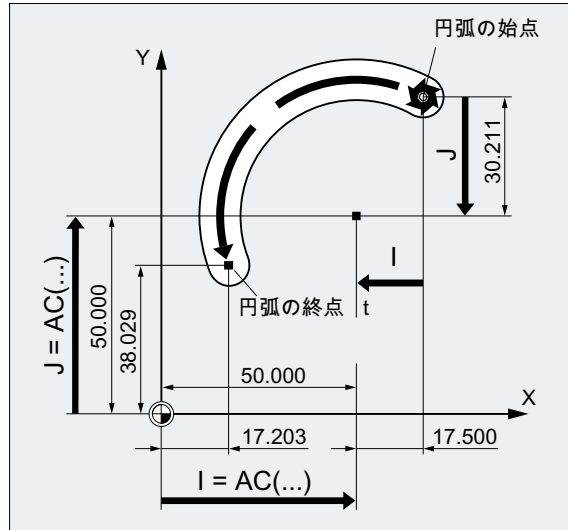
G3:	左周りの円弧補間	
	効果:	モーダル
X...Y...Z...:	直交座標による円弧終点。 現在有効な寸法の表記設定 G90/G91 または...=AC (...) / ...=IC (...) により、円弧終点の座標は、アブソリュート指令またはインクレメンタル指令によって解釈されます。	
I...J...K...:	円弧の中心点座標を X、Y、Z 方向に記述する補間パラメータ 初期設定では、中心点座標は、円弧起点に対するインクレメンタル指令で入力されます。 中心点座標がワーク原点に対するアブソリュート指令で入力される場合、補間パラメータ I、J、K は以下のようにプログラムしてください。 I=AC (...) J=AC (...) K=AC (...) 注 値 0 の補間パラメータは省略できますが、関連する 2 番目のパラメータは常に指定してください。	

注記

初期設定の G90/G91 のアブソリュート指令とインクレメンタル指令は、円弧終点のみに適用されます。

例

例 1: フライス加工



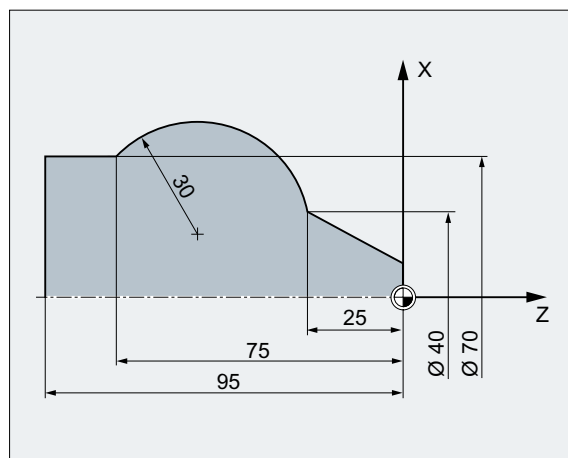
インクレメンタル指令を使用した中心点データ

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 I-.17.5 J-.30.211 F500
```

アブソリュート指令を使用した中心点データ

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 I=AC(50) J=AC(50)
```

例 2:旋削



インクレメンタル指令を使用した中心点データ

N120 G0 X12 Z0
N125 G1 X40 Z-25 F0.2

3.9 動作命令

```
N130 G3 X70 Z-75 I-3.335 K-29.25
N135 G1 Z-95
```

アブソリュート指令を使用した中心点データ

```
N120 G0 X12 Z0
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 I=AC(33.33) K=AC(-54.25)
N135 G1 Z-95
```

3.9.6.3 半径と終点による円弧補間(G2/G3、X... Y... Z...、CR)

補間の円弧軌跡要素の半径と終点を使用する円弧補間のバージョンです。

注記

このバージョンでは、一周円(移動角度 360°)はプログラムできません。

構文

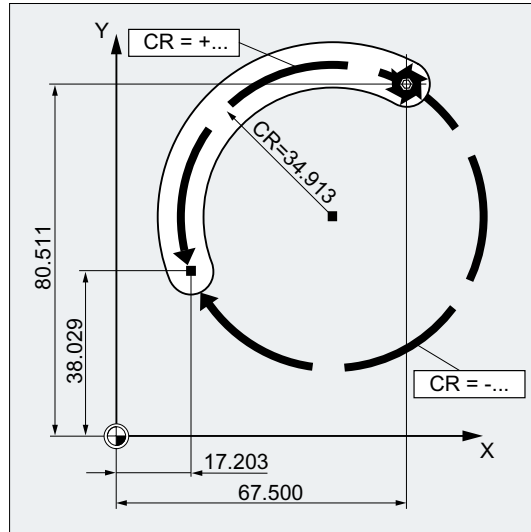
G2/G3 X... Y... Z... CR=±...

意味

G2:	右回りの円弧補間	
	効果:	モーダル
G3:	左回りの円弧補間	
	効果:	モーダル
X...Y...Z...:	直交座標による円弧終点。 現在有効な寸法の表記設定 G90/G91 または...=AC(...) / ...=IC(...)により、終点座標は、アブソリュート指令またはインクレメンタル指令によって解釈されます。	
CR=±...:	円弧半径 符号は、移動角度が 180°より大きいか小さいかを示します。+ 符号は省略できます。	
	CR=+...:	移動角度 ≤ 180°
	CR=-...:	移動角度 > 180°
	注 プログラム可能な半径の最大サイズについて、実質的な制限はありません。	

例

例 1: フライス加工

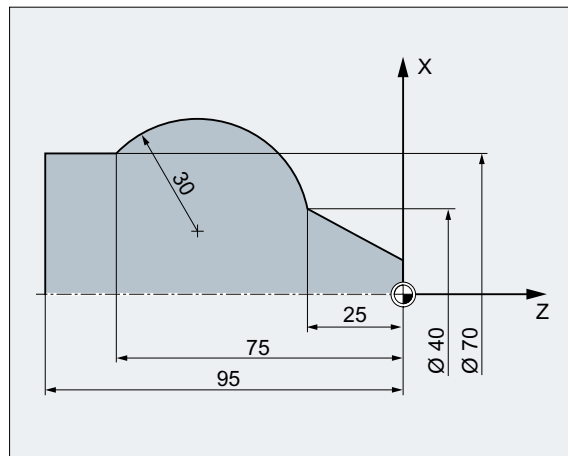


プログラムコード

```

N10 G0 X67.5 Y80.511
N20 G3 X17.203 Y38.029 CR=34.913 F500
...
```

例 2 : 旋削



プログラムコード

```

...
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 CR=30
```

3.9 動作命令

プログラムコード
N135 G1 Z-95
...

3.9.6.4 開口角度と終点/中心点による円弧補間(G2/G3、X...Y...Z... / I...J...K..., AR)

補間の円弧軌跡要素の**開口角度**と、**中心点**または**終点**を使用する円弧補間のバージョンです。

注記
このバージョンでは、一周円(移動角度 360°)はプログラムできません。

構文

G2/G3 X... Y... Z... AR=...

G2/G3 I... J... K... AR=...

意味

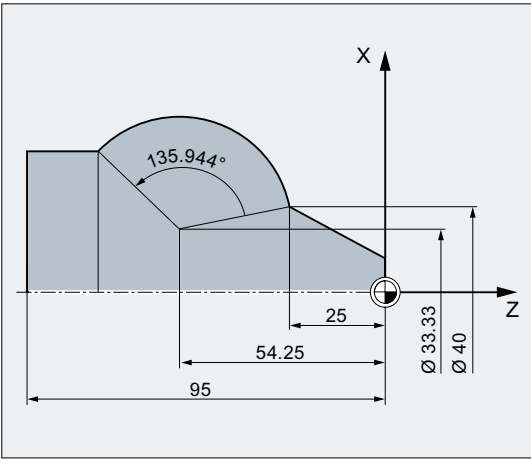
G2:	右回りの円弧補間	
	効果:	モーダル
G3:	左回りの円弧補間	
	効果:	モーダル
X...Y...Z...:	直交座標による円弧終点。 現在有効な寸法の表記設定 G90/G91 または...=AC(...) / ...=IC(...)により、円弧終点の座標は、アブソリュート指令またはインCREMENTAL指令によって解釈されます。	

例

[illegible]

N10	G0	X67.5	Y80.211				
N20	G3	X17.203	Y38.029	AR=140.134	F500		
N20	G3	I-17.5	J-30.211	AR=140.134	F500		

例 2 :旋削



プログラムコード

```
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 AR=135.944
N130 G3 I-3.335 K-29.25 AR=135.944
N130 G3 I=AC(33.33) K=AC(-54.25) AR=135.944
N135 G1 Z-95
```

3.9.6.5 極座標による円弧補間(G2/G3、AP、RP)

補間の**極座標の円弧終点**を使用する円弧補間のバージョンです。

以下の規則が適用されます。

- 極は円弧中心にあります。
- 極半径は円弧半径に対応します。

構文

G2/G3 絶対座=... レシピ処理=...

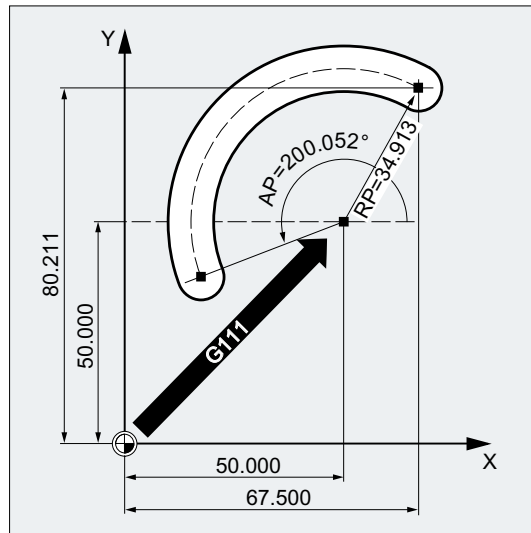
意味

G2:	右回りの円弧補間	
	効果:	モーダル
G3:	左回りの円弧補間	
	効果:	モーダル

絶対圧=...レシピ 処理=...:	極座標による円弧終点。	
	絶対圧=...:	極角度
	レシピ処理 =...:	極座標(≠ 円弧半径)

例

例 1:フライス加工



プログラムコード

```

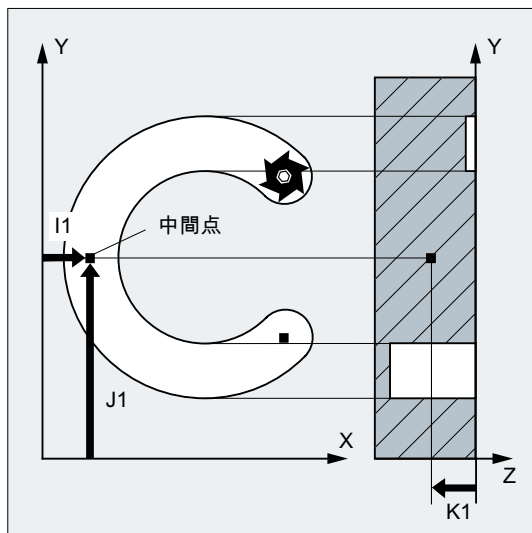
N10 G0 X67.5 Y80.211
N20 G111 X50 Y50
N30 G3 RP=34.913 AP=200.052 F500

```

N125	G1	X40	Z-25	F0.2
N130	G111	X33.33	Z-54.25	
N135	G3	RP=30	AP=142.326	
N140	G1	Z-95		

G 命令 CIP でプログラムした円弧補間バージョンを使用すると、空間の斜面に存在する円弧を補間できます。

移動方向は、起点、中間点、および終点の順に特定されます。



構文

CIP X... Y... Z... I1=AC (...) J1=AC (...) K1=(AC...)

意味

CIP:	中間点経由の円弧補間	
	有効:	モーダル
X...Y...Z...:	直交座標による円弧終点。 現在有効な寸法の表記設定 G90/G91 または...=AC(...) / ...=IC(...) により、円弧終点の座標は、アブソリュート指令またはインクレメンタル指令によって解釈されます。	
I1...J1...K1...:	円弧の中間点座標を X、Y、Z 方向に記述する補間パラメータ 現在有効な寸法の表記設定 G90/G91 または...=AC(...) / ...=IC(...) により、円弧中間点の座標は、アブソリュート指令またはインクレメンタル指令によって解釈されます。 注 値 0 の補間パラメータは省略できますが、関連する 2 番目のパラメータは常に指定してください。	

注記

初期設定の G90/G91(アブソリュート指令またはインクレメンタル指令)は、円弧の中間点または円弧終点のみに適用されます。

インクレメンタル指令 G91 または...=IC(...) が有効な場合、円弧起点が、中間点と終点の基準として使用されます。

注記

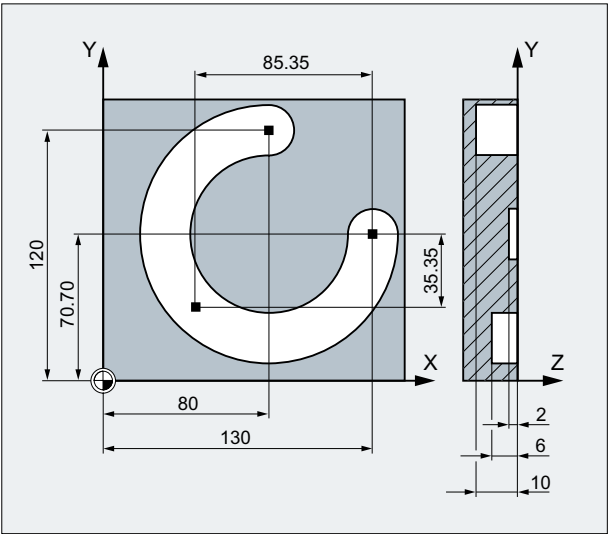
旋削用途

補間パラメータの直径指定は、CIP による円弧軌跡のプログラミングではサポートされていません。そのため、径方向軸の補間パラメータは、**半径**でプログラミングしてください。

例

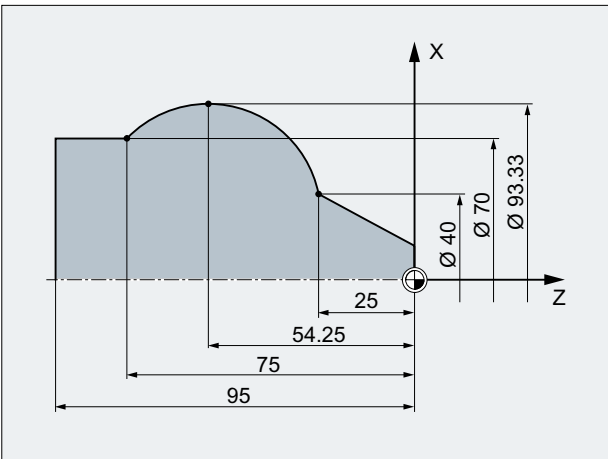
例 1:フライス加工

傾斜した円弧の溝を加工するために、3つの補間パラメータで中間点を指定し、3つの座標で終点を指定して円弧を記述します。



プログラムコード	コメント
N10 G0 G90 X130 Y70.70 S800 M3	; 起点にアプローチします。
N20 G17 G1 Z-2 F100	; 工具の送り速度。
N30 CIP X80 Y120 Z-10 I1=IC(-85.35) J1=IC(-35.35) K1=-6	; 円弧終点と中間点: ; 3つのジオメトリ軸すべての座標
N40 M30	; プログラム終了

例 2 :旋削



プログラムコード	コメント
...	
N125 G1 G90 X40 Z-25 F0.2	
N130 CIP X70 Z-75 I1=IC(26.665) K1=IC(-29.25)	; 径方向軸の補間パラメータ I1 は半径でプログラミングしてください。
; または	
; N130 CIP X70 Z-75 I1=46.665 K1=-54.25	
N135 G1 Z-95	

3.9.6.7 接線方向の遷移による円弧補間(CT、X... Y... Z...)

G 命令 CT でプログラムした円弧補間バージョンを使用すると、以前にプログラムされた輪郭要素と接線方向で接する円弧を補間できます。

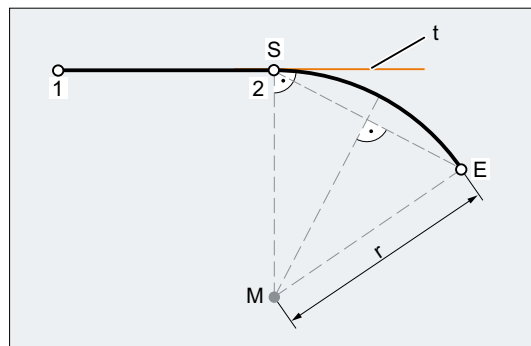
円弧は、**始点**と**終点**、および**始点での接線**の方向によって定義されています。

注記

始点での接線の方向。

CT ブロックの起点での接線の方向は、最後の移動動作ブロックの、プログラム指令輪郭終了部の接線から特定されます。

このブロックと実行中のブロックの間には、いくつかの移動情報のないブロックが存在できます。



S 始点

E 終点

M 円の中心点

R 円弧半径

t 移動動作があった最後のブロックの、プログラム指令輪郭の終点の接線。

図 3-4 円弧軌跡 S-E を接続する直線部 1~2 の接線方向

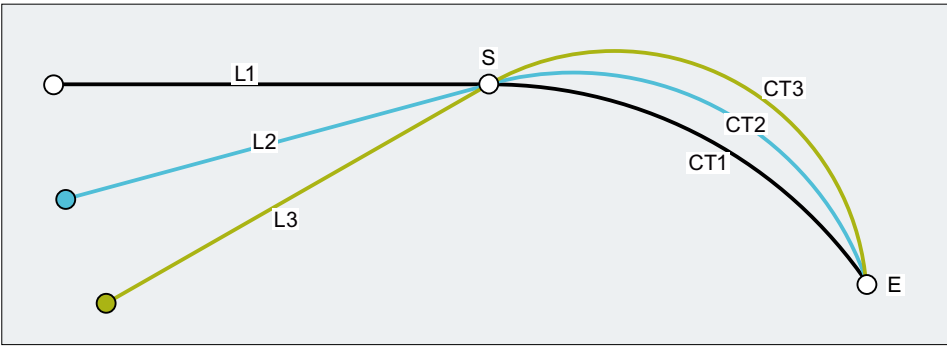


図 3-5 接線方向で接する円弧軌跡は、以前の輪郭要素によって異なります。

構文

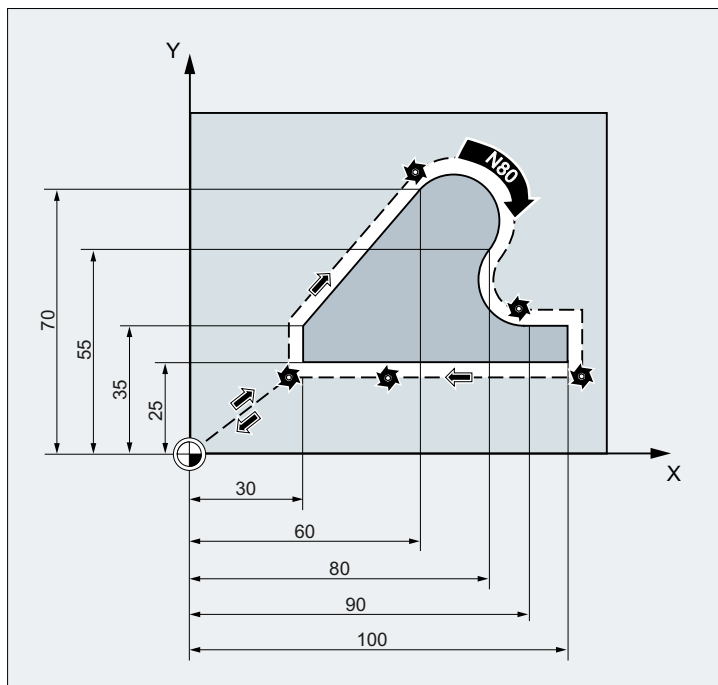
CT X... Y... Z...

意味

CT:	接線方向の遷移による円弧補間	
	効果:	モーダル
X...Y...Z...:	直交座標による円弧終点。 現在有効な寸法の表記設定 G90/G91 または...=AC(...) / ...=IC(...)により、円弧終点の座標は、アブソリュート指令またはインクレメンタル指令によって解釈されます。	

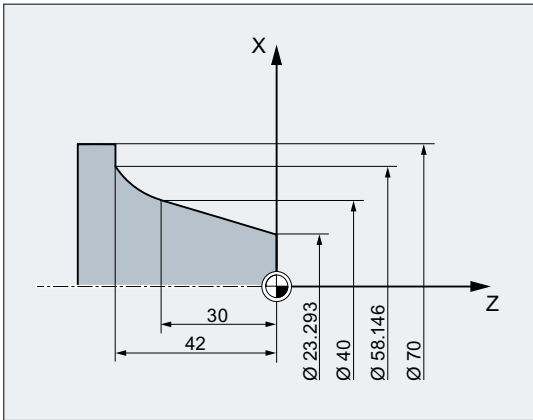
例

例 1: フライス加工



プログラムコード	コメント
N10 G0 Z100	
N20 G17 T1 M6	
N30 G0 X0 Y0 Z2 M3 S300 D1	
N40 Z-5 F1000	; 工具を送ります。
N50 G41 X30 Y25 G1 F1000	; 工具径補正をオンにします。
N60 Y35	; 輪郭をフライス加工します。
N70 X60 Y70	
N80 CT X80 Y55	; 接線方向の遷移による円弧軌跡のプログラムです。
N90 X90 Y35	
N100 G1 X100	
N110 Y25	
N120 X30	
N130 G0 G40 X0 Y0	; 工具径補正をオフにします。
N140 Z100	; 工具を後退します。
N140 M30	

例 2 : 旋削



プログラムコード	コメント
...	
N110 G1 X23.293 Z0 F10	
N115 X40 Z-30 F0.2	
N120 CT X58.146 Z-42	; 接線方向の遷移による円弧軌跡のプログラムです。
N125 G1 X70	
...	

詳細情報

スプライン

スプラインの場合、接線方向は、最後の 2 点を通過する直線で定義されます。動作中の ENAT または EAUTO による A スプラインと C スプラインの場合、この方向は通常、スプラインの終点での方向とは異なります。

B スプラインの遷移は接線方向であり、その接線方向は、A スプライン、または C スプライン、および動作中の ETAN と同じように定義されます。

フレーム変更

接線を定義するブロックと CT ブロックの間でフレーム変更が行われる場合は、接線もこの変更の対象となります。

限界の事例

開始接線の延長線が終点を通過する場合は、円ではなく、直線が生成されます(限界の事例: 無限半径の円)。このような特別の場合は、TURN をプログラム指令しないようにするか、値を TURN=0 にしてください。

注記

値がこの限界の事例の傾向を示すと、無限半径の円弧が生成され、そして通常、TURN が 0 以外のときの加工は、ソフトウェアリミット値の違反によるアラームと共に中止されます。

円弧平面の位置

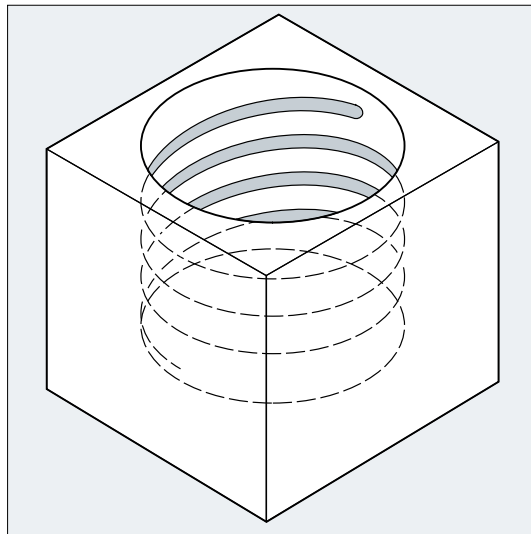
円弧平面の位置は、有効平面(G17~G19)により異なります。

以前のブロックの接線が有効平面にない場合は、その接線が有効平面に投影されたものが使用されます。

始点と終点が、有効平面に垂直で、同じ位置成分ではない場合は、円弧ではなく、ヘリカルが生成されます。

3.9.7 ヘリカル補間(G2/G3、TURN)

ヘリカル補間により、ねじや油溝などを加工できます。



ヘリカル補間では、次の 2 つの移動が重畳されて、並行して実行されます。

- 平面上の円弧移動
- 垂直直線移動(平面上の円弧動作に重畳される移動)

3.9 動作命令

構文

G2/G3 X... Y... Z... I... J... K... TURN=

G2/G3 X... Y... Z... I... J... K... TURN=

G2/G3 AR=... I... J... K... TURN=

G2/G3 AR=... X... Y... Z... TURN=

G2/G3 AP=... RP=... TURN=

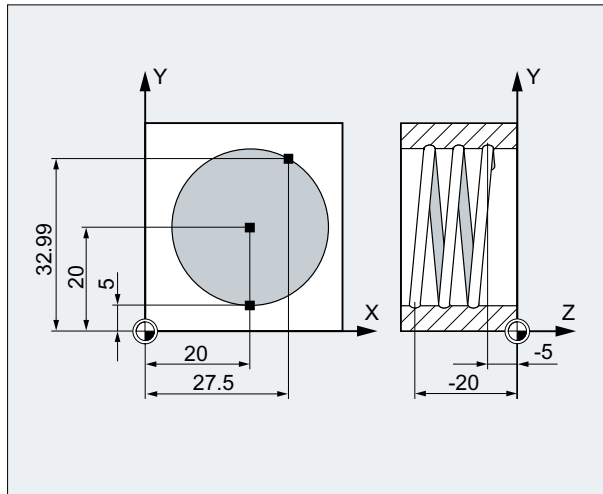
意味

G2:	円弧軌跡の右回り方向の移動
G3:	円弧軌跡の左回り方向の移動
X Y Z :	直交座標の終点
I J K :	直交座標の円弧中心点
AR:	開口角度
TURN= :	0～999 の範囲の追加の円弧軌跡の回数
AP=:	極角度
RP=:	極半径

注記

G2 と G3 はモーダルです。

例



プログラムコード

```

N10 G17 G0 X27.5 Y32.99 Z3
N20 G1 Z-5 F50
N30 G3 X20 Y5 Z-20 I=AC(20) J=AC(20) TURN=2
N40 M30

```

コメント

```

; 開始位置へアプローチします。
; 工具の送り速度。
; ヘリカル指定:開始位置から一周円を
2 回実行して、終点へ移動します。
; プログラム終了

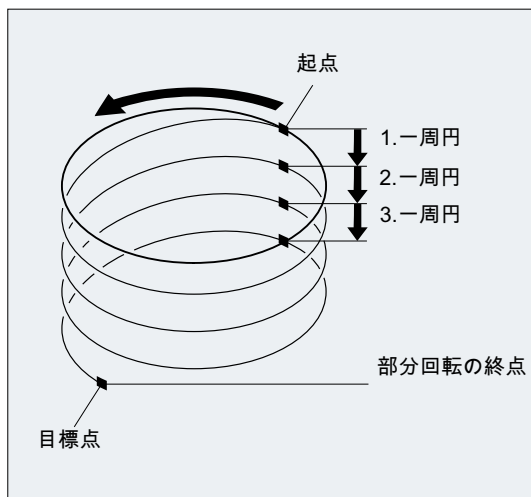
```

他の情報

移動処理

1. アプローチの開始点
2. TURN=でプログラム指令した一周円を実行します。
3. 円弧終了位置まで部分的な回転でアプローチします。
4. 手順 2 と 3 を、切り込みの深さ全体まで実行します。

ヘリカル加工のピッチは、一周円の回数と、プログラム指令した円弧終了位置から計算されます(切り込みの深さ全体まで実行)。



ヘリカル補間の終点のプログラム

補間パラメータの詳しい説明については、円弧補間を参照してください。

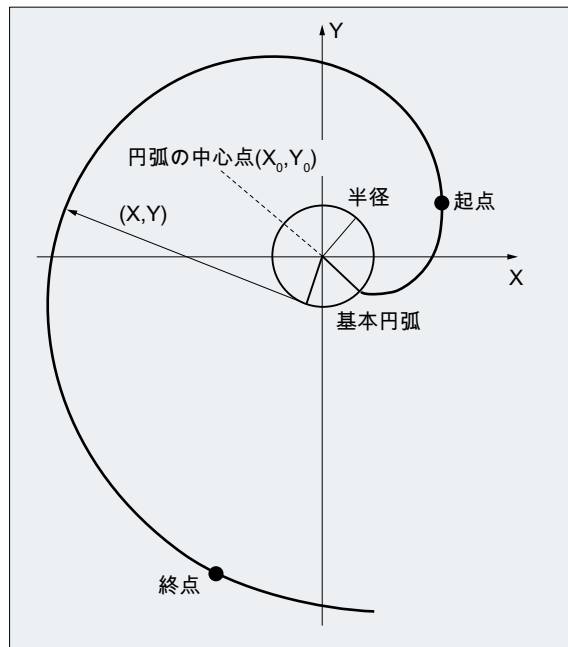
プログラム指令送り速度

ヘリカル補間の場合は、プログラム指令送り速度オーバーライド(CFC)を指定することを推奨します。FGROUP を使用して、プログラム指令送り速度で移動する軸を指定できます。詳細については、「軌跡動作」の章を参照してください。

3.9.8 インボリユート補間(INVCW、INVCCW)

円弧のインボリユートは曲線の周囲をアンラップするため、理想的なストリングの終端の軌跡となります。

インボリユート補間により、インボリユートに沿った軌道を加工できます。これは、基礎円を定義した平面で、プログラム指令の起点から、プログラム指令の終点まで実行されます。



終点は、次の 2 つの方法でプログラム指令できます。

1. 直交座標で直接プログラム指令
2. 開口角度を指定して間接的にプログラム指令(円弧軌跡プログラミングの開口角度のプログラミングも参照してください)

起点と終点が基礎円の平面にない場合は、円弧ヘリカル補間のように、空間で曲線に重畳されます。

有効平面に垂直な軌跡をさらに指定すると、インボリュートを空間で移動できます(円弧に対するヘリカル補間に相当します)。

構文

```
INVCW X...Y...Z...I...J...K...CR=...
INVCCW X...Y...Z...I...J...K...CR=...
INVCW I...J...K...CR=...AR=...
INVCCW I...J...K...CR=...AR=...
```

意味

INVCW:	インボリュート上を右回り方向に移動する命令
INVCCW:	インボリュート上を左回り方向に移動する命令
X...Y...Z...:	直交座標の終点の直接プログラミング

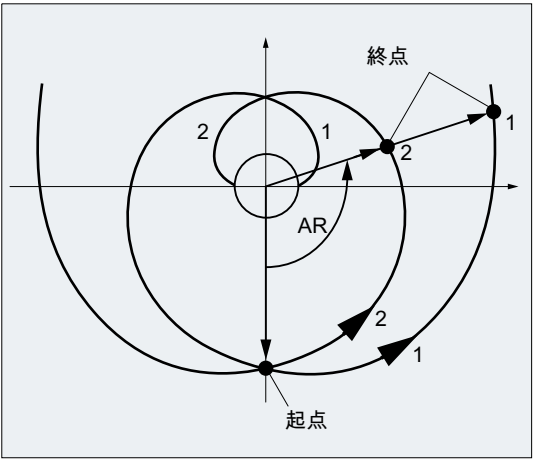
3.9 動作命令

I...J...K...:	直交座標の基礎円の中心点を定義する補間パラメータ 注: 指定される座標は、インボリュートの起点を基準とします。	
CR=...:	基礎円の半径	
AR=...:	開口角度(回転角度)の指定による終点の間接プログラミング 開口角度の開始は、円弧中心点から起点までの線です。	
	AR > 0:	インボリュートの軌跡は、 基礎円から離れる方向 に移動します。
	AR < 0:	インボリュートの軌跡は、 基礎円へ向かう方向 に移動します。 AR < 0 の場合は、終点が常に基礎円の外側である必要があるため、最大回転角度が制限されます。

開口角度の指定による終点の間接プログラミング

通知
開口角度が未定義 開口角度 AR を指定して終点の間接プログラミングをおこなう場合は、角度の符号を考慮してください。これは、符号が変更されると、結果的にインボリュートも変更され、異なる軌跡となるためです。

これは、次の例に図示されています。

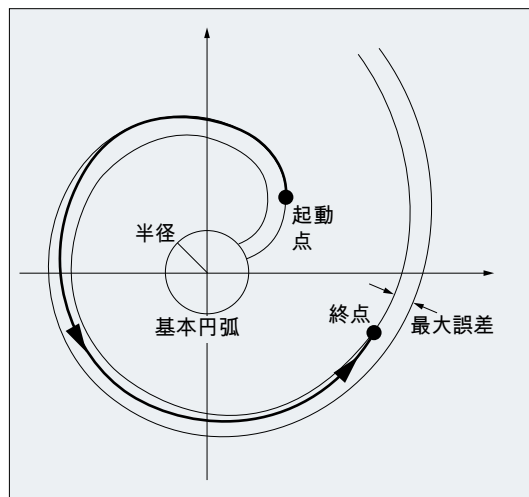


基礎円の半径と中心点、および起点と回転方向の指定(INVCW/INVCCW)は、インボリユート 1 と 2 で同じです。異なっているのは、開口角度の符号だけです。

- $AR > 0$ の場合は、軌跡はインボリユート 1 上にあり、終点 1 へアプローチします。
- $AR < 0$ の場合は、軌跡がインボリユート 2 上にあり、終点 2 へアプローチします。

必要条件

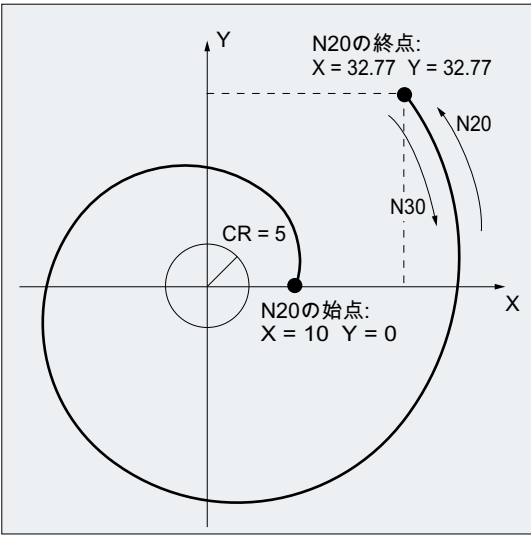
- 起点と終点は両方とも、インボリユートの基礎円の領域(I、J、K で指定された中心点を中心とする半径 CR の円)の外側に指定してください。この条件を満たさない場合は、アラームが発生して、プログラムの処理がキャンセルされます。
- 終点のプログラミングの 2 つのオプション(直交座標による直接プログラミング、および開口角度の指定による間接プログラミング)は、同時に使用できません。したがって、1 つのブロックで利用できるプログラミングオプションは、2 つのうちのいずれかのみです。
- プログラム指令終点が、起点と基礎円で定義したインボリユート上の正しい位置にない場合は、始点と終点で定義した 2 つのインボリユートの間で補間が実行されます(次の図を参照してください)。



終点の最大誤差は、マシンデータにより特定されます(→ 工作機械メーカー)。半径方向のプログラム指令終点の誤差が、マシンデータで定義された値より大きい場合は、アラームが発生して、プログラムの処理がキャンセルされます。

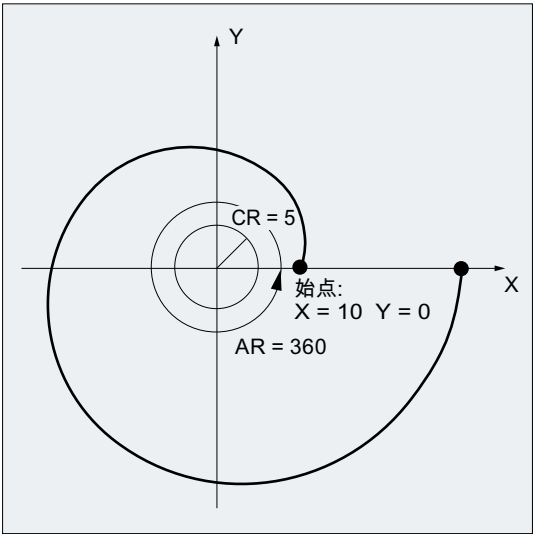
例

例 1:起点からプログラム指令終点までの左回りのインボリュート、および再度右回りに戻るインボリュート



プログラムコード	コメント
N10 G1 X10 Y0 F5000	; 開始位置にアプローチします。
N15 G17	; 加工平面として X/Y 平面を選択。
N20 INVCCW X32.77 Y32.77 CR=5 I-10 J0	; 左回りのインボリュート、直交座標の終点。
N30 INVCW X10 Y0 CR=5 I-32.77 J-32.77	; 右回りのインボリュート、起点は N20 の終点、新しい終点は N20 の起点、新しい円弧中心は新しい起点を基準として、旧円弧中心点と同じです。
...	

例 2:開口角度の指定による終点の間接プログラミングでの左回りのインボリユート



プログラムコード	コメント
N10 G1 X10 Y0 F5000	; 開始位置にアプローチします。
N15 G17	; 加工平面として X/Y 平面を選択。
N20 INVCCW CR=5 I-10 J0 AR=360	; 左回りのインボリユート、完全に 1 回転 (360°) して、基礎円から離れる方向 (正の角度の指定により) へ。
...	

3.9.9 輪郭定義

3.9.9.1 輪郭定義のプログラミング

機能

輪郭定義のプログラミングは、単純な輪郭をすばやく入力するために使用されます。

この方法で指令できるのは、直交座標と角度(ANG または ANG1 と ANG2)のいずれかまたは両方により指定された、遷移要素の面取り、または丸み付けによる 1 個、2 個、または 3 個以上の点による輪郭定義です。

3.9 動作命令

さらに、その他の軸(単独軸、または加工平面に垂直な軸)のアドレス文字、補助機能の指定、G 命令、速度などの任意の NC アドレスも、輪郭定義を記述するブロックで使用できます。

注記

経路計算

輪郭定義は、経路計算を使用すると、簡単にプログラム指令できます。これは、単純な、および複雑なワーク輪郭のプログラミングと画像表示を可能にする操作画面ツールです。経路計算によるプログラム指令輪郭は、パートプログラムに転送されます。

詳細情報:操作マニュアル

パラメータ設定

角度、丸み付け、および面取りの識別子は、次のマシンデータで定義されます。

MD10652 \$MN_CONTOUR_DEF_ANGLE_NAME (輪郭定義の角度の名称)

MD10654 \$MN_RADIUS_NAME (輪郭定義の丸み付けの名称)

MD10656 \$MN_CHAMFER_NAME (輪郭定義の面取りの名称)

注記

工作機械メーカーの仕様書を参照してください。

3.9.9.2 輪郭定義: 1 つの直線

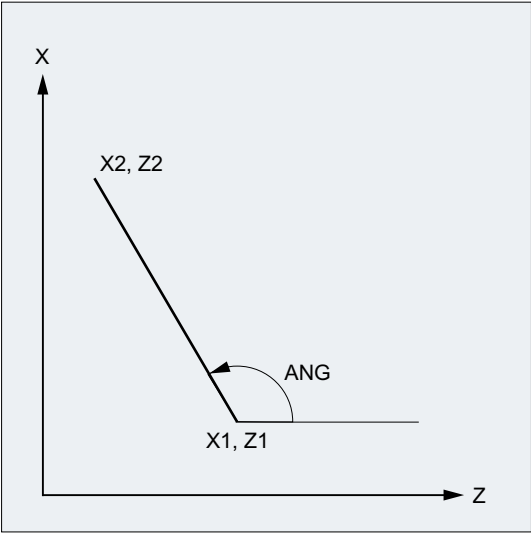
注記

後述の説明では、次のことを前提としています。

- G18 が有効である(⇒ 有効な作業平面は Z/X 平面)。
(ただし、輪郭定義のプログラミングは、G17 と G19 による制限はありません。)
 - 角度、丸み付け、および面取りには、次の識別子が定義されています。
 - ANG (角度)
 - RND(丸み付け)
 - CHR (面取り)
-

直線の終点は、次の指定により定義されます。

- 角度 ANG
- 1 つの 直交座標の終点(X2 または Z2)



ANG: 直線の角度
X1、Z1: 始点座標
X2、Z2: 直線の終点座標

構文

X... ANG=...
Z... ANG=...

意味

X... :	X 方向の終点座標
Z... :	Z 方向の終点座標
ANG:	角度プログラミングの識別子 指定値(角度)は、有効な作業平面の横軸(G18 の Z 軸)を基準とします。

例

プログラムコード	コメント
N10 X5 Z70 F1000 G18	; 開始位置へアプローチします
N20 X88.8 ANG=110	; 角度を指定した直線
N30 ...	

3.9 動作命令

または

プログラムコード	コメント
N10 X5 Z70 F1000 G18	; 開始位置へアプローチします
N20 Z39.5 ANG=110	; 角度を指定した直線
N30 ...	

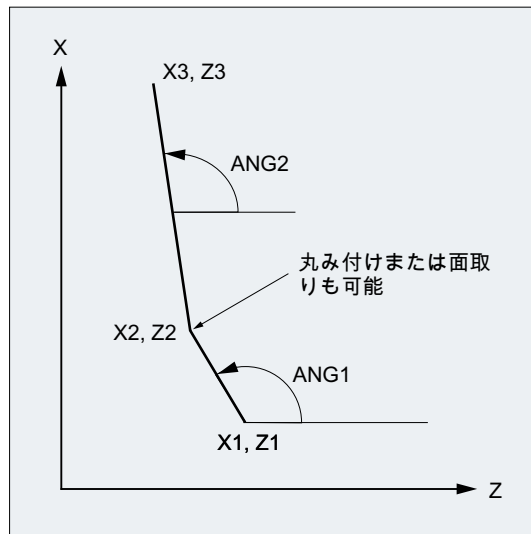
3.9.9.3 輪郭定義: 2 つの直線

注記

後述の説明では、次のことを前提としています。

- G18 が有効である(⇒ 有効な作業平面は Z/X 平面)。
(ただし、輪郭定義のプログラミングは、G17 と G19 による制限はありません。)
- 角度、丸み付け、および面取りには、次の識別子が定義されています。
 - ANG (角度)
 - RND (丸み付け)
 - CHR (面取り)

1 番目の直線の終点は、直交座標の指定、または 2 つの直線の角度の指定によりプログラム指令できます。2 番目の直線の終点は常に、直交座標でプログラム指令してください。2 つの直線の交点は、コーナ、曲線、または面取りとして指定できます。



- ANG1: 1 番目の直線の角度
 ANG2: 2 番目の直線の角度
 X1、Z1: 1 番目の直線の始点座標
 X2、Z2: 1 番目の直線の終点座標または
 2 番目の直線の始点座標
 X3、Z3: 2 番目の直線の終点座標

構文

角度の指定による、1 番目の直線の終点のプログラミング

- 直線間のコーナ遷移:

```
ANG=...
X... Z... ANG=...
```

- 直線間の丸み付け遷移:

```
ANG=... RND=...
X... Z... ANG=...
```

- 直線間の面取り遷移:

```
ANG=... CHR=...
X... Z... ANG=...
```

3.9 動作命令

座標の指定による、1 番目の直線の終点のプログラミング

- 直線間のコーナ遷移:

```
X... Z...  
X... Z...
```

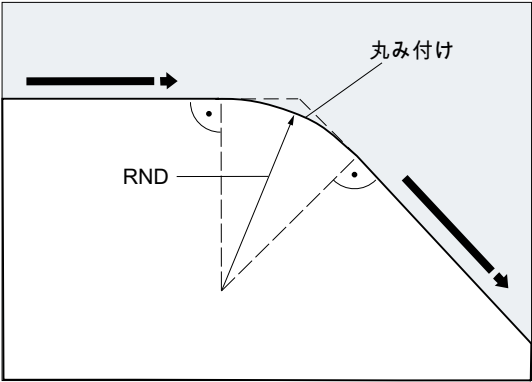
- 直線間の丸み付け遷移:

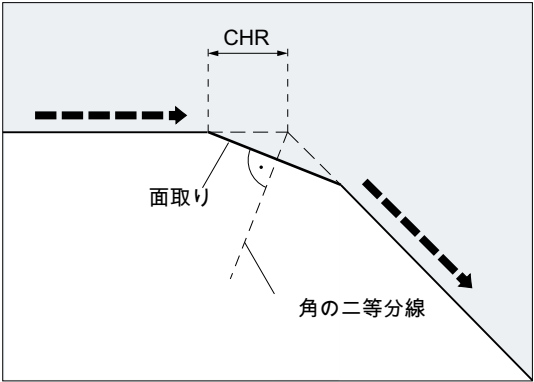
```
X... Z... RND=...  
X... Z...
```

- 直線間の面取り遷移:

```
X... Z... CHR=...  
X... Z...
```

意味

ANG=... :	角度プログラミングの識別子 指定値(角度)は、有効な作業平面の横軸(G18 の Z 軸)を基準とします。
RND=... :	丸み付けのプログラミングの識別子 指定値は、丸み付けの半径に対応します。 

CHR=... :	面取りのプログラミングの識別子 指定値は、移動方向の面取り幅に対応します。 
X... :	X 方向の座標
Z... :	Z 方向の座標

注記
面取りまたは丸み付けのプログラミングについて詳しくは、「面取り、丸み付け(CHF、CHR、RND、RNDM、FRC、FRCM) (ページ 274)」を参照してください。

例

プログラムコード	コメント
N10 X10 Z80 F1000 G18	; 開始位置へアプローチします。
N20 ANG=148.65 CHR=5.5	; 角度と面取りを指定した直線です。
N30 X85 Z40 ANG=100	; 角度と終点を指定した直線です。
N40 ...	

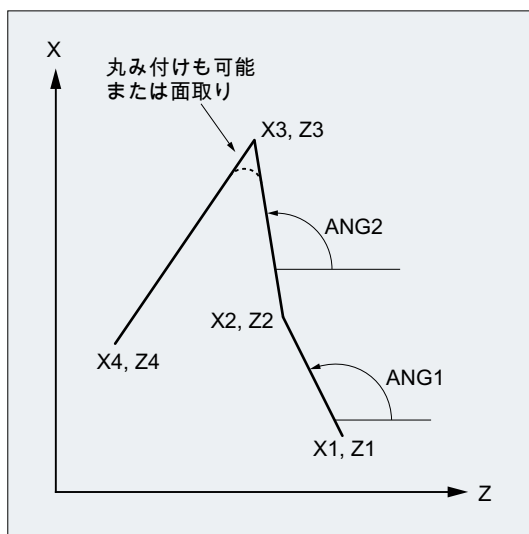
3.9.9.4 輪郭定義: 3 つの直線

注記

後述の説明では、次のことを前提としています。

- G18 が有効である(⇒ 有効な作業平面は Z/X 平面)。
(ただし、輪郭定義のプログラミングは、G17 と G19 による制限はありません。)
- 角度、丸み付け、および面取りには、次の識別子が定義されています。
 - ANG (角度)
 - RND(丸み付け)
 - CHR (面取り)

1 番目の直線の終点は、直交座標の指定、または 2 つの直線の角度の指定によりプログラム指令できます。2 番目と 3 番目の直線の終点は常に、直交座標でプログラム指令してください。これらの直線の交点は、コーナ、曲線、または面取りとして指定できます。



- ANG1: 1 番目の直線の角度
 ANG2: 2 番目の直線の角度
 X1、Z1: 1 番目の直線の始点座標
 X2、Z2: 1 番目の直線の終点座標または
 2 番目の直線の始点座標
 X3、Z3: 2 番目の直線の終点座標または
 3 番目の直線の始点座標
 X4、Z4: 3 番目の直線の終点座標

注記

ここで説明した 3 点輪郭定義のプログラミングは、任意に拡張して、3 点以上の輪郭定義にも使用できます。

構文

角度の指定による、1 番目の直線の終点のプログラミング

- 直線間のコーナ遷移:

```
ANG=...  
X... Z... ANG=...  
X... Z...
```

- 直線間の丸み付け遷移:

```
ANG=... RND=...  
X... Z... ANG=... RND=...  
X... Z...
```

- 直線間の面取り遷移:

```
ANG=... CHR=...  
X... Z... ANG=... CHR=...  
X... Z...
```

座標の指定による、1 番目の直線の終点のプログラミング

- 直線間のコーナ遷移:

```
X... Z...  
X... Z...  
X... Z...
```

- 直線間の丸み付け遷移:

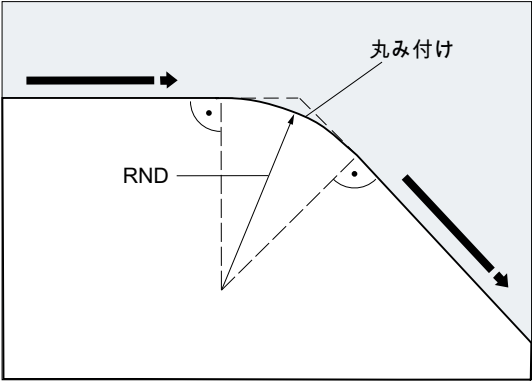
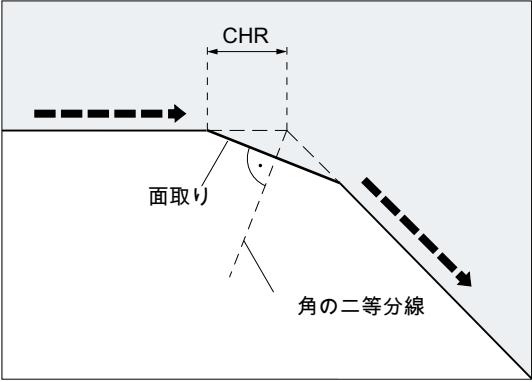
```
X... Z... RND=...  
X... Z... RND=...  
X... Z...
```

- 直線間の面取り遷移:

3.9 動作命令

X... Z... CHR=...
X... Z... CHR=...
X... Z...

意味

ANG=... :	角度プログラミングの識別子 指定値(角度)は、有効な作業平面の横軸(G18 の Z 軸)を基準とします。
RND=... :	丸み付けのプログラミングの識別子 指定値は、丸み付けの半径に対応します。 
CHR=... :	面取りのプログラミングの識別子 指定値は、移動方向の面取り幅に対応します。 
X... :	X 方向の座標
Z... :	Z 方向の座標

注記

面取りまたは丸み付けのプログラミングについては、「面取り、丸み付け(CHF、CHR、RND、RNDM、FRC、FRCM) (ページ 274)」を参照してください。

例

プログラムコード	コメント
N10 X10 Z100 F1000 G18	; 開始位置へアプローチします。
N20 ANG=140 CHR=7.5	; 角度と面取りを指定した直線です。
N30 X80 Z70 ANG=95.824 RND=10	; 角度と面取りを指定した、中間点までの直線です。
N40 X70 Z50	; 終点までの直線です。

3.9.9.5 輪郭定義: 角度による終点プログラミング**機能**

アドレス文字 A が NC ブロックに含まれる場合は、有効平面の軸の両方ともプログラム指令されていないか、軸の一方または両方のいずれかがプログラム指令されています。

プログラム指令軸数

- 有効平面の軸がプログラム指令されていない場合は、この指令のないブロックは、2つのブロックから成る輪郭定義の 1 番目または 2 番目のブロックのいずれかとなります。これが輪郭定義の 2 番目のブロックの場合は、有効平面の起点と終点が同じであることを意味しています。これにより、輪郭定義は単に、有効平面に垂直な移動になります。
- 有効平面の 1 つの軸のみがプログラム指令されている場合は、角度とプログラム指令直交座標、または 2 つのブロックから成る輪郭定義の 2 番目のブロックを使用して終点を明確に定義できる 1 つの直線となります。この 2 番目の場合では、抜けている座標は、到達した最終(モーダル)位置と同じ設定です。
- 有効平面の 2 つの軸がプログラム指令されている場合は、このブロックは 2 つのブロックから成る輪郭定義の 2 番目のブロックとなります。実行中のブロックより以前に、有効平面のプログラム指令軸なしで角度のブロックが指令されていない場合、このブロックは使用できません。

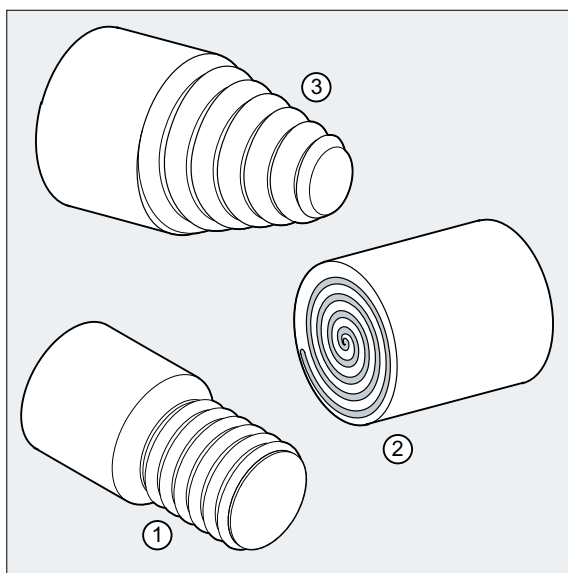
角度 A は、直線補間またはスプライン補間のプログラム指令に使用できます。

3.9.10 ねじ切り

3.9.10.1 固定リードのねじ切り(G33、SF)

固定リードのねじは、G33 を使用して加工できます。

- 円筒ねじ①
- スクロールねじ②
- テーパーねじ③

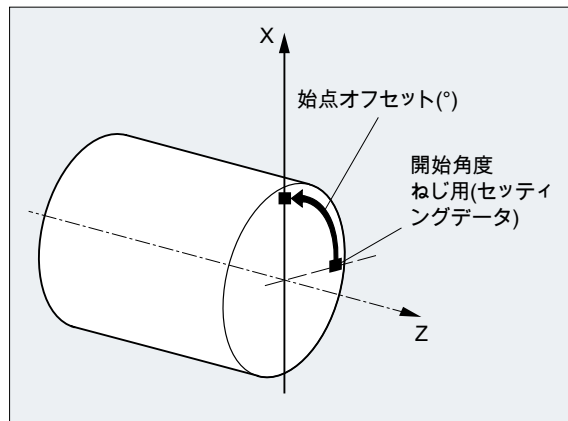


注記

G33 によるねじ切りの加工の必要条件は、位置検出器付き変速主軸です。

多条ねじ

多条ねじ(オフセット切り込みのあるねじ)は、起点オフセットを指定すると、加工できます。プログラミングは、アドレス SF を指令した G33 ブロックでおこなわれます。

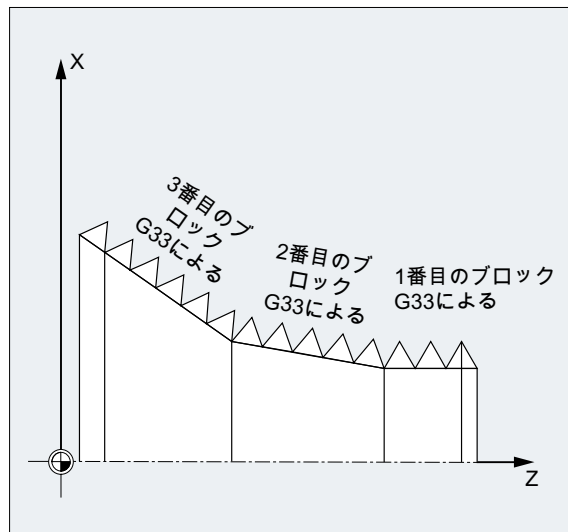


注記

起点オフセットを指定しない場合は、セッティングデータで定義した「ねじの開始角度」が使用されます。

連続ねじ

連続ねじは、複数の G33 ブロックを連続してプログラム指令すると、加工できます。



注記

連続軌跡モード G64 では、速度の不連続変化が発生しないように、先読み速度制御によってブロックがつながっています。

ねじの回転方向

3.9 動作命令

ねじの回転方向は、主軸の回転方向によって特定されます。

- M3 による右回りの場合は、右ねじが加工されます。
- M4 による左回りの場合は、左ねじが加工されます。

構文

ストレートねじ:

G33 Z... K...

G33 Z... K... SF=...

スクロールねじ:

G33 X... I...

G33 X... I... SF=...

テーパねじ:

G33 X... Z... K...

G33 X... Z... K... SF=...

G33 X... Z... I...

G33 X... Z... I... SF=...

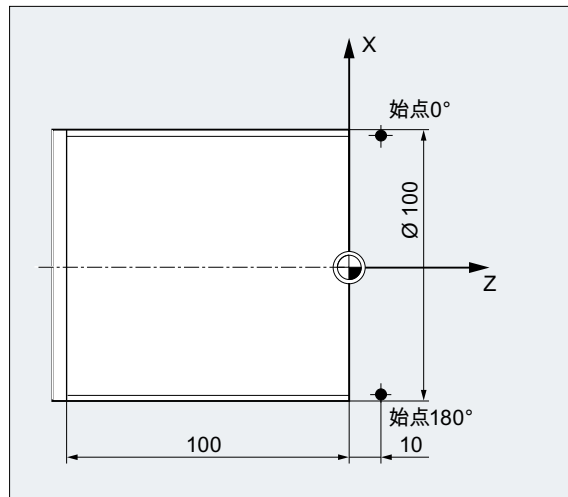
意味

G33:	固定リードのねじ切りをおこなう命令	
X...Y...Z...:	直交座標の終点	
I...:	X 方向のねじリード	
J...:	Y 方向のねじリード	
K...:	Z 方向のねじリード	
Z:	長手軸	
X:	径方向軸	
Z...K...:	ストレートねじの、ねじ長さとのリード	
X...I...:	スクロールねじの、ねじ径とのねじリード	
I...またはK...:	テーパねじのねじリード 指定(I...または K...)はテーパの角度を基準にします。	
	< 45°:	ねじリードは、K... (長手方向のねじリード)で指定します。
	> 45°:	ねじリードは、I... (径方向のねじリード)で指定します。
	= 45°:	ねじリードは、I...またはK...で指定できます。

SF=... :	起点オフセット(多条ねじの場合にのみ必要) 起点オフセットは、アブソリュート回転位置として指定します。	
	値の範囲:	0.0000~359.999°

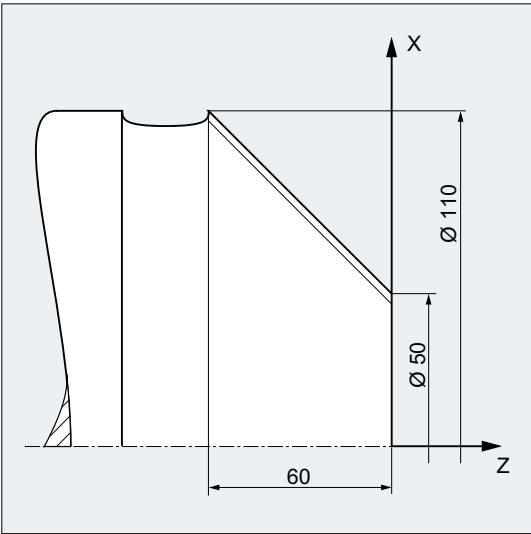
例

例 1:180°の起点オフセットによる二条ストレートねじ



プログラムコード	コメント
N10 G1 G54 X99 Z10 S500 F100 M3	; ワークオフセット、起点へアプローチして、主軸を起動します。
N20 G33 Z-100 K4	; ストレートねじ:Zの終点
N30 G0 X102	; 始点へ後退します。
N40 G0 Z10	
N50 G1 X99	
N60 G33 Z-100 K4 SF=180	; 2 番目の切り込み:起点オフセット 180°
N70 G0 X110	; 工具を後退します。
N80 G0 Z10	
N90 M30	; プログラム終了

例 2 :角度が 45°未満のテーパねじ

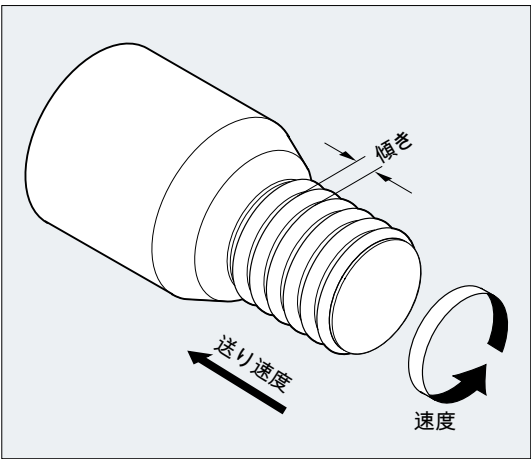


プログラムコード	コメント
N10 G1 X50 Z0 S500 F100 M3	; 起点へアプローチし、主軸を起動します。
N20 G33 X110 Z-60 K4	; テーパーねじ:XとZの終点、Z方向へK...によりねじリードを指定します(角度 < 45°のため)。
N30 G0 Z0 M30	; 後退、プログラム終了

詳細情報

G33 によるねじ切りの送り速度

プログラム指令主軸速度とねじリードから、制御装置は、旋削工具が長手方向か径方向または両方に、ねじ長さだけ移動するために必要な送り速度を計算します。送り速度 F は、G33 の場合には考慮されずに、最大軸速度(早送り)の制限が制御装置で監視されます。



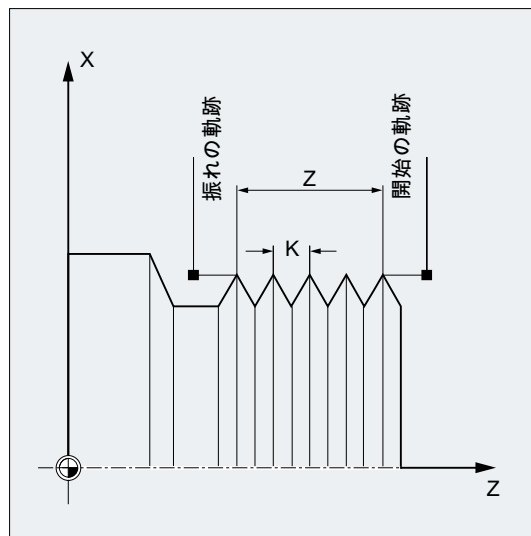
ストレートねじ

ストレートねじは、以下を使用して記述します。

- ねじ長さ
- ねじリード

ねじ長さは、アブソリュート指令またはインクリメンタル指令による直交座標 X、Y、Z のいずれか(旋盤の場合は、Z 方向を推奨します)で入力します。送り速度の加速または減速が発生する、切り始め軌跡と切り上げ軌跡を考慮した値も入力してください。

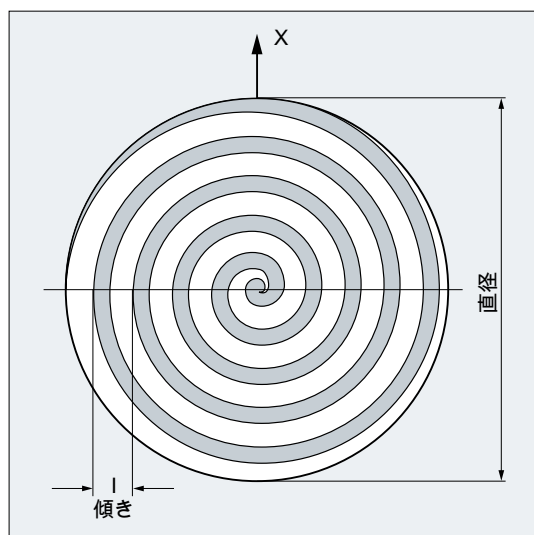
ねじリードは、アドレス I、J、K (旋盤の場合は、K を推奨します)で入力します。



スクロールねじ

スクロールねじは、以下を使用して記述します。

- ねじ直径(X 方向を推奨します)
- ねじリード(I を推奨します)



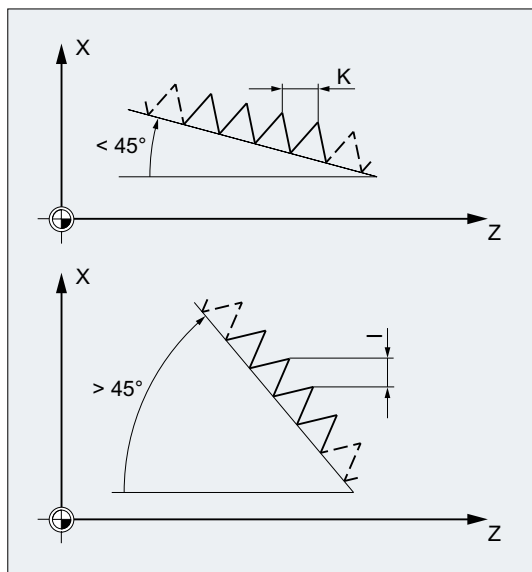
テーパねじ

テーパねじは、以下を使用して記述します。

- 長手方向と径方向の終点(テーパ輪郭)
- ねじリード

テーパ輪郭は、アブソリュート指令またはインクリメンタル指令による直交座標 X、Y、Z(旋盤の加工の場合は、X と Z 方向を推奨します)で入力します。送り速度の加速または減速が発生する、切り始め軌跡と切り上げ軌跡を考慮した値も入力してください。

リードの指定は、テーパの角度(長手軸とテーパ面との成す角度)により異なります。



3.9.10.2 可変リードねじ切り(G34、G35)

G34 命令と G35 命令により、G33 機能が拡張され、アドレス F によるねじピッチの変更をプログラム指令するオプションが追加されました。この結果、G34 の場合は、ねじピッチが単調増加となり、G35 の場合は、単調減少となります。したがって、G34 と G35 命令は、セルフタッピングねじの加工に使用できます。

構文

ピッチが単調増加するストレートねじ:

G34 Z... K... F...

ピッチが単調減少するストレートねじ:

G35 Z... K... F...

ピッチが単調増加するスクロールねじ:

G34 X... I... F...

ピッチが単調減少するスクロールねじ:

G35 X... I... F...

ピッチが単調増加するテーパねじ:

G34 X... Z... K... F...

G34 X... Z... I... F...

ピッチが単調減少するテーパねじ:

G35 X... Z... K... F...

G35 X... Z... I... F...

意味

G34:	ピッチが単調 増加 するねじ切りをおこなう命令
G35:	ピッチが単調 減少 するねじ切りをおこなう命令
X...Y...Z...:	直交座標の終点
I...:	X 方向のねじピッチ
J...:	Y 方向のねじピッチ
K...:	Z 方向のねじピッチ

3.9 動作命令

F...:	ねじピッチの変化	
	ねじの開始ピッチと最終ピッチがわかっている場合は、ねじピッチの変化を計算して、次の数式を使用してプログラム指令できます。	
	$F = \frac{k_e^2 - k_a^2}{2 * l_g} \text{ [mm/rev}^2\text{]}$	
	意味は以下のとおりです。	
	k _e :	最終ねじピッチ(軸目標点座標のねじピッチ) [mm/rev]
	k _a :	(I、JまたはKでプログラム指令された)開始ねじピッチ [mm/rev]
	l _g :	ねじ長さ[mm]

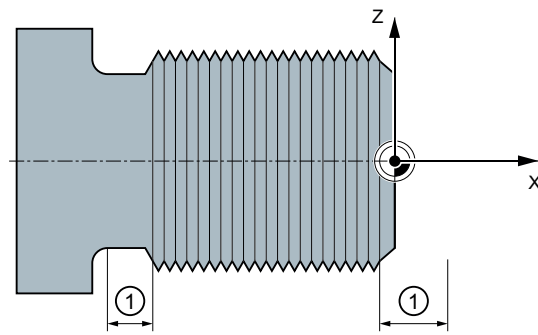
例

プログラムコード	コメント
N1608 M3 S10	; 主軸オン。
N1609 G0 G64 Z40 X216	; 起点にアプローチします。
N1610 G33 Z0 K100 SF=R14	; 一定のピッチ (100 mm/rev) が指定されたねじ切り
N1611 G35 Z-200 K100 F17.045455	; ピッチの減少。17.0454 mm/rev2 ブロック終点のピッチ。50 mm/rev
N1612 G33 Z-240 K50	; 加々速度なしでねじブロックを移動。
N1613 G0 X218	
N1614 G0 Z40	
N1615 M17	

3.9.10.3 G33、G34、および G35(DITS、DITE)でプログラムされた切り始めと切り上げの距離

ねじの切り始め距離と切り上げ距離は、パートプログラムで DITS と DITE アドレスにより指定できます。

ねじ切り軸は、指定された距離で加速、減速されます。



① 加工方向に対応した切り始め軌跡と切り上げ軌跡

短い切り始め軌跡

ねじ切り始めの部分で、工具開始カーブの余地がほとんどありません。このため、DITS を使用して、これをさらに短く指定してください。

短い切り上げ軌跡

ねじの切り上げの段差により、工具の減速カーブのためのスペースがなく、ワークと工具刃先が衝突する可能性を招きます。DITE を使用すると、減速カーブをより短く指定できます。ただし、機械システムの慣性のせいで、まだ衝突が発生することがあります。

対策:より短いねじをプログラム指令して、主軸速度を減速します。

注記

DITE は、ねじ山の最後で、丸み付け間隙として動作します。これにより、軸の移動がなめらかに変化します。

作用

プログラム指令した切り始めと切り上げ軌跡は、軌跡上の加速率だけを増加します。2 つの軌跡の一方の設定が、有効な加速でねじ切り軸が必要とする設定より大きい場合、ねじ切り軸は加速されるか、最高加速度に達すると減速されます。

構文

DITS=<値> DITE=<値>

意味

DITS:	ねじの切り始め軌跡の定義
DITE:	ねじの切り上げ軌跡の定義
<値>:	DITS と DITE では、位置ではなく、軌跡のみをプログラム指令します。 プログラム指令された切り始め/切り上げ距離は、現在の寸法設定に応じて 処理されます(インチ、メトリック)。

例

プログラムコード	コメント
...	
N40 G90 G0 Z100 X10 SOFT M3 S500	
N50 G33 Z50 K5 SF=180 DITS=1 DITE=3	; Z=53 でスミージングを開始します。
N60 G0 X20	

詳細情報

SD42010 \$SC_THREAD_RAMP_DISP

DITS や DITE を含むブロックがメインランに挿入された場合、プログラムされた切り始め/切り上げ時の距離がセッティングデータ SD42010 \$SC_THREAD_RAMP_DISP に伝送されます。

- SD42010 \$SC_THREAD_RAMP_DISP[0] = DITS のプログラム指令値
- SD42010 \$SC_THREAD_RAMP_DISP[1] = DITE のプログラム指令値

最初のねじブロックの前または最初のねじブロック内で切り始め/切り上げ軌跡がプログラム指令されていない場合、セッティングデータの現在の値が使用されます。

チャネル/モードグループ/プログラム終了リセット後の動作

DITS や DITE によって上書きされた SD 42010 値は、チャネル/モードグループ/プログラム終了リセットの後でも有効なままです。

ウォームスタート後の動作

ウォームスタートの場合、セッティングデータは、DITS や DITE による上書きの前に有効だった値にリセットされます(標準動作)。

ただし、ウォーム再起動の後、DITS および DITE によりプログラムされた値が依然として有効である場合、セッティングデータ SD42010 \$SC_THREAD_RAMP_DISP をマシンデータ MD10710 \$MN_PROG_SD_RESET_SAVE_TAB に含める必要があります。

MD10710 \$MN_PROG_SD_RESET_SAVE_TAB[<n>] = 42010

切り始め/切り上げ時の距離が非常に短い場合の特性

切り始め軌跡と切り上げ軌跡の一方または両方が非常に短い場合は、ねじ軸の加減速が設定値より大きくなります。これにより、軸に加減速の過負荷が生じます。

そして、ねじの切り始めに対してアラーム 22280 ("Programmed run-in path too short"(プログラム指令切り始め軌跡が短かすぎます))が発生します

(MD11411 \$MN_ENABLE_ALARM_MASK で当該の設定がされている場合)。このアラームは、情報を示すのみであり、パートプログラムの実行には影響を与えません。

3.9.10.4 ねじ切り時の高速リトラクト(LFON、LFOF、DILF、ALF、LFTXT、LFWP、LFPOS、POLF、POLFMASK、POLFMLIN)

「ねじ切り時の高速リトラクト(G33)」機能を使用すると、次のような状況でも修復が困難な損傷を引き起こさずに、ねじ切りを中断できます。

- DB21, ... DBX7.3 で NC ストップを開始
- アラームで NC ストップを自動的に開始
- 高速入力の切り替え (ページ 621)

後退移動は以下を使用してプログラム指令できます。

- 後退軌跡と後退方向(相対値)
- 後退位置(絶対値)

注記

NC ストップ信号

ねじ切り時に、次の NC ストップ信号は高速リトラクトを起動しません。

- DB21, ... DBX3.4 (NC ストップ軸と主軸)
- DB21, ... DBX7.2 (ブロック境界での NC ストップ)

タッピング

「高速リトラクト」機能はタッピング (G331/G332)では使用できません。

構文

高速リトラクト、後退軌跡と後退方向による後退移動の設定:

G33 ... LFON DILF=<値> LFTXT/LFWP ALF=<値>

高速リトラクト、後退位置による後退移動の設定:

POLF[<軸識別子>]=<値> LFPOS

3.9 動作命令

POLFMASK/POLFMLIN(<軸 1 名称>,<軸 2 名称>,...)
G33 ... LFON

ねじ切り時の高速リトラクトの解除:
LFOF

意味

LFON:	ねじ切り時の高速リトラクトを設定(G33)	
LFOF:	ねじ切り時の高速リトラクトを解除(G33)	
DILF= :	後退軌跡の長さを定義 マシンデータ(MD21200 \$MC_LIFTFAST_DIST)の設定値は、DILF をプログラム指令すると、パートプログラムで変更できます。 注: マシンデータ設定値は常に、NC RESET の後に有効になります。	
LFTXT LFWP:	後退方向が、G 命令 LFTXT および LFWP により、ALF と組み合わせて制御されます。	
	LFTXT:	後退移動が実行される平面は、軌跡タンジェントと工具方向(初期設定)から計算されます。
	LFWP:	後退移動が実行される平面は、現在有効な作業平面です。
ALF= :	この方向は、後退移動の平面で ALF により、ステップ値を使用してプログラム指令します。 LFTXT の場合は、ALF=1 で、工具方向の後退が定義されます。 LFWP の場合、加工平面内の方向が、次の割り当てから導出されます。 • G17 (X/Y 平面) ALF=1 ; X 方向の後退 ALF=3 ; Y 方向の後退 • G18 (Z/X 平面) ALF=1 ; Z 方向の後退 ALF=3 ; X 方向の後退 • G19 (Y/Z 平面) ALF=1 ; Y 方向の後退 ALF=3 ; Z 方向の後退 ALF によるプログラミングオプションについての追加情報は、章「輪郭からの高速リトラクトの移動方向 (ページ 623)」を参照してください。	
LFPOS:	POLFMASK または POLFMLIN を使用して宣言された軸を、POLF でプログラム指令されたアブソリュート軸位置へ後退します。	

POLFMASK:	個別にアブソリュート位置へ後退する軸 (<軸 1 名称>, <軸 1 名称>, ...) の解放	
POLFMLIN:	アブソリュート位置へ直線補間で後退する軸の開放 注: すべての関連軸のダイナミック応答によっては、戻し位置に達する前に直線補間を確立できない場合があります。	
POLF[]:	インデックスでジオメトリ軸または機械軸のアブソリュート後退位置を定義します。	
	効果:	モーダル
	=<値>:	ジオメトリ軸の場合は、割り当てた値はワーク座標系の位置として解釈されます。機械軸の場合は、割り当てた値は機械座標系の位置として解釈されます。 割り当てた値は、インクレメンタル指令としてプログラム指令することもできます。 =IC<値>
<軸識別子>:	ジオメトリ軸または機械軸の識別子です。	

注記

LFON または LFOF は常にプログラム指令できますが、その評価は、ねじ切り (G33) 中のみおこなわれます。

注記

POLFMASK/POLFMLIN を伴う POLF は、ねじ切り以外の用途にも使用されます。

例**例 1: ねじ切り時の高速リトラクトの設定**

プログラムコード	コメント
N55 M3 S500 G90 G18	; 有効な加工平面
...	; 開始位置へアプローチします。
N65 MSG ("thread cutting")	; 工具の切り込み。
MM_THREAD:	
N67 \$AC_LIFTFAST=0	; ねじの開始前にリセットします。
N68 G0 Z5	
N68 X10	

3.9 動作命令

プログラムコード	コメント
N70 G33 Z30 K5 LFON DILF=10 LFWP ALF=7	; ねじ切り時の高速リトラクトを設定します。 後退距離 = 10 mm イニシャル点: Z/X (G18 により) 後退方向: -X (ALF=3 の場合: 後退方向+X)
N71 G33 Z55 X15	
N72 G1	; ねじ切りを選択解除します。
N69 IF \$AC_LIFTFAST GOTOB MM_THREAD	; ねじ切りが中断されている場合。
N90 MSG("")	
...	
N70 M30	

例 2: タッピング前に高速リトラクトを解除します。

プログラムコード	コメント
N55 M3 S500 G90 G0 X0 Z0	
...	
N87 MSG ("tapping")	
N88 LFOF	; タッピング前に高速リトラクトを解除します。
N89 CYCLE...	; G33 によるタッピングサイクル。
N90 MSG("")	
...	
N99 M30	

例 3: アブソリュート後退位置への高速リトラクト

停止の場合、X の軌跡補間がマスクされます。そして、その代わりに POLF[X] 位置へ最大速度で移動します。その他の軸の移動は、プログラム指令輪郭、またはねじリードと主軸速度により特定されて続行されます。

プログラムコード	コメント
N10 G0 G90 X200 Z0 S200 M3	
N20 G0 G90 X170	
N22 POLF[X]=210 LFPOS	
N23 POLFMASK(X)	; 軸 X の高速リトラクトを有効に (設定) します。
N25 G33 X100 I10 LFON	
N30 X135 Z-45 K10	
N40 X155 Z-128 K10	
N50 X145 Z-168 K10	
N55 X210 I10	
N60 G0 Z0 LFOF	
N70 POLFMASK()	; すべての軸の戻しを解除します。

プログラムコード	コメント
M30	

3.9.10.5 凸型ねじ(G335、G336)

G 命令 G335 と G336 を使用して、凸型ねじ(=円筒形状とは異なる)を旋削できます。これは、自重により機械でたるんでいる極めて大きいコンポーネントの加工に使用されます。このような場合、近軸のねじでは、ねじがコンポーネントの中央で小さくなりすぎます。これを凸型ねじで補正できます。

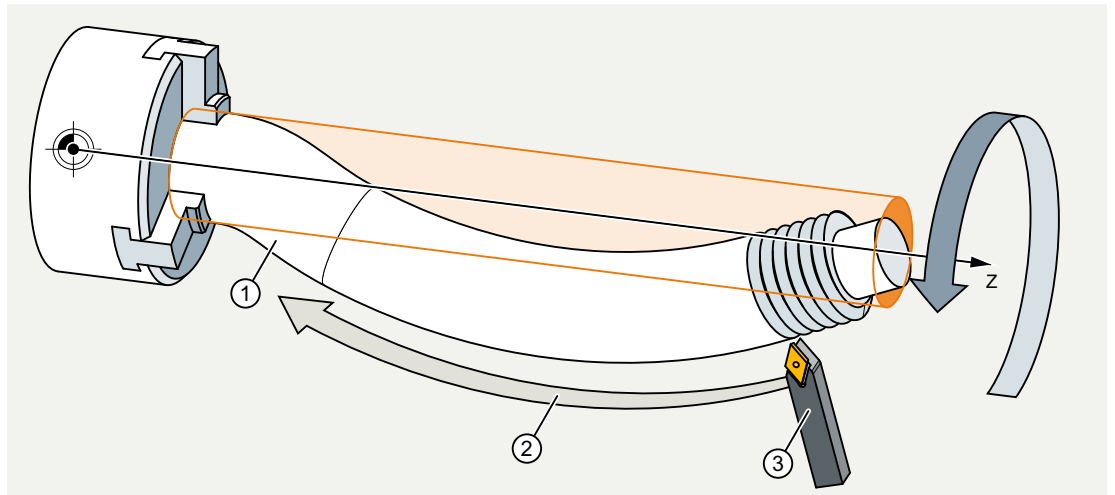


図 3-6 凸型ねじの旋削

プログラミング

凸型ねじの旋削は、G335 または G336 でプログラム指令します。

G335:	右回りの円弧工具軌跡での凸型ねじの旋削
G336:	左回りの円弧工具軌跡での凸型ねじの旋削

プログラミングは、最初はストレートねじと同様に、パラメータ I、J および K によって軸のブロック終点とピッチを指定します(「固定リードのねじ切り(G33、SF)(ページ 248)」を参照してください)。

円弧も指定します。G2/G3 と同様に、これは中心点、半径、開口角度または中間点の指定によってプログラム指令できます(「円弧補間(ページ 212)」を参照してください)。中心点の指定によって凸型ねじをプログラム指令する場合は、以下のことを考慮してください。I、

3.9 動作命令

J および K はねじ切りのピッチに使用されるため、中心点の指定での円弧パラメータは、IR=...、JR=...および KR=...を使用してプログラム指令してください。

IR=...:	円弧中心点の X 方向の直交座標
JR=...:	円弧中心点の Y 方向の直交座標
KR=...:	円弧中心点の Z 方向の直交座標

注記

IR、JR、および KR は、凸型ねじ用の補間パラメータの名称の初期設定で、マシンデータ (MD10651 \$MN_IPO_PARAM_THREAD_NAME_TAB) で設定できます。
初期設定が工作機械メーカーの指定と異なっていないか確認してください。

オプションで、起点オフセット SF も指定できます(「固定リードのねじ切り (G33、SF) (ページ 248)」を参照してください)。

構文

凸型ねじのプログラミングの構文は、一般的には以下のような形式になります。
G335/G336 <軸目標点座標> <ピッチ> <円弧> [<起点オフセット>]

例

例 1: 終点と中心点の指定による右回りの凸型ねじ

プログラムコード	コメント
N5 G0 G18 X50 Z50	; 起点にアプローチします。
N10 G335 Z100 K=3.5 KR=25 IR=-20 SF=90	; 右回り方向に凸型ねじを旋削します。

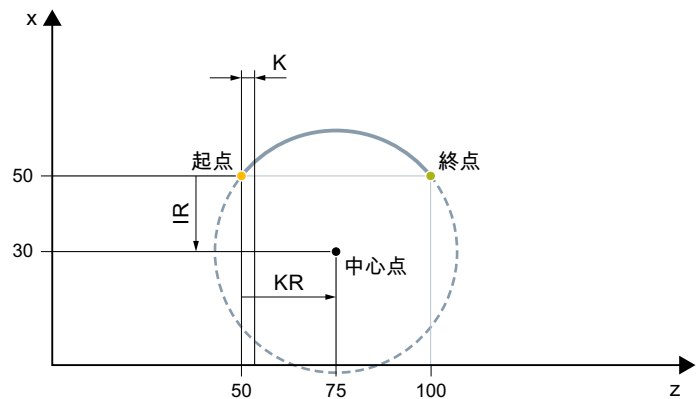


図 3-7 終点と中心点の指定による右回りの凸型ねじ

例 2 : 終点と中心点の指定による左回りの凸型ねじ

プログラムコード	コメント
N5 G0 G18 X50 Z50	; 起点にアプローチします。
N10 G336 Z100 K=3.5 KR=25 IR=20 SF=90	; 凸型ねじを左回り方向に旋削します。

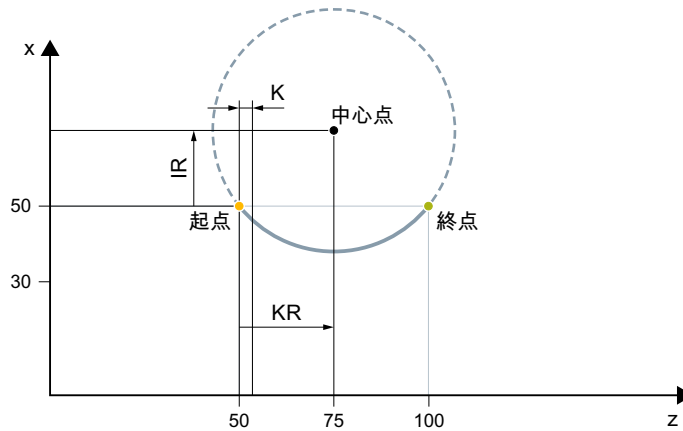


図 3-8 終点と中心点の指定による左回りの凸型ねじ

例 3 : 終点と半径の指定による右回りの凸型ねじ

プログラムコード	コメント
N5 G0 G18 X50 Z50	
N10 G335 Z100 K=3.5 CR=32 SF=90	

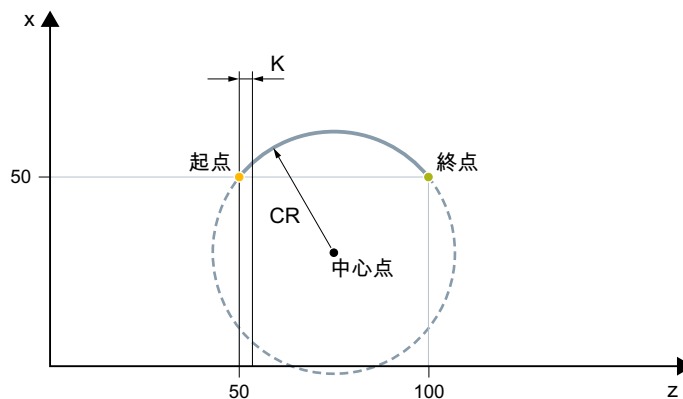


図 3-9 終点と半径の指定による右回りの凸型ねじ

例 4 : 終点と開口角度の指定による右回りの凸型ねじ

プログラムコード

```

N5 G0 G18 X50 Z50
N10 G335 Z100 K=3.5 AR=102.75 SF=90

```

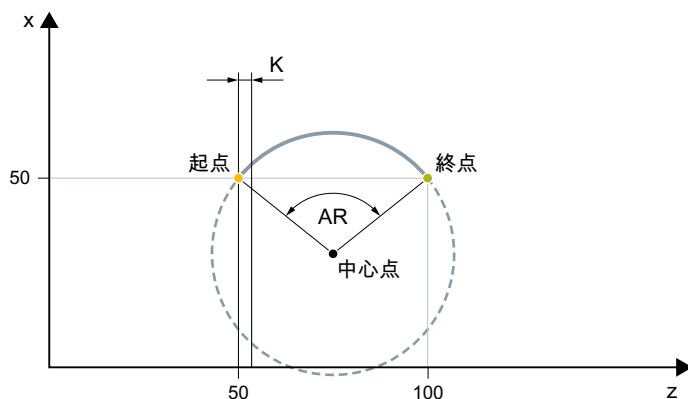


図 3-10 終点と開口角度の指定による右回りの凸型ねじ

例 5 : 中心点と開口角度の指定による右回りの凸型ねじ

プログラムコード

```

N5 G0 G18 X50 Z50
N10 G335 K=3.5 KR=25 IR=-20 AR=102.75 SF=90

```

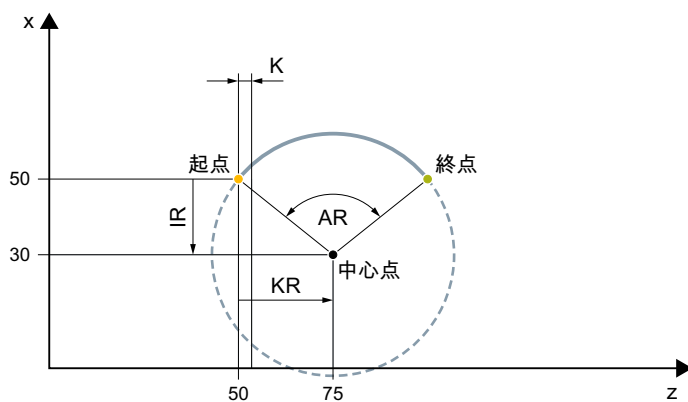


図 3-11 中心点と開口角度の指定による右回りの凸型ねじ

例 6 : 終点と中間点の指定による右回りの凸型ねじ

プログラムコード

```

N5 G0 G18 X50 Z50

```

プログラムコード

```
N10 G335 Z100 K=3.5 I1=60 K1=64
```

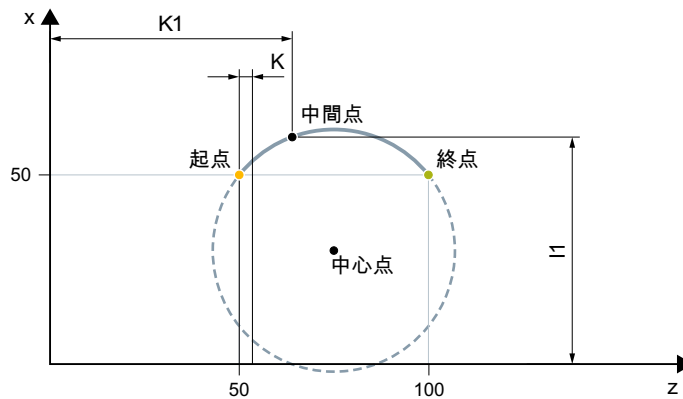
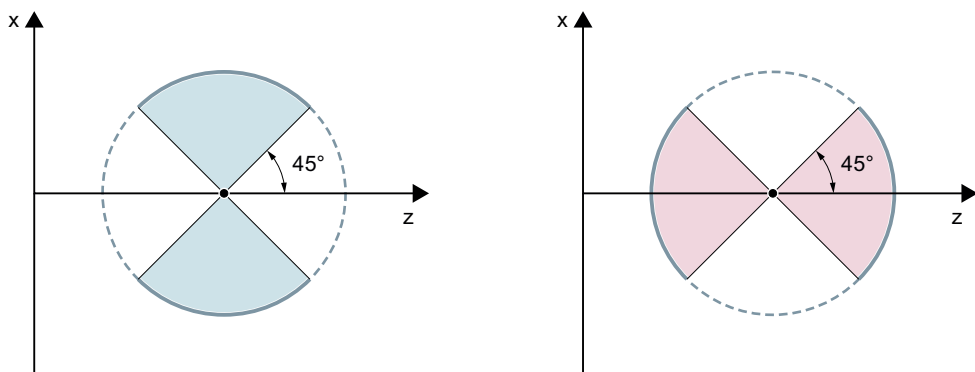


図 3-12 終点と中間点の指定による右回りの凸型ねじ

詳細情報

使用可能な円弧領域

G335/G336 でプログラム指令する円弧は、指定されたねじのメイン軸(I、J または K)が、円弧全体に対する円弧でメイン軸を共有する領域にあるようにしてください。



許容される Z 軸の領域(K でプログラム指令されるピッチ) 許容される X 軸の領域(I でプログラム指令されるピッチ)

次の図で示すようなねじのメイン軸の変更は許容されません。

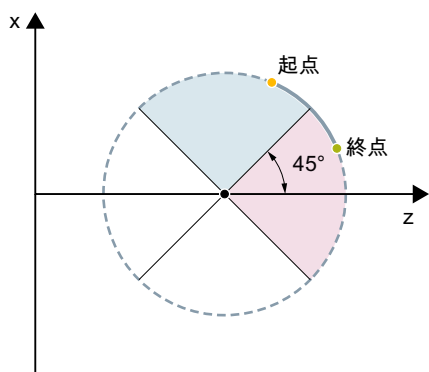


図 3-13 凸型ねじ:許容されない領域

フレーム

G335 と G336 は動作中のフレームでも使用できます。ただし、許容される円弧領域が基本座標系(BCS)で維持されることを確認してください。

円弧軌跡のプログラミングに関する必要条件

G2/G3 での円弧軌跡のプログラミングについて記載した必要条件是、G335/G336 での円弧軌跡のプログラミングにも適用されます(「円弧補間 (ページ 212)」を参照してください)。

3.9.11 フローティングチャックのないタッピング

3.9.11.1 フローティングチャックおよび後退動作を使用しないタッピング(G331、G332)

フローティングチャックのないタッピングでは、G331 命令および G332 命令を使用して次の移動動作を実行できます。

- G331:ねじ終点までのタッピング方向でのタッピング
- G332:主軸回転方向の自動反転を伴うタッピングブロック G331 への後退動作

構文

```
G331 <軸> <ねじピッチ>
G331 <軸> <ねじピッチ> S...
G332 <軸> <ねじピッチ>
```


意味

G331:	タッピング タップ加工される穴は、軸の移動動作(穴あけ深さ)およびねじピッチによって定義されます。	
	有効性:	モーダル
G332:	タッピング時の後退動作 後退動作では、タッピング時(G331)と同じピッチであることが必要です。主軸の回転方向は、自動的に逆になります。	
	有効性:	モーダル
<軸>:	ねじ終点でのジオメトリ軸(X、Y、または Z)の移動距離/位置(たとえば、Z50 など)	
<ねじピッチ>:	ねじピッチ I (X)、J (Y)、または K (Z): • 正のピッチ: 右ねじ(例: K1.25) • 負のピッチ: 左ねじ(例: K-1.25)	
	値の範囲:	±0.001 ~ ±2000.00 mm/回転
S...:	主軸速度 主軸速度を指定しない場合、最後に有効だった主軸速度が使用されます。	

注記

2 番目のギヤ選択データブロック

主軸速度とモータトルクを効果的に適用させて、より短時間で加速するために、その他に、2 つの設定可能なしきい値(最大速度と最小速度)の切り替えのための 2 番目のギヤ選択データブロックを、軸マシンデータに設定しておくことができます。このデータは、1 番目のギヤ選択データブロックとは異なるデータで、1 番目の速度切り替えのしきい値には依存しません。工作機械メーカーの指定に従う必要があります。

詳しくは、『軸と主軸』機能マニュアルを参照してください。

例

- 例: G331/G332 のタッピング (ページ 270)
- 例: 現在のギヤ選択でプログラム指令穴あけ速度を出力 (ページ 270)
- 例: 2 番目のギヤ選択データブロックの用途 (ページ 271)
- 例: 速度はプログラムされておらず、ギヤボックス選択が監視されています (ページ 271)

3.9 動作命令

- 例:ギヤボックスの選択を変更できません、ギヤボックスの選択は監視されています (ページ 271)
- 例:SPOS を使用しないプログラミング (ページ 272)

3.9.11.2 例:G331/G332 のタッピング

プログラムコード	コメント
N10 SPOS[n]=0	; 主軸:位置制御モード ; 開始位置 0 度
N20 G0 X0 Y0 Z2	; 軸:開始位置にアプローチ
N30 G331 Z-50 K-4 S200	; Z のタッピング、 ; ピッチ K-4 が負 => ; 主軸の回転方向:CCW 回転、 ; 主軸速度 200 rpm
N40 G332 Z3 K-4	; Z 方向の後退動作、 ; ピッチ K-4 が負 => (左回り) ; 回転方向の自動逆転 => ; 右回りの主軸回転方向
N50 G1 F1000 X100 Y100 Z100 S300 M3	; 主軸運転中の主軸

3.9.11.3 例:現在のギヤ選択でプログラム指令穴あけ速度を出力

プログラムコード	コメント
N05 M40 S500	; プログラムされた主軸速度:500 rpm => ; ギヤボックス選択 1 (20~1028 rpm)
...	
N55 SPOS=0	; 主軸の位置決め
N60 G331 Z-10 K5 S800	; タッピング ; 主軸速度 800 rpm => ギヤボックス選択 1

プログラム指令主軸速度 S500 に対応するギヤ選択は M40 で、1 番目のギヤ選択データブロックに基づいて特定されます。プログラム指令穴あけ速度 S800 は、現在のギヤ選択で出力され、必要に応じて、ギヤ選択の最大速度に制限されます。SPOS 命令後のギヤ選択の自動変更はできません。ギヤ選択の自動変更をおこなうためには、主軸を速度制御モードにしてください。

注記

主軸速度 800 rpm でギヤボックス選択 2 を選択する場合は、最大速度と最小速度の切り替えのしきい値を、2 番目のギヤ選択データブロックの当該のマシンデータに設定してください(以下の例を参照してください)。

3.9.11.4 例:2 番目のギヤ選択データブロックの用途

2 番目のギヤ選択データブロックにある最大速度と最小速度の切り替えのしきい値は、動作中のメイン主軸の S 値のプログラミング時に、G331/G332 に対して使用されます。自動 M40 ギヤ選択変更を有効にしてください。上記の方法で特定されたギヤ選択は、動作中のギヤ選択と比較されます。両者が異なることが検出された場合、ギヤボックスの選択が変更されます。

プログラムコード	コメント
N05 M40 S500	; プログラムされた主軸速度:500 rpm
...	
N50 G331 S800	; メイン主軸:ギヤ選択 2 を選択します
N55 SPOS=0	; 主軸の位置決め
N60 G331 Z-10 K5	; タッピング
	; 2 番目のギヤボックス選択データブロック 2 からの主軸の加減速

3.9.11.5 例:速度はプログラムされておらず、ギヤボックス選択が監視されています

2 番目のギヤ選択データブロックの使用時に速度を G331 でプログラム指令していない場合は、最後のプログラム指令速度を使用して、ねじが加工されます。ギヤ選択は変更されません。ただし、この場合は監視がおこなわれ、最後のプログラム指令速度が、動作中のギヤ選択の設定速度レンジ(最大速度と最小速度のしきい値で定義されています)内であることが確認されます。それ以外の場合は、アラーム 16748 が出力されます。

プログラムコード	コメント
N05 M40 S800	; プログラムされた主軸速度:800 rpm
...	
N55 SPOS=0	; 主軸の位置決め
N60 G331 Z-10 K5	; タッピング
	; 主軸速度の監視、800 rpm
	; ギヤ選択 1 が有効です
	; ギヤボックス選択 2 は有効である必要があります => アラーム 16748

3.9.11.6 例:ギヤボックスの選択を変更できません、ギヤボックスの選択は監視されています

2 番目のギヤ選択データブロックの使用時に、G331 ブロックではジオメトリに加えて主軸速度をプログラム指令していますが、速度が、動作中のギヤ選択の設定速度レンジ(最大速度と最小速度のしきい値で定義されています)内でない場合でも、ギヤ選択を変更できません。これは、主軸と切り込み軸(複数軸の場合もあります)の軌跡移動が維持されないためです。

3.9 動作命令

上記の例のように、速度とギヤ選択が G331 ブロックで監視され、アラーム 16748 が、必要に応じて発生します。

プログラムコード	コメント
N05 M40 S500	; プログラムされた主軸速度:500 rpm => ; ギヤボックス選択 1
...	
N55 SPOS=0	; 主軸の位置決め
N60 G331 Z-10 K5 S800	; タッピング ; ギヤボックスの選択は変更できません、 ; 主軸速度の監視、800 rpm ; ギヤボックス選択データセット 1: ギヤボックス選択 2 ; は有効であることが必要です => アラーム 16748

3.9.11.7 例:SPOS を使用しないプログラミング

プログラムコード	コメント
N05 M40 S500	; プログラムされた主軸速度:500 rpm => ; ギヤボックス選択 1 (20～1028 rpm)
...	
N50 G331 S800	; メイン主軸: ギヤ選択 2 を選択します
N60 G331 Z-10 K5	; タッピング ; 2 番目のギヤボックス選択データブロック 2 からの主軸の加減速

ギヤ選択が変更された場合など、以前に処理されたパートプログラム区間により特定される現在位置から、主軸のねじ補間が開始されます。したがって、ねじの再加工ができない場合があります。

注記

複数の主軸による加工の場合は、ドリル主軸もメイン主軸であることが必要です。SETMS (主軸番号>) をプログラム指令して、ドリル主軸をメイン主軸として設定できます。

3.9.12 フローティングチャックを使用したタッピング

3.9.12.1 補正チェックおよび後退動作(G63)によるタッピング

フローティングチャックによるタッピングでは、G63 命令を使用して次の移動動作を実行できます。

- G63:ねじ終点までのタッピング方向でのタッピング
- G63:プログラムされた主軸反転回転方向の後退動作

注記

G63 ブロックの後、最後に有効だった補間タイプ G0、G1、G2 が有効になります。

構文

G63 <軸> <回転方向> <速度> <送り速度>

意味

G63:	フローティングチャックによるタッピング	
	効果:	ノンモーダル
<軸>:	ねじ終点でのジオメトリ軸(X、Y、または Z)の移動距離/位置(たとえば、Z50 など)	
<回転方向>:	主軸の回転方向: • M3:右回り、右ネジ • M4:左回り、左ねじ	
<速度>:	タッピング時の最大許容主軸速度、例: S100 など	
<送り速度>:	タッピング軸 F の送り速度、 $F = \text{主軸速度} * \text{ねじピッチ}$	

例

M5 ねじのタッピング:

- 標準に準拠した主軸ピッチ:0.8 mm/rev
- 主軸速度 S:200 rpm
- 送り速度 $F = 200 \text{ rpm} * 0.8 \text{ mm/rev} = 160 \text{ mm/min}$

3.9 動作命令

プログラムコード	コメント
N10 G1 X0 Y0 Z2 F1000 S200 M3	; 起点にアプローチします。 ; 右回りの主軸回転方向、200 rpm
N20 G63 Z-50 F160	; フローティングチャックによるタッピング ; 穴あけ深さ:絶対値 Z=50mm ; 送り速度:160 mm/min
N30 G63 Z3 M4	; 後退移動:絶対値 Z=3mm ; 回転方向の逆転 ; 左回りの主軸回転方向、200 rpm

3.9.13 面取り、丸み付け(CHF、CHR、RND、RNDM、FRC、FRCM)

有効な作業平面内の輪郭のコーナでは、丸み付けまたは面取りを実行できます。

最適な加工面品質にするために、面取り/丸み付けのための用途別送り速度をプログラム指令できます。送り速度をプログラム指令しない場合は、標準軌跡送り速度 F が適用されます。

「モーダル丸み付け」機能を使用すると、複数の輪郭のコーナを同じ方法で連続して丸み付けできます。

構文

輪郭のコーナの面取り:

```
G...X...Z...CHF=<値> FRC/FRCM=<値>
G...X...Z...
```

輪郭のコーナの丸み付け:

```
G...X...Z...RND=<値> FRC=<値>
G...X...Z...
```

モーダル丸み付け:

```
G...X...Z...RNDM=<値> FRCM=<値>
...
RNDM=0
```

注記

面取り/丸み付けのためのテクノロジー(送り速度、送り速度タイプ、M 命令など)は、マシンデータ MD20201 \$MC_CHFRND_MODE_MASK (面取り/丸み付け動作)のビット 0 の設定に応じて、前または次のブロックの指令が適用されます。推奨設定は、前のブロックからのテクノロジーの使用(ビット 0 = 1)です。

意味

CHF=... :	輪郭のコーナの面取り	
	<値>:	面取りの長さ(G70/G71 に対応する単位)
CHR=... :	輪郭のコーナの面取り	
	<値>:	最初の移動方向の面取りの幅(G70/G71 に対応する単位)
RND=... :	輪郭のコーナの丸み付け	
	<値>:	丸み付けの半径(G70/G71 に対応する単位)
RNDM=... :	モーダル丸み付け(複数の輪郭のコーナを同じ方法で連続して丸み付け)	
	<値>:	丸み付けの半径(G70/G71 に対応する単位)
	モーダル丸み付けは RNDM=0 で無効になります。	
FRC=... :	面取り/丸み付けのノンモーダル送り速度	
	<値>:	mm/min (G94 が有効な場合)、または mm/rev(G95 が有効な場合)単位の送り速度
FRCM=... :	面取り/丸み付けのモーダル送り速度	
	<値>:	mm/min (G94 が有効な場合)、または mm/rev(G95 が有効な場合)単位の送り速度
	FRCM=0 を設定すると、面取り/丸み付けのモーダル送り速度が解除され、F のプログラム指令送り速度が有効になります。	

注記

面取り/丸み付けが大きすぎる

面取り (CHF/CHR) または丸み付け (RND/RNDM) のためのプログラム指令値が、関連する輪郭要素に対して大きすぎる場合は、面取りまたは丸み付けが自動的に調整されます。

1. MD11411 \$MN_ENABLE_ALARM_MASK ビット 4 が設定されている場合、アラーム「面取りまたは丸み付けを縮小してください」が出力されます(キャンセルアラーム)。
2. 面取り/丸み付けは、輪郭のコーナに適合するまで縮小されます。その結果、少なくとも 1 つのブロックが移動しなくなります。このブロックで、必要な移動が停止されます。

注記

面取り/丸み付け不可

次の場合には、面取り/丸み付けは実行されません。

- 直線軌跡と円弧軌跡のいずれも対象平面で使用されない
- 対象平面外で移動が発生する
- 対象平面が変更された
- 移動に関する情報を含まない(命令の出力のみなど)ブロックの数が、マシンデータで指定された数を超えている

注記

FRC/FRCM

面取りを G0 で移動した場合は、FRC/FRCM は動作しません。この命令は、F 値に従って、エラーメッセージを表示せずにプログラム指令できます。

FRC は、面取り/丸み付けをブロックにプログラム指令している場合、または RNDM が有効な場合にのみ効果があります。

FRC は、実行中のブロックの F または FRCM 値に上書きします。

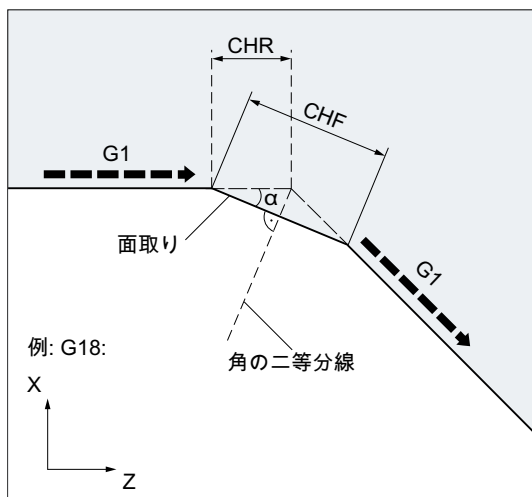
FRC のプログラム指令送り速度は、ゼロより大きい値にしてください。

FRCM=0 を設定すると、F でプログラム指令された面取り/丸み付けの送り速度が動作します。

FRCM をプログラム指令した場合は、FRCM 値を、G94 ⇄ G95 などの変更時に F と同様に再プログラム指令する必要があります。F のみが再プログラム指令され、変更前の送り速度タイプ FRCM > 0 の場合は、エラーメッセージが出力されます。

例

例 1: 直線と直線の間の面取り



- MD20201 ビット 0=1 (前のブロックからテクノロジーを使用)
- G71 が有効です。
- 移動方向(CHR)への面取りの幅は 2 mm、面取りの送り速度は 100 mm/min にします。

プログラミングは、次の 2 つの方法でおこなわれます。

- CHR によるプログラミング

プログラムコード

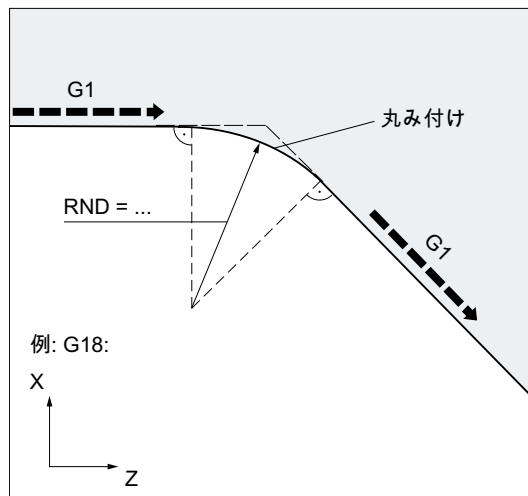
```
...
N30 G1 Z... CHR=2 FRC=100
N40 G1 X...
...
```

- CHF によるプログラミング

プログラムコード

```
...
N30 G1 Z... CHF=2(cosα*2) FRC=100
N40 G1 X...
...
```

例 2:2 つの直線間の丸み付け



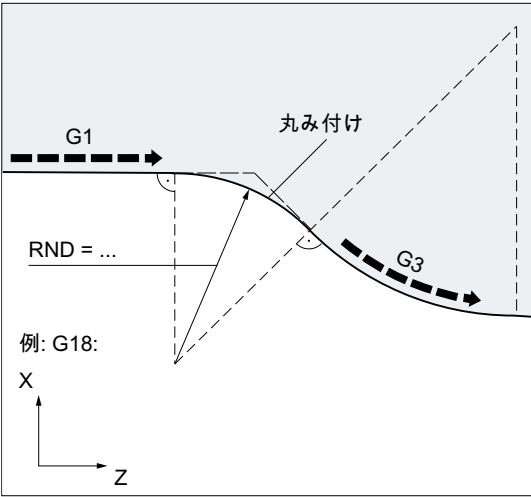
- MD20201 ビット 0=1 (前のブロックからテクノロジーを使用)
- G71 が有効です。
- 丸み付けの半径は 2 mm、丸み付けの送り速度は 50 mm/min にします。

プログラムコード

```
...
N30 G1 Z... RND=2 FRC=50
N40 G1 X...
...
```

例 3:直線と円弧の間の丸み付け

RND 機能を使用すると、直線軌跡と円弧軌跡の間に、接線方向の接続による円弧の輪郭要素をあらゆる組み合わせで挿入できます。



- MD20201 ビット 0 = 1 (前のブロックからテクノロジーを使用)
- G71 が有効です。
- 丸み付けの半径は 2 mm、丸み付けの送り速度は 50 mm/min にします。

プログラムコード

```
...
N30 G1 Z... RND=2 FRC=50
N40 G3 X... Z... I... K...
...
```

例 4:ワークの鋭利な端面をバリ取りするモーダル丸み付け

プログラムコード	コメント
...	
N30 G1 X... Z... RNDM=2 FRCM=50	; モーダル丸み付けを有効にします。 丸み付けの半径:2 mm 丸み付けの送り速度:50 mm/min
N40 ...	
N120 RNDM=0	; モーダル丸み付けを解除します。
...	

例 5:次のブロックのテクノロジー、または前のブロックからテクノロジーを適用

- MD20201 ビット 0 = 0:次のブロックのテクノロジーを使用(初期設定)

プログラムコード	コメント
N10 G0 X0 Y0 G17 F100 G94	
N20 G1 X10 CHF=2	; F=100 mm/min による面取り N20-N30
N30 Y10 CHF=4	; FRC=200 mm/min による面取り N30-N40
N40 X20 CHF=3 FRC=200	; FRCM=50 mm/min による面取り N40-N60
N50 RNDM=2 FRCM=50	
N60 Y20	; FRCM=50 mm/min によるモーダル丸み付け N60-N70
N70 X30	; FRCM=50 mm/min によるモーダル丸み付け N70-N80
N80 Y30 CHF=3 FRC=100	; FRC=100 mm/min による面取り N80-N90
N90 X40	; F=100 mm/min によるモーダル丸み付け N90-N100 (FRCM の選択解除)
N100 Y40 FRCM=0	; G95 FRC=1 mm/ rev によるモーダル丸み付け N100-N120
N110 S1000 M3	
N120 X50 G95 F3 FRC=1	
...	
M02	

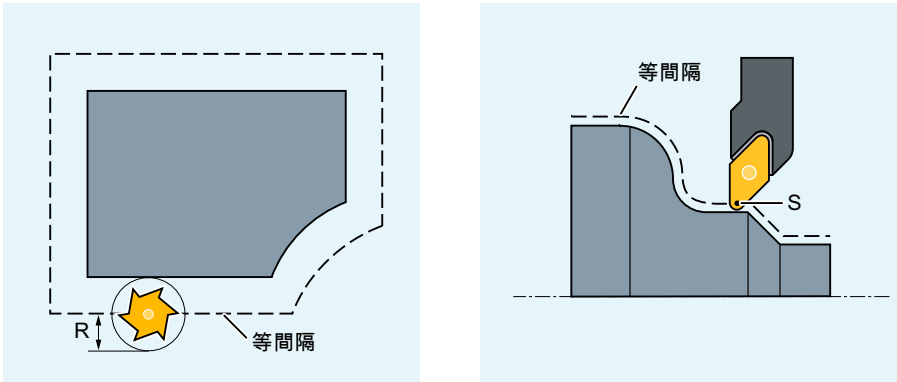
- MD20201 ビット 0 = 1:前のブロックからテクノロジーを使用(推奨設定)

プログラムコード	コメント
N10 G0 X0 Y0 G17 F100 G94	
N20 G1 X10 CHF=2	; F=100 mm/min による面取り N20-N30
N30 Y10 CHF=4 FRC=120	; FRC=120 mm/min による面取り N30-N40
N40 X20 CHF=3 FRC=200	; FRC=200 mm/min による面取り N40-N60
N50 RNDM=2 FRCM=50	
N60 Y20	; FRCM=50 mm/min によるモーダル丸み付け N60-N70
N70 X30	; FRCM=50 mm/min によるモーダル丸み付け N70-N80
N80 Y30 CHF=3 FRC=100	; FRC=100 mm/min による面取り N80-N90
N90 X40	; FRCM=50 mm/min によるモーダル丸み付け N90-N100
N100 Y40 FRCM=0	; F=100 mm/min によるモーダル丸み付け N100-N120
N110 S1000 M3	
N120 X50 CHF=4 G95 F3 FRC=1	; G95 FRC=1 mm/ rev による面取り N120-N130
N130 Y50	; F=3 mm/rev によるモーダル丸み付け N130-N140
N140 X60	
...	
M02	

3.10 工具径補正

3.10.1 工具径補正(G40、G41、G42、OFFN)

工具径補正(TRC)が有効なときは、制御装置が自動的に、さまざまな工具の等距離工具軌跡を計算します。



構文

G0/G1 X...Y... Z...G41/G42 [OFFN=<値>]	
...	
G40 X...Y... Z...	

意味

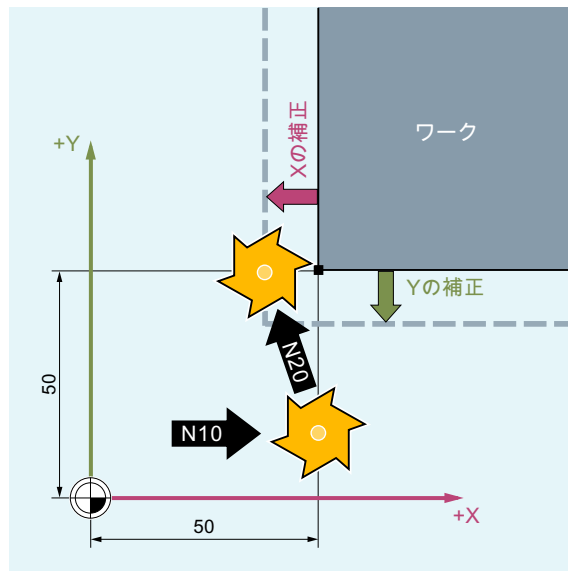
G41:	輪郭の加工方向の左側で TRC を有効にします。
G42:	輪郭の加工方向の右側で TRC を有効にします。
OFFN=<値>:	プログラム指令輪郭の仕上げ代(通常の輪郭オフセット) (任意選択) (荒仕上げのための等距離軌跡を生成するためなど)。
G40:	TRC を解除します。

注記

G40/G41/G42、G0、またはG1を含むNCブロックが有効であり、選択した作業平面で少なくとも1つの軸が指定されている必要があります。

このGコードを適用時に軸が1つだけ指定されている場合は、2番目の軸の最終位置が自動的に追加され、**両軸**で移動します。

この2つの軸は、チャンネルのジオメトリ軸として有効にしてください。これは、GEOAXプログラミングを使用して有効にできます。

例**例 1: フライス加工**

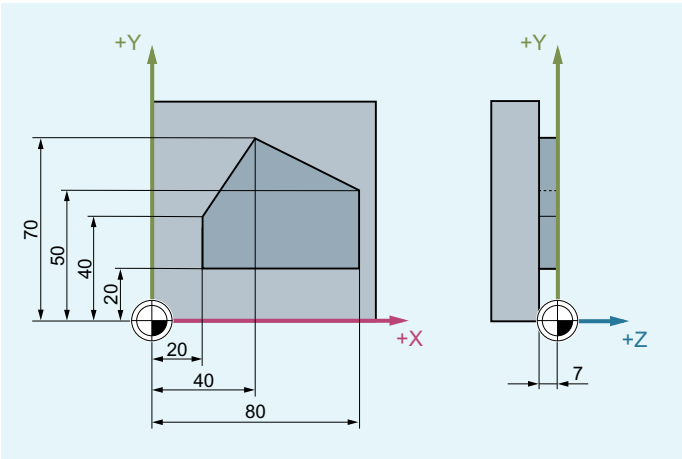
プログラムコード	コメント
N10 G0 X50 T1 D1	; 工具長補正のみを有効にします。径補正なしで X50 へアプローチします。
N20 G1 G41 Y50 F200	; 径補正が有効になり、点 X50/Y50 の点へ径補正有りでアプローチします。
N30 Y100	
...	

3.10 工具径補正

例 2:フライス加工を例とした「通常の」手順

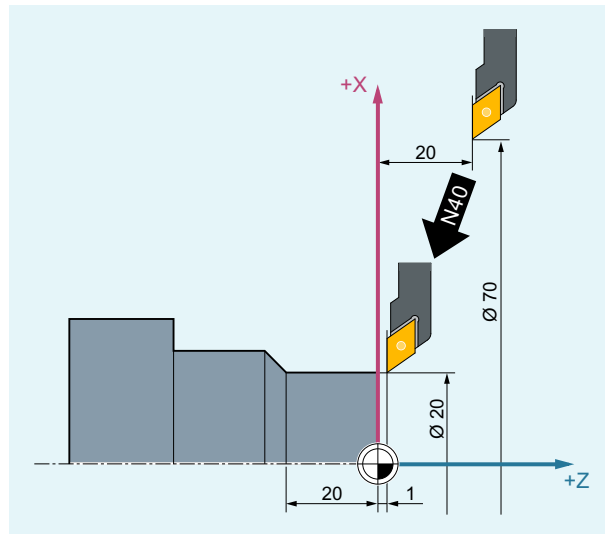
「通常の」手順:

- 1. 工具を呼び出します。
- 2. 工具を交換します。
- 3. 作業平面と工具径補正を有効にします。



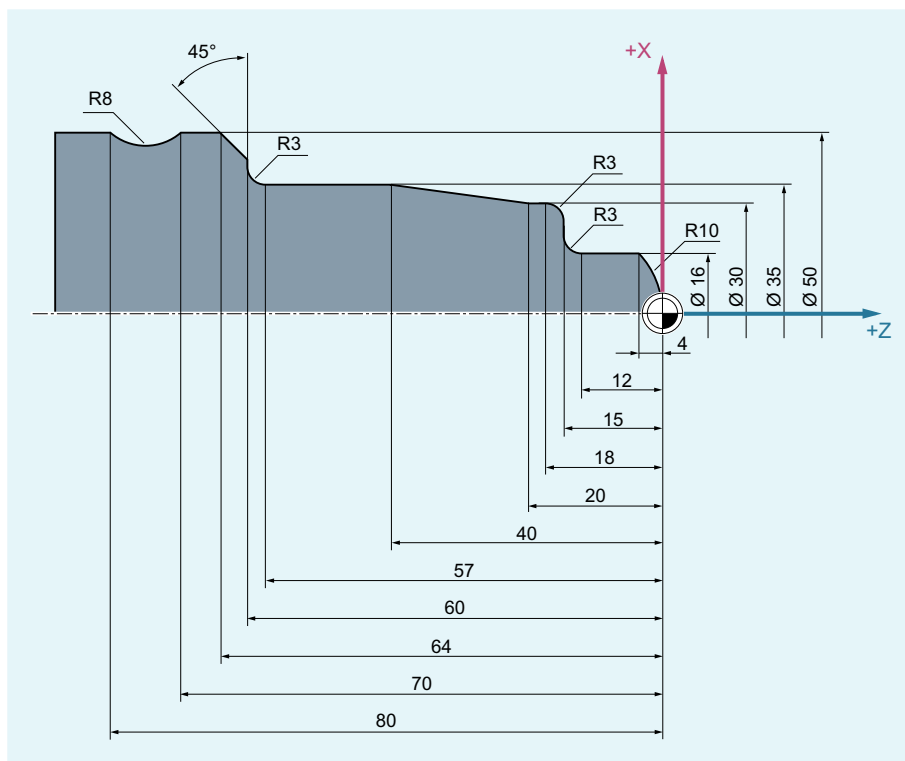
プログラムコード	コメント
N10 G0 Z100	; 工具交換のための後退
N20 G17 T1 M6	; 工具交換
N30 G0 X0 Y0 Z1 M3 S300 D1	; 工具オフセット値を呼び出し、工具長補正を選択します。
N40 Z-7 F500	; 工具を送ります。
N50 G41 X20 Y20	; 工具径補正を有効にして、工具は輪郭の左側を加工します。
N60 Y40	; 輪郭をフライス加工します。
N70 X40 Y70	
N80 X80 Y50	
N90 Y20	
N100 X20	
N110 G40 G0 Z100 M30	; 工具の後退、プログラム終了。

例 3: 旋削



プログラムコード	コメント
...	
N20 T1 D1	; 工具長補正のみを有効にします。
N30 G0 X70 Z20	; 径補正なしで X70 Z20 へアプローチします。
N40 G42 X20 Z1	; 径補正が有効になり、径補正ありで X20/Z1 の点へアプローチします。
N50 G1 Z-20 F0.2	
...	

例 4: 旋削



プログラムコード

コメント

```

N5 G0 G53 X280 Z380 D0
N10 TRANS X0 Z250
N15 LIMS=4000
N20 G96 S250 M3
N25 G90 T1 D1 M8
N30 G0 G42 X-1.5 Z1
N35 G1 X0 Z0 F0.25
N40 G3 X16 Z-4 I0 K-10
N45 G1 Z-12
N50 G2 X22 Z-15 CR=3
N55 G1 X24
N60 G3 X30 Z-18 I0 K-3
N65 G1 Z-20
N70 X35 Z-40
N75 Z-57
N80 G2 X41 Z-60 CR=3
N85 G1 X46
N90 X52 Z-63
N95 G0 G40 G97 X100 Z50 M9

```

```

; 起点。
; ゼロオフセット。
; 速度制限 (G96)。
; 周速一定制御を選択します。
; 工具とオフセットを選択します。
; 工具にノーズ R 補正を設定します。

; 半径 10 の円弧を旋削します。

; 半径 3 の円弧を旋削します。

; 半径 3 の円弧を旋削します。

; 半径 3 の円弧を旋削します。

; ノーズ R 補正を選択解除し、工具交換ロケーションへ
; アプローチします。

```


プログラムコード	コメント
N100 T2 D2	; 工具を呼び出してオフセットを選択します。
N105 G96 S210 M3	; 周速一定制御を選択します。
N110 G0 G42 X50 Z-60 M8	; 工具にノーズ R 補正を設定します。
N115 G1 Z-70 F0.12	; 直径 50 で旋削します。
N120 G2 X50 Z-80 I6.245 K-5	; 半径 8 の円弧を旋削します。
N125 G0 G40 X100 Z50 M9	; 工具を後退させてノーズ R 補正を選択解除します。
N130 G0 G53 X280 Z380 D0 M5	; 工具交換ロケーションへアプローチします。
N135 M30	; プログラム終了。

詳細情報

工具軌跡の計算

輪郭には、工具軌跡を計算するために次の情報が必要です。

- 工具番号(T...)、刃先番号(D...)
- 加工方向(G41/G42)
- 作業平面(G17/G18/G19)

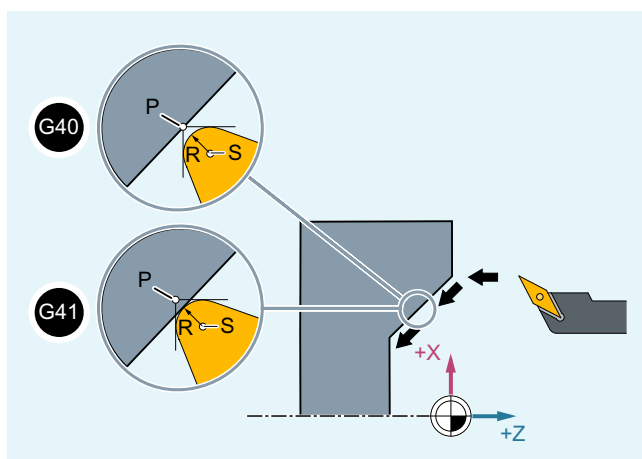
工具番号(T...)、刃先番号(D...)

工具軌跡とワーク輪郭の間の距離は、フライス工具半径、または刃先半径と刃先位置の指定から計算します。

加工方向(G41/G42)

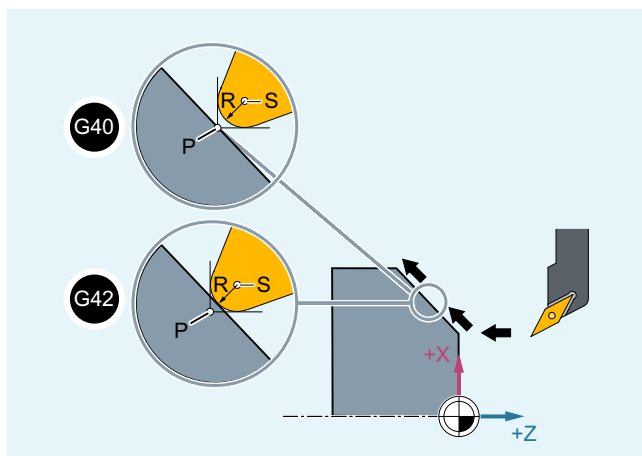
この情報から、コントローラは、工具軌跡をシフトする方向を検出します。

3.10 工具径補正



P 工具先端
S 刃先の中心点
R 刃先半径

図 3-14 輪郭(G41)の左側の加工方向に旋削するときの工具径補正



P 工具先端
S 刃先の中心点
R 刃先半径

図 3-15 輪郭(G42)の右側の加工方向に旋削するときの工具径補正

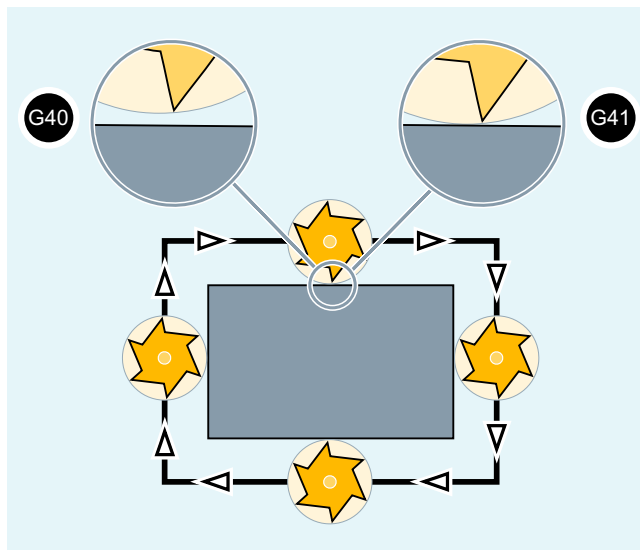


図 3-16 輪郭(G41)の左側の加工方向にフライス削りするときの工具径補正

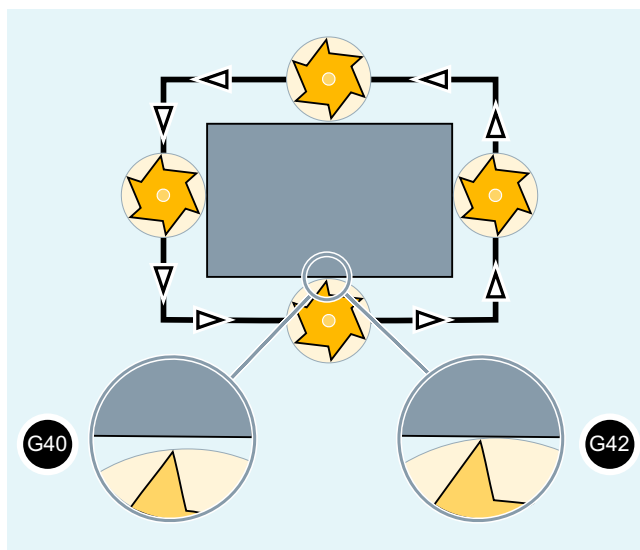


図 3-17 輪郭(G42)の右側の加工方向にフライス削りするときの工具径補正

注記

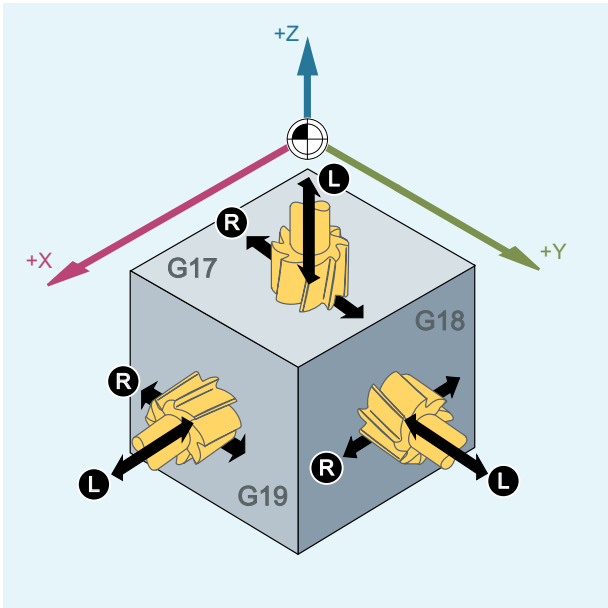
負の補正値は、オフセット方向の変更と同じ意味です(G41 ⇔ G42)。

作業平面(G17/G18/G19)

この情報から、コントローラは作業平面を検出し、軸の補正方向も検出します。

例:フライス工具

3.10 工具径補正



L 長さ
R 半径

プログラムコード	コメント
...	
N10 G17 G41 ...	; 工具径補正が X/Y 平面でおこなわれ、工具長補正が Z 方向におこなわれます。
...	

注記

2 軸の機械ではノーズ R 補正は「実」平面でのみ可能です。通常は G18 です。

工具長補正

選択した工具の直径軸に割り当てられた摩耗パラメータは、マシンデータを使用して直径値として定義できます。この割り当ては、以降の平面の変更時に自動的に変更されません。変更するためには、平面を変更した後に再度、工具を選択してください。

閉じた輪郭による工具径補正動作

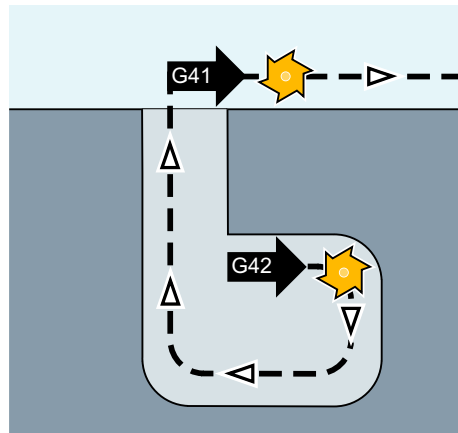
閉じた輪郭による工具径補正動作は設定日を使用して設定できます。

SD42496 \$SC_CUTCOM_CLSD_CONT

値	意味
FALSE	2つの連続した円弧ブロック、または1つの円弧ブロックに続く1つの直線ブロックから成る(ほとんど)閉じた輪郭の補正時に、内側に2つの交点がある場合は、1番目の輪郭部分のブロック終点に近い方の交点が、標準処理に従って選択されます。 1番目のブロックの起点と2番目のブロックの終点の間の距離が有効な補正半径の10%より短く、1000軌跡ステップ単位以下(小数点以下3桁で1mmに対応)である場合に、輪郭は(ほとんど)閉じていると見なされます。
TRUE	上記と同様の事例では、最もブロック開始点に近い1番目の輪郭部分に位置する交点を選択されます。

補正方向の変更(G41 ⇄ G42)

補正方向の変更(G41 ⇄ G42)は、中間に G40 を使用せずにプログラム指令できます。

**作業平面の変更**

G41/G42 が有効なとき、作業平面(G17/G18/G19)を変更できません。

工具オフセットデータブロックの変更(D...)

工具オフセットデータブロックは、補正モードで変更できます。

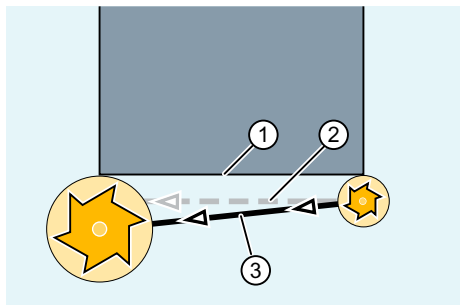
変更した工具半径は、新しい D 番号の入ったブロックから有効になります。

注記

半径の変更または補正移動はブロック全体にわたっておこなわれ、それは、プログラム指令終点で、新しい等距離位置に到達するだけです。

3.10 工具径補正

直線移動の場合は、工具が、起点と終点の間で斜めの軌跡に沿って移動します。



- ① プログラム指令輪郭
- ② 有効なブロックで工具オフセットデータブロックを変更しない場合の工具軌跡
- ③ 有効なブロックの工具オフセットデータブロックの変更時の工具軌跡と結果としての工具半径の変更

円弧補間では、らせん移動がおこなわれます。

工具半径の変更

変更は、システム変数の使用などにより、実行できます。その順序は、工具オフセットデータブロックを変更するときと同じです(D...)。

注記

変更した値は、次回の T または D のプログラム指令時にのみ有効になります。変更は、次のブロックまで適用されません。

補正モード

補正モードは、補正平面の移動命令、または位置データを含まない特定の数の連続ブロック、または M 機能によってのみ中断できます。

注記

連続割り込みブロックまたは M 命令の最大数は、マシンデータで設定できます(工作機械メーカーの情報を参照してください)。

注記

軌跡距離がゼロのブロックも、中断の条件として数えられます。

下記も参照

輪郭へのアプローチと後退(NORM、KONT、KONTC、KONTT) (ページ 291)

3.10.2 輪郭へのアプローチと後退(NORM、KONT、KONTC、KONTT)

必要条件

KONTC と KONTT 命令は、「多項式補間」オプションがコントローラで有効になっている場合にのみ使用できます。

機能

工具径補正が有効(G41/G42)な場合は、NORM、KONT、KONTC、または KONTT 命令を使用して、必要な輪郭形状または素材形状に対して、工具のアプローチ軌跡と後退軌跡を補正できます。

KONTC または KONTT を使用すると、3 軸すべての連続条件が確実に満たされます。したがって、補正平面に垂直な軌跡成分を同時にプログラミングできます。

構文

G41/G42 NORM/KONT/KONTC/KONTT X...Y...Z...	
...	
G40 X...Y...Z...	

意味

NORM:	直線へ、または直線からの直接アプローチ/後退を有効にします。 工具は、輪郭点に対して垂直な向きになります。
KONT:	プログラム指令コーナの動作 G450 または G451 に従って、始点/終点周りの移動によってアプローチ/後退を有効にします。
KONTC:	一定の曲率でアプローチ/後退を有効にします。
KONTT:	一定の接線でアプローチ/後退を有効にします。

注記

G1 ブロックのみが、KONTC と KONTT の最初のアプローチ/後退ブロックとして使用できます。コントローラはこれらのブロックを、当該のアプローチ/後退軌跡の多項式に置き換えます。

必要条件

KONTT と KONTC は、工具径補正の 3 次元タイプ(CUT3DC、CUT3DCC、CUT3DF)では使用できません。これらをプログラム指令すると、コントローラは、エラーメッセージを表示せずに、内部で NORM に切り替えます。

例

KONTC

円弧中心点から開始して、一周円へアプローチします。アプローチブロックのブロック終点の方向と曲率半径は、その次の円弧の値と同じです。切り込みは、アプローチ/後退ブロックの両方で同時に、Z 方向へおこなわれます。以下に、工具軌跡の垂直投影図を示します。

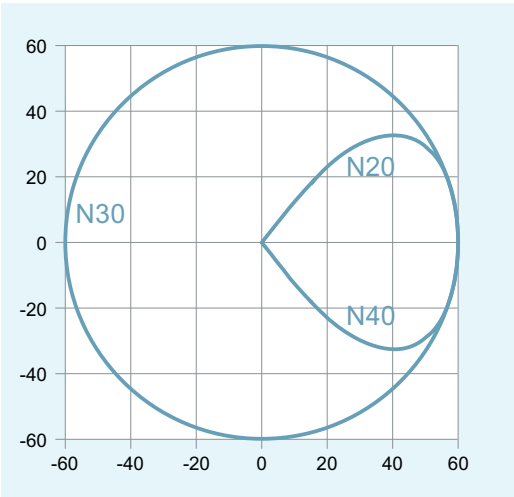


図 3-18 垂直投影図

関連する NC プログラム部分は次のとおりです。

プログラムコード	コメント
\$TC_DP1[1,1] = 121	; フライス工具
\$TC_DP6[1,1]=10	; 半径 10 mm
N10 G1 X0 Y0 Z60 G64 T1 D1 F10000	
N20 G41 KONTC X70 Y0 Z0	; アプローチ
N30 G2 I-70	; 一周円
N40 G40 G1 X0 Y0 Z60	; 後退
N50 M30	

曲率が一周円の円弧軌跡に合わせて補正されると同時に、Z60 から円弧平面の Z0 へ移動がおこなわれます。

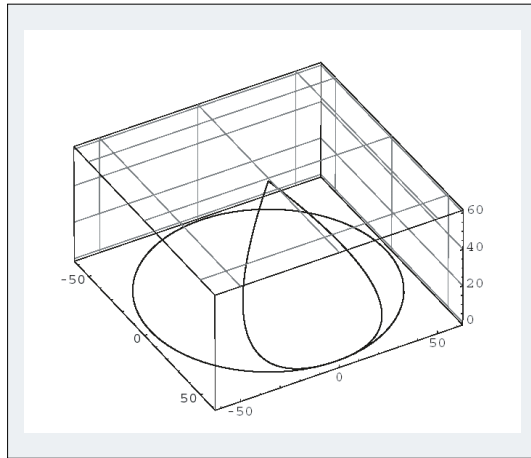


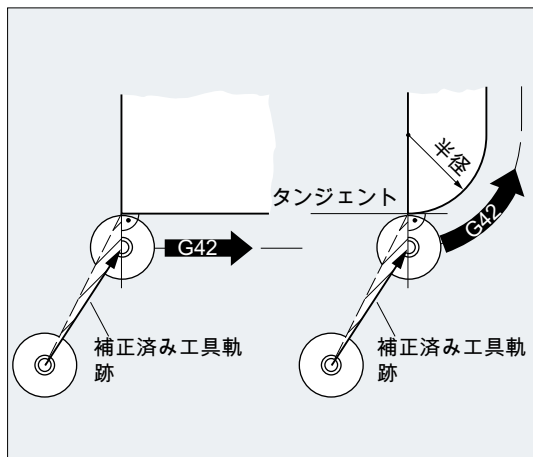
図 3-19 3次元表示

詳細情報

NORM によるアプローチ/後退

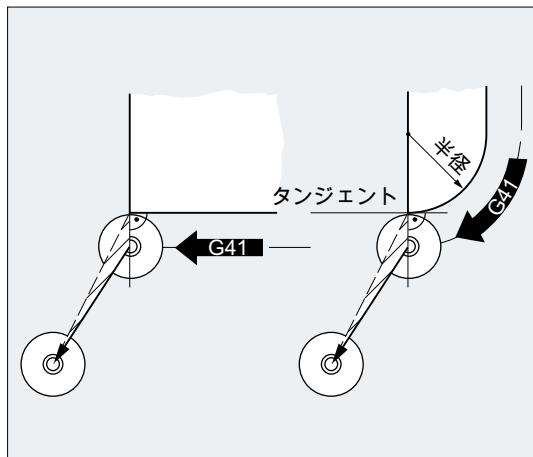
1. アプローチ:

NORM が有効な場合は、(移動のために設定アプローチ角度がプログラム指令された場合でも) 直接、補正された開始位置へ直線上を工具が移動して、起点で軌跡タンジェントに垂直に位置決めされます。

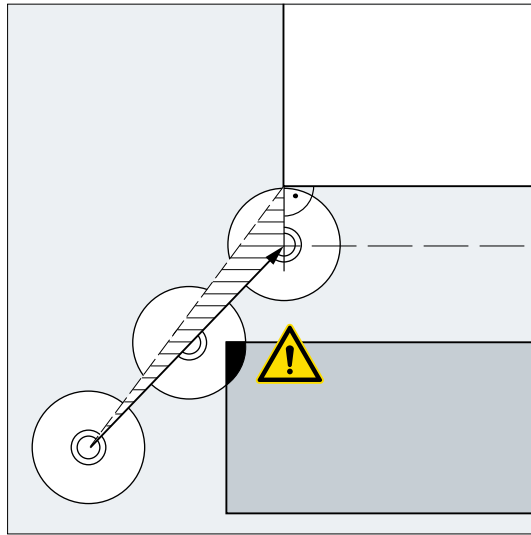


2. 後退:

工具は、(移動のために設定アプローチ角度がプログラム指令された場合でも)最後に補正された軌跡終点に対して垂直となり、次の径補正無効位置(工具交換位置など)へ直接、直線移動します。



アプローチ/後退角度を変更すると、干渉発生の可能性があります。



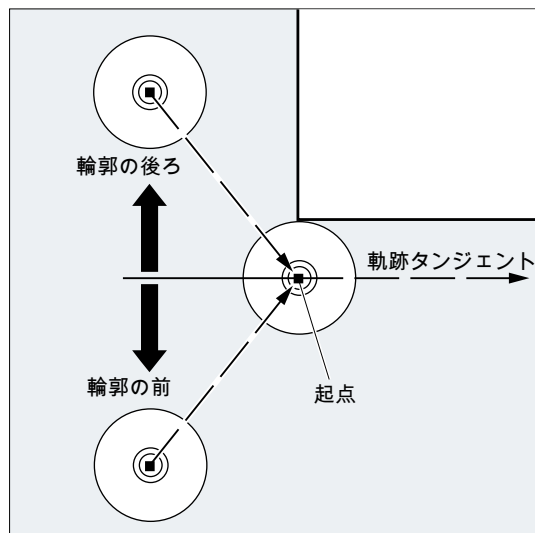
通知

干渉の可能性

アプローチ/後退角度の変更は、干渉が発生しないように、プログラミングのときに考慮してください。

KONT によるアプローチ/後退

アプローチの前に、工具を輪郭の**前**または**後ろ**に配置できます。起点での軌跡タンジェントは境界線として機能します。



3.10 工具径補正

これに従い、KONT によるアプローチ/後退に関して、次の 2 つの場合があります。

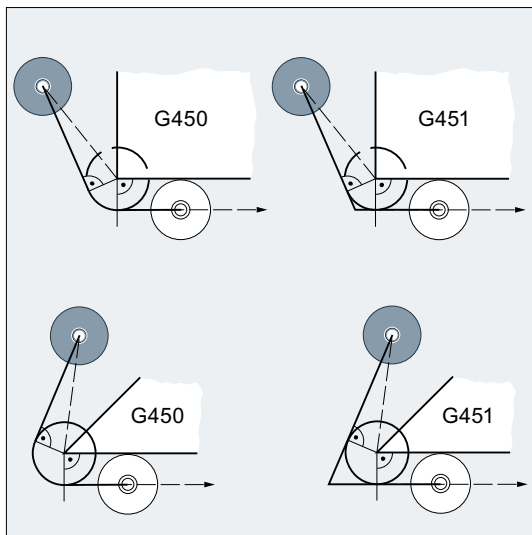
1. 工具が輪郭の前にあります。
→ アプローチ/後退の方法は **NORM** と同じです。

2. 工具が輪郭の後ろにあります。

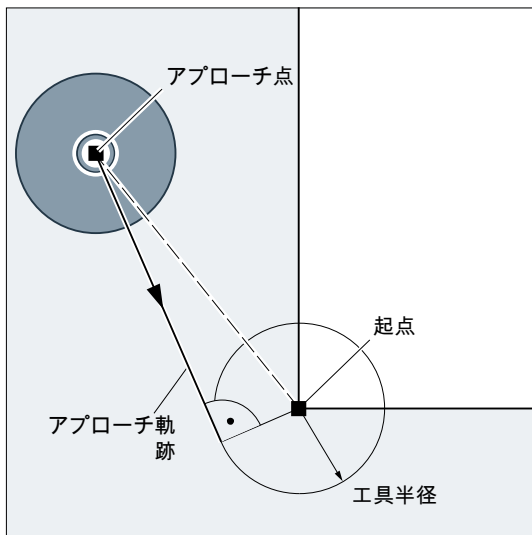
－ アプローチ:

工具は、円弧軌跡に沿って、またはプログラム指令したコーナ動作(G450/G451)に応じて、交点のまわりを等距離軌跡で、起点を中心に移動します。

G450/G451 命令は、実行中のブロックから次のブロックへの遷移に適用されます。



両方(G450/G451)の場合で、次のアプローチ軌跡が生成されます。



直線が、径補正無効のアプローチ点から引かれます。この線は、円弧半径= 工具半径で円に接しています。円弧中心点は起点です。

－ 後退:

後退にはアプローチと同じ規則が適用されますが、適用の順番が逆です。

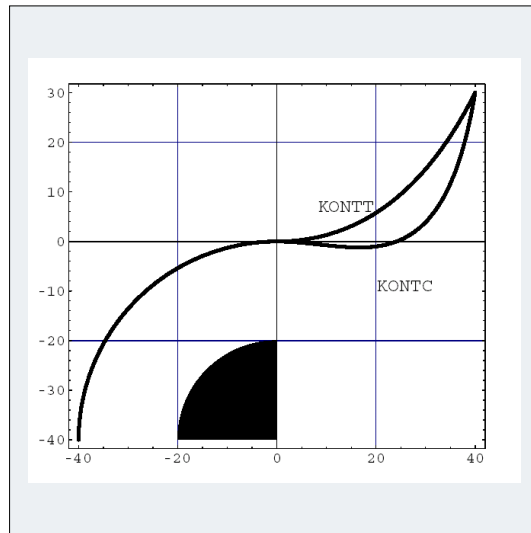
KONTC によるアプローチ/後退

輪郭点は、一定の曲率でアプローチ/後退がおこなわれます。輪郭点では加減速に不連続変化は発生しません。始点から輪郭点への軌跡は多項式として補間されます。

KONTT によるアプローチ/後退

輪郭点へ、一定接線によりアプローチ/後退がおこなわれます。輪郭点で加減速に不連続変化が発生する場合があります。始点から輪郭点への軌跡は多項式として補間されます。

KONTC と KONTT の違い

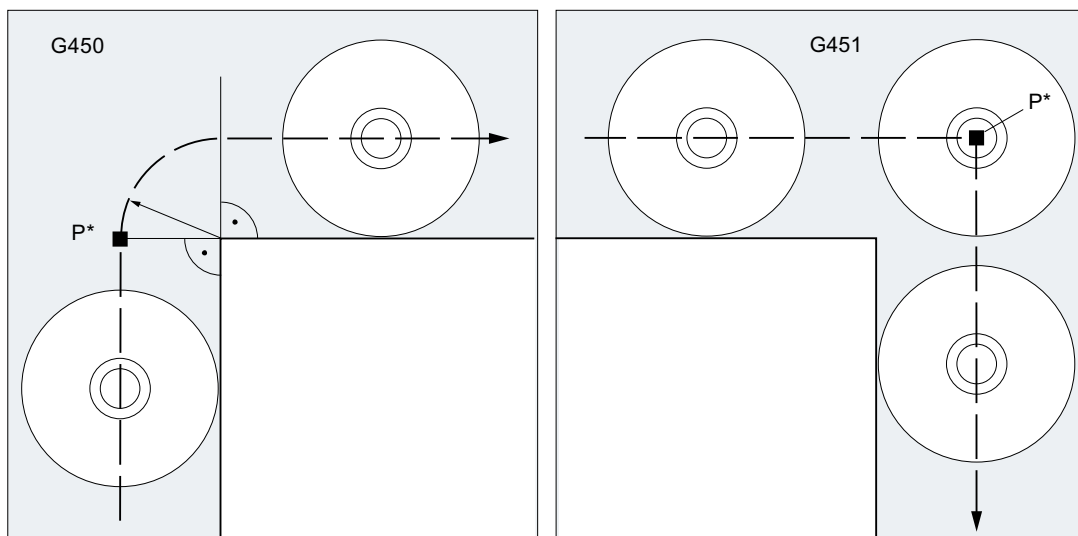


以下の図は、KONTT と KONTC 間のアプローチ/後退動作の違いを示します。X0 Y-40 の中心点を中心とする半径 20 mm の円は、外側半径 20 mm の工具で補正されます。したがって、工具の中心点は、半径 40 mm の円弧軌跡に沿って移動します。アプローチブロックの終点は X40 Y30 にあります。円弧ブロックと後退ブロック間の遷移はゼロ点でおこなわれます。KONTC に関連した曲率はその後も継続するため、後退ブロックは最初に、負の Y 成分で移動を実行します。これは、多くの場合、望ましくない状況です。この動作は、KONTT 後退ブロックでは発生しません。ただし、このブロックの場合は、ブロック遷移のとき加減速の不連続変化が発生します。

KONTT または KONTC ブロックが後退ブロックでなく、アプローチブロックの場合は、輪郭はまったく同一で、反対方向に加工されます。

3.10.3 外側コーナの補正(G450、G451、DISC)

工具径補正が有効(G41/G42)な場合は、G450 または G451 命令を使用して、外側コーナの周りを移動時に補正された工具軌跡の進路を定義できます。



G450 の場合は、工具中心点が、ワークコーナの周りを工具半径の円弧を描いて移動します。

G451 の場合は、工具中心点は、工具とプログラム指令輪郭の間の間隔が工具半径と等距離の、2つの線の交点へアプローチします。G451 は、円弧と直線にのみ適用されます。

注記

G450/G451 は、KONT が有効な場合のアプローチ軌跡と輪郭の後ろのアプローチ点の定義にも使用されます(「輪郭へのアプローチと後退(NORM、KONT、KONTC、KONTT) (ページ 291)」を参照してください)。

DISC 命令を使用すると、G450 による挿入円を変形させ、それによって輪郭コーナをより鋭角にすることができます。

構文

G450 [DISC=<値>]

G451

意味

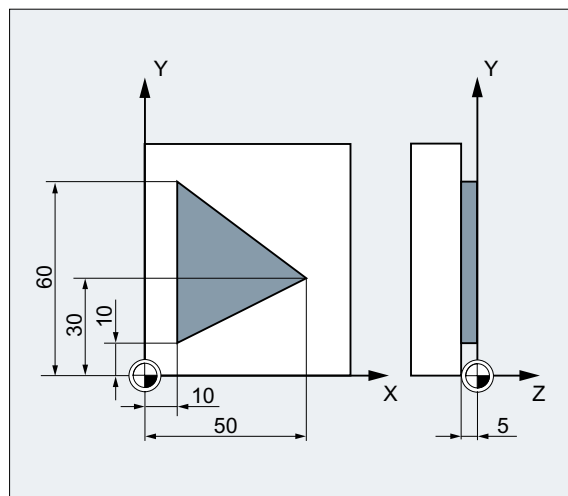
G450:	G450 は、ワークコーナの周りの円弧軌跡の移動に使用されます。				
DISC:	G450 による円弧軌跡のフレキシブルなプログラミング(任意選択)				
	<値>:	タイ プ:	INT		
		値の範囲:		0, 1, 2, ... 100	
		意味		0	挿入円
				100	等距離軌跡の交点(理論値)
G451:	G451 を使用して、ワークコーナの場合は、2 つの等距離軌跡の交点へアプローチします。工具はワークのコーナから離れます。				

注記

DISC は、G450 の呼び出しを使用する場合にのみ適用されますが、G450 を使用していない、それ以前のブロックでプログラム指令できます。これらの命令は両方ともモーダルです。

例

以下の例では、(N30 ブロックのコーナ動作のプログラムによって)、挿入円をすべての外側コーナに対してプログラムしています。こうすることで、方向変更時の工具の停止と後退が防止されます。



プログラムコード	コメント
N10 G17 T1 G0 X35 Y0 Z0 F500	; 起動条件.
N20 G1 Z-5	; 工具を送ります。
N30 G41 KONT G450 X10 Y10	; KONT アプローチ/後退モードおよび コーナ動作 G450 で TRC を有効にします。
N40 Y60	; 輪郭をフライス加工します。
N50 X50 Y30	
N60 X10 Y10	
N80 G40 X-20 Y50	; 補正モードを解除し、挿入円で後退します。
N90 G0 Y100	
N100 X200 M30	

詳細情報

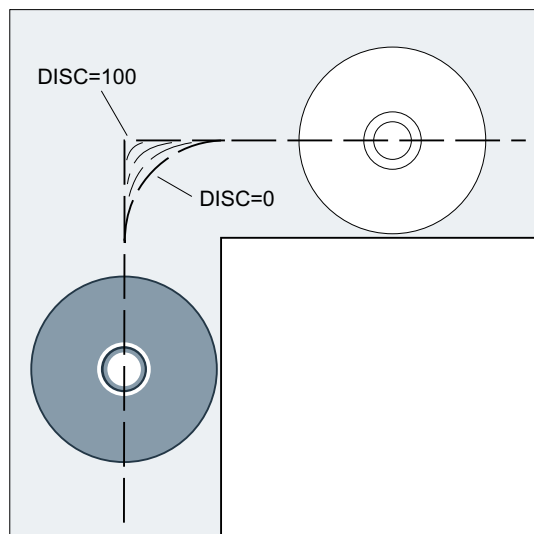
G450/G451

中間点 P*で、制御装置は、切り込み移動や機能切り替えなどの命令を実行します。これらの命令はコーナを形成する 2 つのブロックの間に挿入されたブロックでプログラム指令されます。

G450 の場合、挿入円は、そのデータに関しては次の移動指令に属します。

DISC

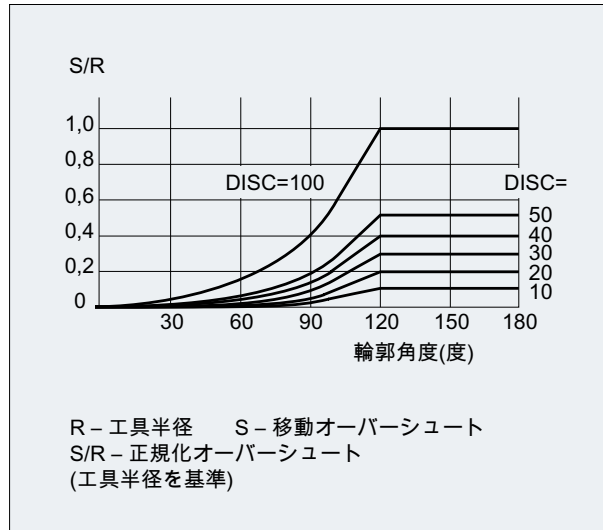
0 より大きい DISC 値を指定すると、その高さが拡大された中間円弧が示され、結果は挿入楕円、挿入放物線、または挿入双曲線になります。



上限は、マシンデータで定義できます(通常は DISC=50)。

移動動作

G450 が鋭角の輪郭角度と大きな DISC 値で有効な場合、コーナで工具が輪郭から離れます。輪郭角度が 120°以上の場合は、輪郭の周りを均一な S/R 値で移動します。

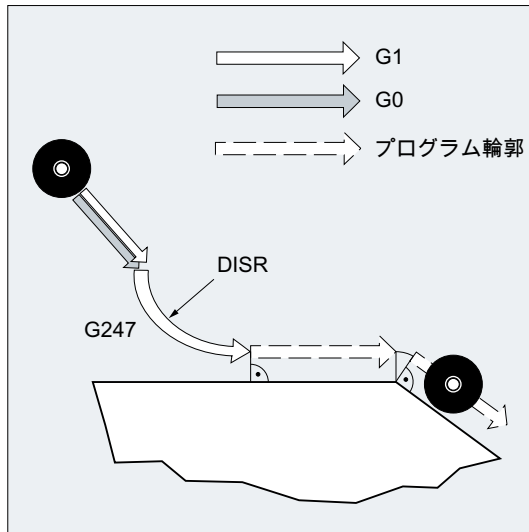


G451 が鋭角の輪郭角度で有効になると、軌跡から離れる移動により、余分な非切削工具軌跡が発生することがあります。このような場合の挿入円への自動切り替えは、マシンデータのパラメータを使用して定義できます。

3.10.4 滑らかなアプローチと後退

3.10.4.1 アプローチと後退(G140～G143、G147、G148、G247、G248、G347、G348、G340、G341、DISR、DISCL、DISRP、FAD、PM、PR):

SAR (Smooth Approach and Retraction: 滑らかなアプローチと後退)機能を使用して、始点の位置に関係なく、輪郭の始点へ接線方向のアプローチを実行します。



この機能は、通常は工具径補正と組み合わせて使用します。

この機能を有効にすると、制御装置は、指定したパラメータに応じて次のブロックへの移行（または後退時には前のブロックからの移行）が実行されるように、中間点を計算します。

アプローチ移動は、4 つまでの移動区分から成ります。移動の起点は P_0 、終点は P_4 と呼ばれます。この 2 つのポイント間に、 P_1 、 P_2 、 P_3 の最大 3 つの中間点を設定できます。点 P_0 、 P_3 、および P_4 は常に定義されます。中間点 P_1 と P_2 は、定義したパラメータとジオメトリ条件に従って省略できます。後退時には、これらの点を逆方向に通過します。つまり、 P_4 から始まり、 P_0 で終わります。

構文

滑らかなアプローチ:

- 直線による:
G147 G340/G341 ... DISR=..., DISCL=..., DISRP=...FAD=...
- 4 分円/半円による:
G247/G347 G340/G341 G140/G141/G142/G143 ...
DISR=...DISCL=...DISRP=...FAD=...

滑らかな後退:

- 直線による:
G148 G340/G341 ... DISR=..., DISCL=..., DISRP=...FAD=...
- 4 分円/半円による:
G248/G348 G340/G341 G140/G141/G142/G143 ...
DISR=...DISCL=...DISRP=...FAD=...

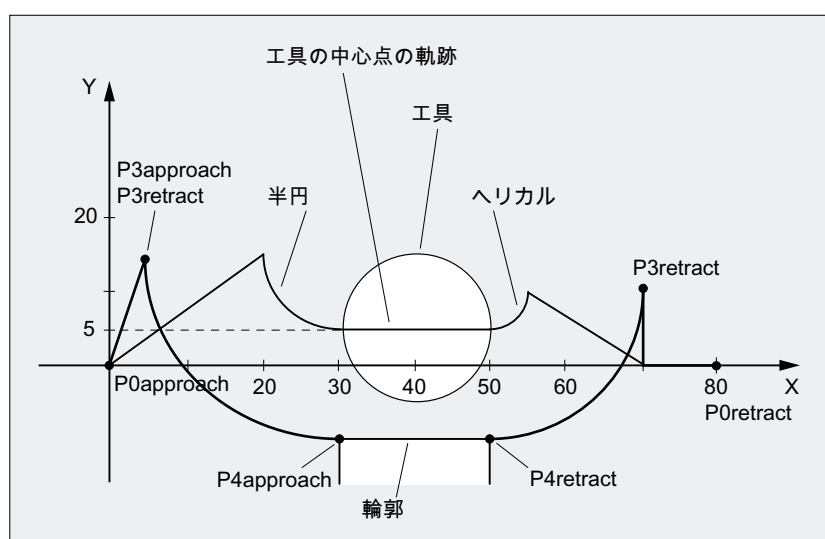
意味

G147:	直線によるアプローチ
G148:	直線による後退
G247:	4 分円によるアプローチ
G248:	4 分円による後退
G347:	半円によるアプローチ
G348:	半円による後退
G340:	空間でのアプローチと後退(初期設定)
G341:	平面でのアプローチと後退
G140:	現在の補正サイドに応じたアプローチと後退方向(初期設定)
G141:	左からのアプローチまたは左への後退
G142:	右からのアプローチまたは右への後退
G143:	接線の方向の始点と終点の相対位置に対応したアプローチと後退方向
DISR=...:	1. 直線によるアプローチと後退の場合(G147/G148): 輪郭の起点から刃先までの距離 2. 円弧(G247、G347/G248、G348)によるアプローチと後退の場合: 工具中心点の軌跡の半径 重要: 半円による REPOS の場合、DISR は円弧の直径
DISCL=...:	加工平面から高速切り込み移動の終点までの距離 DISCL=AC(...)高速切り込み移動の終点の絶対位置の指定
DISCL=AC(...) :	高速切り込み移動の終点の絶対位置の指定
DISRP:	加工平面から点 P1(イニシャル点)までの距離
DISRP=AC(...) :	点 P1 の絶対位置の指定

3.10 工具径補正

FAD=...:	滑らかな送り移動の速度 プログラム指令値は、現在の送り速度タイプ(G グループ 15)に基づいて動作します。
FAD=PM(...):	プログラム指令値は、現在の送り速度タイプに関係なく、毎分送り(G94 のように)として解釈されます。
FAD=PR(...):	プログラム指令値は、現在の送り速度タイプに関係なく、毎回転送り速度(G95 のように)として解釈されます。

例



- 滑らかなアプローチ(N20 ブロックが動作)です。
- 4 分円によるアプローチ(G247)です。
- アプローチ方向がプログラム指令されていないため、G140 が適用されます。つまり、TRC が有効です(G41)。
- 輪郭オフセット OFFN=5 (N10)。
- 現在の工具半径=10。そして、有効な TRC 補正半径=15、SAR 輪郭の半径=25 です。その結果、工具中心軌跡の半径は DISR=10 と同じです。
- Z 位置のみが N20 でプログラム指令されているため、円弧の終点は N30 から取得されます。

- 切り込み移動
 - 早送りで Z20 から Z7 へ(DISCL=AC(7))移動します。
 - その後、FAD=200 で Z0 へ移動します。
 - X/Y 平面のアプローチ円弧、および F1500 を含む後続ブロック(この速度を後続のブロックで有効にするには、N30 の動作中の G0 を G1 で上書きします。上書きしない場合は、さらに輪郭が G0 で加工されます)。
- 滑らかな後退が(N60 ブロックで有効)です。
- 4 分円(G248)とヘリカル(G340)による後退です。
- FAD は G340 には無関係のため、FAD はプログラム指令されていません。
- DISCL=6 のため、起点では Z=2、終点では Z=8 です。
- DISR=5 の場合は、SAR 輪郭の半径=20、工具中心点軌跡の半径=5 です。

Z8 から Z20 への後退移動、および X/Y 平面に平行に X70 Y0 へ移動します。

プログラムコード	コメント
\$TC_DP1[1,1] = 120	; 工具定義 T1/D1
\$TC_DP6[1,1]=10	; 半径
N10 G0 X0 Y0 Z20 G64 D1 T1 OFFN=5	; (P0 アプローチ)
N20 G41 G247 G341 Z0 DISCL=AC(7) DISR=10 F1500 FAD=200	; アプローチ (P3 アプローチ)
N30 G1 X30 Y-10	; (P4 アプローチ)
N40 X40 Z2	
N50 X50	; (P4 後退)
N60 G248 G340 X70 Y0 Z20 DISCL=6 DISR=5 G40 F10000	; 後退 (P3 後退)
N70 X80 Y0	; (P0 後退)
N80 M30	

詳細情報

アプローチ輪郭と後退輪郭の選択

アプローチ輪郭と後退輪郭は、次に示す 2 番目の G グループに属する当該の G 命令によって選択されます。

G147:	直線によるアプローチ
G247:	4 分円によるアプローチ
G347:	半円によるアプローチ
G148:	直線による後退
G248:	4 分円による後退
G348:	半円による後退

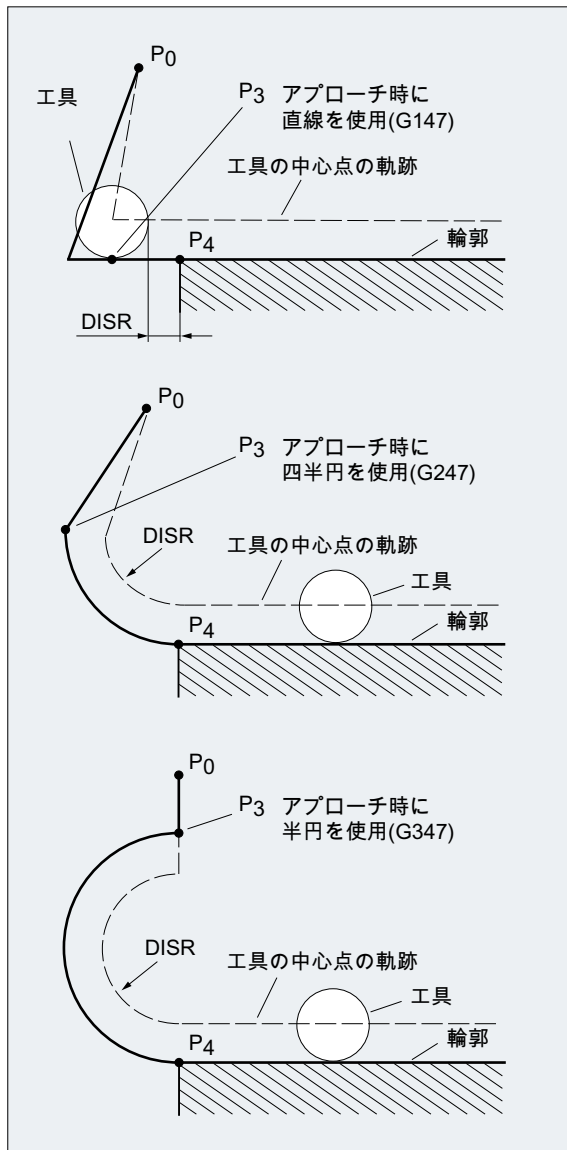


図 3-20 同時に工具径補正を実施するアプローチ移動

アプローチ方向と後退方向の選択

工具径補正(G140、初期設定)を使用して、正の工具半径でアプローチ方向と後退方向を特定します。

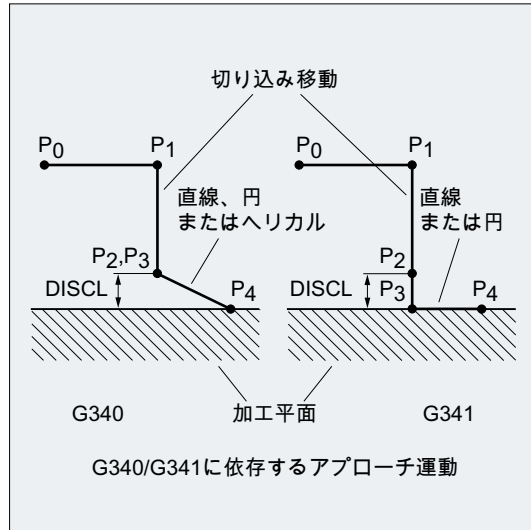
- G41 が有効→左からのアプローチ
- G42 が有効→右からのアプローチ

さらに、G141、G142、および G143 によるアプローチオプションを使用できます。

この G コードは、アプローチ輪郭が 4 分円または半円のときにのみ有効です。

始点と終点の間の動作ステップ(G340 と G341)

どのような場合でも、移動は、複数の直線と、アプローチ輪郭を特定する G 命令に対応して、追加の直線や 4 分円または半円から構成されます。以下の図に 2 つのタイプの軌跡分割を示します。



G340:	点 P_0 から点 P_1 まで直線でアプローチします。パラメータ DISRP をプログラム指令していない場合は、この直線は加工平面に平行です。 点 P_1 から、DISCL パラメータで定義した加工平面までの安全距離である点 P_3 まで、加工平面に垂直に切り込みます。 2 番目のグループの G 命令(直線、円弧、ヘリカル)によって定義されたカーブで、終点 P_4 までアプローチします。G247 または G347 が有効であり(4 分円または半円)、開始点 P_3 が終点 P_4 によって定義される加工平面から離れている場合は、円ではなくヘリカルが挿入されます。点 P_2 は定義されないか、または P_3 と一致します。 円弧平面またはヘリカル軸は、SAR ブロックで有効な平面(G17G18/G19)によって特定されます。つまり、次のブロックで、開始接線自体ではなく接線の投影を使用して円が定義されます。 点 P_0 から点 P_3 への移動は、SAR ブロックの前に有効だった速度で、2 本の直線に沿っておこなわれます。
G341:	点 P_0 から点 P_1 まで直線でアプローチします。パラメータ DISRP をプログラム指令していない場合は、この直線は加工平面に平行です。 点 P_1 から、DISCL パラメータで定義した加工平面までの安全距離である点 P_2 まで、加工平面に垂直に切り込みます。 点 P_2 から点 P_3 まで加工平面に垂直に切り込みます。2 番目のグループの G 命令によって定義されたカーブで終点までアプローチします。P_3 と P_4 は加工平面内にあるため、G247 または G347 ではヘリカルではなく常に円弧が挿入されます。

有効平面 G17G18/G19(円弧平面、ヘリカル軸、有効平面に垂直な切り込み移動)の位置が含まれる場合は常に、動作中の回転フレームがすべて考慮されます。

アプローチ直線の長さまたはアプローチ円弧の半径(DISR)

- 直線によるアプローチ/後退

DISR で、輪郭の起点からの刃先の距離を指定します。つまり、TRC が有効なときの直線の長さは、工具半径と DISR のプログラム指令値の合計です。工具半径は、正の場合にのみ考慮されます。

結果の直線の長さは正になるようにしてください。つまり、DISR に負の値を使用できるのは、DISR の絶対値が工具半径未満の場合のみです。

- 円弧によるアプローチ/後退

DISR で、工具中心点軌跡の半径を指定します。TRC が有効な場合は、特定の半径の円弧が生成され、この円弧がプログラム指令半径による工具中心点軌跡となります。

加工平面から点 P2 までの距離(DISCL)

点 P₂ の位置が、円弧平面に垂直な軸上でアブソリュート基準により指定される場合は、その値を、DISCL=AC(...) という形式でプログラム指令してください。

以下の場合には DISCL=0 で適用されます。

- G340 の場合:アプローチ移動全体の構成が、2 つのブロックのみとなる(P₁、P₂、および P₃ が一緒になります)。アプローチ輪郭が P₁~P₄ で形成される。
- G341 の場合:アプローチ輪郭全体の構成が、3 つのブロックとなる(P₂ と P₃ が一緒になります)。また、P₀ と P₄ が同じ平面上にある場合は、2 つのブロックのみとなる(P₁ から P₃ への切り込み移動は省略されます)。
- DISCL で定義した点が、P₁ と P₃ の間に必ず位置するようになっている。つまり、加工平面に垂直な成分では、この成分をもつすべての移動で、符号は同じになります。
- 方向の逆転の検出時には、マシンデータ MD20204 \$MC_SAR_CLEARANCE_TOLERANCE で定義した許容範囲が使用されます。

加工平面から点 P1(イニシャル点)までの距離(DISRP)

点 P₁ の位置が、加工平面に垂直な軸上でアブソリュート基準により指定される場合は、その値を、DISRP=AC(...) という形式でプログラム指令してください。

このパラメータをプログラム指令しない場合は、加工平面から点 P₁ までの距離は点 P₀ までの距離と同じになります。つまり、アプローチ直線 P₀ → P₁ は加工平面に対して平行になります。

DISCL によって定義される点が、P₀ と P₂ の間にいることがシステムによって確認されます。つまり、加工平面に垂直な成分をもつすべての移動(切り込み方向移動、P₃ から P₄ へのアプローチ動作)で、この成分の符号は同じになります。方向の反転はできません。この条件に違反すると、アラームが出力されます。

方向の逆転の検出時には、マシンデータ MD20204 \$MC_SAR_CLEARANCE_TOLERANCE で定義した許容範囲が使用されます。ただし、P₁ が P₀ と P₂ によって定義される範囲外にあり、誤差がこの許容範囲以下の場合、P₁ は P₀ または P₂ によって定義される平面内にあるとみなされます。

終点のプログラミング

終点は通常、X...Y...Z...でプログラム指令します。

アプローチ時の輪郭終点のプログラミングは、後退時のプログラミングとは全く違います。そのため、ここでは 2 つの場合を個別に扱います。

アプローチの終点 P4 のプログラミング

3.10 工具径補正

終点 P_4 は、SAR ブロック自体にプログラム指令できます。あるいは、次の移動ブロックの終点によって P_4 を決定することもできます。SAR ブロックと次の移動ブロックの間には、さらにジオメトリ軸の移動なしブロックを挿入できます。

例:

プログラムコード	コメント
\$TC_DP1[1,1] = 120	; フライス工具 T1/D1
\$TC_DP6[1,1]=7	; 半径 7 mm の工具
N10 G90 G0 X0 Y0 Z30 D1 T1	
N20 X10	
N30 G41 G147 DISCL=3 DISR=13 Z=0 F1000	
N40 G1 X40 Y-10	
N50 G1 X50	
...	

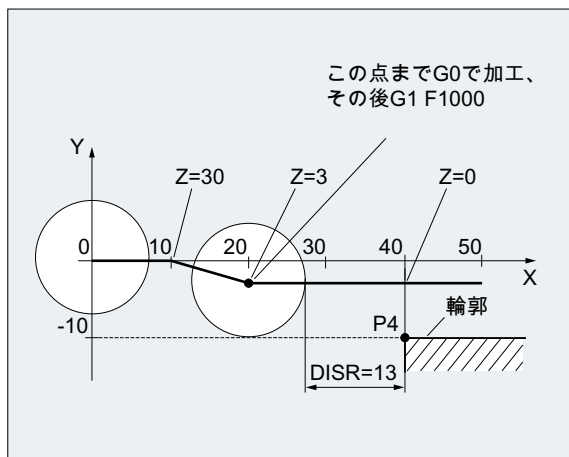
N30/N40 は、次のように置換できます。

N30 G41 G147 DISCL=3 DISR=13 X40 Y-10 Z0 F1000

または

N30 G41 G147 DISCL=3 DISR=13 F1000

N40 G1 X40 Y-10 Z0



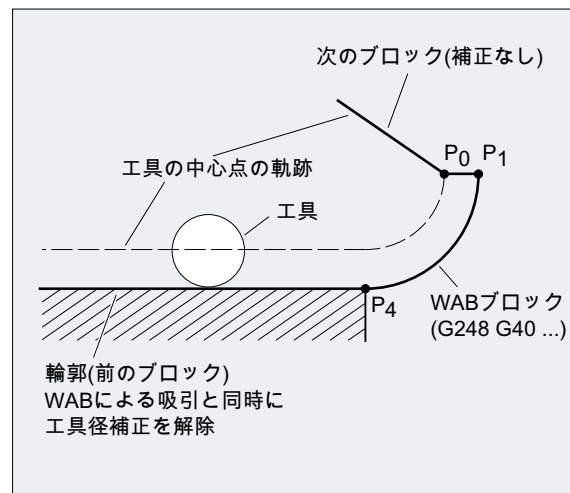
後退の終点 P_0 のプログラミング

後退の場合、SAR 輪郭の終点は、後続のブロックではプログラム指令できません。つまり、終了位置は、プログラム指令軸の数に関係なく、常に SAR ブロックから取得します。終点の特定時には、次の 3 つの場合があります。

1. SAR ブロックにジオメトリ軸がプログラム指令されていない場合。この場合、輪郭は点 P_1 (DISRP のプログラム指令がある場合)、点 P_2 (DISCL のプログラム指令はあるが、DISRP のプログラム指令がない場合)、または点 P_3 (DICLS と DISRP のどちらのプログラム指令もない場合) で終了します。
加工平面を記述する軸の位置は、後退輪郭によって決定されます(直線または円弧の終点)。これに垂直な軸成分は、DISCL または DISRP で定義します。この場合に DISCL=0 かつ DISRP=0 であれば、完全に平面上での移動になります。つまり点 P_0 から P_3 までは一致します。
2. 加工平面に対して垂直な軸のみが SAR ブロックでプログラム指令されている場合。この場合、輪郭は点 P_0 で終了します。DISRP がプログラム指令されている(つまり、点 P_0 と P_1 が一致しない)場合、直線 $P_1 \rightarrow P_0$ は加工平面に対して垂直になります。他の 2 つの軸の位置は、1 と同じ方法で特定されます。
3. 加工平面の 1 つ以上の軸がプログラム指令されている場合。加工平面の 2 番目の軸は、前のブロックでのその軸の最終位置からモデルに決定できます。

加工平面に対して垂直な軸がプログラム指令されているかどうかに応じて、この軸の位置が 1 または 2 の説明のとおり生成されます。この方法で生成される位置によって、終点 P_0 が定義されます。SAR 後退ブロックが工具径補正の解除にも使用される場合、最初の 2 つのケースでは、加工平面に P_1 から P_0 までの追加軌跡成分が挿入され、後退輪郭の終点で工具径補正が解除される時に移動が生成されないようにします。つまり、この点は、補正された輪郭上の位置ではなく、工具中心点を表します。ケース 3 では、プログラム指令点 P_0 によって、全輪郭の終点での工具中心点の位置がすでに直接定義されているため、工具径補正を解除するための特別な対策は不要です。

ケース 1 と 2、つまり、工具径補正の同時解除とともに加工平面の終点が明確にプログラム指令されていない場合の動作を、次の図に示します。



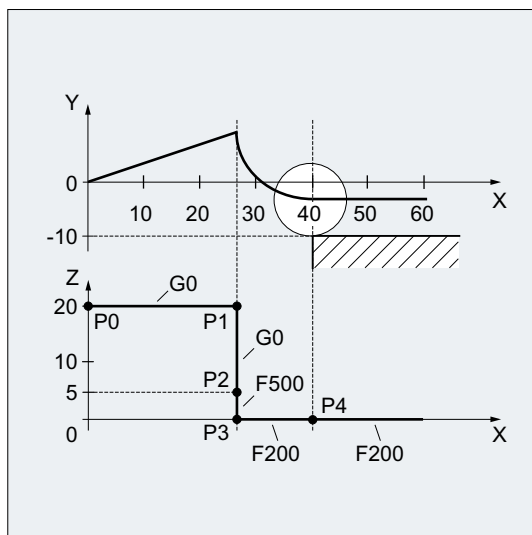
3.10 工具径補正

アプローチ速度と後退速度

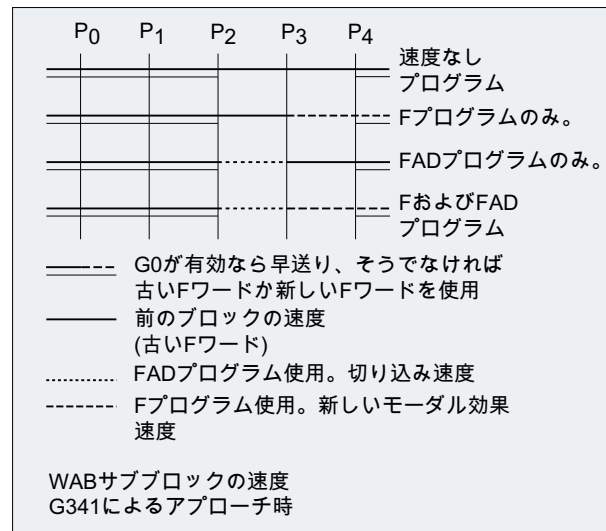
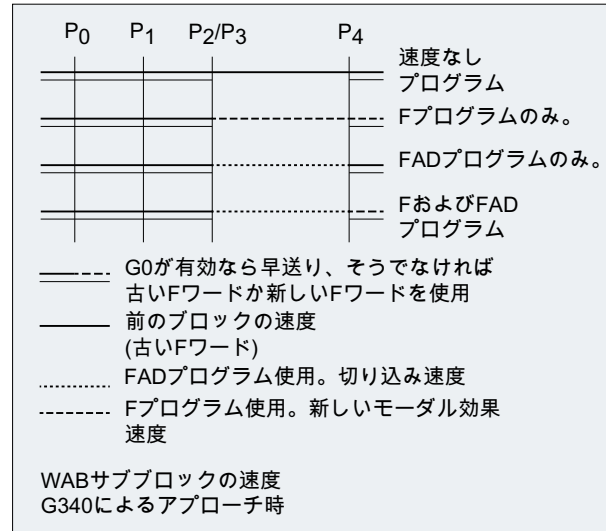
- 前のブロックの速度(G0)
 P_0 から P_2 までのすべての移動、つまり、加工平面に平行な移動、および安全距離までの切り込み移動部分は、この速度で実行されます。
- FAD によるプログラミング
 以下の送り速度を指定します。
 - G341: P_2 から P_3 までの加工平面に垂直な切り込み移動
 - G340: P_2 または P_3 から P_4 への切り込み移動
 FAD がプログラム指令されていない場合、速度を定義する F 命令が SAR ブロックでプログラム指令されないと、前のブロックからモーターに有効になった速度で輪郭のこの部分を移動します。
- プログラム指令送り速度 F
 この送り速度の値は、FAD をプログラム指令していない場合に、 P_3 または P_2 で有効です。F ワードを SAR ブロックでプログラム指令していない場合は、前のブロックの速度が有効です。

例:

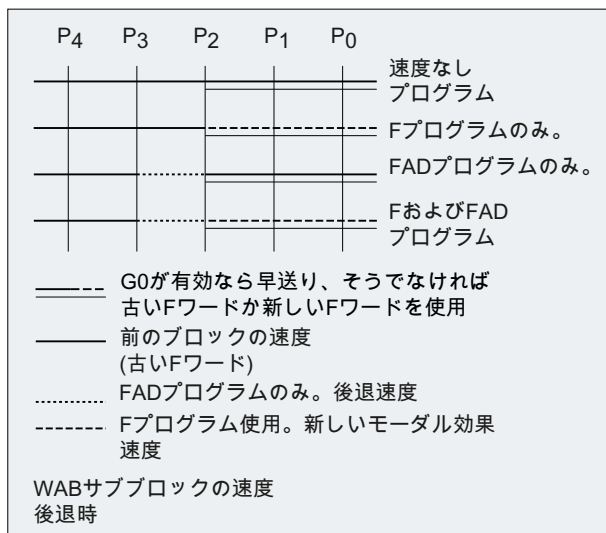
プログラムコード	コメント
\$TC_DP1[1,1] = 120	; フライス工具 T1/D1
\$TC_DP6[1,1]=7	; 半径 7 mm の工具
N10 G90 G0 X0 Y0 Z20 D1 T1	
N20 G41 G341 G247 DISCL=AC(5) DISR=13 FAD 500 X40 Y-10 Z=0 F200	
N30 X50	
N40 X60	
...	



後退中には、前のブロックからモーダルで動作中の送り速度と、SARブロックでプログラム指令した送り速度の値の役割が逆転します。つまり、実際の後退輪郭は旧送り速度で移動し、Fワードでプログラム指令された新しい速度が P_2 から P_0 までに適用されます。



3.10 工具径補正



位置の読み取り

点 P_3 と P_4 はアプローチ中に、WCS でシステム変数として読み取ることができます。

- $\$P_APR:P$ の読み取り
- $_3$ (イニシャル点)
- $\$P_AEP:P$ の読み取り
- $_4$ (輪郭の起点)
- $\$P_APDV:\P_APR と $\$P_AEP$ に有効なデータが含まれるかどうかの読み取り

3.10.4.2 拡張された後退方法によるアプローチと後退(G460、G461、G462)

特定の特別な形状の事例の場合、工具径補正を起動または解除するためには、アプローチと後退ブロックに対して以前から使用されている、衝突検出による実行方法と比べて、特別に拡張されたアプローチと後退方法が必要です。衝突検出を使用すると、輪郭の一部で加工が完全でない等の結果になることがあります。次の図を参照してください。

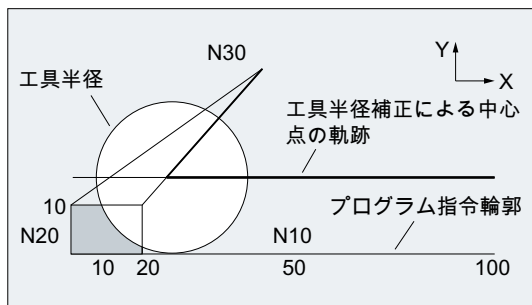


図 3-21 G460 による後退動作

構文

G460

G461

G462

意味

G460:	従来と同様(アプローチと後退ブロックの衝突検出の適用)。
G461:	TRC ブロックに円弧を挿入(中心点が補正なしのブロックの終点にあり、交点の半径が工具半径と同じため、交点ができない場合)。 交点まで、輪郭の終点を中心とした 補助円弧 により (輪郭の終点まで)加工がおこなわれます。
G462:	TRC ブロックに直線を挿入(交点ができない場合; ブロックはその終了時の接線によって拡張されます)(初期設定)。 加工は、最後の輪郭要素の 拡張部分 (輪郭の終点の少し手前)までおこなわれます。

注記

アプローチ動作は後退動作とは対称に動作します。

アプローチ/後退動作は、アプローチ/後退ブロックの G 命令の状態で特定されます。したがって、アプローチ動作は、後退動作とは無関係に設定できます。

例

例 1:G460 による後退動作

次の例は、工具径補正解除のときの事例のみを説明します。アプローチの動作はまったく同じです。

プログラムコード	コメント
G42 D1 T1	; 工具半径 20 mm
...	
G1 X110 Y0	
N10 X0	
N20 Y10	
N30 G40 X50 Y50	

例 2 : G461 によるアプローチ

プログラムコード	コメント
N10 \$TC_DP1[1,1]=120	; フライス工具のタイプ
N20 \$TC_DP6[1,1]=10	; 工具半径
N30 X0 Y0 F10000 T1 D1	
N40 Y20	
N50 G42 X50 Y5 G461	
N60 Y0 F600	
N70 X30	
N80 X20 Y-5	
N90 X0 Y0 G40	
N100 M30	

詳細情報

G461

最後の TRC ブロックと先行するブロックの間に交点ができない場合は、このブロックのオフセット曲線は、中心点が補正なしのブロックの終点にあり、半径が工具半径と同じである円弧により拡張されます。

制御装置は、先行するブロックのいずれかを使用して、この円弧を切削しようとしています。

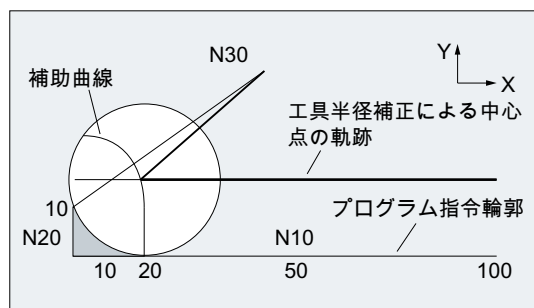


図 3-22 G461 による後退動作

衝突監視 CDON、CDOF

CDOF が有効な場合(「衝突監視、CDON、CDOF」の章を参照してください)は、交点が見つかり、検索が中止されます。つまり、システムでは、先行するブロックにそれ以上の交点があるかどうかをチェックしません。

CDON が有効な場合は、最初の交点が見つかった後も、さらに交点の検索を続行します。

このようにして見つかった交点が、先行ブロックの新しい終点であり、解除ブロックの始点です。挿入された円弧は、交点の計算専用としてのみ使用され、移動が生じることはありません。

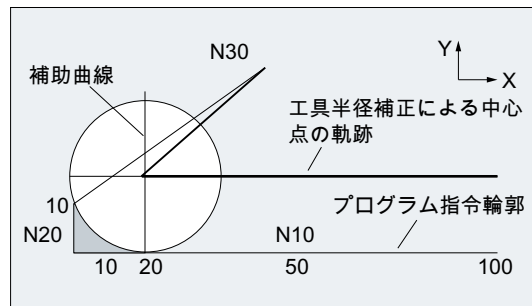
注記

交点が見つからない場合は、アラーム 10751(衝突の危険)が発生します。

G462

最後の TRC ブロックと先行ブロックの間で交点ができない場合は、工具径補正によって、最後のブロックの終点で、G462 による後退時(初期設定)に直線が挿入されます(ブロックはその終了時の接線によって拡張されます)。

その後の交点の検索は、G461 の場合の手順と同じです。



G462 による後退動作(例を参照してください)

G462 の場合は、プログラム例のなかで N10 と N20 により生成されたコーナは、使用工具では実際に移動可能な範囲の限界までは加工されません。ただし、例のなかで N20 の左に対して(プログラム指令輪郭とは違う)輪郭部分で、y 値の高さが 10 mm より大きいために干渉して加工できない場合は、この動作が必要となる場合があります。

KONT によるコーナ動作

KONT が有効な場合(始点または終点で輪郭の周りを移動)は、終点が輪郭の前か後ろかにより、動作が異なります。

- **終点が輪郭の前の場合**

終点が輪郭の前にある場合、後退動作は **NORM** の場合と同じです。この機能は、**G451** の最後の輪郭ブロックが直線または円弧に拡張された場合でも変わりません。したがって、輪郭の終点付近の経路干渉を避けるために、さらに回り込みの移動方法を設定する必要はありません。

- **終点が輪郭の後ろの場合**

終点が輪郭の後ろにある場合は、**G450/G451** に応じて円弧または直線が常に挿入されます。この場合、**G460~462** は無効です。この事例で、最後の移動ブロックに先行ブロックとの交点がない場合は、挿入された輪郭要素による、またはプログラム指令終点へのバイパス円弧の終点の直線による交点ができる可能性があります。

挿入された輪郭要素が円弧(**G450**)で、先行ブロックとの接合点を形成する場合、これは、**NORM** と **G461** により発生する接合点と同じです。ただし通常は、円弧の残りの部分は、まだ移動する必要があります。後退ブロックの直線部分については、交点をこれ以上、計算する必要はありません。

2 番目の例では、挿入した輪郭要素に、先行ブロックとの接合点が見つからない場合、後退直線と先行ブロックの間の交点が移動します。

したがって、**NORM** が有効、または **KONT** による動作が **NORM** による動作と幾何学的に同じであれば、**G461** または **G462** が動作中の場合にのみ、**G460** と異なる動作が発生します。

3.10.5 衝突検出(「ボトルネック検出」)の起動/解除(CDON、CDOF、CDOF2)

有効な工具径補正による衝突監視(「ボトルネック検出」)は、**G** グループ 23 の命令によって **NC** プログラムで起動または解除されます。

構文

```
G41/G42 CDON  
...  
CDOF/CDOF2
```

意味

CDON:	衝突検出(「ボトルネック検出」)の起動 CDON を使用して、非隣接または非連続的なブロックの工具軌跡が交差するかどうか調整可能な(MD20240)ブロック数を確認できます。この先読み機能を使用すると、発生する可能性のある衝突が事前に検出されるため、コントローラが能動的にそれを回避できます。
CDOF:	衝突検出(「ボトルネック検出」)の解除 CDOF では、実行中のブロックの共通の交点に対する検索が(内側コーナの)先行の移動ブロックで実行され、必要に応じて、さらに先行のブロックに対しても実行されます。交点が検出されると、それ以上のブロックは調べられません。外側コーナーでは、交点は常に 2 つの連続したブロック間で検出されます。 注: CDOF を使用すると、必要な情報が NC プログラムに存在しない場合などに発生するボトルネックの検出ミスを回避できます。
CDOF2:	3 次元外周加工用の衝突検出の無効化 工具オフセット方向は、CDOF2 の隣接ブロック部分により特定されます。CDOF2 は 3 次元外周削りにのみ有効で、他のすべてのタイプの加工(2 次元削り(CUT2D)や 3 次元正面削り(CUT3DFD)など)に対しては CDOF と同じです。

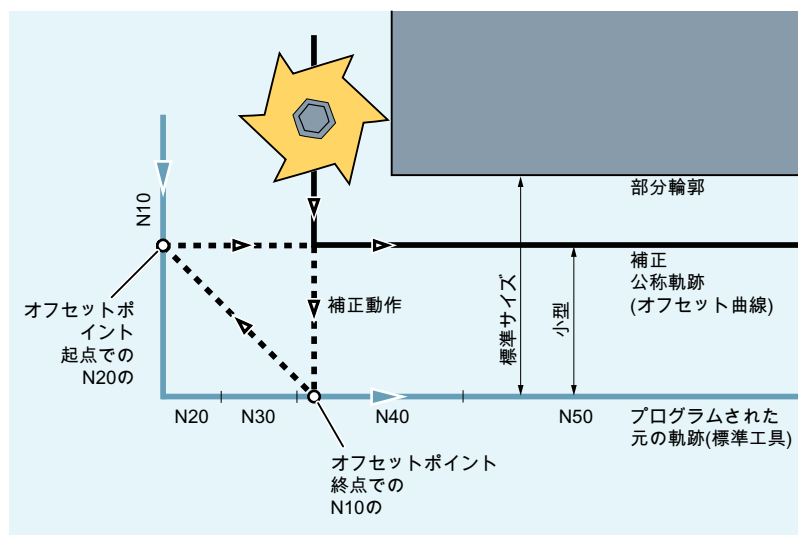
一例を使用した衝突検出の働き

NC プログラムで、標準工具の中心点軌跡を記述します。以下の図では、ジオメトリ関係を図示するために、輪郭が実際以上に大きく表示されているため、実際に使用する工具の輪郭は小さくなります。

例では、制御は 3 つのブロックの概要のみです。

MD20240 \$MC_CUTCOM_MAXNUM_CHECK_BLOCKS = 3

3.10 工具径補正



交点は、N10 と N40 の 2 つのブロックのオフセット曲線間にのみ存在するため、N20 と N30 の 2 つのブロックは省略されます。例では、N40 ブロックでは、コントローラは、N10 を完全に処理する必要があるかどうかを認識していません。したがって、省略できるブロックは 1 つのみです。

CDOF2 が動作中の場合は、図に示す補正動作が実行されます。また、この動作は停止しません。この事例では、CDOF または CDON が動作中の場合は、結果的にアラームが発生します。

3.10.6 2 1/2 次元工具オフセット (CUT2D、CUT2DD、CUT2DF、CUT2DFD)

2½次元工具径補正は、傾斜面の加工時にワークを回転させる場合に使用します。工具の割り出しではありません。この機能は、命令 CUT2D、CUT2DD、CUT2DF、CUT2DFD を使用して有効にします。

工具長オフセット

工具長補正は常に、固定された回転しない加工平面を基準として考慮されます。

輪郭工具の 2½次元工具径補正

輪郭工具の 2½次元工具径補正は、CUT2D、CUT2DD、CUT2DF、または CUT2DFD と共に、次の 2 つの命令 G41 (輪郭の左側の工具径補正) または G42 (輪郭の右側の工具径補正) のいずれかがプログラムされている場合に有効になります。これは、個々の輪郭セグメントの

ワーク単位加工に使用できる、軸対称でない工具の場合に、自動刃先選択に使用できません。

注記

2½次元工具径補正が無効の場合は、輪郭工具が、1番目の刃先のみを備えた標準工具と同様に動作します。

標準工具とは違う工具を基準にした 2½次元工具径補正

標準工具とは違う工具を基準にした 2½次元工具径補正は、CUT2DD または CUT2DFD 命令を使用して有効にします。この工具径補正は、プログラムされた輪郭が標準工具とは違う工具の中心点軌跡を基準にしている、標準工具とは違う工具以外が加工に使用されている場合に適用します。2½次元工具径補正を計算する場合は、有効な工具の半径の摩耗(\$TC_DP_15)とプログラムされていると思われる工具オフセット OFFN (ページ 280)および TOFFR (ページ 94)のみが考慮されます。動作中の工具の基本半径(\$TC_DP6)は**考慮されません**。

構文

```
CUT2D
CUT2DD
CUT2DF
CUT2DFD
```

意味

CUT2D:	2½次元工具径補正の有効化
CUT2DD:	標準工具とは違う工具を基準にした 2½次元工具径補正の有効化
CUT2DF:	2½次元径補正、および現在のフレームまたは傾斜面に対する工具径補正の有効化
CUT2DFD:	2½次元径補正、および現在のフレームまたは傾斜面に対する工具径補正の有効化

詳細情報

輪郭工具

- 有効化

輪郭工具の工具径補正は、以下を使用してチャンネル別に有効にされます。

MD28290 \$MC_MM_SHAPED_TOOLS_ENABLE

- 工具タイプ

輪郭工具タイプは、以下を使用してチャンネル別に定義されます。

MD20370 \$MC_SHAPED_TOOL_TYPE_NO

- 刃先

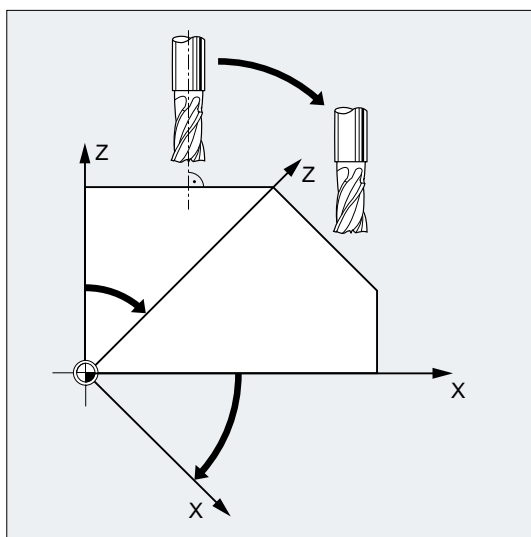
各輪郭工具に刃先の番号(D 番号)を任意の順序で割り当てることができます。1 工具あたりの最大刃先数は以下を使用してパラメータ設定されます。

MD18106 \$MN_MM_MAX_CUTTING_EDGE_PERTOOL

輪郭工具の 1 番目の刃先は、工具を有効にしたときに選択される刃先です。たとえば、プログラムで命令 T3 D5 を使用して 3 番目の工具(T3)の 5 番目の刃先(D5)が有効になっている場合、D5 およびそれ以降の刃先は輪郭工具の一部、または全体と共に定義されます。D5 の前に位置する刃先は無視されます。

補正平面の回転なしの 2½次元工具径補正(CUT2D、CUT2DD)

回転が含まれているフレームがプログラムされている場合、CUT2D または CUT2DD では、工具径補正が行われる平面(補正平面)は同時に回転されません。工具径補正は、非回転加工平面を基準にして考慮されます(G17、G18、G19)。工具長補正は補正平面に対して機能します。

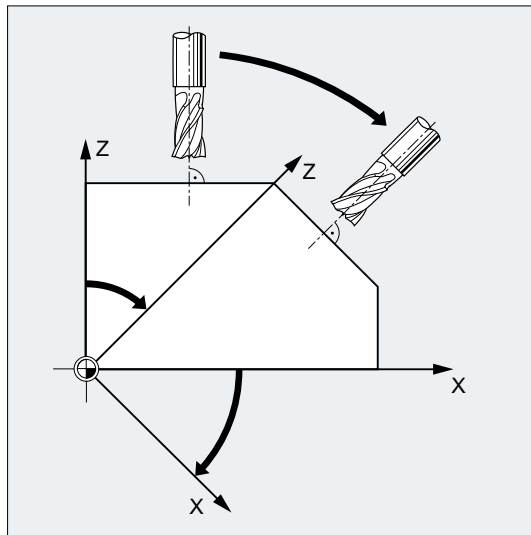


傾斜面を加工する場合、工具オフセットを「オリエンテーション可能な工具の工具長補正」の機能に基づいて正しく設定するか計算する必要があります。

補正平面の回転ありの 2½次元工具径補正(CUT2DF、CUT2DFD)

回転が含まれているフレームがプログラムされている場合、CUT2DF または CUT2DFD では、工具径補正が行われる平面(補正平面)も**回転**されます。工具径補正は、**回転**加工平面を基準にして考慮されます(G17、G18、G19)。ただし、工具長補正はまだ**非回転**加工平面に対して機能します。

必要条件:機械で、工具オリエンテーションが回転される加工平面に直交して調整可能であり、加工のために設定可能であること。

**注記**

工具長補正は、回転しない作業平面に対して、そのまま有効です。

詳しくは、『工具』機能マニュアルを参照してください。

3.10.7 工具径補正の抑制(CUTCONON、CUTCONOF)

「工具径補正の抑制」機能を使用して、多くのブロックの工具径補正をマスクします。これに対し、前のブロックの工具径補正で設定された、プログラム指令工具中心軌跡と実際に移動した工具中心軌跡との差は、補正として保持されます。反転点での直線フライス加工のときに複数の移動ブロックが必要ですが、工具径補正(方法に従い)生成した輪郭が不要であるときは、この方式を使用すると、役に立つ場合があります。これは、工具径補正のタイプ(2½次元、3次元正面削り、3次元外周削り)とは無関係に使用できます。

3.10 工具径補正

構文

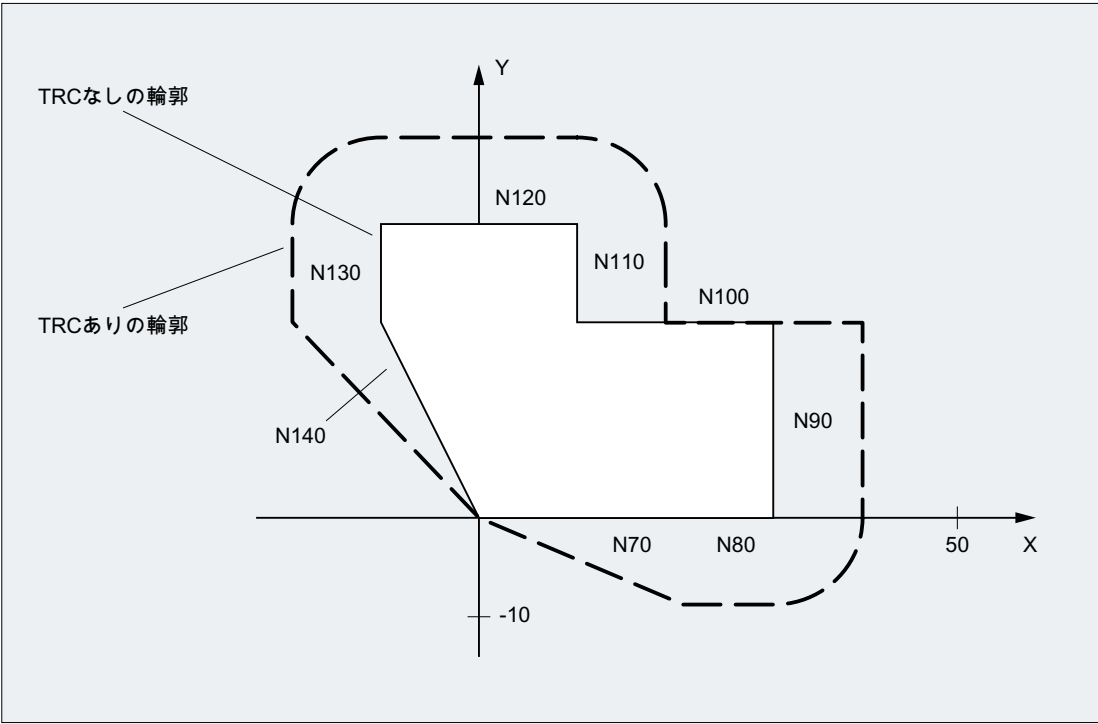
CUTCONON

CUTCONOF

意味

CUTCONON:	「工具径補正の抑制」機能を有効にする命令
CUTCONOF:	「工具径補正の抑制」機能を解除する命令

例



プログラムコード	コメント
N10	; 工具 d1 の定義
N20 \$TC_DP1[1,1] = 110	; タイプ
N30 \$TC_DP6[1,1]= 10.	; 半径
N40	
N50 X0 Y0 Z0 G1 G17 T1 D1 F10000	
N60	
N70 X20 G42 NORM	
N80 X30	

プログラムコード	コメント
N90 Y20	
N100 X10 CUTCONON	; 補正マスクを適用します。
N110 Y30 KONT	; 必要に応じて、補正マスクの解除時にバイパス円弧を挿入します。
N120 X-10 CUTCONOF	
N130 Y20 NORM	; TRC 解除時にはバイパス円弧がありません。
N140 X0 Y0 G40	
N150 M30	

詳細情報

工具径補正は、通常は補正マスクの前に有効で、補正マスクを再度解除するときも、まだ有効です。CUTCONON の前の最後の移動ブロックでは、ブロック終点のオフセット点へアプローチします。後続のすべてのブロックは、オフセットマスクが有効となり、オフセットなしで移動します。ただし、これらのブロックは、最後のオフセットブロックの終点からそのオフセット点へのベクトルによりオフセットされます。これらのブロックでは、あらゆるタイプの補間(直線、円弧、多項式)ができます。

補正マスクの解除ブロック、つまり、CUTCONOF を含むブロックは、通常どおりに補正されます。これは、起点のオフセット点で開始されます。1つの直線ブロックが、前のブロック(つまり、動作中の CUTCONON を含む、最後のプログラム指令移動ブロック)の終点とこの点の間に挿入されます。

円弧平面が補正平面に垂直な円弧ブロック(垂直円弧)は、CUTCONON のプログラム指令ブロックと同様に扱われます。このオフセットマスクの自動的な適用は、補正平面での移動動作を含み、かつ、円弧ではない最初の移動ブロックで自動的にキャンセルされます。この意味で、垂直円弧が発生する可能性があるのは、外周削りのときのみです。

3.10.8 当該の工具の刃先位置を持つ工具

当該の工具のコントロールポイントを持つ工具(旋削工具と研削工具-工具タイプ 400 ~-599、「磨耗の符号評価」の章を参照してください)の場合、G40 から G41/G42 へ、またはその逆の変更は工具交換として扱われます。このため、座標変換が有効な(TRANSMIT など)場合は、先読み停止がおこなわれ(解読停止)、これにより、目的の輪郭部分からの誤差が発生する可能性があります。

下記のオリジナルの機能が変化します。

1. TRANSMIT による先読み停止
2. KONT によるアプローチと後退の交点計算

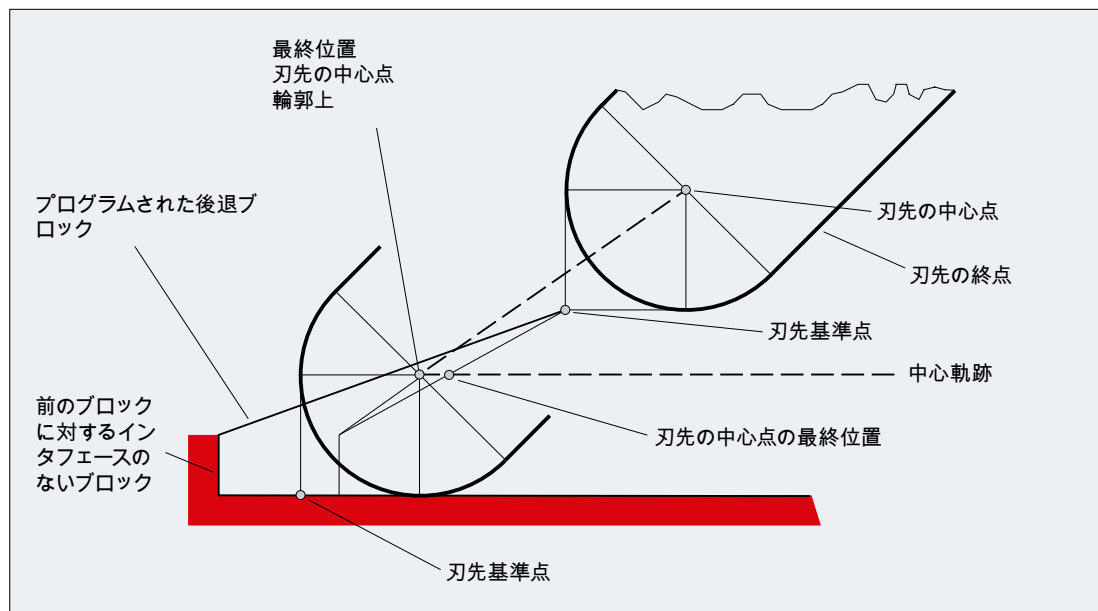
3. 動作中の工具径補正での工具交換
4. 座標変換時の、可変工具オリエンテーションでの工具径補正

詳細情報

オリジナルの機能が、次のように変更されています。

- G40 から G41/G42 へ、およびその逆の変更が、工具交換として扱われなくなります。したがって、TRANSMIT で先読み停止は発生しなくなります。
- ブロック始点とブロック終点の工具刃先中心点間の直線を使用して、アプローチブロックと後退ブロックの交点が計算されます。工具刃先基準点と工具刃先中心点の差分は、この移動に重畳されます。

KONT によるアプローチと後退(工具は輪郭点の周囲を移動します; 前述の「輪郭アプローチと後退」の章を参照してください)時に、アプローチと後退移動の直線部分のブロックで重畳がおこなわれます。したがって、工具形状の条件は、当該のコントロールポイントの有無にかかわらず、同じです。前の動作からの誤差が発生するのは、アプローチまたは後退ブロックが、隣接する移動ブロックと交わらないような、比較的稀な状況でのみです。以下の図を参照してください。



- 円弧ブロック、および分母次数が 4 を超える多項式を含む移動ブロックでは、工具刃先中心点と工具刃先基準点の間隔が変わる場合は、工具径補正が動作中に工具を交換することはできません。その他のタイプの補間を使用すると、座標変換が有効(TRANSMIT など)なときの工具交換が可能となります。
- 可変工具オリエンテーションでの工具径補正の場合は、工具刃先基準点から工具刃先中心点への座標変換は、単純なゼロオフセットを使用しておこなうことができなくなります。したがって、当該の工具のコントロールポイントをもつ工具は、3 次元外周削りには使用できません(アラームが発生します)。

注記

正面削りの場合は、この運転に使用できるのが、当該の工具のコントロールポイントなしで定義した工具タイプのみであるため、この条件は無意味です。(区別して認識されていないタイプの工具は、指定された半径を持つボールエンドミルとして扱われます。コントロールポイントのパラメータは無視されます)

3.11 軌跡動作

3.11.1 イグザクトストップ(G60、G9、G601、G602、G603)

イグザクトストップ移動モードでは、移動動作に関与するすべての軌跡軸と付加軸が連続的には移動せず、各ブロックの終点で減速して停止状態になります。

イグザクトストップは、鋭角の外側コーナの加工、または精密な寸法で内側コーナの仕上げが必要な場合に使用します。

3.11 軌跡動作

イグザクトストップは、コーナ点へのアプローチの精度と次のブロックへの遷移のタイミングを指定します。

- 「精密イグザクトストップ」

ブロック切り替えは、移動動作に関与するすべての軸が、「精密イグザクトストップ」の軸別の許容範囲の制限値に達したタイミングでおこなわれます。

「精密イグザクトストップ」は以下を介して設定されます。MD36010

`$MA_STOP_LIMIT_FINE[<Axis>]`
- 「汎用イグザクトストップ」

ブロック切り替えは、移動動作に関与するすべての軸が、「汎用イグザクトストップ」の軸別の許容範囲の制限値に達したタイミングでおこなわれます。

「汎用イグザクトストップ」は以下を介して設定されます。MD36000

`$MA_STOP_LIMIT_COARSE[<Axis>]`
- 「補間終了」

ブロック切り替えは、移動動作に関与するすべての軸で、コントローラが、設定されたゼロ速度の計算を完了したタイミングでおこなわれます。関与する軸の実位置も追従誤差も考慮されません。

構文

G60...
 G9...
 G601/G602/G603、など

意味

G60:	モーダルイグザクトストップの適用命令
G9:	ノンモーダルイグザクトストップの適用命令
G601:	イグザクトストップ条件「精密イグザクトストップ」の適用命令
G602:	イグザクトストップ条件「汎用イグザクトストップ」の適用命令
G603:	イグザクトストップ条件「補間終了」の適用命令

注記

イグザクトストップ条件(G601/G602/G603)の起動命令が有効となるのは、G60 または G9 が有効な場合のみです。

例

プログラムコード	コメント
N5 G602	; 「汎用イグザクトストップ」条件を選択します。
N10 G0 G60 Z...	; イグザクトストップがモーダルに有効です。
N20 X...Z...	; G60 は引き続き有効です。
...	
N50 G1 G601	; 「精密イグザクトストップ」条件を選択します。
N80 G64 Z...	; 連続軌跡モードへ切り替えます。
...	
N100 G0 G9	; イグザクトストップはこのブロックでのみ動作します。
N110 ...	; 連続軌跡モードが再度有効になります。

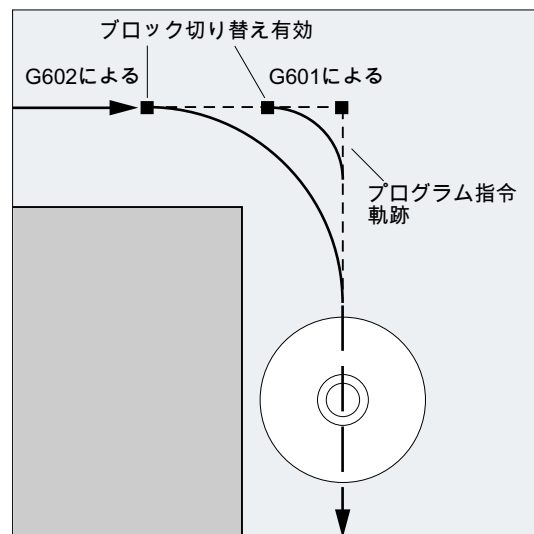
詳細情報

G60、G9

G9 は実行中のブロックで、G60 は実行中のブロックとすべての後続ブロックで、イグザクトストップをおこないます。

連続軌跡モード命令 G64 または G641～G645 を使用して、G60 を解除します。

G601、G602



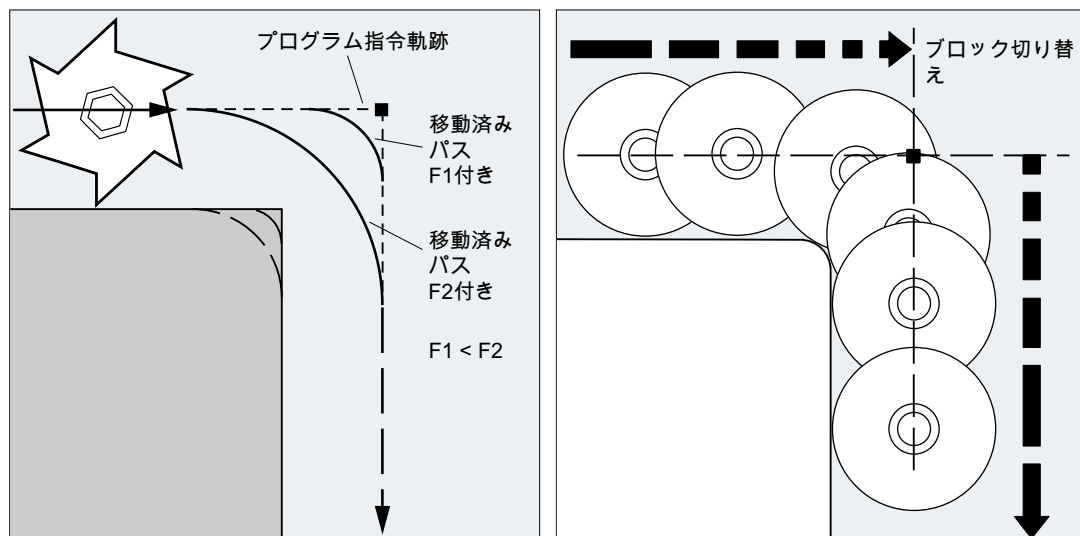
移動が減速し、コーナ点で短時間停止します。

注記

イグザクトストップ条件の制限値を、必要以上に小さく設定しないでください。制限値が小さくなるほど、目標位置への位置決めとアプローチにかかる時間が長くなります。

G603

ブロック切り替えは、制御装置が、関連軸に設定されたゼロ速度の計算を完了すると開始されます。この時点で、軸のダイナミック応答と軌跡速度に応じて、フィードバック値が、比例係数だけ遅れた値となります。これで、ワークコーナが丸くなる場合があります。

**イグザクトストップ設定条件**

G0、および第 1G 機能グループのその他の命令は、プログラム指令したイグザクトストップ条件に反して、設定条件が自動的に使用されるよう、チャネル別に設定できます(工作機械メーカーの仕様書を参照してください)。

3.11.2 連続軌跡モード(G64、G641、G642、G643、G644、G645、ADIS、ADISPOS)

連続軌跡モードでは、ブロック終点の(ブロック切り替えのための)軌跡速度は、イグザクトストップ条件を満たすことのできるレベルまで減速しません。このモードの実際の目的は、ブロック切り替え点で軌跡軸が急激に減速しないようにして、プログラムが次のブロックに移動するときの軸速度を、できるだけ一定に保つことです。この目的を果たすために、連続軌跡モードの選択時に「Look-head」機能も有効です。

スムージングありの連続軌跡モードを使用すると、接線方向の成形や、プログラム指令輪郭の部分的な変化による角度のあるブロック遷移部、または両方のスムージングが容易になります。

連続軌跡モード:

- 輪郭を丸み付けします。
- イグザクトストップ条件を満たすために必要な減速処理と加速処理が無くなるため、加工時間を短縮します。
- さらに速度変動がなくなり、切削条件が向上します。

連続軌跡モードは、次の場合は適していません。

- 輪郭をできるだけ速く移動する必要がある (早送りなど)。
- 連続輪郭を得るために、輪郭精度に、プログラム指令輪郭から指定の許容範囲内の誤差があってもよい。

連続軌跡モードは、次の場合に適しません。

- 輪郭を正確に移動する必要がある。
- 一定速度が絶対に必要である。

注記

連続軌跡モードは、次の理由などで、自動的に先読み停止をおこなうブロックでは中断されます。

- 特定の機械状態データ(\$A...)へのアクセス
- 補助機能出力

構文

```
G64 ...
G641 ADIS=...
G641 ADISPOS=...
G642 ...
G643 ...
G644 ...
G645 ...
```

意味

G64:	過負荷係数に従って減速する連続軌跡モード
G641:	距離条件に従ってスムージングをおこなう連続軌跡モード
ADIS=...:	軌跡機能 G1、G2、G3 などの、G641 による距離条件
ADISPOS=...:	早送り G0 の、G641 による距離条件

3.11 軌跡動作

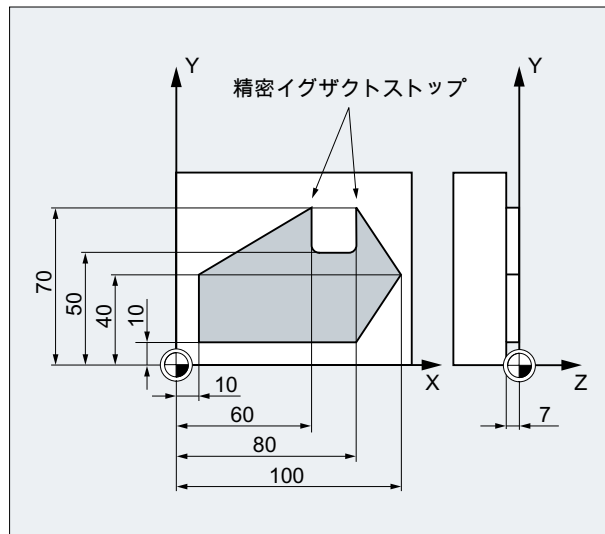
	距離条件(= 丸み付き隙間) ADIS または ADISPOS はそれぞれ、丸み付けブロックがそのブロック終了までに進むことができる最大距離、またはブロック終了後から丸み付けブロックの終了までの距離を示します。 注: ADIS/ADISPOS をプログラム指令していない場合は、「ゼロ」値が適用されるため、移動動作は G64 に対応します。丸み付き隙間は、移動距離が短くなるように、自動的に小さくなります(最大 36%)。
G642:	定義した許容範囲内のスムージングをおこなう連続軌跡モード このモードの場合、通常の場合では、最大許容軌跡誤差内でスムージングがおこなわれます。ただし、これらの軸別の許容範囲の代わりに、最大輪郭誤差(輪郭の許容範囲)または工具オリエンテーションの最大角度誤差(旋回の許容範囲)を監視するように設定できます。 注: 輪郭と旋回の許容範囲を含む拡張は、「多項式補間」オプションを備えたシステムのみでサポートされています。
G643:	定義した許容範囲内のスムージングをおこなう連続軌跡モード(ブロック内部) G643 は、個別の丸み付けブロックの生成には使用されないという点で、G642 と異なります。代わりに、軸別にブロック内部の丸み付け移動が挿入されます。丸み付き隙間は軸毎に変えることができます。
G644:	最大ダイナミック応答でスムージングをおこなう連続軌跡モード 注: G644 は、キネマティックトランスフォーメーションが動作中には、使用できません。システムは内部で G642 に切り替えます。
G645:	スムージング、および定義許容範囲内で接線方向のブロック遷移をおこなう連続軌跡モード G645 は、コーナに対して G642 と同じ働きがあります。G645 を使用すると、オリジナルの輪郭の曲率が少なくとも 1 つの軸で不連続変化を示す場合に、丸み付けブロックは、接線方向のブロック遷移部でのみ生成されます。

注記

丸み付けは、スムージングの代わりに使用できません(RND)。ユーザーは、丸み付け領域内の輪郭の形状に関して、推測に基づいて運転しないでください。丸み付けのタイプは、工具軌跡速度などの動的な条件により変わる場合があります。したがって、輪郭の丸み付けが実用的なのは、ADIS 値が小さい場合のみです。コーナを定義した輪郭で移動する場合は、RND を使用してください。

注記

G641、G642、G643、G644、または G645 で開始された丸み付け移動が中断された場合は、以降の再位置決め(REPOS)は中断点ではなく、(REPOS モードに対応する) オリジナルの移動ブロックの始点または終点を使用しておこなわれます。

例

溝の 2 つの外側コーナへ正確にアプローチします。それ以外の加工は、加工を連続軌跡モードでおこないます。

プログラムコード	コメント
N05DIAMOF	; 半径指定寸法
N10 G17 T1 G41 G0 X10 Y10 Z2 S300 M3	; 開始位置へアプローチし、主軸を起動して軌跡補正を有効にします。
N20 G1 Z-7 F8000	; 工具を送ります。
N30 G641 ADIS=0.5	; 輪郭遷移を滑らかにします。
N40 Y40	
N50 X60 Y70 G60 G601	; 精密イグザクトストップで正確な位置へアプローチします。

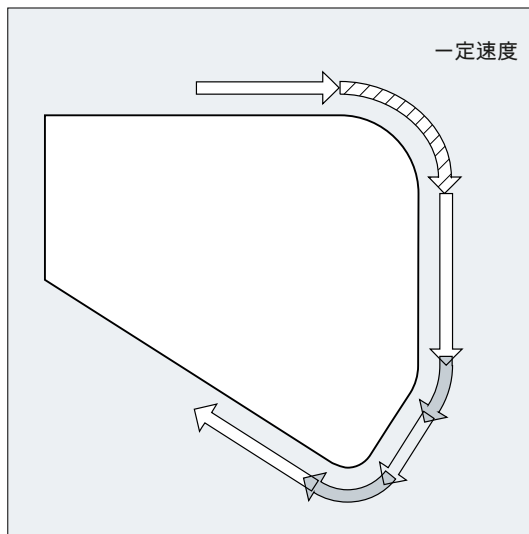
3.11 軌跡動作

プログラムコード	コメント
N60 Y50	
N70 X80	
N80 Y70	
N90 G641 ADIS=0.5 X100 Y40	; 輪郭遷移を滑らかにします。
N100 X80 Y10	
N110 X10	
N120 G40 G0 X-20	; 軌跡補正を解除します。
N130 Z10 M30	; 工具の後退、プログラム終了。

詳細情報

連続軌跡モード G64

連続軌跡モードでは、工具は、接線方向の輪郭遷移で、全体にわたりできるだけ一定の軌跡速度で移動します(ブロック境界で減速はありません)。LookAhead 減速は、イグザクトストップで各コーナとブロックの手前で適用されます。



コーナ部も、一定速度で移動します。輪郭誤差を最小限にするため、速度は、加減速制限と過負荷係数に従って減速します。

注記

輪郭遷移のスムージングの程度は、送り速度と過負荷係数に応じて変わります。過負荷係数は、MD32310 \$MA_MAX_ACCEL_OVL_FACTOR で設定できます。

MD20490 \$MC_IGNORE_OVL_FACTOR_FOR_ADIS を設定すると、設定した過負荷係数にかかわらず、ブロック遷移部が常に丸み付けされます。

軌跡移動中に不必要に停止しないよう、次の点に注意してください(レリーフカット)。

- 補助機能が、移動の終了後または次の移動の前に有効になり、連続軌跡モードが中断されます(例外:高速補助機能)。
- 位置決め軸は常に、イグザクトストップ仕様、および精密位置決め範囲に従って(G601と同様に)移動します。NCブロックが位置決め軸を待つ場合は、連続軌跡モードはその軌跡軸で中断します。

ただし、コメントのみを含む中間ブロック、計算ブロック、またはサブプログラム呼び出しは、連続軌跡モードには影響しません。

注記

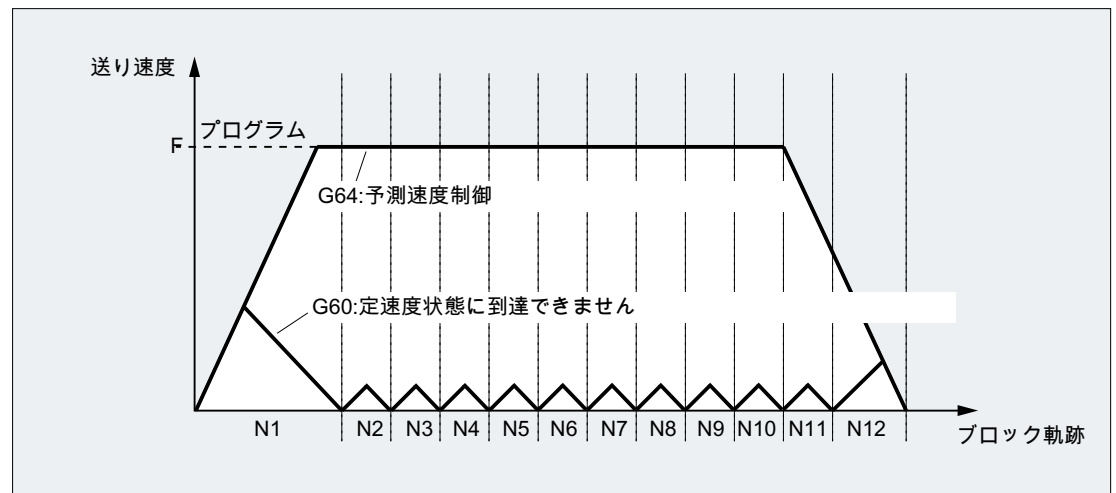
FGROUPに含まれない軌跡軸がある場合は、ブロック遷移部で、FGROUPに含まれない軸に速度の不連続変化がよく発生します。コントローラは、この速度の変化を、MD32300 \$MA_MAX_AX_ACCEL と MD32310 \$MA_MAX_ACCEL_OVL_FACTOR で設定した許容値に制限します。この減速動作は、スムージング機能を使用して軌跡軸間の特定位置の相互関係を滑らかにすることで回避できます。

LookAhead 予測速度制御

連続軌跡モードでは、コントローラが、事前に複数の NC ブロックの速度制御を自動的に特定します。これにより、ほとんどの接線方向の遷移で、複数のブロックにわたり加速と減速が可能となります。

LookAhead は特に、移動軌跡が短く、軌跡送り速度が高速の移動処理の加工に適しています。

LookAhead 計算に含まれる NC ブロックの数はマシンデータで定義できます。



3.11 軌跡動作

距離条件に応じたスムージングをおこなう連続軌跡モード(G641)

G641 を使用すると、コントローラが輪郭遷移部で遷移要素を挿入します。スムージング隙間 ADIS (または G0 の場合の ADISPOS)は、コーナをスムージングできる最大範囲を指定します。このスムージング隙間内で、制御装置は設定された軌跡を無視して動的に最適化された軌跡と置き換えることができます。

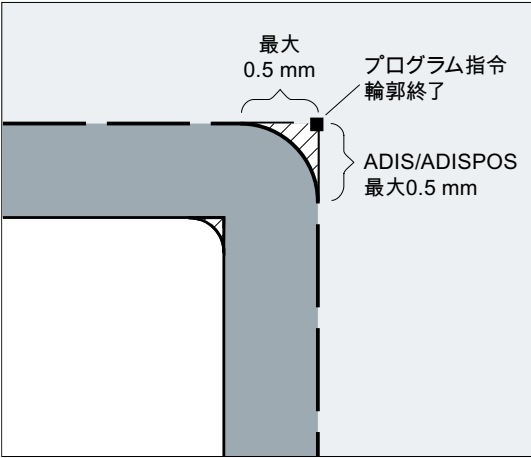
制限事項:すべての軸に対して使用できる ADIS 値は 1 つだけです。

G641 の動作は、RNDM の動作に似ていますが、作業平面の軸のみに限定されません。

G64 と同様に、G641 は LookAhead の予測速度制御で動作します。大きな曲率のスムージングブロックへは、減速してアプローチします。

例:

プログラムコード	コメント
N10 G641 ADIS=0.5 G1 X...Y...	; スムージングブロックは、プログラム指令ブロック終点の前に 0.5 mm 以下で開始し、ブロック終点の後に 0.5 mm 以下で終了します。この設定は、そのままモダルです。



注記

このスムージングは、定義されたスムージング機能(RND、RNDM、ASPLINE、BSPLINE、CSPLINE)で代替えはできません、このため定義されたスムージングを代わりに使用しないでください。

G642 によるスムージングの軸精度

G642 を使用すると、スムージングは定義した ADIS の範囲内でおこなわれるのではなく、MD33100 \$MA_COMPRESS_POS_TOL で定義した軸の許容範囲が適用されます。スムージング隙間は、すべての軸の中の最短のスムージング軌跡により特定されます。この値は、スムージングブロックの生成時に考慮されます。

G643 によるブロック内部のスムージング

G643 によるスムージングの場合の正確な輪郭からの最大誤差は、マシンデータ MD33100 \$MA_COMPRESS_POS_TOL を使用して軸ごとに定義します。

個別のスムージングブロックの生成では G643 を使用しませんが、軸別のブロック内部のスムージング移動が挿入されます。G643 の場合は、軸ごとに異なるスムージング隙間を使用できます。

G642/G643 による輪郭と旋回の許容範囲のあるスムージング

MD20480 \$MC_SMOOTHING_MODE を使用して、G642 と G643 によるスムージングを設定できます。こうすると、軸別の許容範囲の代わりに、輪郭の許容範囲、および旋回の許容範囲を適用できます。

輪郭の許容範囲と旋回の許容範囲は、次のようにチャンネル別のセッティングデータで設定します。

SD42465 \$SC_SMOOTH_CONTUR_TOL (最大輪郭誤差)

SD42466 \$SC_SMOOTH_ORI_TOL (工具オリエンテーションの最大角度誤差)

セッティングデータは、NC プログラムでプログラム指令できます。つまり、ブロック遷移毎に異なるデータを使用できます。輪郭の許容範囲と工具オリエンテーションの許容範囲の指定が大きく異なった場合、G643 のみ有効です。

注記

輪郭と旋回の許容範囲を含む拡張は、「多項式補間」オプションを備えたシステムのみでサポートされています。

注記

旋回の許容範囲内のスムージングに対しては、方向座標変換を有効にしてください。

G644 の可能な最大ダイナミック応答によるスムージング

最大ダイナミック応答によるスムージングは、MD20480 \$MC_SMOOTHING_MODE の 1000 の位に設定されます。

値	意味
0	最大軸誤差の指定: MD33100 \$MA_COMPRESS_POS_TOL
1	プログラミングによる最大スムージング軌跡の指定: ADIS=...または ADISPOS=...

3.11 軌跡動作

値	意味
2	スムージング範囲で発生する各軸の最大振動数の指定: MD32440 \$MA_LOOKAH_FREQUENCY スムージング移動時に振動数が指定された最大数を超えないよう、スムージング範囲が定義されます。
3	G644 によるスムージング時は、許容範囲もスムージング隙間も監視されません。 各軸は、最大ダイナミック応答によりコーナの周りを移動します。 SOFT を使用すると、各軸の最大加減速度と最大加々速度の両方が維持されます。 BRISK 命令を使用すると、加々速度は制限されずに、各軸の最大加減速度で移動します。

G645 による接線方向のブロック遷移部のスムージング

G645 を使用してスムージング移動を定義し、関与するすべての軸の加減速の滑らかさ(不連続変化なし)を維持し、元の輪郭からのパラメータ設定最大誤差(MD33120 \$MA_PATH_TRANS_POS_TOL)を超えないようにします。

角度がある接線方向のブロック遷移部の場合は、スムージング動作が G642 と同じになります。

中間スムージングブロックなし

中間スムージングブロックは、次の場合は挿入されません。

- 軸が 2 つのブロック間で停止する。
これは、次の場合に発生します。
 - 次のブロックに、移動の前の補助機能出力が含まれるとき。
 - 以下のブロックに、軌跡移動が含まれないとき。
 - 位置決め軸であった軸が、初めて次のブロックの軌跡軸として移動する。
 - 軌跡軸であった軸が、初めて次のブロックの位置決め軸として移動する。
 - 前のブロックにジオメトリ軸の移動があり、次のブロックでは移動がない。
 - 次のブロックにジオメトリ軸の移動があり、前のブロックでは移動がない。
 - タッピングの前に、次のブロックに準備機能として G33 を使用して、前のブロックでは使用していない。
 - BRISK と SOFT の間に変化がある。
 - 座標変換に関与するすべての軸が、軌跡移動に割り当てられていない(揺動軸、位置決め軸など)。
- スムージングブロックが、パートプログラムの実行速度を低下させる。
これは、次の場合に発生します。
 - 2 つのブロックの間が非常に短い。
各ブロックには少なくとも 1 回の補間クロックサイクルが必要なので、中間ブロックが追加されると加工時間は 2 倍になります。
 - ブロック遷移 G64 (スムージングのない連続軌跡モード)が、速度を低下させずに移動できた場合。
スムージングにより加工時間が増加します。これは、許容過負荷係数(MD32310 \$MA_MAX_ACCEL_OVL_FACTOR)の値が、ブロック遷移部が丸み付けされるかどうかに影響することを意味します。過負荷係数は、G641/G642 によるスムージングの場合にのみ考慮されます。過負荷係数は、G643 によるスムージングの場合には効果がありません(この特性は、MD20490 \$MC_IGNORE_OVL_FACTOR_FOR_ADIS を TRUE に設定することで、G641 と G642 に対しても設定できます)。

3.12 座標変換(フレーム)

- スムージングがパラメータ設定されない。
これは、次の場合に発生します。
 - G0 ブロックの G641 に対して ADISPOS=0 (初期設定)の場合
 - 非 G0 ブロックの G641 に対して ADISPOS=0 (初期設定)の場合
 - G641 の場合、G0 から非 G0 または非 G0 から G0 への移行では、ADISPOS および ADIS からの下位値が適用されます。
 - G642/G643 の場合に、すべての軸別の許容範囲がゼロのとき。
- ブロックに移動動作が含まれない(ゼロブロック)。
これは、次の場合に発生します。
 - シンクロナイズドアクションが有効であるとき。
通常、インタプリタによりゼロブロックは削除されます。ただし、シンクロナイズドアクションが有効の場合は、このゼロブロックが含まれ、実行もされます。このとき、動作中のプログラム指令に対応して、イグザクトストップが開始されます。これにより、シンクロナイズドアクションを切り替えることもできます。
 - プログラムのジャンプによりゼロブロックが生成されるとき。

早送り G0 の連続軌跡モード

早送り移動の場合も、リストされた機能 G60/G9 または G64 - G641～G645 のそれぞれ - のどれかを指定してください。指定しない場合は、マシンデータの初期設定が使用されます。

3.12 座標変換(フレーム)

3.12.1 フレーム

フレーム

フレームは、1 つの直交座標系を別の直交座標系に変換する内蔵された計算規則です。

基本フレーム(基本オフセット)

基本フレームは、基本座標系(BCS)から基本ゼロオフセットシステム(BZS)への座標変換を記述し、設定可能フレームと同じ働きがあります。

基本座標系(BCS) (ページ 46)を参照してください。

設定可能フレーム

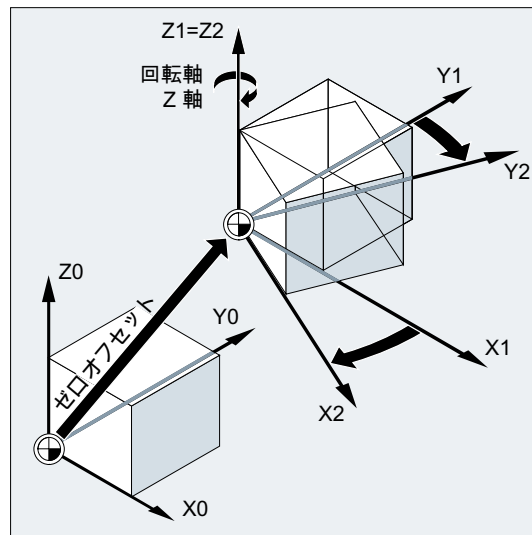
設定可能フレームは、任意の NC プログラム内から G54～G57 と G505～G599 の各命令で呼び出せる、設定可能なゼロオフセットです。オフセット値はユーザーにより事前に設定され、コントローラのゼロオフセットメモリに格納されます。これらの値を使用して、設定可能ゼロオフセットシステム(SZS)を定義します。

参照:

- 設定可能ゼロオフセットシステム(SZS) (ページ 48)
- 設定可能ゼロオフセット(G54～G57、G505～G599、G53、G500、SUPA、G153) (ページ 163)

プログラマブルフレーム

NC プログラム内で、最初を選択されたワーク座標系(または「設定可能ゼロオフセットシステム」)を別の位置に移動したり、必要に応じて、回転、反転、拡大/縮小を行ったりすることが便利な場合、または必要な場合があります。これは、プログラマブルフレームを使用して実現できます。



フレーム命令 (ページ 342)を参照してください。

3.12 座標変換(フレーム)

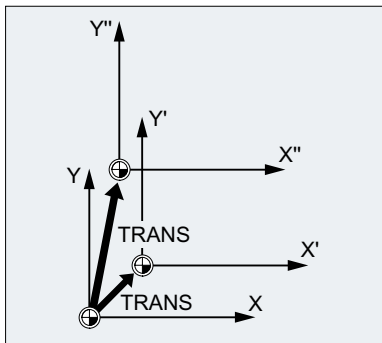
3.12.2 フレーム命令

機能

プログラマブルフレームの命令は、実行中の NC プログラムに適用されます。これらのフレームは、追加要素または代替要素として機能します。

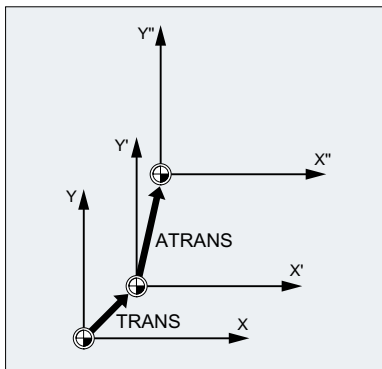
- 代替命令

以前にプログラム指令したフレーム命令をすべて解除します。基準となるのは、呼び出した最後の設定可能ゼロオフセットです(G54～G57、G505～G599)。



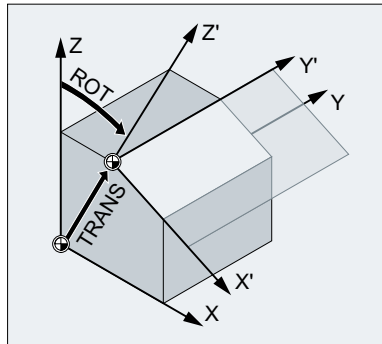
- 追加命令

既存のフレームに追加されます。基準となるのは、現在設定されているワーク原点、またはフレーム命令の最後のプログラム指令ワーク原点です。



用途例

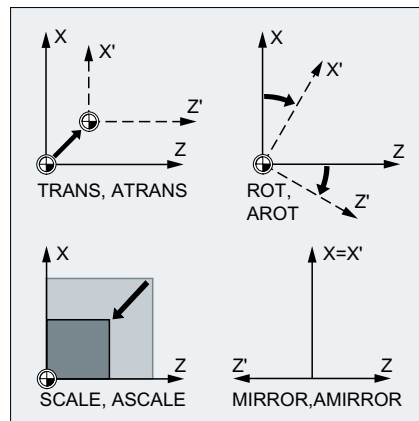
1. ワーク座標系(WCS)の原点を移動します。
2. ワーク座標系(WCS)を、目的の作業平面に平行な向きになるように回転します。



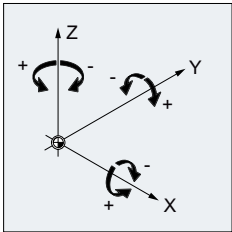
構文

代替命令	追加命令
TRANS X... Y... Z...	ATRANS X... Y... Z...
ROT X... Y... Z...	AROT X... Y... Z...
ROT RPL=...	AROT RPL=...
ROTS/CROTS X...Y...	AROTS X...Y...
SCALE X... Y... Z...	ASCALE X... Y... Z...
MIRROR X0/Y0/Z0	AMIRROR X0/Y0/Z0

意味



3.12 座標変換(フレーム)

TRANS/ATRANS:	指定した 1 つ、または複数のジオメトリ軸方向のワーク座標系オフセット		
ROT/AROT:	以下のようなワーク座標系の回転があります。 - 指定した 1 つ、または複数のジオメトリ軸を中心とした個々の回転の結合による回転 または - 現在の作業平面の回転角度 RPL=...(G17/G18/G19)による回転		
	回転の方向:		
	回転順序:	RPY 表記による:	Z、Y'、X"
		オイラー角による:	Z、X'、Z"
	値の範囲:	回転角度は、次の範囲内でのみ一義的に定義されます。	
		RPY 表記による:	-180 ≤ x ≤ 180
			-90 < y < 90
			-180 ≤ z ≤ 180
		オイラー角による:	0 ≤ x < 180
			-180 ≤ y ≤ 180
	-180 ≤ z ≤ 180		
ROTS/AROTS:	立体角の指定によるワーク座標系の回転 空間の平面の向きは、2 つの立体角を指定して一義的に定義します。したがって、2 つまでの立体角をプログラム指令してください。 ROTS/AROTS X...Y... / Z...X... / Y...Z...		
CROTS:	CROTS は ROTs と同じように機能しますが、CROTS はデータベースの有効フレームを基準としています。		
SCALE/ASCALE:	輪郭のサイズを拡大/縮小する、指定した 1 つ、または複数のジオメトリ軸方向のスケーリング		

MIRROR/AMIRROR:	指定したジオメトリ軸の反転(方向変更)によるワーク座標系のミラーリング	
	値:	自由に選択可能(この場合:"0")

必要条件

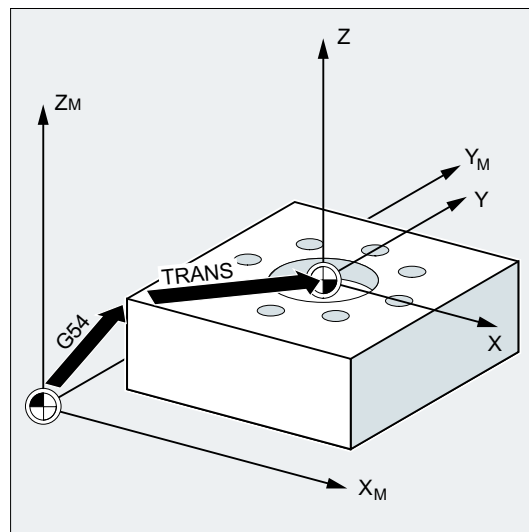
- フレーム命令は、個別の NC ブロックでプログラム指令してください。
- フレーム命令は、個々に使用することも、必要に応じて組み合わせて使用することもできます。
- フレーム命令は、プログラム指令順に実行されます。
- 追加命令は、サブプログラムでしばしば使用されます。サブプログラムを **SAVE** 属性でプログラム指令している場合は、メインプログラムで定義した基本命令がサブプログラム終了後に失われません。

3.12.3 プログラマブルゼロオフセット(TRANS、ATRANS)

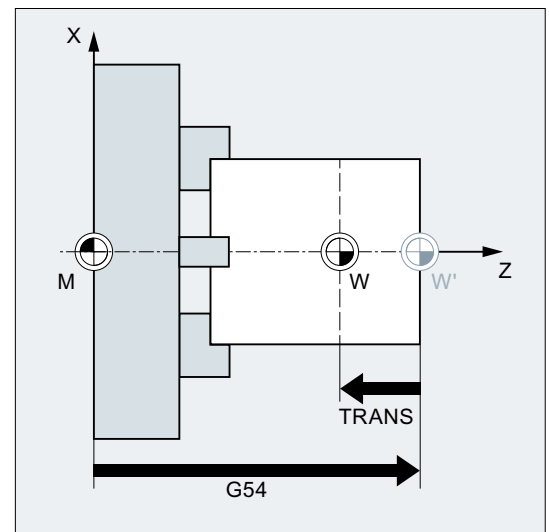
TRANS 命令は、設定可能なゼロオフセット(G54 ... G57、G505 ... G599)で設定された SZS を基準として、WCS をアブソリュートで移動します。

ATRANS 命令は、TRANS で設定された WCS を追加で移動します。

フライス加工:



旋削:



3.12 座標変換(フレーム)

構文

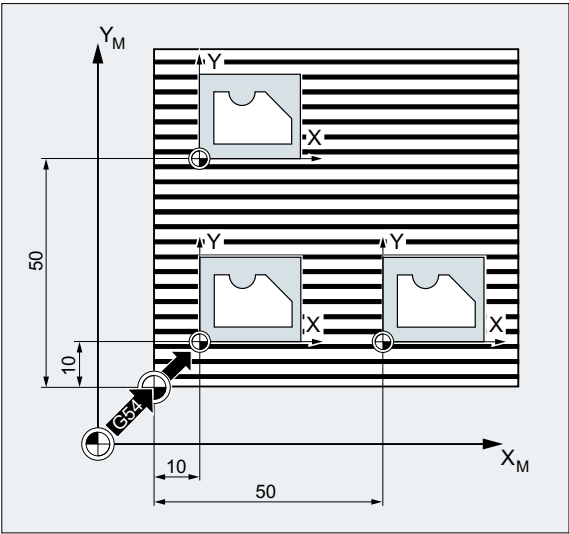
```
TRANS X... Y... Z...
ATRANS X... Y... Z...
```

意味

TRANS:	設定可能なゼロオフセット(G54 ... G57、G505 ... G599)で設定された、ワーク原点(SZS)を基準とする WCS のアブソリュートオフセット	
	単独ブロック指令:	あり
ATRANS:	TRANS で設定されたパラメータ設定したワーク原点を基準とした WCS の追加ゼロオフセット	
	単独ブロック指令:	あり
X...Y...Z...:	指定したジオメトリ軸方向のオフセット値	

例

例 1:フライス加工



このワークでは、図に示す形状が、プログラムの中に繰り返してできます。

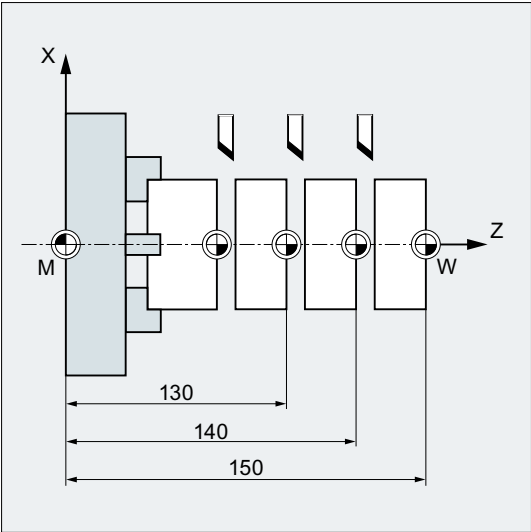
この形状の加工処理はサブプログラムに格納されます。

ゼロオフセットを使用して、それぞれの場合に必要なワーク原点を設定し、その後にサブプログラムを呼び出します。

プログラムコード	コメント
N10 G1 G54	; 作業平面 X/Y、ワーク原点。
N20 G0 X0 Y0 Z2	; 起点にアプローチします。
N30 TRANS X10 Y10	; アブソリュートオフセット
N40 L10	; サブプログラム呼び出し

プログラムコード	コメント
N50 TRANS X50 Y10	; アブソリュートオフセット
N60 L10	; サブプログラム呼び出し
N70 M30	; プログラム終了

例 2 :旋削



プログラムコード	コメント
...	
N10 TRANS X0 Z150	; アブソリュートオフセット
N15 L20	; サブプログラム呼び出し
N20 TRANS X0 Z140 (または ATRANS Z-10)	; アブソリュートオフセット
N25 L20	; サブプログラム呼び出し
N30 TRANS X0 Z130 (または ATRANS Z-10)	; アブソリュートオフセット
N35 L20	; サブプログラム呼び出し
...	

詳細情報

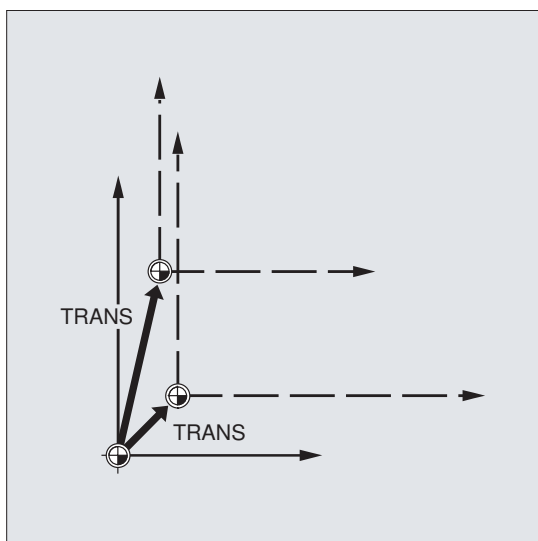
TRANS X... Y... Z...

3.12 座標変換(フレーム)

指定した軸方向にプログラム指令したオフセット値による平行移動(軌跡軸、同期軸、および位置決め軸)。基準となるのは、呼び出した最後の設定可能ゼロオフセットです(G54～G57、G505～G599)。

通知**オリジナルフレームなし**

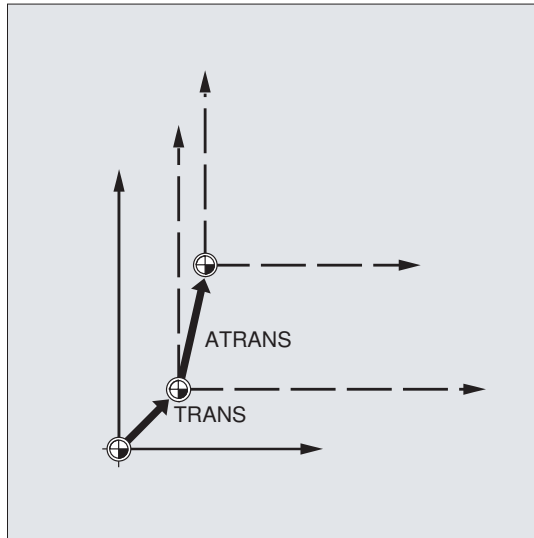
TRANS 命令は、以前に有効にしたプログラマブルフレームのすべてのフレーム成分をリセットします。

**注記**

ATrans を使用すると、オフセットを既存のフレームに追加するようにプログラム指令できます。

ATrans X... Y... Z...

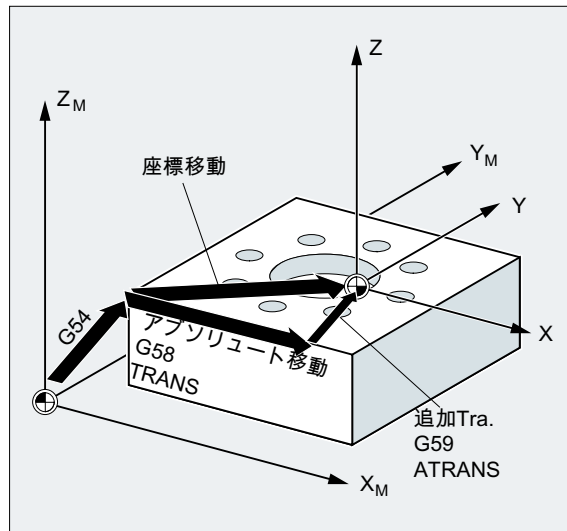
指定軸方向にプログラム指令したオフセット値による平行移動。現在設定されている、または最後のプログラム指令原点を、基準として使用します。



3.12.4 プログラマブルゼロオフセット(G58、G59)

G58 と G59 機能を使用すると、特定の軸で、プログラマブルゼロオフセット(TRANS/ATRANS) (ページ 345)の平行移動成分を代用できます。

- G58:アブソリュート平行移動成分(荒削りオフセット)
- G59:追加平行移動成分(仕上げオフセット)



3.12 座標変換(フレーム)

必要条件

G58 と G59 機能は、仕上げオフセットの設定が
(MD24000 \$MC_FRAME_ADD_COMPONENTS = 1)の場合にのみ使用できます。

構文

```
G58 <axis_1><value_1> ... <axis_3><value_3>
G59 <axis_1><value_1> ... <axis_3><value_3>
```

意味

G58:	G58 を使用すると、指定軸のプログラマブルゼロオフセットの絶対リユート平行移動成分を置き換えますが、プログラム指令追加オフセットは、そのまま有効です。参照先は最後に呼び出された設定可能ゼロオフセットによって提供されます(G54 ... G57, G505 ... G599)。
	単独ブロック指令:あり
G59:	G59 を使用すると、指定軸のプログラマブルゼロオフセットの追加平行移動成分を置き換えますが、プログラム指令絶対オフセットは、そのまま有効です。
	単独ブロック指令:あり
<axis_n>:	チャネルのジオメトリ軸
<value_n>:	指定ジオメトリ軸の方向のオフセット値

例

プログラムコード	コメント
...	
N50 TRANS X10 Y10 Z10	; 絶対リユート平行移動成分 X10 Y10 Z10
N60 ATRANS X5 Y5	; 追加平行移動成分 X5 Y5
	→ サムオフセット:X15 Y15 Z10
N70 G58 X20	; 絶対リユート平行移動成分 X20
	→ サムオフセット X25 Y15 Z10
N80 G59 X10 Y10	; 追加平行移動成分 X10 Y10
	→ サムオフセット X30 Y20 Z10
...	

詳細情報

アブソリュート平行移動成分(荒削りオフセット)は、次の命令で変更します。

- TRANS
- G58
- CTRANS
- CFINE
- \$P_PFRAME[X,TR]

追加平行移動成分(仕上げオフセット)は、次の命令で変更します。

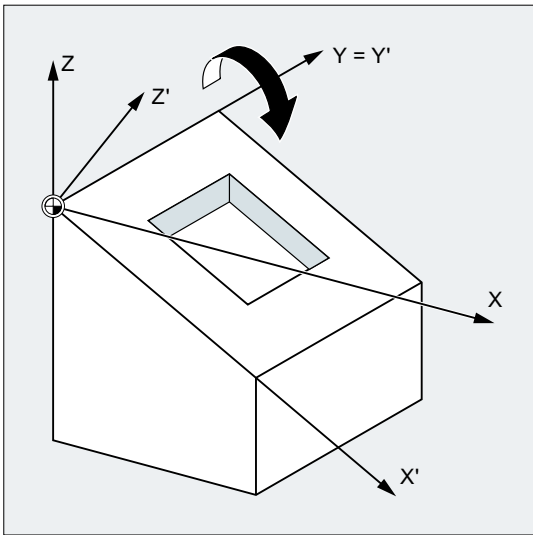
- ATRANS
- G59
- CTRANS
- CFINE
- \$P_PFRAME[X,FI]

例

命令	荒削りオフセット V_C	仕上げオフセット V_F
TRANS X10	$V_C = 10$	変更なし
G58 X10	$V_C = 10$	変更なし
\$P_PFRAME[X,TR] = 10	$V_C = 10$	変更なし
ATrans X10	変更なし	$V_F = V_F + 10$
G59 X10	変更なし	$V_F = 10$
\$P_PFRAME[X,FI] = 10	変更なし	$V_F = 10$
CTrans (X,10)	$V_C = 10$	$V_F = 0$
CTrans ()	$V_C = 0$	$V_F = 0$
CFINE (X,10)	$V_C = 0$	$V_F = 10$

3.12.5 プログラマブル回転(ROT、AROT、RPL)

ROT/AROT 命令により空間でワーク座標系を回転できます。命令はプログラマブルフレーム \$P_PFRAME のみを基準とします。



構文

ROT <1 番目のジオメトリ軸><角度> <2 番目のジオメトリ軸><角度> <3 番目のジオメトリ軸><角度>
ROT RPL=<角度>
AROT <1 番目のジオメトリ軸><角度> <2 番目のジオメトリ軸><角度> <3 番目のジオメトリ軸><角度>
AROT RPL=<角度>

注記

ワーク座標系の回転は、オイラー角によって実行されます。

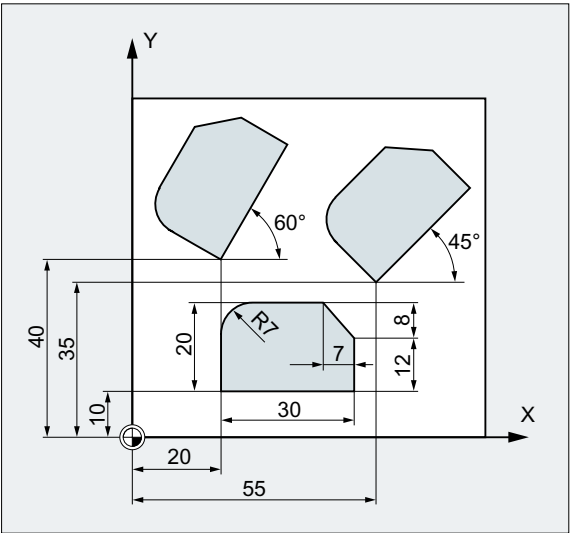
意味

ROT:	アブソリュート回転	
	基準フレーム:	プログラマブルフレーム \$P_PFRAME
	基準点	G54 ... G57、G505 ... G599 で設定した現在のワーク座標系の原点

AROT:	追加回転	
	基準フレーム:	プログラマブルフレーム\$P_PFRAME
	基準点	G54 ... G57、G505 ... G599 で設定した現在のワーク座標系の原点
<n 番目のジオメトリ軸>:	指定された角度でおこなわれる回転の中心となる n 番目のジオメトリ軸の識別子 プログラム指令されていないジオメトリ軸の回転角度には、0°の値が自動的に設定されます。	
RPL:	有効な平面(G17、G18、G19)に垂直なジオメトリ軸を中心として指定角度だけ回転	
	基準フレーム:	プログラマブルフレーム\$P_PFRAME
	基準点	G54 ... G57、G505 ... G599 で設定した現在のワーク座標系の原点
<角度>	角度指定(°単位)	
	値の範囲:	$-360^{\circ} \leq \text{角度} \leq 360^{\circ}$

例

例 1:G17 平面での回転



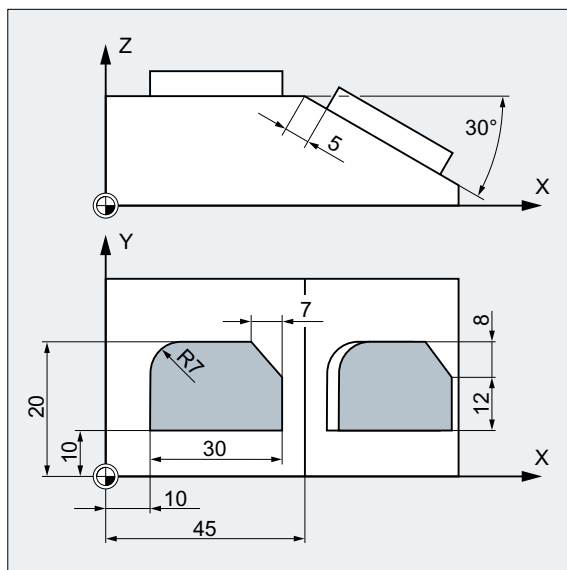
このワークでは、図に示す形状が、プログラムの中に繰り返しでできます。これらの形状は軸に平行に配置されていないため、ゼロオフセットの他に回転をおこなう必要があります。

プログラムコード	コメント
N10 G17 G54	; 作業平面 x/Y、ワーク原点
N20 TRANS X20 Y10	; アブソリュートオフセット

3.12 座標変換(フレーム)

プログラムコード	コメント
N30 L10	; サブプログラム呼び出し
N40 TRANS X55 Y35	; アブソリュートオフセット
N50 AROT RPL=45	; G17 平面に垂直な Z 軸を中心とした 45°の追加回転
N60 L10	; サブプログラム呼び出し
N70 TRANS X20 Y40	; アブソリュートオフセット (以前のオフセットをすべてリセットします)
N80 AROT RPL=60	; G17 平面に垂直な Z 軸を中心とした 60°の追加回転
N90 L10	; サブプログラム呼び出し
N100 G0 X100 Y100	; 後退
N110 M30	; プログラム終了

例 2:Y 軸を中心とした空間の回転



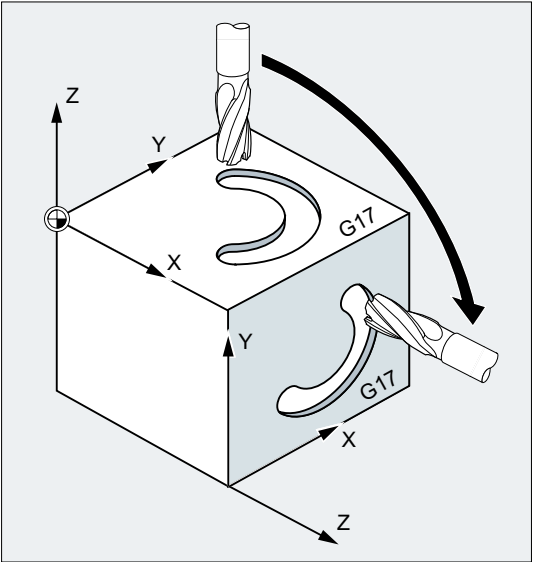
この例では、軸に平行なワーク面と傾斜ワーク面を1回のクランプで加工します。

条件:

工具は、回転した Z 方向の傾斜面に垂直に位置合わせしてください。

プログラムコード	コメント
N10 G17 G54	; 作業平面 X/Y、ワーク原点
N20 TRANS X10 Y10	; アブソリュートオフセット
N30 L10	; サブプログラム呼び出し
N40 ATRANS X35	; 追加オフセット
N50 AROT Y30	; Y 軸を中心とした追加回転
N60 ATRANS X5	; 追加オフセット
N70 L10	; サブプログラム呼び出し
N80 G0 X300 Y100 M30	; 後退、プログラム終了

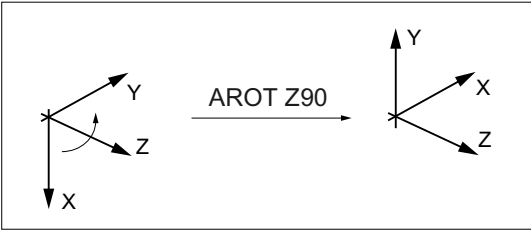
例 3:多面加工



この例では、互いに直交している 2 つのワーク面で、同じ形状をサブプログラムで加工します。右側のワーク面の新しい座標系では、切り込み方向、作業平面、および原点が上部の面と同様に設定されています。したがって、サブプログラムの実行に必要な条件が引き続き適用されます。作業平面 G17、座標平面 X/Y、切り込み方向 Z。

プログラムコード	コメント
N10 G17 G54	; 作業平面 X/Y、ワーク原点
N20 L10	; サブプログラム呼び出し
N30 TRANS X100 Z-100	; WCS のアブソリュートオフセット
N40 AROT Y90	; Y 軸を中心とした WCS の 90° の追加回転

3.12 座標変換(フレーム)

プログラムコード	コメント
N50 AROT Z90	; Z 軸を中心とした WCS の 90° の追加回転
	
N60 L10	; サブプログラム呼び出し
N70 G0 X300 Y100 M30	; 後退、プログラム終了

詳細情報

有効な平面内での回転

RPL=...を使用してプログラミングする場合、WCS は有効な平面に垂直な軸を中心として回転されます。

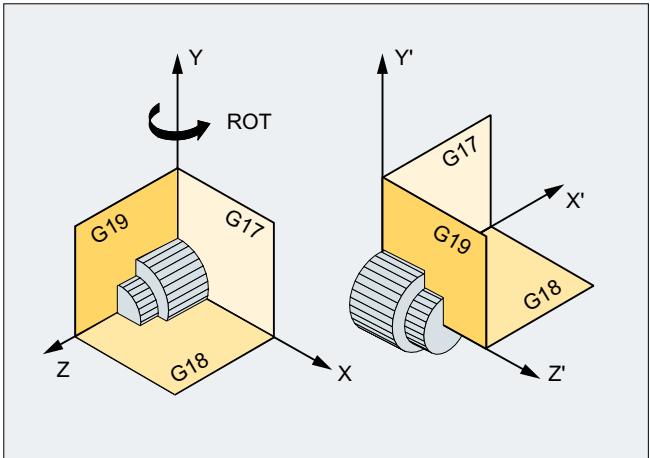


図 3-23 Y 軸を中心とした、または G18 平面での回転

 **警告**

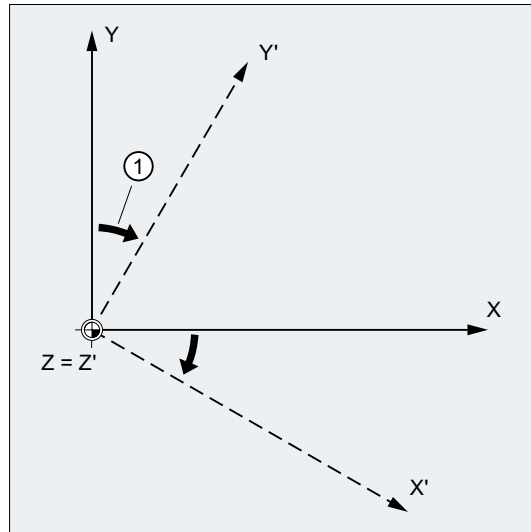
平面の変更

回転後の平面の変更(G17、G18、G19)がプログラム指令されている場合、各軸の現在の回転角度は保持され、新しい平面でも有効になります。そのため、平面の変更前に現在の回転角度を 0 にリセットしておくことを強くお勧めします。

- N100 ROT X0 Y0 Z0 ; 明示的な角度のプログラミング
- N100 ROT ; 自動的な角度のプログラミング

ROT X... Y... Z...によるアブソリュート回転

WCS は、指定した軸を中心として、プログラム指令回転角度だけ回転されます。

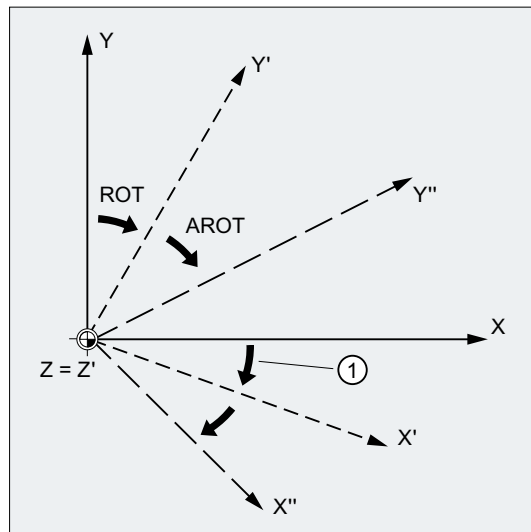


① 回転角度

図 3-24 Z 軸を中心としたアブソリュート回転

AROT X... Y... Z...による追加回転

WCS は、指定した軸を中心として、プログラム指令回転角度だけさらに回転されます。



① 回転角度

図 3-25 Z 軸を中心としたアブソリュート回転と追加回転

作業平面の回転

ROT/AROT を使用した回転時には、作業平面(G17、G18、G19)も回転します。

3.12 座標変換(フレーム)

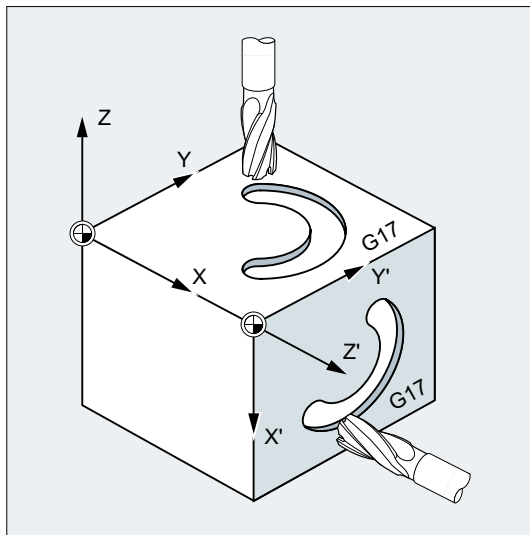
例:作業平面 G17

WCS はワークの最上面に位置付けられます。オフセットと回転を使用して、この座標系をいずれかの側面に移動します。作業平面 G17 も回転します。このようにして、G17 平面での X と Y による移動動作と Z による切り込みをプログラム指令できます。

必要条件:

工具は作業平面に垂直にし、切り込み軸の正方向が工具ベースの方向を指すようにしてください。

CUT2DF を指定すると、回転平面で工具径補正を起動します。



3.12.6 立体角によるプログラマブルフレーム回転(ROTS、AROTS、CROTS)

ワーク座標系の回転は、ROTS、AROTS、および CROTS 命令を使用して立体角で指定できます。立体角は、空間で回転された平面と、まだ回転されていない WCS の主平面との交点によって形成される角度です。

注記

ジオメトリ軸識別子

詳細説明用の例として次のように定義します。

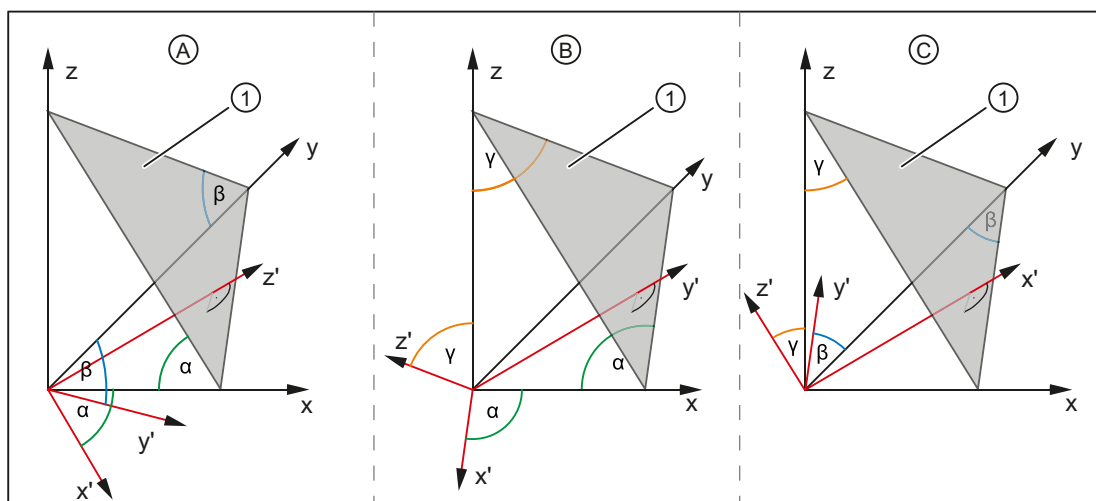
- 1 番目のジオメトリ軸:X
- 2 番目のジオメトリ軸:Y
- 3 番目のジオメトリ軸:Z

次の図に示すように、ROTS α β のプログラミングの結果、WCS の G17 平面は、表示されている傾斜面に平行に配置されます。WCS の原点の位置はそのまま変更されません。

3.12 座標変換(フレーム)

回転した WCS の向きは、1 番目の回転軸がこれによって形成される平面上に、3 番目の軸がオリジナルの座標系に配置されるように定義されます。例では、X'はオリジナルの X/Z 平面にあります。

3.12 座標変換(フレーム)



① 傾斜面

α 、 β 、立体角

Y

A 傾斜面に平行な G17 平面の配置:

- 1 番目の回転
y を中心に x を角度 α まで回転 $\Rightarrow$
傾斜面に平行な x' 軸
- 2 番目の回転
x' を中心に y' を角度 β まで回転 $\Rightarrow$
傾斜面に平行な y' 軸
 $\Rightarrow$ 傾斜面に平行な y' 軸
 $\Rightarrow$ 傾斜面に平行な G17

- B 傾斜面に平行な G18 平面の配置:
- 1 番目の回転
x を中心に z を角度 γ まで回転 $\Rightarrow$
傾斜面に平行な z' 軸
 - 2 番目の回転
z' を中心に x' を角度 α まで回転 $\Rightarrow$
傾斜面に平行な x' 軸
 $\Rightarrow$ 傾斜面に平行な y' 軸
 $\Rightarrow$ 傾斜面に平行な G18
- C 傾斜面に平行な G19 平面の配置:
- 1 番目の回転
z を中心に y を角度 β まで回転 $\Rightarrow$
傾斜面に平行な y' 軸
 - 2 番目の回転
y' を中心に z' を角度 γ まで回転 $\Rightarrow$
傾斜面に平行な z' 軸
 $\Rightarrow$ 傾斜面に平行な x' 軸
 $\Rightarrow$ 傾斜面に平行な G19

構文

必要条件

空間の平面の位置は、2 つの立体角により明確に定義してください。3 番目の立体角を指定すると、平面は「過剰に定義」されます。そのため、3 番目の立体角は指定しないでください。

1 つの立体角のみをプログラム指令する場合、WCS の回転は ROT、AROT と同じになります(「プログラマブル回転(ROT、AROT、RPL) (ページ 352)」の章を参照してください)。

プログラム指令された 2 つの軸により、平面は G17、G18、G19 の平面定義に従って指定されます。これは座標軸の順番(平面の 1 番目の軸/2 番目の軸)または立体角による回転の順番を次のように定義します。

平面	1 番目の軸	2 番目の軸
G17	X	Y
G18	Z	X
G19	Y	Z

3.12 座標変換(フレーム)

G17 平面の配置 ⇒ X と Y の立体角

- 最初の回転:X を Y を中心に角度 α 回転
- 2 番目の回転:Y を X'を中心に角度 β 回転
- 向き:X'はオリジナルの Z/X 平面にあります。

ROTS X< α > Y< β >AROTS X< α > Y< β >CROTS X< α > Y< β >**G18 平面の配置 ⇒ Z と X の立体角**

- 回転:Z を X を中心に角度 γ 回転
- 2 番目の回転:X を Z'を中心に角度 α 回転
- 向き:Z' はオリジナルの Y/Z 平面にあります。

ROTS Z< γ > X< α >AROTS Z< γ > X< α >CROTS Z< γ > X< α >**G19 平面の配置 ⇒ Y と Z の立体角**

- 最初の回転:Y を Z を中心に角度 β 回転
- 2 番目の回転:Z を Y'を中心に角度 γ 回転
- 向き:Y'はオリジナルの X/Z 平面にあります。

ROTS Y< β > Z< γ >AROTS Y< β > Z< γ >CROTS Y< β > Z< γ >

意味

ROTS:	立体角によるアブソリュートフレーム回転、 基準フレーム:プログラマブルフレーム\$P_PFRAME
AROTS:	立体角による追加フレーム回転、 基準フレーム:プログラマブルフレーム\$P_PFRAME
CROTS:	立体角によるアブソリュートフレーム回転、 基準フレーム:プログラマブルフレーム\$P_...
X、Y、Z:	ジオメトリ軸識別子(上述の注:ジオメトリ軸識別子を参照してください)
A、 β 、 γ :	当該のジオメトリ軸に対応する立体角: • $\alpha \rightarrow X$ • $\beta \rightarrow Y$ • $\gamma \rightarrow Z$

3.12.7 プログラマブルスケーリング係数(SCALE、ASCALE)

SCALE/ASCALE を使用すると、すべての軌跡軸、同期軸、および位置決め軸に対して、それぞれの場合に指定した軸の方向へ拡大または縮小のスケーリング係数をプログラム指令できます。これにより、幾何学的に類似した形状、またはさまざまな縮み代をプログラミングで考慮することができます。

構文

```
SCALE X... Y... Z...
ASCALE X... Y... Z...
```

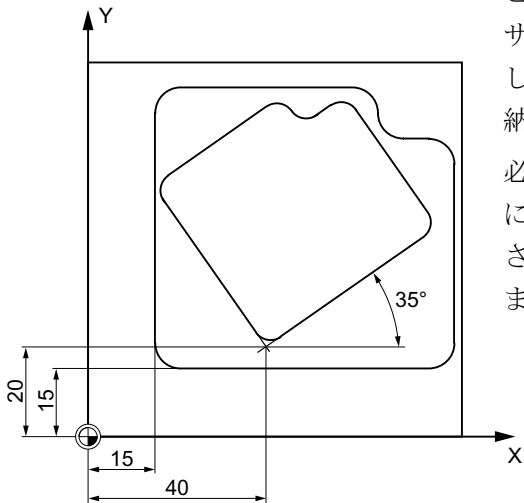
注記

各フレーム命令は、個別の NC ブロックにプログラム指令します。

意味

SCALE:	G54～G57、G505～G599 で設定した現在有効な座標系に対するアプソリュートの拡大/縮小。
ASCALE:	現在有効な設定座標系、またはプログラム指令座標系に対する追加の拡大/縮小
X... Y... Z...:	指定したジオメトリ軸方向のスケーリング係数

例



このワークではポケットが 2 回ありますが、サイズが異なり、お互いに関係をもって回転しています。加工処理はサブプログラムに格納されています。

必要なワーク原点は、ゼロオフセットと回転により設定され、輪郭がスケーリングで縮小され、その後にサブプログラムが呼び出されます。

3.12 座標変換(フレーム)

プログラムコード	コメント
N10 G17 G54	; 作業平面 X/Y、ワーク原点
N20 TRANS X15 Y15	; アブソリュートオフセット
N30 L10	; 大きいポケットを加工
N40 TRANS X40 Y20	; アブソリュートオフセット
N50 AROT RPL=35	; この平面の 35°の回転
N60 ASCALE X0.7 Y0.7	; 小さいポケットのスケーリング係数
N70 L10	; 小さいポケットを加工
N80 G0 X300 Y100 M30	; 後退、プログラム終了

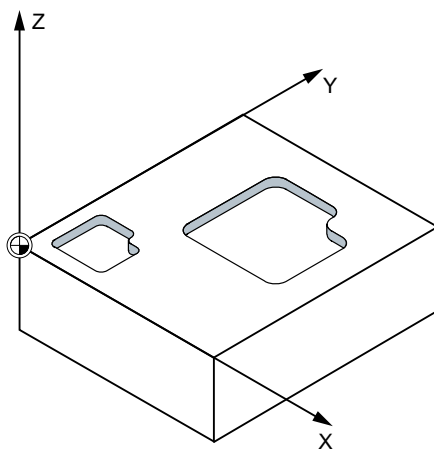
詳細情報

SCALE X... Y... Z...

個々のスケーリング係数を軸毎に指定できます。この係数に従って、形状が縮小または拡大されます。拡大/縮小は、G54～G57、G505～G599 で設定したワーク座標系を基準とします。

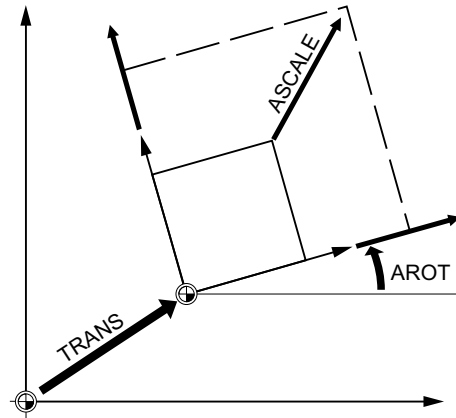
通知**オリジナルフレームなし**

SCALE 命令は、以前に有効にしたプログラマブルフレームのすべてのフレーム成分をリセットします。

**ASCALE X... Y... Z...**

ASCALE 命令を使用してプログラム指令すると、拡大/縮小の変更が既存のフレームに追加されます。この場合、有効な最後のスケーリング係数に新しい係数を掛けた値が使用されます。

現在設定された、または最後のプログラム指令座標系が、拡大/縮小の変更の基準として使用されます。



スケーリングとオフセット

注記

SCALE の後に ATRANS でオフセットをプログラム指令した場合は、オフセット値も拡大/縮小されます。

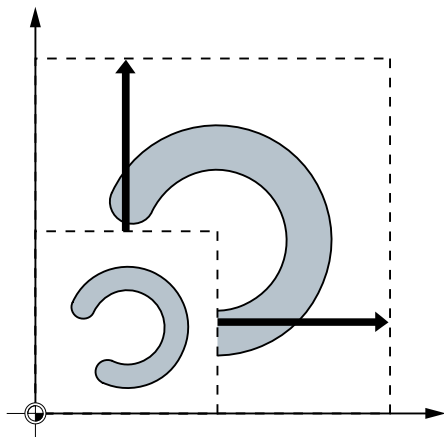
異なるスケーリング係数

通知

干渉の可能性

異なるスケーリング係数の使用時には、さらに注意してください。たとえば、円弧補間を拡大/縮小できるのは、同じ係数を使用する場合のみです。

3.12 座標変換(フレーム)

**注記**

ただし、異なるスケーリング係数を意図的に使用して、歪んだ円弧をプログラム指令することもできます。

3.12.8 プログラマブルミラーリング(MIRROR、AMIRROR)

MIRROR/AMIRROR を使用すると、座標軸上のワーク形状を反転できます。反転呼び出し(サブプログラムなど)の後にプログラム指令したすべての移動は、ミラーリングにより実行されます。

構文

```
MIRROR X...Y...Z...  
AMIRROR X...Y...Z...
```

注記

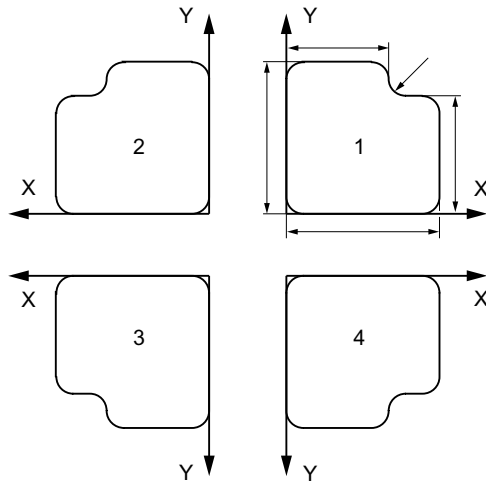
各フレーム命令は、個別の NC ブロックにプログラム指令します。

意味

MIRROR:	G54～G57、G505～G599 で設定した現在有効な座標系に対するアブソリュートの反転です。
AMIRROR:	現在有効な設定座標系、またはプログラム指令座標系を基準とする追加のミラーイメージです。
X...Y...Z...:	方向を変更するジオメトリ軸です。ここで指定値は自由に選択できます(例: X0 Y0 Z0)。

例

例 1: フライス加工

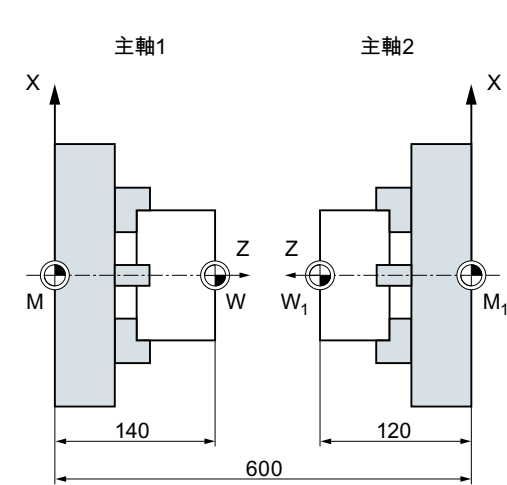


ここに示す輪郭は、サブプログラムとして 1 回だけプログラム指令します。他の 3 つの輪郭は、ミラーリングによって生成されます。ワーク原点は、輪郭の中央に配置されます。

プログラムコード	コメント
N10 G17 G54	; 作業平面 X/Y、ワーク原点
N20 L10	; 右上の最初の輪郭を加工します
N30 MIRROR X0	; X 軸を反転します (X の方向を変更します)
N40 L10	; 左上の 2 番目の輪郭を加工します
N50 AMIRROR Y0	; Y 軸を反転します (Y の方向を変更します)
N60 L10	; 左下の 3 番目の輪郭を加工します
N70 MIRROR Y0	; MIRROR で以前のフレームをリセットします。Y 軸を反転します (Y の方向を変更します)
N80 L10	; 右下の 4 番目の輪郭を加工します
N90 MIRROR	; ミラーリングの解除
N100 G0 X300 Y100 M30	; 後退、プログラム終了

3.12 座標変換(フレーム)

例 2 :旋削



実際の加工はサブプログラムとして格納され、各主軸での実行はミラーリングとオフセットで実行されます。

プログラムコード	コメント
N10 TRANS X0 Z140	; W のゼロオフセット
...	; 主軸 1 による 1 番目の側面の加工
N30 TRANS X0 Z600	; 主軸 2 のゼロオフセット
N40 AMIRROR Z0	; Z 軸のミラーリング
N50 ATRANS Z120	; W1 のゼロオフセット
...	; 主軸 2 による 2 番目の側面の加工

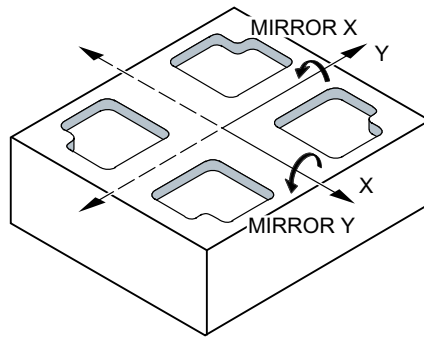
詳細情報

MIRROR X... Y... Z...

反転は、選択した作業平面の軸方向の変更でプログラム指令します。

例: 作業平面 G17 X/Y

(ミラー軸 Y による)反転には、X の方向変更が必要です。したがって、MIRROR X0 でプログラム指令します。その後、輪郭が、ミラー軸 Y の反対側に反転します。



ミラーリングが、G54～G57、G505～G599 で設定した現在有効な座標系基準で実行されます。

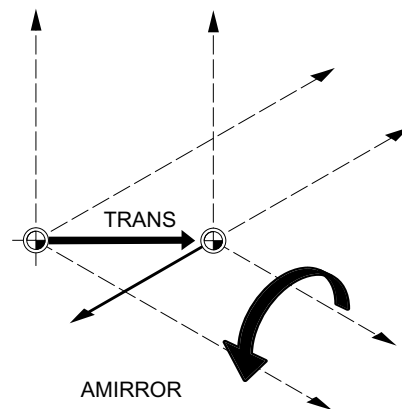
通知

オリジナルフレームなし

MIRROR 命令は、以前に有効にしたプログラマブルフレームのすべてのフレーム成分をリセットします。

AMIRROR X... Y... Z...

既存の座標変換に追加するミラーイメージは、AMIRROR でプログラム指令します。現在設定されている、または最後のプログラム指令座標系を、基準として使用します。



ミラーリングの解除

すべての軸に対して:MIRROR (軸パラメータなし)

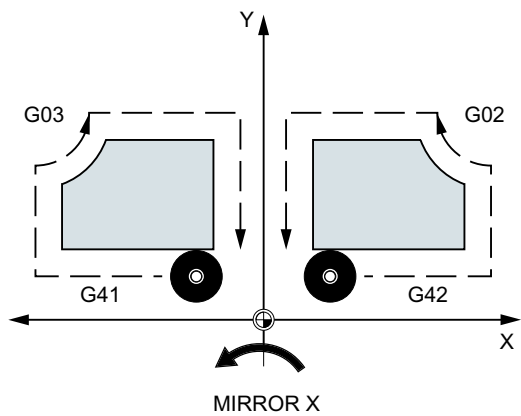
以前にプログラム指令したフレームのフレーム成分はすべて、リセットされます。

3.12 座標変換(フレーム)

工具径補正

注記

反転命令を使用すると、コントローラは自動的に、新しい加工方向に従って軌跡補正命令 (G41/G42 or G42/G41)を変更します。



同じことが、円弧回転の方向(G2/G3 or G3/G2)にも適用されます。

注記

MIRROR の後に AROT で追加回転をプログラム指令した場合は、逆方向の回転(正/負または負/正)処理がおこなわれる可能性があります。ジオメトリ軸の反転は、コントローラにより自動的に回転に変換され、対応できる場合は、マシンデータで指定されたミラー軸での反転にも変換されます。これは、設定可能ゼロオフセットにも適用されます。

ミラー軸

ミラー軸は、次のマシンデータで設定できます。

MD10610 \$MN_MIRROR_REF_AX = <値>

規格値	意味
0	ミラーリングは、プログラム指令軸が反転します(値の反転)。
1	基準軸は X 軸です。
2	基準軸は Y 軸です。
3	基準軸は Z 軸です。

プログラム指令値の解釈

次のマシンデータを使用して、プログラム指令値をどのように解釈するかを指定します。

MD10612 \$MN_MIRROR_TOGGLE = <値>

規格値	意味
0	プログラム指令軸の値は使用されません。
1	プログラム指令軸の値が使用されます。 - プログラム指令軸の値≠0 では、軸を反転していない場合は、反転します。 - プログラム指令軸の値=0 の場合は、ミラーリングを無効にします。

3.12.9 工具オリエンテーションに従ったフレーム生成(TOFRAME、TOROT、PAROT):

TOFRAME により、Z 軸が現在の工具オリエンテーションと一致する矩形フレームを生成します。これは、ユーザーが、衝突の回避して、工具を Z 方向に後退させる(5 軸プログラムの工具の停止後など)ことができるということです。

X 軸と Y 軸の位置は、マシンデータ MD21110 \$MC_X_AXES_IN_OLD_X_Z_PLANE(自動フレーム定義の座標系)の設定により特定されます。新しい座標系は、機械のキネマティクスから生成された状態で残るか、または新しい X 軸が以前の Z/X 平面にあるように、新しい Z 軸を中心としてさらに回転します(工作機械メーカーの仕様書を参照してください)。

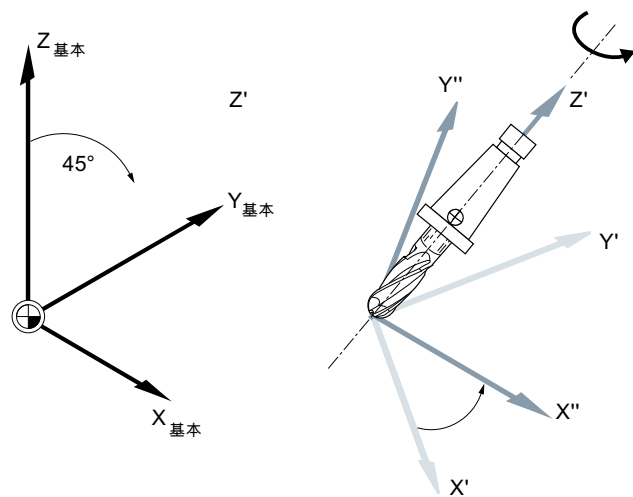
その結果、向きを示すフレームが形成されて、プログラマブルフレームのシステム変数に書き込まれます(\$P_PFRAME)。

TOROT は、プログラム指令フレームの回転成分に上書きするだけです。他のすべての成分は、そのまま変更されません。

TOFRAME と TOROT は、一般的には G17 (作業平面 X/Y)が有効なフライス加工運転用に設計されています。ただし、旋削運転の場合、あるいは通常 G18 または G19 が有効なときは、X 軸または Y 軸が工具の向きに一致する場合にフレームが必要です。これらのフレームは、TOFRAMEX/TOROTX または TOFRAMEY/TOROTY の各命令でプログラム指令します。

PAROT により、ワーク上にワーク座標系を配置します。

3.12 座標変換(フレーム)



構文

TOFRAME/TOFRAMEZ/TOFRAMEY/TOFRAMEX

...

TOROTOF

TOROT/TOROTZ/TOROTY/TOROTX

...

TOROTOF

PAROT

...

PAROTOF

意味

TOFRAME:	フレームを回転して、WCS の Z 軸を工具オリエンテーションに平行に配置します
TOFRAMEZ:	TOFRAME と同じです
TOFRAMEY:	フレームを回転して、WCS の Y 軸を工具オリエンテーションに平行に配置します
TOFRAMEX:	フレームを回転して、WCS の X 軸を工具オリエンテーションに平行に配置します

TOROT:	フレームを回転して、WCS の Z 軸を工具オリエンテーションに平行に配置します TOROT で定義した回転は、TOFRAME で定義した回転と同じです。
TOROTZ:	TOROT と同じ
TOROTY:	フレームを回転して、WCS の Y 軸を工具オリエンテーションに平行に配置します
TOROTX:	フレームを回転して、WCS の X 軸を工具オリエンテーションに平行に配置します
TOROTOF:	工具オリエンテーションに平行な向きを解除します
PAROT:	フレームを回転して、ワークにワーク座標系を配置します 動作中のフレームの平行移動、スケーリング、およびミラーリングは、そのまま有効です。
PAROTOF:	PAROT で有効にしたワーク用のフレーム回転を PAROTOF で無効にします。

注記

TOROT 命令により、キネマティックタイプ毎に、有効な旋回工具ホルダとの整合性のあるプログラミングを確実に実行できます。

旋回工具ホルダの事例と同様に、PAROT を使用して、作業台の回転を起動できます。これにより、フレームが定義されて、機械で補正移動がおこなわれないように、ワーク座標系の位置を変更します。言語命令 PAROT は、旋回工具ホルダが無効な場合でも受け付けます。

例

プログラムコード	コメント
N100 G0 G53 X100 Z100 D0	
N120 TOFRAME	
N140 G91 Z20	; TOFRAME が計算に含まれ、すべてのプログラム指令ジオメトリ軸の移動は、新しい座標系を基準にします。
N160 X50	
...	

3.12 座標変換(フレーム)

詳細情報

軸方向の割り当て

TOFRAMEX、TOFRAMEY、TOROTX、TOROTY の各命令のいずれかを、TOFRAME/
TOFRAMEZ または TOROT/TOROTZ の代わりにプログラム指令した場合は、次の表に記載された軸方向命令が適用されます。

命令	工具方向(垂直軸)	2 番目の軸 (横軸)	2 番目の軸 (縦軸)
TOFRAME/TOFRAMEZ / TOROT/TOROTZ	Z	X	Y
TOFRAMEY/TOROTY	Y	Z	X
TOFRAMEX/TOROTX	X	Y	Z

TOFRAME または TOROT の個別のシステムフレーム

TOFRAME または TOROT の結果のフレームは、個別のシステムフレーム \$P_TOOLFRAME に書き込むことができます。このために、マシンデータ

MD28082 \$MC_MM_SYSTEM_FRAME_MASK のビット 3 を有効にしてください。プログラマブルフレームは、変更されずにそのままです。プログラマブルフレームが他のどこかで追加処理されると、違いが発生します。

3.12.10 フレームの選択解除(G53、G153、SUPA、G500)

工具交換位置へのアプローチなどの特定の処理の実行時には、それぞれの場合に、さまざまなフレーム成分の定義とマスクが必要となる場合があります。

設定可能フレームは、モーダルに無効にするか、またはノンモーダルにマスクすることができます。

プログラマブルフレームは、マスクするか、またはノンモーダルに解除することができます。

構文

G53
G153
SUPA
G500
TRANS
ROT
SCALE
MIRROR

意味

G53:	プログラマブルと設定可能フレームのすべてをノンモーダルマスクします。
G153:	G153 には、G53 と同じ働きがあり、同様に、基本フレーム全体(\$P_ACTBFRAME)もマスクします。
SUPA:	SUPA には、G153 と同じ働きがあり、同様に、次のものもマスクします。 • ハンドルオフセット(DRF) • 重畳移動 • 外部ゼロオフセット • PRESET オフセット
G500:	G500 に値が含まれない場合は、すべての設定可能フレーム(G54～G57、G505～G599)のモーダルの解除をおこないます。
TRANS ROT SCALE MIRROR:	軸の詳細がない場合、プログラマブルフレームを削除します。

3.12.11 プログラミング軸別の重畳の解除(CORROF)

CORROF 手順により、以下の軸別重畳(オーバーライド)が解除されます:

- ハンドル移動により設定された追加ゼロオフセット(DRF オフセット)
- \$AA_OFF システム変数によってプログラムされた位置オフセット

先読み停止は、重畳(オーバーライド)値の解除によって開始され、解除された重畳動作の位置成分が基本座標系の位置に伝送されます。それによって、移動する軸はありません。

\$AA_IM システム変数(軸の現在の **MCS** 指令)によって読み取ることができる位置データは、**機械座標系**で変化しません。

重畳動作の解除された成分が含まれるようになったために、\$AA_IW システム変数(軸の現在の **WCS** 指令)によって読み取ることができる位置データは**ワーク座標系**では変化します。

注記

CORROF は、**NC プログラム**でプログラムできます。

CORROF は、**シンクロナイズドアクション**ではプログラムしないでください。

3.12 座標変換(フレーム)

構文

```
CORROF(<Axis>,"<String>"[,<Axis>,"<String>"])
```

意味

CORROF:	軸の以下のオフセットおよび重畳の解除の手順:	
	• DRF オフセット • 位置オフセット(\$AA_OFF)	
	有効性:	モーダル
<軸>:	軸識別子(チャンネル、ジオメトリ軸、または機械軸の識別子)	
	データタイプ:	AXIS
<String>:	重畳タイプを定義する文字列	
	データタイプ:	BOOL
	値	意味
	DRF	DRF オフセット
	AA_OFF	位置オフセット(\$AA_OFF)

例

例 1:DRF オフセットの軸別解除(1)

DRF オフセットは、DRF 手動パルス発生器の移動により X 軸で生成されます。チャンネルの他の軸に対しては、DRF オフセットは機能しません。

プログラムコード	コメント
N10 CORROF (X,"DRF")	; ここでは CORROF は、DRFOF と同じ働きがあります。
...	

例 2:DRF オフセットの軸別解除(2)

DRF オフセットは、DRF 手動パルス発生器の移動により X 軸と Y 軸で生成されます。チャンネルの他の軸に対しては、DRF オフセットは機能しません。

プログラムコード	コメント
; X 軸の DRF オフセットのみが選択解除されます。Y 軸の DRF オフセットは保持されます	
; DRFOF では、両方のオフセットが解除されます。	
N10 CORROF (X,"DRF")	
...	

例 3:\$AA_OFF 位置オフセットの軸別解除

プログラムコード	コメント
;	位置オフセット == 10 が X 軸に対して補間されます。
N10 WHEN TRUE DO \$AA_OFF[X]=10 G4 F5	
...	
;	X 軸の位置オフセットの選択解除: \$AA_OFF[X]=0
;	X 軸は移動しません。
;	位置オフセットを、X 軸の現在位置に加算します。
N80 CORROF(X,"AA_OFF")	
...	

例 4:DRF オフセットと\$AA_OFF 位置オフセットの軸別選択解除 (1)

DRF オフセットは、DRF 手動パルス発生器の移動により X 軸で生成されます。チャネルの他の軸に対しては、DRF オフセットは機能しません。

プログラムコード	コメント
;	位置オフセット 10 が X 軸に対して補間されます。
N10 WHEN TRUE DO \$AA_OFF[X]=10 G4 F5	
...	
;	X 軸の DRF オフセットと位置オフセットのみを選択解除します。
;	Y 軸の DRF オフセットは保持されます。
N70 CORROF(X,"DRF",X,"AA_OFF")	
...	

例 5:DRF オフセットと\$AA_OFF 位置オフセットの軸別選択解除 (2)

DRF オフセットは、DRF 手動パルス発生器の移動により X 軸と Y 軸で生成されます。チャネルの他の軸に対しては、DRF オフセットは機能しません。

プログラムコード	コメント
;	位置オフセット == 10 が X 軸に対して補間されます。
N10 WHEN TRUE DO \$AA_OFF[X]=10 G4 F5	
...	
;	Y 軸の DRF オフセットと X 軸の位置オフセットのみを選択解除します。
;	X 軸の DRF オフセットは保持されます。
N70 CORROF(Y,"DRF",X,"AA_OFF")	
...	

詳細情報

\$AA_OFF_VAL

位置オフセットを\$AA_OFF で選択解除すると、対応する軸のシステム変数\$AA_OFF_VAL (軸重畳の統合距離)はゼロになります。

JOG モードの\$AA_OFF

JOG モードでも、\$AA_OFF が変更されると、マシンデータ MD 36750

\$MA_AA_OFF_MODE でこの機能を有効にしている場合は、位置オフセットは重畳移動として補間されます。

シンクロナイズドアクションの\$AA_OFF

位置オフセットを CORROF (<軸>, "AA_OFF") で選択解除時に、\$AA_OFF (DO \$AA_OFF [<軸>]=<値>)を直ちにリセットするシンクロナイズドアクションが有効である場合、\$AA_OFF は選択解除されますが、リセットされません。また、アラーム 21660 が表示されます。ただし、CORROF の後のブロックなど、後の方のブロックでシンクロナイズドアクションが有効になる場合は、\$AA_OFF が設定されたままとなり、位置オフセットは補間されます。

自動チャネル軸入れ替え

別のチャネルで有効な軸が CORROF のプログラム指令対象である場合、その軸は、軸入れ替え時にそのチャネルに取り込まれ(条件:MD30552 \$MA_AUTO_GET_TYPE > 0)、位置オフセットか DRF オフセットのいずれか、または両方が選択解除されます。

3.12.12 追加ゼロオフセットの選択解除(DRFROF)

移動ハンドルによって指令された追加ゼロオフセット(DRF オフセット)は、DRFOF 手順によって選択解除されます。

先読み停止は選択解除によって実施され、解除された DRF オフセットの位置決め構成部分は、基本座標系の中の位置に転送され、それによって軸が移動することはありません。
\$AA_IM システム変数の値(軸の現在の MCS 指令値)は変わりませんが、\$AA_IW システム変数の値(軸の現在の WCS 指令値)は変わります。これは、重畳された移動から選択解除された構成部分が含まれるためです。

構文

DRFOF

意味

DRFOF:	チャンネルで動作中のすべての軸の DRF ハンドルオフセットを選択解除する手順	
	有効性:	モーダル

例

例 1:DRF オフセットの軸別の選択解除(1)

DRF オフセットは、DRF 手動パルス発生器の移動により X 軸で生成されています。チャンネルの他の軸に対しては、DRF オフセットは機能しません。

プログラムコード	コメント
N10 CORROF(X,"DRF")	; ここでは CORROF は、DRFOF と同じ働きがあります。
...	

例 2 :DRF オフセットの軸別の選択解除(2)

DRF オフセットは、DRF 手動パルス発生器の移動により X 軸と Y 軸で生成されています。チャンネルの他の軸に対しては、DRF オフセットは機能しません。

プログラムコード	コメント
N10 CORROF(X,"DRF")	; X 軸の DRF オフセットのみが選択解除されます。Y 軸の DRF オフセットは保持されます (DRFOF の場合は、両方のオフセットが選択解除されます)。
...	

例 3 :DRF オフセットと\$AA_OFF 位置オフセットの軸別の選択解除(1)

DRF オフセットは、DRF 手動パルス発生器の移動により X 軸で生成されます。チャンネルの他の軸に対しては、DRF オフセットは機能しません。

プログラムコード	コメント
N10 WHEN TRUE DO \$AA_OFF[X]=10 G4 F5	; 位置オフセット== 10 が X 軸に対して補間されます。
...	
N70 CORROF(X,"DRF",X,"AA_OFF")	; X 軸の DRF オフセットと位置オフセットのみを選択解除します。Y 軸の DRF オフセットは保持します。
...	

3.12 座標変換(フレーム)

例 4 :DRF オフセットと\$AA_OFF 位置オフセットの軸別の選択解除(2)

DRF オフセットは、DRF 手動パルス発生器の移動により X 軸と Y 軸で生成されます。チャネルの他の軸に対しては、DRF オフセットは機能しません。

プログラムコード	コメント
N10 WHEN TRUE DO \$AA_OFF[X]=10 G4 F5	; 位置オフセット== 10 が X 軸に対して補間されます。
...	
N70 CORROF(Y,"DRF",X,"AA_OFF")	; Y 軸の DRF オフセットと X 軸の位置オフセットを選択解除します。X 軸の DRF オフセットは保持します。
...	

3.12.13 研削固有ゼロオフセット(GFRAME0, GFRAME1 ... GFRAME100)

チャネル内で研削フレームを有効にする命令

GFRAME<n>命令をプログラミングすると、\$P_GFR[<n>]データ管理の対応する研削フレームがチャネル内で有効になります。これにより、データ管理の\$P_GFR[<n>]研削フレームと同一の有効な\$P_GFRAME 研削フレームが設定されます。

GFRAME<n> ⇒ \$P_GFRAME = \$P_GFR[<n>]

命令	チャネル内で有効になった研削フレーム
GFRAME0	\$P_GFR[0] (ヌルフレーム)
GFRAME1	\$P_GFR[1]
...	...
GFRAME100	\$P_GFR[100]

構文

GFRAME<n>

意味

GFRAME<n>:	データ管理の研削フレーム<n>の有効化	
	G グループ:	64
	基本設定:	MD20150 \$MC_GCODE_RESET_VALUES[63]
	有効性:	モーダル

<n>:	研削フレームの番号	
	値の範囲:	0, 1, 2, ... 100

3.13 補助機能出力

機能

工作機械で、NC プログラムで、PLC が指定の切り替え動作をおこなうために、補助機能出力によって、指定の PLC へ情報を送信します。補助機能が、そのパラメータと共に PLC インタフェースへ出力されます。伝送される値と信号は、PLC ユーザープログラムで処理してください。

補助機能

次の補助機能を PLC へ渡すことができます。

補助機能	アドレス
工具選択	T
工具オフセット	D、DL
送り速度	F/FA
主軸速度	S
M 機能	M
H 機能	H

各機能グループまたは単独機能に対して、マシンデータを使用して、移動動作の**前**か**同時**、および**後**の、どのタイミングで出力をおこなうかを定義します。

PLC は、さまざまな方法で補助機能出力に応答するようにプログラム指令ができます。

3.13 補助機能出力

機能

補助機能の重要な機能の概要を、次の表に示します。

機能	アドレス拡張子		値			意味	ブロック 毎の最大 指令数
	意味	範囲	範囲	タイプ	意味		
M	-	0 (固有値)	0 ... 99	INT	機能	0～99 の範囲に対しては、アドレス拡張子は 0 です。 アドレス拡張子なしの予約コード: M0、M1、M2、M17、M30	5
	主軸番号	1 - 12	1 ... 99	INT	機能	M3、M4、M5、M19、M70 では、アドレス拡張子は主軸番号です (例: M2=5; 主軸 2 の主軸停止)。 主軸番号がない場合は、機能はメイン主軸に適用されます。	
	指定なし	0 - 99	100 ... 2147483647	INT	機能	ユーザー用 M 機能*	
S	主軸番号	1 - 12	0 ... $\pm 1.8 \times 10^{308}$	REAL	速度	主軸番号がない場合は、機能はメイン主軸に適用されます。	3
H	指定なし	0 - 99	0 ... ± 2147483647 $\pm 1.8 \times 10^{308}$	INT REAL	指定なし	機能は NC では無効です。PLC でのみ実行されます。*	3
T	主軸番号 (工具管理機能が動作中の場合)	1 - 12	0～32000 (または、工具管理機能が動作中の場合は工具名称)	INT	工具選択	工具名称は PLC インタフェースには渡りません。	1

機能	アドレス拡張子		値			意味	ブロック 毎の最大 指令数
	意味	範囲	範囲	タイプ	意味		
D	-	-	0 - 12	INT	工具オフ セットの 選択	D0:解除 初期設定:D1	1
DL	ロケーシ ョンによ るオフセ ット	1 - 6	0 ... $\pm 1.8 \times 10^{308}$	REAL	工具仕上 げオフセ ットの選 択	以前に選択した D 番号 を参照します。	1
F	-	-	0.001 - 999 999.999	REAL	軌跡送り 速度		6
FA	軸番号	1 - 31	0.001 - 999 999.999	REAL	軸の送り 速度		
* 機能の意味は工作機械メーカー(工作機械メーカーの仕様書を参照してください)によって定義されます。							

詳細情報

NC ブロック毎の機能出力数

1 つの NC ブロックに 10 個までの機能出力をプログラム指令できます。補助機能は、シンクロナイズドアクションの動作部からも出力できます。

グループ化

記述した機能はグループ化できます。一部の M 命令では、グループの割り当てが予約されています。グループ毎に応答動作を定義できます。

高速機能出力(QU)

高速出力としてプログラム指令していない機能は、個々の出力毎に、キーワード QU で高速出力として定義できます。プログラムの実行はその他の機能の応答を待つことなく続行されます(プログラムは送信応答を待ちます)。これにより、移動に対する不要な待機と中断を回避できます。

注記

「高速機能出力」機能には、当該のマシンデータを設定してください(→ 工作機械メーカー)。

移動指令の機能出力

情報の伝送だけでなく、当該の応答の待機も同様に時間がかかるために、移動に影響します。

3.13 補助機能出力

ブロック切り替え遅延のない高速応答

ブロック切り替え動作がマシンデータで調整できます。「ブロック切り替え遅延なし」設定を選択すると、高速補助機能に関するシステム動作は次のようになります。

補助機能出力	動作
移動の前	高速補助機能によるブロック間のブロック遷移は、 中断することなく、減速されずに 実行されます。補助機能出力は、ブロックの最初の補間サイクルでおこなわれます。後続のブロックは、応答遅延なしで実行されます。
移動中	高速補助機能によるブロック間のブロック遷移は、 中断することなく、減速されずに 実行されます。補助機能出力は、ブロックの途中でおこなわれます。後続のブロックは、応答遅延なしで実行されます。
移動後	移動はブロック終点で停止します。補助機能出力は、ブロック終点でおこなわれます。後続のブロックは、応答遅延なしで実行されます。



注意

連続軌跡モードの機能出力

前のブロックの移動動作では、連続軌跡モード(G64/G641)が中断されて、イグザクトストップがおこなわれる**前**に機能が出力されます。

実行中のブロックの移動動作では、連続軌跡モード(G64/G641)が中断されて、イグザクトストップの**後**に機能が出力されます。

重要:連続軌跡モードは、PLC からの未処理の応答信号の待機によって中断される場合があります。たとえば、非常に短い軌跡長のブロックに **M** 命令処理がある場合です。

3.13.1 M 機能

M 機能は、「クーラント ON/OFF」、および機械のその他の機能等の切り替え動作を開始します。

構文

M<値>

M[<アドレス拡張子>] = <値>

意味

M:	M 機能のプログラミングのためのアドレスです。	
<アドレス拡張子>:	一部の M 機能(主軸機能に対する主軸番号の指定など)には拡張アドレス表記が適用されます。	
<値>:	割り当て値(M 機能番号)を、特定の運転機能に割り当てます。	
	タイプ:	INT
	値の範囲:	0 ... 2147483647(最大 INT 値)

予約 M 機能

制御装置は、次に示すような、プログラムを実行するために重要な特定の M 機能を標準で備えています。

M 機能	意味
M0*	プログラムストップ
M1*	オプションストップ
M2*	プログラムの終了、メインプログラム(M30)
M3	主軸の右回り
M4	主軸の左回り
M5	主軸停止
M6	工具交換(初期設定)
M17*	サブプログラム終了
M19	主軸位置決め
M30*	プログラムの終了、メインプログラム(M2)
M40	自動ギヤ切り替え
M41	ギヤ選択 1
M42	ギヤ選択 2
M43	ギヤ選択 3
M44	ギヤ選択 4
M45	ギヤ選択 5
M70	主軸の軸モードへの切り替え

3.13 補助機能出力

注記

拡張アドレス表記は、*が付いている機能では使用できません。

機能 M0、M1、M2、M17、および M30 は常に移動動作後に起動されます。

工作機械メーカーが定義する M 機能

未使用の M 機能番号はすべて、工作機械メーカーが、クランプ機器制御のための機能の切り替え、追加の運転機能の適用/解除などに使用できます。

注記

未使用の M 機能番号に割り当てられた機能は、機械別に異なります。したがって、特定の M 機能が、機械毎に異なる機能になる場合があります。

機械で利用できる M 機能とそれらの機能については、工作機械メーカーの仕様書を参照してください。

例**例 1:ブロック内の M 機能の最大数**

プログラムコード	コメント
N10 S...	
N20 X...M3	; 軸移動を伴うブロック内の M 機能、 ; X 軸の移動前の主軸の加速。
N180 M789 M1767 M100 M102 M376	; ブロックに最大 5 個の M 機能があります。

例 2 :高速出力としての M 機能

プログラムコード	コメント
N10 H=QU(735)	; H735 用の高速出力。
N10 G1 F300 X10 Y20 G64	
N20 X8 Y90 M=QU(7)	; M7 用の高速出力。

M7 は、連続軌跡モード(G64)が中断されないように、高速出力としてプログラム指令されます。

注記

この機能は、他の機能出力との組み合わせにより時系列に割り当てを変更するときなど、特別な場合にのみ使用してください。

予約 M 命令に関する詳細情報

プログラムストップ M0

M0 のある NC ブロックで、加工が停止します。これで、切屑を取り除いたり、再計測ができます。

プログラムストップ 1-オプションルストップ:M1

M1 は、以下を使用して設定できます。

- HMI/対話ボックスの「プログラム制御」
または
- NC/PLC インタフェース

NC プログラムの実行が、プログラム指令ブロックで停止します。

プログラムストップ 2 - M1 に関連したプログラムの実行を停止する補助機能

プログラムストップ 2 は、HMI/対話ボックスの「プログラム制御」で設定でき、加工部の終了時に、いつでも加工処理を中断できます。これにより、オペレータは製造を中断して、切屑の除去などをおこなうことができます。

プログラム終了:M2、M17、M30

プログラムは M2、M17、または M30 によって終了します。メインプログラムが別のプログラムから(サブプログラムとして)呼び出された場合、M2/M30 および M17 は同じ働きになります。つまり、M17 はメインプログラムでは、M2/M30 と同じ働きをします。

主軸機能:M3、M4、M5、M19、M70

主軸番号の指定の拡張アドレス表記は、すべての主軸に適用されます。

例:

プログラムコード	コメント
M2 = 3	; 2 番目の主軸の右回り主軸回転

アドレス拡張子をプログラム指令しない場合は、機能はメイン主軸に適用されます。

3.14 補助命令

3.14.1 メッセージ出力(MSG)

MSG () 命令を使用して、パートプログラムの任意の文字列をメッセージとしてオペレータに出力できます。

構文

```
MSG ("<メッセージテキスト>" [, <実行>])  
...  
MSG ( )
```

意味

MSG:	メッセージの出力用の予約サブプログラム呼び出し		
<メッセージテキスト>:	メッセージとして表示される任意の文字列		
	タイプ:	STRING	
	最大長さ:	124 文字で、最大 2 行で表示されます(2*62 文字)。	
	リンク演算子「<<」を使用して、変数もメッセージテキストに出力することができます。		
<実行>:	メッセージを書き込むタイミングを定義するパラメータ(任意選択)		
	タイプ:	INT	
	値:	0(初期設定)	メッセージを書き込むために、専用メインランブロックを生成しません。この専用メインランブロックは、実行可能な次の NC ブロックで生成されます。動作中の連続軌跡モードは中断されません。
		1	メッセージを書き込むために、専用のメインランブロックを生成します。動作中の連続軌跡モードが中断されます。
MSG () :	実際のメッセージは、メッセージテキストを指定せずに MSG () をプログラム指令すると、解除できます。解除しない場合は、次のメッセージが表示されるまで表示されたままになります。		

注記

操作画面で有効になっている言語でメッセージを出力する場合、現在 HMI で設定されている言語についての情報を要求できます。この情報は、システム変数\$AN_LANGUAGE_ON_HMI (ページ 1582)を使用してパートプログラムとシンクロナイズドアクションで問い合わせることができます。

例

例 1:メッセージの出力/解除

プログラムコード	コメント
N10 G91 G64 F100	; 連続軌跡モード
N20 X1 Y1	
N...X...Y...	
N20 MSG ("Machining part 1")	; メッセージを最初に N30 で出力します。 ; 連続軌跡モードを維持します。
N30 X...Y...	
N...X...Y...	
N400 X1 Y1	
N410 MSG ("Machining part 2",1)	; メッセージを N410 で出力します。 ; 連続軌跡モードが中断されます。
N420 X1 Y1	
N...X...Y...	
N900 MSG ()	; メッセージを解除します。

例 2:変数を使用したメッセージテキスト

プログラムコード	コメント
N10 R12=\$AA_IW [X]	; R12 に X 軸の現在位置を格納します。
N20 MSG ("Check position of X axis"<<R12<<)	; 変数 R12 を使用してメッセージを出力します。
...	
N90 MSG ()	; N20 からメッセージを解除します。

3.14.2 OPI 変数での文字列の書き込み(WRTPR)

WRTPR () 機能を使用すると、パートプログラムから任意の文字列を OPI 変数 progProtText に書き込むことができます。

3.14 補助命令

構文

```
WRTPR(<String>[,<ExecTime>])
```

意味

WRTPR:	文字列を出力するためのファンクションコール。			
<文字列>:	OPI 変数 progProtText に書き込まれる任意の文字列。			
	タイプ:	STRING		
	最大長さ:	128 文字		
<ExecTime>:	文字列が書き込まれるタイミングを定義するための選択パラメータ。			
	タイプ:	INT		
	データ範囲:	0, 1		
		0 (初期設定)	文字列を書き込むための専用のメインランブロックは生成されません。この専用メインランブロックは、実行可能な次の NC ブロックで生成されます。動作中の連続軌跡モードは中断されません。	
		1	文字列を書き込むための専用のメインランブロックが生成されます。動作中の連続軌跡モードが中断されません。	

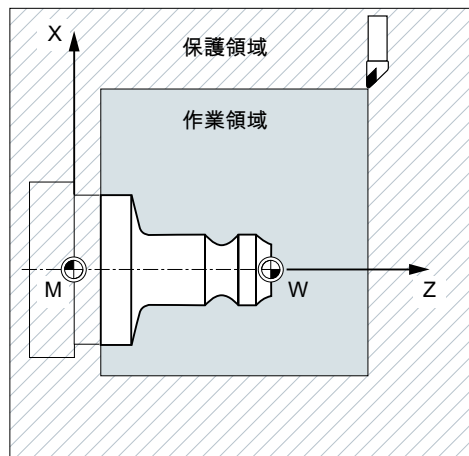
例

プログラムコード	コメント
N10 G91 G64 F100	; 連続軌跡モード
N20 X1 Y1	
N30 WRTPR("N30")	; まず、文字列「N30」が N40 に書き込まれます。 ; 連続軌跡モードを維持します。
N40 X1 Y1	
N50 WRTPR("N50",1)	; 文字列「N50」が N50 に書き込まれます。 ; 連続軌跡モードが中断されます。
N60 X1 Y1	

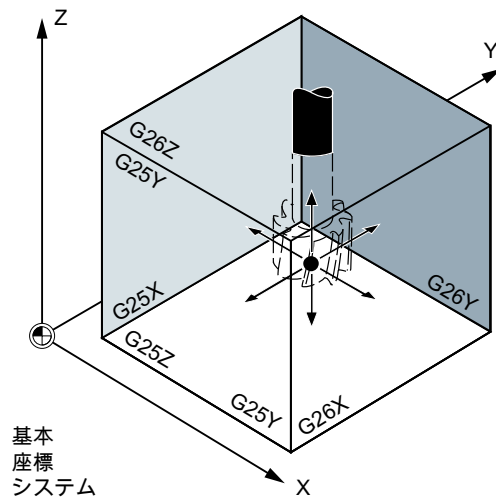
3.14.3 ワーキングエリアリミット

3.14.3.1 BCS のワーキングエリアリミット(G25/G26、WALIMON、WALIMOF)

G25/G26 は、工具が移動できる作業領域(作業フィールド、作業空間)を制限します。G25/G26 で定義した作業領域リミットの外側のエリアでは、あらゆる工具動作が禁止されます。



個々の軸の座標は、基本座標系で適用されます。



WALIMON 命令を使用して、有効なすべての軸の作業領域リミットをプログラム指令してください。WALIMOF 命令は、作業領域リミットを解除します。初期設定は WALIMON です。したがって、WALIMON のプログラム指令が必要なのは、作業領域リミットが事前に解除されている場合のみです。

3.14 補助命令

構文

```
G25 X... Y... Z...
G26 X... Y... Z...
WALIMON
...
WALIMOF
```

意味

G25:	作業領域リミットの下限 基本座標系のチャネル軸で値を割り当てます
G26:	作業領域リミットの上限 基本座標系のチャネル軸で値を割り当てます
X... Y... Z...:	個々のチャネル軸の作業領域リミットの下限と上限 指定したリミットは基本座標系を基準としています。
WALIMON:	作業領域リミットを、すべての軸について オン にします
WALIMOF:	作業領域リミットを、すべての軸について オフ にします

G25/G26 を使用して値をプログラム指令するだけでなく、次の軸セッティングデータを使用して値を入力することもできます。

SD43420 \$SA_WORKAREA_LIMIT_PLUS (正の作業領域リミット)

SD43430 \$SA_WORKAREA_LIMIT_MINUS (負の作業領域リミット)

SD43420 と SD43430 でパラメータ設定した作業領域リミットの起動と解除は、軸別セッティングデータを使用して、特定の方向に対して実行され、直ちに有効となります。

SD43400 \$SA_WORKAREA_PLUS_ENABLE (正方向の作業領域リミットが有効)

SD43410 \$SA_WORKAREA_MINUS_ENABLE (負方向の作業領域リミットが有効)

方向別に起動/解除を実行すると、軸の作業範囲を 1 方向のみに制限できます。

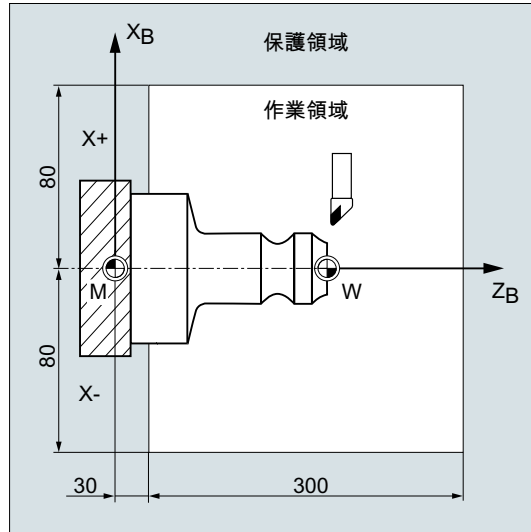
注記

G25/G26 でプログラム指令作業領域リミットは、SD43420 と SD43430 に入力された値より優先され、この値を上書きします。

注記

G25/G26 は、アドレス S で主軸速度の制限をプログラム指令するためにも使用できます。詳しくは、「プログラマブル主軸速度制限(G25、G26) (ページ 118)」を参照してください。

例



作業領域リミット G25/26 を使用すると、旋盤の作業領域が制限され、周囲の機器と装置(回転装置、計測ステーションなど)が損傷から保護されます。

初期設定: WALIMON

プログラムコード	コメント
N10 G0 G90 F0.5 T1	
N20 G25 X-80 Z30	; 個々の座標軸の下限を定義します
N30 G26 X80 Z330	; 上限を定義します
N40 L22	; 切削プログラム
N50 G0 G90 Z102 T2	; 工具交換ロケーションへ
N60 X0	
N70 WALIMOF	; 作業領域リミットを解除します
N80 G1 Z-2 F0.5	; ドリル
N90 G0 Z200	; 背面
N100 WALIMON	; 作業領域リミットのオン
N110 X70 M30	; プログラム終了

詳細情報

工具の基準点

工具長補正が有効な場合は、工具の先端が基準点として監視されます。有効でない場合は、工具ホルダの基準点が監視されます。

3.14 補助命令

工具径の監視は、個別に有効にしてください。これは、チャンネルマシンデータを使用して行います。

MD21020 \$MC_WORKAREA_WITH_TOOL_RADIUS

工具基準点が、作業領域リミットで定義されたワーキングエリアの外側にある場合、またはこのワーキングエリアの外に移動した場合は、プログラム処理が停止します。

注記

座標変換が有効な場合は、工具データ(工具長と工具半径)が考慮され、記述した動作とは異なる動作となる場合があります。

プログラム指令可能な作業領域リミット、G25/G26

作業領域リミットの上限(G26)と下限(G25)を軸ごとに定義できます。これらの値は直ちに有効となり、RESET 後と再電源投入後も、対応するマシンデータ設定(→ MD10710 \$MN_PROG_SD_RESET_SAVE_TAB)に対して、そのまま有効です。

注記

あらかじめ設定されている CALCPOSI (ページ 664)タグ機能を使用すると、作業領域リミットやプロテクションゾーンを考慮しながら予測軌跡に沿って移動しているかどうかを確認できます。

3.14.3.2 WCS/SZS の作業領域リミット(WALCS0 ... WALCS10)

「WCS/SZS の作業領域リミット」により、ワーク座標系(WCS)または設定可能ゼロオフセットシステム(SZS)におけるチャンネル軸の移動範囲をチャンネル別に、ワーク毎に柔軟に制限することができます。これは主として、従来の旋盤で使用するためのものです。

必要条件

チャンネル軸を原点確立してください。

作業領域リミットグループ

座標変換の切り換えや有効なフレームのオン/オフ時などのように、軸割り当てを変更するときに、すべてのチャンネル軸に対して軸固有の作業領域リミットを書き換えなくてもよいように、作業領域リミットグループが用意されています。

作業領域リミットグループには、以下のデータが含まれます。

- すべてのチャンネル軸の作業領域リミット
- 作業領域リミットの基準座標系

構文

```
...
$P_WORKAREA_CS_COORD_SYSTEM[<WALimNo>]=<値>
$P_WORKAREA_CS_PLUS_ENABLE[<WALimNo>,<Ax>]=<値>
$P_WORKAREA_CS_LIMIT_PLUS[<WALimNo>,<Ax>]=<値>
$P_WORKAREA_CS_MINUS_ENABLE[<WALimNo>,<Ax>]=<値>
$P_WORKAREA_CS_LIMIT_MINUS[<WALimNo>,<Ax>]=<値>
...
WALCS<n>
...
WALCS0
```

意味

\$P_WORKAREA_CS_COORD_SYSTEM[<WALimNo>]=<値>		
作業領域リミットグループが基準とする座標系		
<WALimNo> :	作業領域リミットグループ	
	タイプ:	INT
	データ範囲:	0 (グループ 1) ... 9 (グループ 10)
<値>:	タイプ INT の値	
	1	ワーク座標系(WCS)
	3	設定可能ゼロオフセットシステム(SZS)

\$\$P_WORKAREA_CS_PLUS_ENABLE[<WALimNo>,<Ax>]=<値>		
指定したチャンネル軸に対して 正 の軸方向の作業領域リミットを有効にします。		
<WALimNo> :	作業領域リミットグループ	
	タイプ:	INT
	値の範囲:	0 (グループ 1) ... 9 (グループ 10)
<Ax>:	チャンネル軸の名称	

3.14 補助命令

<値>:	タイプ BOOL の値	
	0 (FALSE)	リリースなし
	1 (TRUE)	表示

\$P_WORKAREA_CS_MINUS_ENABLE [<WALimNo>, <Ax>] = <値>		
指定したチャネル軸に対して 負 の軸方向の作業領域リミットを有効にします。		
<WALimNo> :	作業領域リミットグループ	
	タイプ:	INT
	値の範囲:	0 (グループ 1) ... 9 (グループ 10)
<Ax>:	チャネル軸名称	
<値>:	タイプ BOOL の値	
	0 (FALSE)	これはリリースされていません。
	1 (TRUE)	イネーブル

\$P_WORKAREA_CS_LIMIT_PLUS [<WALimNo>, <Ax>] = <値>		
指定したチャネル軸の 正 の方向の作業領域リミット		
<WALimNo> :	作業領域リミットグループ	
	タイプ:	INT
	値の範囲:	0 (グループ 1) ... 9 (グループ 10)
<Ax>:	チャネル軸の名称	
<値>:	タイプ REAL の値	

\$P_WORKAREA_CS_LIMIT_MINUS [<WALimNo>, <Ax>] = <値>		
指定したチャネル軸の 負 の方向の作業領域リミット		
<WALimNo> :	作業領域リミットグループ	
	タイプ:	INT
	値の範囲:	0 (グループ 1) ... 9 (グループ 10)
<Ax>:	チャネル軸の名称	
<値>:	タイプ REAL の値	

WALCS<n>:	作業領域リミットグループの作業領域リミットの有効化	
<n>:	作業領域リミットグループの番号	
	値の範囲:	1 ... 10

WALCS0:	チャンネル内で有効になっている作業領域リミットの解除
---------	----------------------------

注記

実際に使用可能な作業領域リミットグループの番号は、設定によって異なります(→ 工作機械メーカーの詳細情報を参照してください)。

例

次の3つの軸がチャンネルで定義されます。X、Y、およびZ

作業領域リミットグループ No. 2 を定義して有効にします。次の指定に従って、軸が WCS で制限されます。

- X 軸の正方向:10 mm
- X 軸の負方向:制限なし
- Y 軸の正方向:34 mm
- Y 軸のマイナス方向: -25 mm
- Z 軸の正方向:制限なし
- Z 軸のマイナス方向: -600 mm

プログラムコード	コメント
...	
N51 \$P_WORKAREA_CS_COORD_SYSTEM[1]=1	; 作業領域リミットグループ 2 の作業領域リミットは、WCS 内で適用されます。
N60 \$P_WORKAREA_CS_PLUS_ENABLE[1,X]=TRUE	
N61 \$P_WORKAREA_CS_LIMIT_PLUS[1,X]=10	
N62 \$P_WORKAREA_CS_MINUS_ENABLE[1,X]=FALSE	
N70 \$P_WORKAREA_CS_PLUS_ENABLE[1,Y]=TRUE	
N73 \$P_WORKAREA_CS_LIMIT_PLUS[1,Y]=34	
N72 \$P_WORKAREA_CS_MINUS_ENABLE[1,Y]=TRUE	
N73 \$P_WORKAREA_CS_LIMIT_MINUS[1,Y]=-25	
N80 \$P_WORKAREA_CS_PLUS_ENABLE[1,Z]=FALSE	

3.14 補助命令

プログラムコード	コメント
N82 \$P_WORKAREA_CS_MINUS_ENABLE[1,Z]=TRUE	
N83 \$P_WORKAREA_CS_LIMIT_PLUS[1,Z]=-600	
...	
N90 WALCS2	; 作業領域リミットグループ 2 の起動
...	

詳細情報

有効性

WALCS1～WALCS10 によるワーキングエリアリミットは、WALIMON によるワーキングエリアリミットとは無関係に動作します。この両方の機能が有効な場合は、最初に到達した軸移動の制限が有効になります。

工具の基準点

工具データ(工具長と工具径)を使用する場合は、ワーキングエリアリミット監視時の工具の基準点は、WALIMON によるワーキングエリアリミットの動作に対応します。

3.14.4 リファレンス点復帰(G74)

機械に電源を投入したとき(インクレメンタル位置検出器を使用する場合)は、すべての軸送り台は、その原点マークにリファレンス復帰をしてください。この後に、はじめて移動動作をプログラム指令できます。

レファレンス点には、NC プログラムで G74 を使用してアプローチできます。

構文

G74 X1=0 Y1=0 Z1=0 A1=0 ...; 個別の NC ブロックでプログラム指令します

意味

G74:	G 命令呼び出しのレファレンス点復帰
X1=0 Y1=0 Z1=0 ...:	直線軸 に指定された機械軸アドレス X1、Y1、Z1…は、レファレンス点としてアプローチされます。
A1=0 B1=0 C1=0 ...:	回転軸 に指定された機械軸アドレス A1、B1、C1…は、レファレンス点としてアプローチされます。

注記

G74 でレファレンス点へアプローチする軸には、座標変換をプログラム指令しないでください。

座標変換は TRAFOOF 命令で無効にします。

例

検出器を変更するときは、レファレンス点へアプローチして、ワーク原点を設定します。

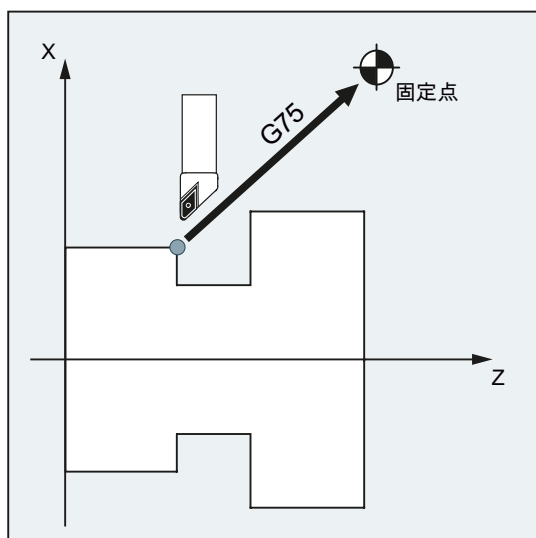
プログラムコード	コメント
N10 SPOS=0	; 位置制御の主軸
N20 G74 X1=0 Y1=0 Z1=0 C1=0	; 直線軸と回転軸のレファレンス点復帰
N30 G54	; ゼロオフセット
N40 L47	; 切削プログラム
N50 M30	; プログラム終了

3.14.5 固定点へのアプローチ(G75)

ノンモーダル命令 G75/を使用すると、軸を個々に、互いに無関係に、機械空間の固定点(工具交換位置、取り付け位置、パレット交換位置など)に移動できます。

固定点は、マシンデータ(MD30600 \$MA_FIX_POINT_POS[n])に格納された、機械座標系の位置です。軸毎に最大 4 つまでの固定点を定義できます。

固定点には、現在の工具またはワークの位置にかかわらず、すべての NC プログラムからアプローチできます。内部の先読み停止は、軸の移動前に実行されます。



要件

G75 で固定点へアプローチするには、次の条件を満たしてください。

- 正確に計算された固定点座標がマシンデータに書き込まれている。
- 固定点が有効移動範囲(→ソフトウェアのリミットスイッチ制限に注意してください)内にある。
- 移動する軸は原点確立済みである。
- 工具径補正が無効。
- キネマティックトランスフォーメーションが有効になっていない。
- 移動する軸のいずれも、動作中の座標変換に関与していない。
- 移動する軸のいずれも、動作中の連結のスレーブ軸ではない。
- 移動する軸のいずれも、ガントリグループの軸ではない。
- コンパイルサイクルが移動成分を起動しない。

構文

G75 <軸名称><軸位置> ... FP=<n>

意味

G75:	固定点アプローチ		
<軸名称>:	固定点へ移動する機械軸の名称 すべての軸識別子を使用できます。		
<軸位置>:	位置値は意味を持ちません。したがって、通常は「0」という値を指定します。		
FP=:	アプローチする固定点		
	<n>:	固定点番号	
		値の範囲:	1, 2, 3, 4
	注: FP=<n>および固定点番号のどちらも指定しない場合、または FP=0 をプログラム指令した場合は、FP=1 を指定したと解釈され、固定点 1 へアプローチします。		

注記

1 つの G75/ブロックに複数の軸をプログラム指令できます。プログラム指令した軸は、同時に指定した固定点へ移動します。

注記

アドレス FP の値が、各プログラム指令軸(MD30610 \$MA_NUM_FIX_POINT_POS)に指定した固定点の数を超えないようにしてください。

例

工具交換のときに、軸 X (=AX1) と Z (=AX3) が、X = 151.6 と Z = -17.3 の固定機械軸位置 1 に移動する必要があります。

マシンデータ:

- MD30600 \$MA_FIX_POINT_POS[AX1,0] = 151.6
- MD30600 \$MA_FIX_POINT[AX3,0] = 17.3

NC プログラム:

プログラムコード	コメント
...	
N100 G55	; 設定可能ゼロオフセットを有効にします。
N110 X10 Y30 Z40	; WCS の位置にアプローチします。

3.14 補助命令

プログラムコード	コメント
N120 G75 X0 Z0 FP=1 M0	; X 軸が 151.6 に移動し、 ; Z 軸が 17.3 に移動します (MCS)。 ; 各軸が最大速度で移動します。 ; このブロックでは、追加の移動は有効になりません。 ; 終了位置に達した後で ; 追加の移動が起こらないように ; ここに停止が挿入されます。
N130 X10 Y30 Z40	; N110 の位置へ再びアプローチします。 ; ゼロオフセットが再度有効になります。
...	

注記

「マガジンによる工具管理機能」機能が有効な場合は、補助機能 T...または M... (通常は M6)だけでは、G75 動作の終了でブロック切り替えの禁止はおこなわれません。

理由: 「マガジンによる工具管理機能」が有効な場合は、工具交換の補助機能が PLC へ出力されません。

詳細情報

G75

軸が、機械軸として早送りで移動します。この動作は内部で、「SUPA」(すべてのフレームをマスク)と「GO RTLI OF」(単独軸補間による早送り移動)機能を使用して割り当てられます。

「RTLI OF」(単独軸補間)の条件が満たされない場合は、固定点へ軌跡移動でアプローチします。

固定点に到達すると、軸が「精密イグザクトストップ」許容範囲内で停止状態になります。

G75 に対するパラメータ設定可能なダイナミック応答

次のマシンデータにより、固定点位置への位置決め移動(G75)に必要なダイナミック応答モードを設定できます。

MD18960 \$MN_POS_DYN_MODE (位置決め軸のダイナミック応答のタイプ)

追加の軸移動

G75 ブロックが解釈される時点で、以下の追加の軸移動がおこなわれます。

- 外部ゼロオフセット
- DRF
- 同期オフセット(\$AA_OFF)

この後は、G75 ブロックが移動終了に到達するまで、追加の軸移動は変更できません。

G75 ブロックの解釈後の追加の移動に従って、固定点へのアプローチがオフセットされます。

以下に示した追加の移動は、補間がおこなわれるときも、考慮されません。そのため、その後、目標位置がオフセットされます。

- オンライン工具補正
- BCS と機械座標系のコンパイルサイクルからの追加の移動

動作中のフレーム

すべての動作中のフレームが無視されます。そして機械座標系で移動がおこなわれます。

WCS/SZS の作業領域リミット

座標系別作業領域リミット(WALCS0 ... WALCS10)は、G75 を含むブロックでは無効です。終点は、次のブロックの始点として監視されます。

POSA/SPOSA による軸/主軸移動

プログラム指令の軸/主軸が以前に POSA または SPOSA によって移動している場合は、固定点へアプローチする前に、これらの移動を完了します。

G75 ブロックの主軸機能

主軸が「固定点アプローチ」を禁じられている場合は、追加の主軸機能(SPOS/SPOSA による位置決めなど)を G75 ブロックでプログラム指令できます。

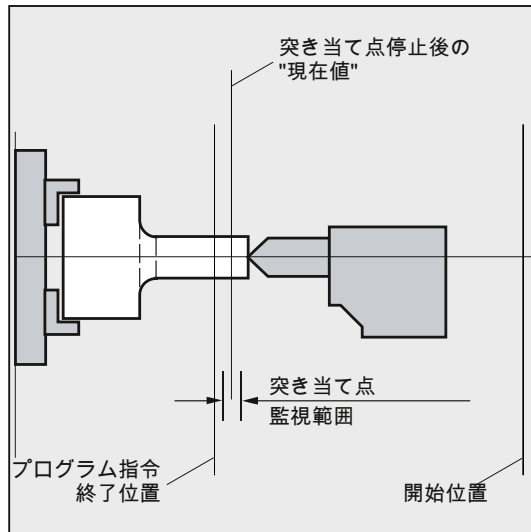
モジュロ軸

モジュロ軸の場合は、最短距離に沿って固定点へアプローチします。

3.14.6 突き当て点停止(FXS、FXST、FXSW)

機能

「突き当て点停止」機能を使用すると、心押し台、クイル、グリッパなどで必要なワーククランプの推力を定義することができます。この機能は、機械のレファレンス点へのアプローチにも使用できます。



トルクを十分に小さくすると、プローブを接続することなく、簡単な計測操作をおこなうこともできます。「突き当て点停止」機能は、軸だけでなく、軸移動機能を持つ主軸としても実行できます。

構文

```

FXS [<軸>]=...
FXST [<軸>]=...
FXSW [<軸>]=...
FXS [<軸>]=... FXST [<軸>]=...
FXS [<軸>]=... FXST [<軸>]=... FXSW [<軸>]=...

```

意味

FXS: 「突き当て点停止」機能の適用命令と解除命令
FXS [<軸>]=1: 機能を有効にします
FXS=<軸>=0: 機能を解除します
FXST: クランプトルクを設定するための任意選択命令
最大ドライブトルクに対する割合を%で指定します

FXSW: 突き当て点監視用の範囲幅を設定するための任意選択命令
mm、inch、または°単位で指定します

<軸>: 機械軸名称
機械軸(X1、Y1、Z1 など)をプログラム指令します。

注記

FXS、FXST、および FXSW 命令はモーダルです。

FXST と FXSW のプログラミングは任意選択です。この指定をしない場合は、最後のプログラム指令値または当該のマシンデータの設定値が適用されます。

突き当て点停止の起動:FXS[<軸>] = 1

終点への移動は、軌跡軸、または位置決め軸の移動として記述できます。位置決め軸の場合は、ブロックの境界を越えてこの機能をおこなうことができます。

突き当て点停止は、複数の軸に対して同時に、他の軸の移動と同じようにおこなうことができます。突き当て点は、開始位置と終了位置の間に配置してください。

通知

干渉の可能性

「突き当て点停止」機能がすでに軸/主軸に対して有効になっていた場合は、軸に対して新しい位置をプログラム指令することはできません。

この機能を選択する前に、主軸を位置制御モードに切り替えてください。

例:

プログラムコード	コメント
X250 Y100 F100 FXS[X1]=1 FXST[X1]=12.3 FXSW[X1]=2	; 軸 X1 は、送り速度 F100 (この指定は任意選択です) で目標位置 X=250 mm へ移動します。
...	クランプトルクは最大ドライブトルクの 12.3% で、監視は 2 mm の範囲幅でおこなわれます。

突き当て点停止の解除:FXS[<軸>] = 0

機能を選択解除すると、先読み停止がおこなわれます。

3.14 補助命令

FXS[<軸>]=0 を含むブロックは、移動を含む場合があります。

通知**干渉の可能性**

後退位置への移動は、突き当て点から離れる方向に動かしてください。そうでない場合は、突き当て点、または機械が損傷するおそれがあります。

ブロック切り替えは、後退位置に達したときに実行されます。後退位置を指定しない場合は、トルク制限が解除されると直ちに、ブロック切り替えが実行されます。

例:

プログラムコード	コメント
X200 Y400 G01 G94 F2000 FXS[X1]=0	; 軸 X1 が突き当て点から位置 X = 200 mm へ後退します。他の指定はすべて任意選択です。
...	

クランプトルク(FXST)と監視範囲(FXSW)

プログラム指令のトルク制限 FXST はいずれも、ブロックの先頭から有効です。つまり、突き当て点にも、トルクを小さくしてアプローチします。FXST と FXSW は、パートプログラムの中でいつでもプログラム指令と変更ができます。変更箇所は、同じブロックで移動を実行する前に有効になります。

通知**干渉の可能性**

再プログラム指令をする前に軸が移動していた場合、新しい突き当て点の監視範囲をプログラム指令すると、範囲幅の変更だけでなく、範囲の中心の基準点の変更もおこなわれます。範囲の変更時の機械軸の実位置は、新しい範囲の中心点です。

範囲は、突き当て点から外れた場合にのみ、突き当て点監視が異常を検出するように、選択してください。

詳細情報**上昇カーブ**

新しいトルク制限の上昇カーブの割合をマシンデータで定義して、トルク制限指令への急激な変化を防止できます。(クイルの挿入など)

アラームのマスク

マシンデータ項目でアラームをマスクして、新しいマシンデータ設定を **NEW_CONF** で起動することで、アプリケーションのために、パートプログラムで突き当て点アラームをマスクできます。

起動

突き当て点への移動命令は、シンクロナイズドアクションまたはテクノロジーサイクルから呼び出すことができます。これらの命令は、動作を開始することなく有効にでき、トルクが直ちに制限されます。指令値を介して軸が移動されるとただちに制限停止モニタが有効になります。

シンクロナイズドアクションからの起動

例:

予測される事象(\$R1)が発生したが、突き当て点停止がまだ実行されていない場合に、軸 Y に対して FXS を有効にします。トルクは、定格トルク値の 10%の値にします。監視範囲幅は初期設定されています。

プログラムコード

```
N10 IDS=1 WHENEVER (($R1=1) AND ($AA_FXS[Y]==0)) DO $R1=0 FXS[Y]=1
FXST[Y]=10
```

通常のパートプログラムでは、\$R1 が、目標のタイミングで確実に設定されるよう作成してください。

シンクロナイズドアクションからの解除

例:

予想される事象(\$R3)が発生し、「制限停止に接触」状態(システム変数\$AA_FXS)に達した場合に、FXS を選択解除します。

プログラムコード

```
IDS=4 WHENEVER (($R3==1) AND ($AA_FXS[Y]==1)) DO FXS[Y]=0
FA[Y]=1000 POS[Y]=0
```

突き当て点に到達

突き当て点に達した場合下記の処理をおこないます。

- 残移動距離が削除され、指令値が補正されます。
- ドライブトルクが、プログラム指令制限値 FXSW まで増加した後、一定に保持されます。
- 突き当て点監視が、指定した範囲幅で有効になります。

必要条件

- 残移動距離を削除する計測
「残移動距離削除による計測」(MEAS 命令)と「突き当て点停止」は、1つのブロックで同時にプログラム指令できません。
例外:1つの計測は1つの軌跡軸に作用します、そして、その他の計測は1つの位置決め軸に作用します。または両方の計測が位置決め軸に作用します。
- 軌跡誤差監視
輪郭誤差監視は、「突き当て点停止」が有効な間はおこなわれません。
- 位置決め軸
位置決め軸による「突き当て点停止」の場合は、ブロック切り替えは突き当て点移動に影響されずにおこなわれます。
- リンク軸とコンテナ軸
突き当て点停止は、リンク軸とコンテナ軸にも使用できます。
割り当てた機械軸の状態は、コンテナが回転しても維持されます。これは、FOCON によるモーダルなトルク制限にも適用されます。
- 次の場合は、突き当て点停止を使用**できません**。
 - ガントリ軸の場合
 - PLC でのみ制御される同時位置決め軸の場合(FXS は NC プログラムで選択されるためです)。
- トルク制限の値を下げすぎると、位置コントローラが限界状態になって輪郭誤差が大きくなり、軸が指令値に追従できなくなります。この動作状態で、トルク制限の値を上げると、突発的に、ガタガタするような移動が発生する場合があります。軸が必ず指令値に追従できるように、輪郭誤差をチェックして、無制限のトルクにより、誤差が大きくなり過ぎないようにしてください。

3.14.7 ドウエル時間(G4)

命令 G4 では、メインランで実行されると同時に期限切れになるブロックで時間(ドウエル時間)がプログラムされます。次のブロックへのブロック変更は、時間が完全に時間切れになると同時に実行されます。

注記

G4 は、連続軌跡モードを中断します。

構文

G4 F<時間>

```
G4 S<NumSpi>
G4 S<n> = <NumSpi>
```

意味

G4:	ドウェル時間を有効にします。	
	単独ブロック 指令:	可
F<時間>:	ウェル時間 <時間>をアドレス F で秒単位で指定します。	
S<NumSpi>:	ドウェル時間をアドレス S で、現在の主軸を基準にした主軸の回転数<NumSpi>でプログラムします。	
S<n>=NumSpi >:	ドウェル時間をアドレス S で、アドレス拡張子<n>でアドレス指定された主軸を基準にした回転数<NumSpi>でプログラムします。	

注記

ドウェルブロック G4 で指定された時間に使用されるアドレス F および S は、プログラムの送り速度 F...および主軸速度 S...に影響を及ぼしません。

必要条件

シンクロナイズドアクション

2つのシンクロナイズドアクションを、ドウェル時間を持つ後続のブロックがシンクロナイズドアクションが実行されるアクションブロックとなるよう1つのプログラムでプログラム指令します。一方のシンクロナイズドアクションはモーダルシンクロナイズドアクションです。もう一方のシンクロナイズドアクションはノンモーダルシンクロナイズドアクションです。ノンモーダルシンクロナイズドアクションがモーダルシンクロナイズドアクションに影響を及ぼすよう意図されている場合、たとえば、UNLOCKでの実行のためにリリースするような場合は、**最低でも2つの補間サイクル**、たとえばG4

F<interpolator_cycle * 2>を有効なドウェル時間として提供する必要があります。

有効なドウェル時間は、マシンデータ MD10280 \$MN_PROG_FUNCTION_MASK, Bit 4 = <値>の設定によって異なります。

値	意味
0	有効なドウェル時間は、プログラムされたドウェル時間と等しくなります。
1	有効なドウェル時間は、次に大きい補間周期の倍数に丸められたプログラムされたドウェル時間と等しくなります(MD10071 \$MN_IPO_CYCLE_TIME)。

3.14 補助命令

プログラム例:

- MD10071 \$MN_IPO_CYCLE_TIME == 8 ms
- MD10280 \$MN_PROG_FUNCTION_MASK, Bit 4 = 1

プログラムコード	コメント
N10 WHEN TRUE DO LOCK(1)	; ノンモーダル SynAct.; モーダル SynAct.の LOCKID=1
N20 G4 F2 ; SynAct のアクションブロック	WHEN TRUE DO UNLOCK(1) の N10 から N30 の ; ノンモーダル SynAct:UNLOCK ; モーダル SynAct.の ID=1
N40 ID=1 WHENEVER TRUE DO \$R0=1 RDISABLE	; モーダル SynAct ID=1 ; R 変数 R0=1 ; 読み込み停止の設定
N50 G4 F0.012	; SynAct from N40 および N50 のアクションブロック ; 下記のパラグラフ「説明」を参照
N60 G4 F10	

説明

望ましい動作は、N30 チャンネルからのモーダルシンクロナイズドアクションが N40 からの ID=1 のモーダルシンクロナイズドアクションの有効なロック(LOCK)をキャンセルし、R 変数が N50 に書き込まれて、読み込み停止が有効になるようにすることです。この動作は、有効なドウェル時間の長さが最低でも 2 補間周期である場合にのみ実現されます。

有効なドウェル時間は、プログラムされたドウェル時間、補間周期、MD10280 \$MN_PROG_FUNCTION_MASK, Bit 4 の設定によってもたらされます。有効なドウェル時間の長さが最低でも 2 補間周期となることを保証するには、以下のドウェル時間をプログラムする必要があります。

- ビット 4 == 0: プログラムされたドウェル時間 $\geq 2 * \text{補間周期}$
- ビット 4 == 1: プログラムされたドウェル時間 $\geq 1.5 * \text{補間周期}$

有効なドウェル時間が 2 補間周期よりも短い場合、R 変数の書き込みと読み込み停止はブロック N60 まで実行されません。

例

プログラムコード	コメント
N10 G1 F200 Z-5 S300 M3	; 送り速度 F、主軸速度 S
N20 G4 F3	; ドウェル時間: 3 秒
N30 X40 Y10	
N40 G4 S30	; 主軸の 30 回転のドウェルです (S=300 rpm、および 100% の速度オーバーライドによる、 $t = 0.1 \text{ min}$ に相当)。

プログラムコード	コメント
N50 X...	; N10 のプログラム指令送り速度と主軸速度が、引き続き適用されます。

3.14.8 内部先読み停止

機能

制御装置は、機械状態データ(\$A...)へのアクセス時に内部の先読み停止をおこないます。次のブロックは、先読みがおこなわれて保存されたすべてのブロックが全て実行されるまで、実行されません。前のブロックは(G9 として)イグザクトストップで停止します。

例

プログラムコード	コメント
...	
N40 POSA[X]=100	
N50 IF \$AA_IM[X]==R100 GOTOF MARKE1	; 機械状態データ(\$A...)へのアクセス、制御装置は内部の先読み停止をおこないます。
N60 G0 Y100	
N70 WAITP(X)	
N80 LABEL1:	
...	

3.15 その他の情報

3.15.1 軸

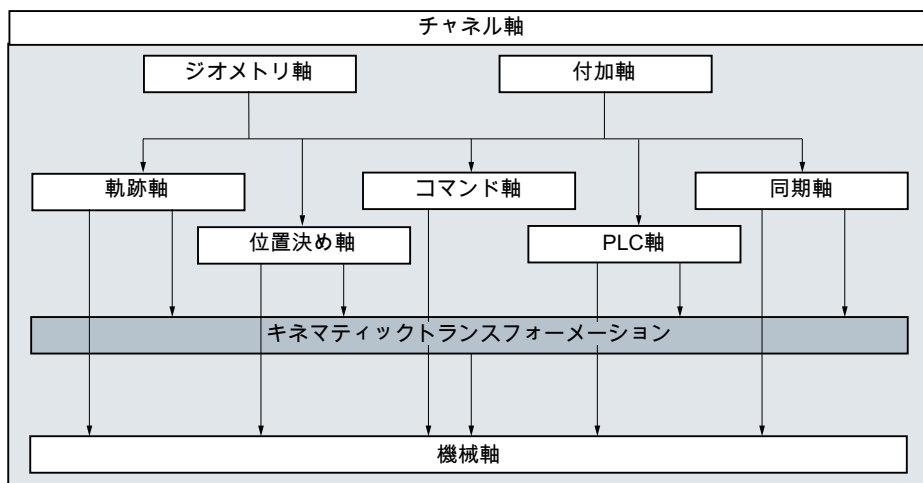
軸タイプ

プログラム指令時には、次の軸タイプに区別されます。

- メイン軸/ジオメトリ軸
- 付加軸
- 主軸、メイン主軸
- 機械軸

3.15 その他の情報

- チャネル軸
- 軌跡軸
- 位置決め軸
- 同期軸
- コマンド軸
- PLC 軸/競合位置決め軸
- リンク軸
- マスタリンク軸



3.15.1.1 メイン軸/ジオメトリ軸

メイン軸は、右手直交座標系を定義します。工具の移動は、この座標系でプログラム指令します。

NC 加工では、メイン軸をジオメトリ軸と呼びます。この用語は、このプログラミング説明書でも使用します。

置換可能なジオメトリ軸

「置換可能なジオメトリ軸」機能(プログラミング説明書、上級編を参照してください)を使用すると、パートプログラムから、マシンデータを使用して設定したジオメトリ軸グループを変更できます。この場合、ジオメトリ軸はどれも、同期付加軸として定義されたチャネル軸に置換できます。

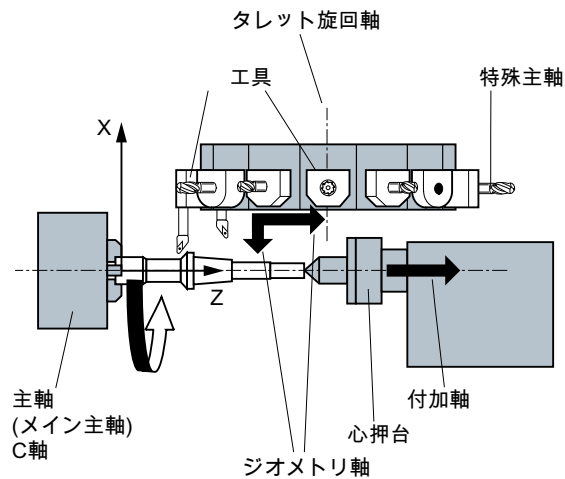
軸識別子

ジオメトリ軸の名称/識別子は、次のマシンデータを使用して定義できます。

MD20060 \$MC_AXCONF_GEOAX_NAME_TAB (チャンネル内でのジオメトリ軸の名称)

旋盤の標準識別子

1. ジオメトリ軸:X
2. ジオメトリ軸:Z



フライス盤の標準識別子

1. ジオメトリ軸:X
2. ジオメトリ軸:Y
3. ジオメトリ軸:Z

詳細情報

最大3つのジオメトリ軸を使用して、フレームとワーク形状(輪郭)をプログラム指令します。

ジオメトリ軸とチャンネル軸の識別子は、基準が同じであれば、同一にできる場合もあります。

ジオメトリとチャンネル軸名称は、すべてのチャンネルで同一してください。つまり、プログラムはどのチャンネルでも実行できます。

3.15.1.2 付加軸

ジオメトリ軸とは違って、付加軸間では、幾何学的な関係は定義されていません。

代表的な付加軸は以下の軸です。

- 工具リボルバ軸
- 旋回テーブル軸

3.15 その他の情報

- 旋回ヘッド軸
- ローダー軸

軸識別子

円形マガジンを装備した旋盤の例は、次のとおりです。

- レボルバ位置 U
- 心押し台 V

プログラミング例

プログラムコード	コメント
N10 G1 X100 Y20 Z30 A40 F300	; 軌跡軸の移動
N20 POS[U]=10POS[X]=20 FA[U]=200 FA[X]=350	; 位置決め軸の移動
N30 G1 X500 Y80 POS[U]=150FA[U]=300 F550	; 軌跡軸と位置決め軸
N40 G74 X1=0 Z1=0	; 基準点にアプローチします。

3.15.1.3 主軸、メイン主軸

機械のキネマティックスにより、どの主軸がメイン主軸であるかを特定します。一般に、この主軸は、マシンデータで第1主軸（メイン主軸）として宣言されます。

この割り当ては、SETMS (<主軸番号>) プログラム命令で変更できます。主軸番号を指定せずに SETMS を使用すると、マシンデータで定義したメイン主軸に割り当てを戻すことができます。

メイン主軸は、ねじ切りなどの応用機能をサポートしています。

主軸識別子

S または S0

3.15.1.4 機械軸

機械軸は、機械に物理的に存在する軸です。

プログラム指令された軌跡軸または付加軸の移動は、チャンネルで有効な座標変換 (TRANSMIT、TRACYL、または TRAORI) により、複数の機械軸で実行できます。

機械軸は、特別な場合(レファレンス点復帰や固定点アプローチなど)にのみ、プログラムで直接アドレス指令されます。

軸識別子

機械軸の名称/識別子は、次の NC 別マシンデータを使用して定義できます。

MD10000 \$MN_AXCONF_MACHAX_NAME_TAB (機械軸名称)

初期設定 X1、Y1、Z1、A1、B1、C1、U1、V1

さらに、機械軸には次の固定の軸識別子があり、これはマシンデータで設定された名称とは関係なく、常に使用できます。

AX1、AX2、…、AX<n>

3.15.1.5 チャネル軸

チャネルに割り当てられたジオメトリ軸、付加軸、および機械軸すべてを、チャネル軸と呼びます。

軸識別子

ジオメトリ軸と付加軸のチャネル別名称/識別子は、次のマシンデータを使用して定義できます。

MD20080 \$MC_AXCONF_CHANAX_NAME_TAB (チャネル軸名称)

初期設定 X、Y、Z、A、B、C、U、V

チャネルのジオメトリ軸または付加軸をエミュレーションする機械軸の割り当ては、次のマシンデータで定義されます。

MD20070 \$MC_AXCONF_MACHAX_USED (使用される機械軸)

3.15.1.6 軌跡軸

軌跡軸は、軌跡を定義し、空間の工具の移動を定義します。

この軌跡に対しては、プログラム指令送り速度が有効です。この軌跡に関与する各軸は、その位置に同時に到達します。一般に、これらはジオメトリ軸です。

ただし、初期設定により、どの軸が軌跡軸であるかが定義され、速度が特定されます。

軌跡軸は、NC プログラムで FGROUP を使用して指定できます。

FGROUP の詳細については、送り速度(G93、G94、G95、F、FGROUP、FL、FGREF) (ページ 119)を参照してください。

3.15.1.7 位置決め軸

位置決め軸は、個別に補間されます。つまり、各位置決め軸には、個別に軸補間器と送り速度があります。位置決め軸は軌跡軸と一緒に補間しません。

3.15 その他の情報

位置決め軸は、NC プログラムまたは PLC により移動します。軸が、NC プログラムと PLC の両方で同時に移動する場合は、エラーメッセージが表示されます。

代表的な位置決め軸は、次のとおりです。

- ワークを機械へ搬送するローダ
- ワークを機械から取り出すローダ
- 工具マガジン/タレット

タイプ

同期制御位置決め軸は、ブロックの終点で同期する場合、または複数のブロックにまたがる場合で区別されます。

POS 軸

このブロックでプログラム指令したすべての軌跡軸と位置決め軸が、プログラム指令終点に到達すると、ブロックの終点でブロック切り替えがおこなわれます。

POSA 軸

これらの位置決め軸の移動は、複数のブロックに拡張できます。

POSP 軸

終了位置へアプローチするためのこれらの位置決め軸の移動は、この区間でおこなわれます。

注記

位置決め軸を特別な POS/POSA 識別子なしで移動した場合は、同期軸となります。

軌跡軸の連続軌跡モード(G64)を使用できるのは、位置決め軸(POS)がその最終位置に、軌跡軸より先に到達する場合のみです。

POS/POSA でプログラム指令された軌跡軸は、このブロックが持続する間は軌跡軸グループから削除されます。

POS、POSA、および POSP について詳しくは、「位置決め軸の移動(POS、POSA、POSP、FA、WAITP、WAITMC) (ページ 129)」を参照してください。

3.15.1.8 同期軸

同期軸は、軌跡に同期して、開始位置から、プログラム指令終了位置まで移動します。

F のプログラム指令送り速度は、ブロックでプログラム指令されたすべての軌跡軸に適用されますが、同期軸には適用されません。同期軸の移動にかかる時間は、軌跡軸と同じです。

同期軸は、軌跡補間に同期して移動する回転軸でも可能です。

3.15.1.9 コマンド軸

コマンド軸は、事象(命令)に応答して、シンクロナイズドアクションから起動します。この軸は、パートプログラムとはまったく非同期で位置決め、起動、および停止できます。軸は、パートプログラムとシンクロナイズドアクションから同時に移動できません。

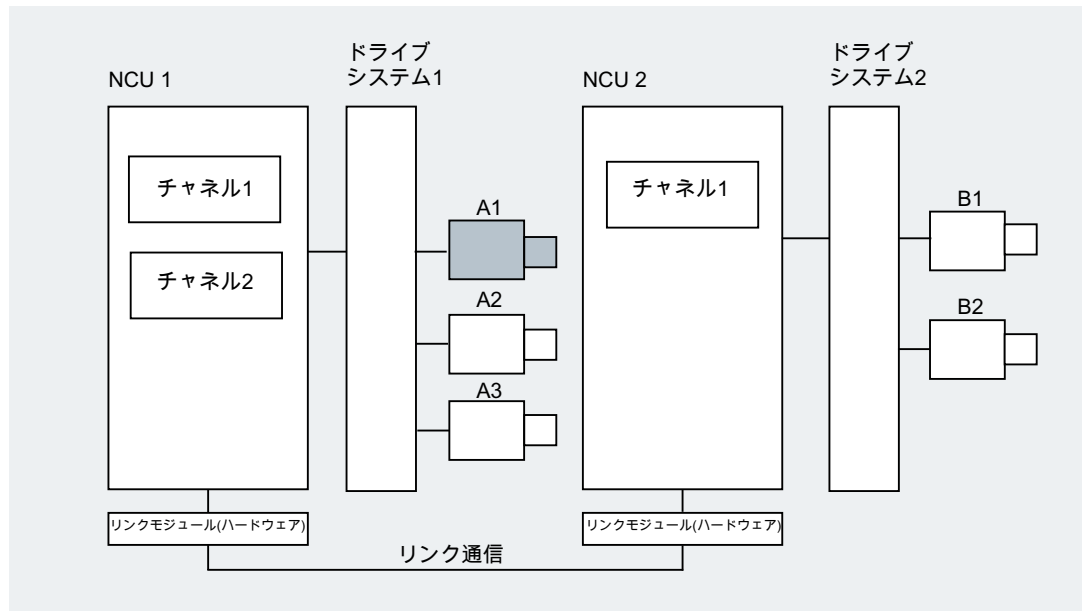
コマンド軸は、別に補間されます。つまり、各コマンド軸には、個別に軸補間器と送り速度があります。

3.15.1.10 PLC 軸

PLC 軸は PLC で、基本プログラムの応用ファンクションブロックによって移動されて、その移動は、その他のすべての軸と非同期になることができます。この移動は、軌跡移動と同期移動と無関係におこなわれます。

3.15.1.11 リンク軸

リンク軸は、別の NCU に物理的に接続し、その位置が当該 NCU から制御される軸です。リンク軸は、別の NCU のチャンネルにダイナミックに割り当てることができます。リンク軸は、特定の NCU の観点からするとローカル軸ではありません。



軸コンテナの概念は、NCU への割り当てのダイナミックな変更のために使用されます。パートプログラムからの GET と RELEASE による軸入れ替えは、リンク軸には**使用できません**。

詳細情報

必要条件

- 関与する NCU、NCU1、および NCU2 は、リンクモジュールによる高速通信を使用して接続してください。
- この軸は、マシンデータによって適切に設定してください。
- 「リンク軸」オプションをインストールしてください。

説明

位置制御は、その軸がドライブに物理的に接続された NCU で実行されます。この NCU はまた、関連する軸の VDI インタフェースを含んでいます。リンク軸の位置指令値は、別の NCU で生成され、NCU リンクによって伝送されます。

リンク通信は、補間器と、位置コントローラまたは PLC インタフェース間の連携動作を保証する必要があります。補間器が計算した指令値が、ホーム NCU の位置制御ループに送られ、逆に、フィードバック値がホーム NCU から補間器に戻される必要があります。

軸コンテナ

軸コンテナはリングバッファデータ構造であり、ここでローカル軸とリンク軸のいずれか、または両方がチャンネルに割り当てられます。リングバッファの入力値は、**周期的にシフト**できます。

ローカル軸またはリンク軸への直接参照に加えて、論理機械軸イメージでのリンク軸設定により、軸コンテナへの参照も可能です。このタイプの参照は、以下の項目から構成されます。

- コンテナ番号および
- スロット(コンテナ内のリングバッファの位置)

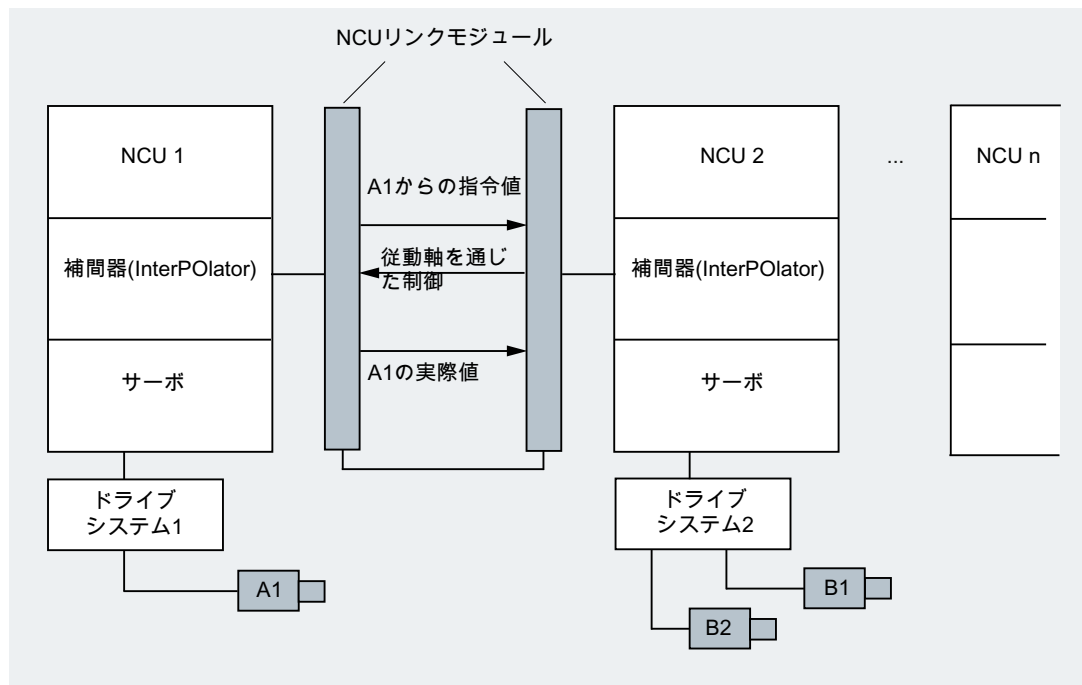
リングバッファロケーションの入力には、以下が含まれます。

- ローカル軸または
- リンク軸

1 基の NCU からみて、軸コンテナの入力値はローカル機械軸またはリンク軸を含みます。1 基の NCU の論理機械軸イメージ(MD10002 \$MN_AXCONF_LOGIC_MACHAX_TAB)の入力値は固定値です。

3.15.1.12 マスタリンク軸

マスタリンク軸とは、1 基の NCU が補間し、1 基または複数の他の NCU がスレーブ軸の制御用メイン軸として使用するリンク軸です。



軸位置コントローラアラームは、対応する軸にマスタリンク軸経由で接続されている他のすべての NCU に送信されます。

マスタリンク軸に依存する NCU は、この軸により、次の連結関係を使用できます。

- マスタ値(指令値、実マスタ値、シミュレートマスタ値)
- 連結移動
- 法線方向制御
- 電子ギア(ELG)
- 主軸同期

プログラミング

マスタ NCU:

マスタ値軸に物理的に割り当てられた NCU のみが、この軸の移動動作をプログラム指令できます。移動プログラムには、応用機能または命令を入れないでください。

スレーブ軸の NCU:

スレーブ軸の NCU の移動プログラムには、マスタリンク軸(マスタ値軸)の移動指令を入れないでください。この規則に違反するとアラームが発生します。

マスタリンク軸は、チャンネル軸識別子によって通常の方法で処理されます。マスタリンク軸のステータスには、選択したシステム変数を使用してアクセスできます。

詳細情報

必要条件

- 関連する NCU、つまり、NCU1～NCU<n> (<n>は 8 以下の数字)は、高速通信用リンクモジュールで内部接続してください。
参照先:
セットアップ説明書 NCU
- この軸は、マシンデータによって適切に設定してください。
- 「リンク軸」オプションをインストールしてください。
- 関与するすべての NCU に対して同じ補間クロックサイクルを設定してください。

制限事項

- メイン軸がマスタリンク軸の場合、メイン軸はリンク軸にすることはできません。つまり、メイン軸のあるホーム NCU 以外の NCU では、メイン軸を移動できません。
- メイン軸がマスタリンク軸の場合、メイン軸はコンテナ軸にすることはできません。つまり、異なる NCU が交互にメイン軸にアクセスすることはできません。
- マスタリンク軸は、ガントリグループのプログラム指令マスタ軸として機能することはできません。
- マスタリンク軸との連結は、カスケードにすることはできません。
- 軸入れ替えは、マスタリンク軸のホーム NCU 内でのみ実行できます。

システム変数

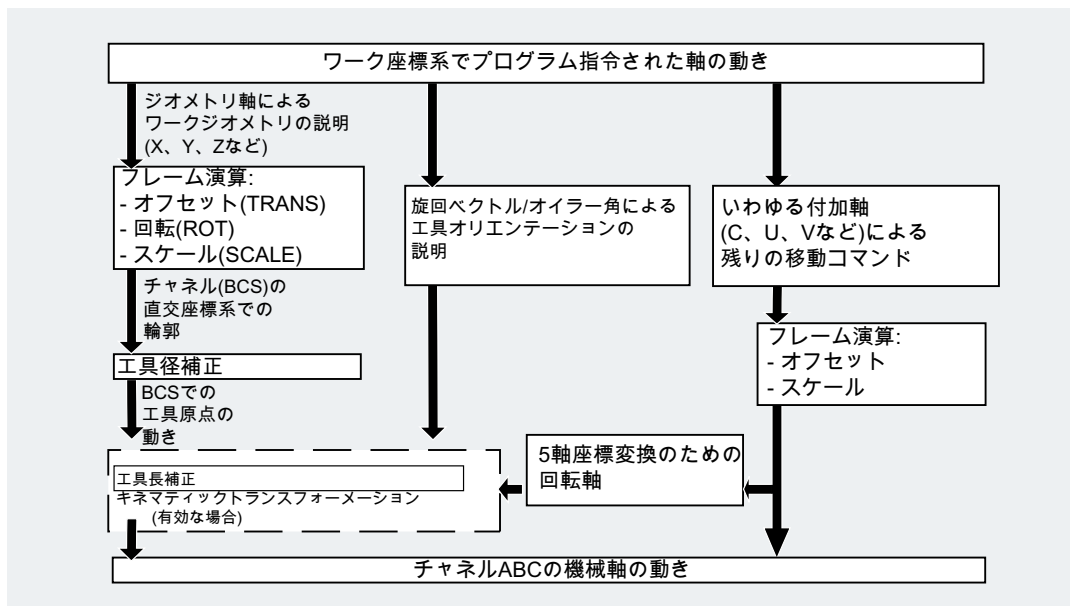
次のシステム変数は、マスタリンク軸のチャンネル軸識別子と組み合わせて使用できます。

システム変数	意味
\$AA_LEAD_SP	シミュレーションされたマスタ値-位置
\$AA_LEAD_SV	シミュレーションされたマスタ値-速度

これらのシステム変数がマスタ軸のホーム NCU によって更新される場合、このマスタ軸に応じてスレーブ軸の制御を必要とするその他の NCU にも同様に新しい値が伝送されます。

3.15.2 移動指令から機械移動まで

プログラム指令軸移動(移動指令)とその結果である機械移動の関係を、次の図に示します。

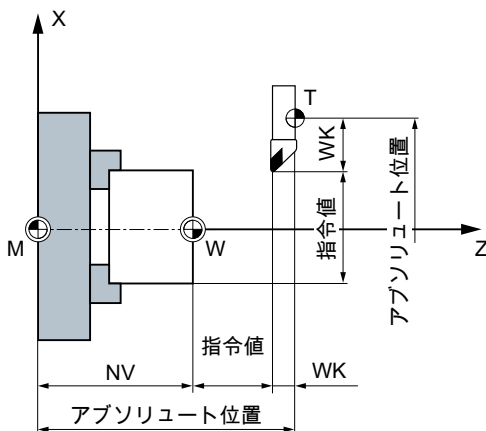


3.15.3 軌跡演算

軌跡演算は、すべてのオフセットと補正を考慮して、ブロックの移動距離を特定します。

一般的に:

$$\text{距離} = \text{指令値} - \text{現在位置} + \text{ゼロオフセット(ZO)} + \text{工具オフセット(TO)}$$

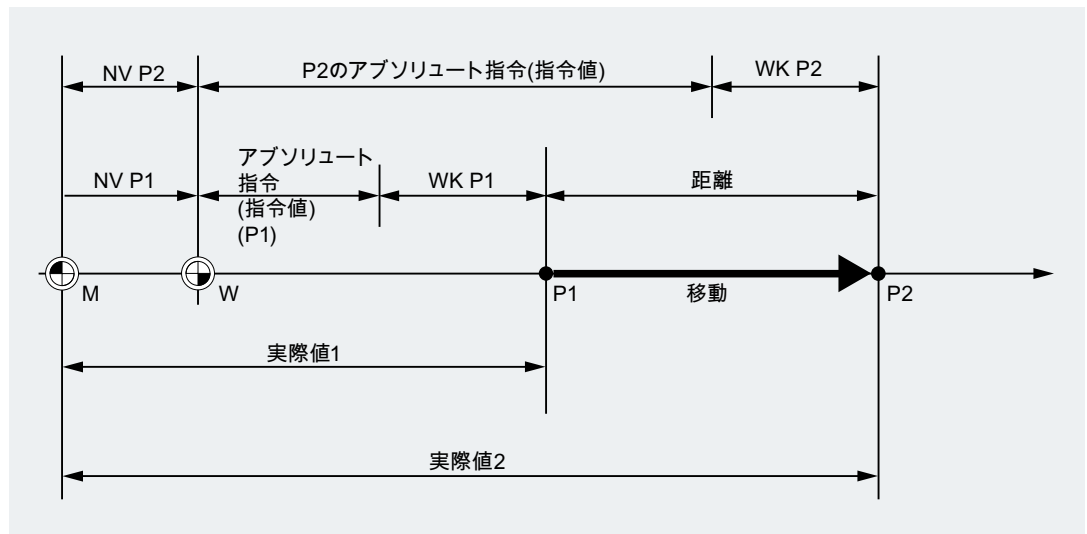


新しいゼロオフセットと新しい工具オフセットを新しいプログラムブロックでプログラム指令した場合は、次の条件が適用されます。

- アブソリュート指令を使用する場合:

$$\text{距離} = (\text{アブソリュート指令 P2} - \text{アブソリュート指令 P1}) + (\text{WO P2} - \text{WO P1}) + (\text{TO P2} - \text{TO P1})$$
- インクリメンタル指令を使用する場合:

$$\text{距離} = \text{インクリメンタル指令} + (\text{WO P2} - \text{WO P1}) + (\text{TO P2} - \text{TO P1})$$



3.15.4 アドレス

固定アドレス

このアドレスは、固定値として設定されています。つまり、アドレス文字を変更できません。

一覧は表「固定アドレス (ページ 1486)」を参照してください。

3.15 その他の情報

設定可能アドレス

工作機械メーカーは、マシンデータで、このアドレスに別の名称を割り当てることが出来ます。

注記

設定可能アドレスは、制御装置内では固有にしてください。つまり、異なるアドレスタイプ(軸値と終点、工具オリエンテーション、補間パラメータなど)に同じアドレス名称を使用しないでください

一覧は表「設定可能アドレス (ページ 1492)」を参照してください。

モーダルアドレス/ノンモーダルアドレス

プログラム指令モーダルアドレスは、新しい値を同じアドレスにプログラム指令するまでは(後続のすべてのブロックで)有効です。

ノンモーダルアドレスは、それがプログラム指令されたブロックでのみ適用されます。

例:

プログラムコード	コメント
N10 G01 F500 X10	
N20 X10	; N10 からの送り速度 F は、新しい送り速度を入力するまで、そのまま有効です。

軸拡張子を含むアドレス

軸拡張子を含むアドレスでは、軸名称が、アドレスの後に角括弧で囲んで入れられます。軸名称によって軸を割り当てます。

例:

プログラムコード	コメント
FA[U]=400	; U 軸の軸別送り速度。

表「固定アドレス (ページ 1486)」も参照してください。

拡張アドレス表記

拡張アドレス表記により、より多くの軸、および主軸をシステムで構成できます。

拡張アドレスは、数値拡張子と「=」記号で割り当てた算術式で構成されます。数値拡張子は 1 桁または 2 桁で、常に正の数です。

拡張アドレス表記は、以下の直接アドレスにのみ使用できます。

アドレス	意味
X、Y、Z、...	軸アドレス
i、j、k	補間パラメータ
S	主軸速度
SPOS、SPOSA	主軸位置決め
M	応用機能
H	補助機能
T	工具番号
F	送り速度

例:

プログラムコード	コメント
X7	; 「=」は不要で、7は値ですが、ここで「=」を使用することもできます
X4=20	; 軸 X4; 「=」が必要です
CR=7.3	; 2文字の英字; 「=」が必要です
S1=470	; 1番目の主軸の速度:470 rpm
M3=5	; 3番目の主軸の主軸停止

数値拡張子の代わりにアドレス M、H、S、および SPOS と SPOSA 用の変数を使用できます。変数識別子は角括弧で囲みます。

例:

プログラムコード	コメント
S[SPINU]=470	; 主軸の速度で、その数値が SPINU 変数に保存されます。
M[SPINU]=3	; 主軸の右回転で、その数値が SPINU 変数に保存されます。
T[SPINU]=7	; 主軸の工具選択で、その番号が SPINU 変数に保存されます。

3.15.5 名称

DIN 66025 準拠の命令には、NC 高機能言語により、識別子が補足されます。

3.15 その他の情報

識別子の例:

- システム変数
- ユーザー定義変数
- 軸/主軸
- サブプログラム
- キーワード
- ジャンプマーク
- マクロ

注記

識別子は一義的にしてください。異なる複数の対象には同じ識別子を使用でき **ません**。

命名規則

名称は、以下の規則に従って自由に選択できます。

- 使用可能な文字:
 - 英字:A ... Z、a ... z
 - 数字:0 ... 9
 - アンダースコア:_
- 最初の 2 文字は、英字またはアンダースコアにしてください。
- 最大長さ:
 - プログラム名称 (ページ 50):24 文字
 - 軸名称:8 文字
 - 変数名:31 文字

注記

予約キーワードは識別子として使用しないでください。

サイクル

名前の競合を避けるために、ユーザーサイクルの名称の割り当てには、次の仕様を遵守することをお勧めします。

文字列	以下の名称に予約済み
• CYCLE • CUST_ • GROUP_ • _ • S_ • E_ • F_	SIEMENS サイクル
• CCS_	SIEMENS コンパイルサイクル
• CC_	ユーザーコンパイルサイクル

ユーザーサイクル

ユーザーサイクルの名称の先頭に U_ を付けることをお勧めします。

変数

変数の名前割り付けに関する情報は、以下の章に記載されています。

- システムデータ (ページ 434)
- ユーザー変数の定義(DEF) (ページ 443)

3.15.6 定数

定数(一般)

定数とは、アドレスへの値割り当てなど、プログラムの実行中に値を変更できないデータ要素です。

10 進定数

10 進定数の数値は、10 進法で表示されます。

INTEGER 定数

INTEGER 定数は整数値です。つまり、符号ありまたは符号なしの小数点のない数字列です。

例:

X10	アドレス X へ値+10 を割り当て
X -35	アドレス X へ値-35 を割り当て
X0	アドレス X へ値 0 を割り当て 注: X0 の代わりに X を使用することはできません。

REAL 定数

REAL 定数は、小数点あり、符号ありまたは符号なし、および指数ありまたは指数なしの数字列です。

例:

X10.25	アドレス X へ値+10.25 を割り当て
X -10.25	アドレス X へ値-10.25 を割り当て
X0.25	アドレス X へ値+0.25 を割り当て
X.25	先頭の「0」なしで、アドレス X へ値+0.25 を割り当て
X=-.1EX-3	アドレス X へ値-0.1*10 ⁻³ を割り当て

注記

小数点の入力が可能なアドレスで、アドレスに使用できる数より多くの小数点以下の桁数を指定した場合は、有効な桁数に丸められます。

16 進数定数

定数は、16 進法、つまり 16 を基数として解釈することもできます。文字 A~F は、10 進値 10~15 に相当する 16 進数字です。

16 進数定数は、一重引用符で囲まれた、先頭の「H」の後に 16 進法表記の値が続く数値です。英字と数字の間には、区切り文字を使用できます。

例:

プログラムコード	コメント
\$MC_TOOL_MANAGEMENT_MASK='H7F'	; 16 進数定数を割り当てて、マシンデータのビット 0～7 を設定します。

注記

最大文字数は、整数データタイプの数値の範囲に制限されます。

2 進数定数

定数は、2 進法で解釈することもできます。この場合は、「0」と「1」のみを使用します。

2 進数定数は、一重引用符で囲まれた、先頭の「B」の後に 2 進法表記の値が続く数値です。各桁の間には、区切り文字を使用できます。

例:

プログラムコード	コメント
\$MN_AUXFU_GROUP_SPEC='B10000001'	; 2 進数定数を割り当てて、マシンデータのビット 0 とビット 7 を設定します。

注記

最大文字数は、整数データタイプの数値の範囲に制限されます。

3.15.7 演算子および算術機能

演算子

算術演算子

REAL および INT タイプのシステム変数は以下の演算子によってリンクすることができます。

演算子	意味
+	加算
-	減算
*	乗算

3.15 その他の情報

演算子	意味
/	- シンクロナイズドアクションでの除算: $\text{INT} / \text{INT} \Rightarrow \text{INT}$ - ファンクション <code>ITOR()</code> を使用したシンクロナイズドアクションの除算、結果は <code>REAL</code>: $\text{ITOR}(\text{INT}) / \text{ITOR}(\text{INT}) \Rightarrow \text{REAL}$ - NC プログラムでの除算: $\text{INT} / \text{INT} \Rightarrow \text{REAL}$
DIV	整数の除算: $\text{INT} / \text{INT} \Rightarrow \text{INT}$
MOD	モジュロ除算(<code>INT</code> タイプの場合のみ)では、 <code>INT</code> 除算の余りとなります。 例: $3 \text{ MOD } 4 = 3$

注記

同一タイプの変数のみこれらの演算によりリンクされます。

関係演算子

演算子	意味
==	等しい
>	等しくない
<	より小さい
>	不等号 (より大)
<=	等しいかそれより小さい
>=	等しいかそれより大きい

ブール演算子

演算子	意味
NOT	NOT
および	および
または	または
XOR	排他的論理和

ビット論理演算子

演算子	意味
B_OR	ビットごとの OR
B_AND	ビットごとの AND

演算子	意味
B_XOR	ビットごとの排他的 OR
B_NOT	ビットごとの否定

演算子の優先度

シンクロナイズドアクションで演算子は次の優先度を持ちます(最優先:1):

優先順位	演算子	意味
1	NOT, B_NOT	否定、ビットごとの否定
2	*, /, DIV, MOD	乗算、除算
3	+, -	加算、減算
4	B_AND	ビットごとの AND
5	B_XOR	ビットごとの排他的 OR
6	B_OR	ビットごとの OR
7	および	および
8	XOR	排他的論理和
9	または	または
10	<<	文字列の結合、結果は STRING タイプ
11	==, <>, <, >, >=, <=	関係演算子

注記

式で複数の演算子を使用する場合は、括弧"(...)"を使用して個々の演算子の優先度を明確にすることを強く推奨します。

複数の演算子を含む式の条件の例:

プログラムコード

```
...WHEN ($AA_IM[X] > VALUE) AND ($AA_IM[Y] > VALUE1) DO ...
```

算術機能

演算子	意味
SIN()	サイン
COS()	コサイン

3.15 その他の情報

演算子	意味
TAN()	タンジェント
ASIN()	アークサイン
ACOS()	アークコサイン
ATAN2()	アークタンジェント 2
SQRT()	平方根
ABS()	絶対値
POT()	2 乗(平方)
TRUNC()	整数要素 比較演算命令の精度は TRUNC で設定できます。
ROUND()	整数に丸める
LN()	自然対数
EXP()	指数関数

インデックス

タイプ「…の配列」のシステム変数のインデックスをシステム変数にすることができます。インデックスは補間器クロックサイクルのメインランで使用することもできます。

例

プログラムコード <pre>...WHEN ... DO \$AC_PARAM[\$AC_MARKER[1]]=3</pre>

制限事項

- インデックスをその他のシステム変数とネストすることは許可されていません。
- インデックスは事前処理変数から作成してはいけません。以下の例は、\$P_EP が事前処理変数なので許可されていません:
\$AC_PARAM[1] = \$P_EP[\$AC_MARKER[0]]

下記も参照

算術機能 (ページ 494)

命令 (ページ 494)

作業の準備

4.1 フレキシブルな NC プログラミング

4.1.1 変数

特に算術機能とチェック命令を組み合わせることでシステムデータおよびユーザーデータの領域からの変数を使用すると、非常に高いレベルの柔軟性を持った NC プログラムとサイクルの作成が可能になります。

警告

変数の変更によって引き起こされる物的損害と人身事故

NC プログラムで変数を使用する場合、対応するアクセス権を持つ機械オペレータまたは第 3 者が変数を変更し、プログラムの実行に影響を受ける可能性があることを考慮してください。これにより、物的損害および人身事故が生じるおそれがあります。

- 変数の変更によってプログラムの実行に悪影響が出るのを避けるために、適切なデータチェック(「入力の検証」)を NC プログラムでおこなってください。

4.1 フレキシブルなNCプログラミング

- システムデータ

システムデータには、システムで事前定義された変数が含まれています。これらの変数には定義された意味があります。これらの変数はおもにシステムソフトウェアで使用されます。ユーザーはNCプログラムおよびサイクルでこれらの変数の読み取りと書き込みができます。例:マシンデータ、セッティングデータ、システム変数。

システムデータアイテムの意味は固定されていますが、ユーザーは再定義によって特定の制限内で特性を変更することができます。

「システムデータ、ユーザーデータ、NC 命令(REDEF)の再定義 (ページ 450)」を参照してください

- ユーザーデータ

ユーザーデータには、ユーザーによって定義された変数が含まれています。変数の意味は、ユーザーだけが定義できます。これらの変数はシステムに評価されません。

ユーザーデータは以下に分けられます。

- 予約ユーザー変数

予約ユーザー変数はシステム内で定義済みの変数で、その番号はマシンデータでパラメータ設定されています。ユーザーはこれらの変数のプロパティを変更できます。

「システムデータ、ユーザーデータ、NC 命令(REDEF)の再定義 (ページ 450)」を参照してください

- ユーザー定義変数

ユーザー定義変数は、ユーザーが定義し、実行時まではシステムが作成しない変数です。その番号、データタイプ、可視性、およびその他の機能は、ユーザーが定義します。

「ユーザー変数の定義(DEF) (ページ 443)」を参照してください

4.1.1.1 システムデータ

システムデータには、システムで事前定義され、NCプログラムとサイクルのなかで、コントローラの現在のパラメータ設定へのアクセス、加えて機械状態、制御状態、および処理状態へのアクセスを可能にする変数が含まれています。

先読み変数

先読み変数は先読み中に、つまり、この変数が含まれているブロックが解釈されるときに読み取りおよび書き込みされるシステムデータです。先読み変数は、先読み停止をおこないません。

メインラン変数

先読み変数はメインラン中に、つまり、この変数が含まれているブロックが実行されるときに読み取りおよび書き込みされるシステムデータです。以下にメインラン変数を示します。

- シンクロナイズドアクションでプログラム指令できる変数(読み取り/書き込み)
- NC プログラムでプログラム指令して、先読み停止ができる変数です(読み取り/書き込み)。
- NC プログラムでプログラム指令でき、値は先読み中に計算されますが、メインランまでは書き込まれない変数です(メインランと同期:書き込み専用)

接頭語方式

システムデータとそれ以外のデータを区別するために、システムデータの前には通常、\$記号の後に 1 文字または 2 文字とアンダーバーのついた接頭語が付けられます。

\$ + 1 文字	意味:データタイプ
先読みデータ (先読み中に読み取り/書き込みがおこなわれるシステムデータ)	
\$M	マシンデータ ¹⁾
\$S	セッティングデータ、プロテクションゾーン ¹⁾
\$T	工具管理データ
\$P	プログラム指令値
\$C	ISO サイクルのサイクル変数
\$O	オプションデータ
R	R 変数(算術変数) ²⁾
メインランデータ (メインラン中に読み取り/書き込みがおこなわれるシステムデータ)	
\$\$M	マシンデータ ¹⁾
\$\$S	セッティングデータ ¹⁾
\$A	現在のメインランデータ
\$V	位置コントローラデータ

4.1 フレキシブルな NC プログラミング

\$ + 1 文字	意味:データタイプ
\$R	R 変数(算術変数) ²⁾
¹⁾ 機械データとセッティングデータを先読み変数とメインラン変数のどちらとして扱うかは、データに付けた\$記号が 1 つか 2 つかによって決まります。表記はその用途に応じて自由に選択できます。 ²⁾ パートプログラム/サイクルで R 変数を先読み変数として使用するときは、たとえば R10 のように接頭語を省略してください。シンクロナイズドアクションでメインラン変数として使用するときは、\$R10 のように、\$記号を接頭語として書いてください。	

2 番目の文字	意味:可視性
N	NC グローバル変数(NC)
C	チャンネル別変数(チャンネル)
A	軸別変数(軸)

必要条件

接頭語方式での例外

以下の変数方式は、上記で指定した接頭語方式とは異なっています。

- \$TC_...:ここでは、2 番目の文字 C はチャンネル別システム変数ではなく、工具ホルダ用システム変数を表しています(TC=工具ホルダ)。
- \$P_...:チャンネル別システム変数

シンクロナイズドアクションでマシンデータとセッティングデータに使用

マシンデータとセッティングデータをシンクロナイズドアクションで使用するときは、接頭語を使用して、マシンデータまたはセッティングデータを、先読みの実行とメインランのどちらに同期して読み取るまたは書き込むかを定義できます。

加工中にデータを変更しない場合は、先読みの実行に同期して読み取ることができます。このために、マシンデータまたはセッティングデータの接頭語は、1 個の\$記号で書いてください。

```
ID=1 WHENEVER $AA_IM[z] < $SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```


加工中にデータを変更する場合は、メインランに同期して読み取り/書き込みをおこなってください。このために、マシンデータまたはセッティングデータの接頭語は、2 個の\$記号で書いてください。

```
ID=1 WHENEVER $AA_IM[z] < $$SA_OSCILL_REVERSE_POS2[Z]-6 DO $AA_OVR[X]=0
```

注記

マシンデータとセッティングデータの書き込み

マシンデータまたはセッティングデータの項目を書き込むときは、パートプログラム/サイクルの実行時に有効なアクセスレベルで書き込みが可能であり、データが直ちに有効となるように「IMMEDIATE」設定されていることが重要です。

下記も参照

変数 (ページ 433)

4.1.1.2 予約ユーザー変数:チャンネル別算術変数(R)

チャンネル別算術変数または R 変数は、予約された名称 R のユーザー変数で、REAL データタイプの配列として定義されています。互換性の理由から、R[10] などの配列インデックスによる表記、および R10 などの配列インデックスなしの表記の両方を、R 変数に使用できます。

シンクロナイズドアクションを使用する場合は、\$R10 のように、\$記号を含む接頭語を使用してください。

構文

先読み変数として使用する場合:

R<n>

R[<式>]

メインラン変数として使用する場合:

\$R<n>

\$R[<式>]

4.1 フレキシブルな NC プログラミング

意味

R:	パートプログラムなどで、先読み変数として使用する場合の識別子	
\$R:	シンクロナイズドアクションなどで、メインラン変数として使用する場合の識別子	
	タイプ:	REAL
	値の範囲:	非指数表記の場合: ± (0.000 0001 ... 9999 9999) 注: 小数点以下 8 桁までの数値を使用できます。
		指数表記の場合: ± (1*10 ⁻³⁰⁰ ... 1*10 ⁺³⁰⁰) 注: - 表記: <仮数>EX<指数> 例: 8.2EX-3 - 符号と小数点を含め、10 文字までの文字を使用できます。
<n>:	R 変数の番号	
	タイプ:	INT
	値の範囲:	0~MAX_INDEX 注: MAX_INDEX は、次のように、R 変数で設定した番号から計算します。 MAX_INDEX = (MD28050 \$MN_MM_NUM_R_PARAM) - 1
<式>:	配列インデックス 式の結果が INT データタイプ(INT、REAL、BOOL、CHAR)に変換できるかぎり、任意の式を配列インデックスとして使用できます。	

例

R 変数の割り当ておよび数学関数での R 変数の使用:

プログラムコード	コメント
R0=3.5678	; 先読み変数の割り当て
R[1]=-37.3	; 先読み変数の割り当て
R3=-7	; 先読み変数の割り当て
\$R4=-0.1EX-5	; メインプログラムの実行での割り当て: R4 = -0.1 * 10 ⁻⁵

4.1 フレキシブルな NC プログラミング

プログラムコード	コメント
\$R[6]=1.874EX8	; メインプログラムの実行での割り当て:R6 = 1.874 * 10 ⁸
R7=SIN(25.3)	; 先読み変数の割り当て
R[R2]=R10	; R 変数を使用した間接アドレス指定
R[(R1+R2)*R3]=5	; 数式を使用した間接アドレス指定
X=(R1+R2)	; R1 と R2 の合計から得られる位置に軸 X を移動します
Z=SQRT(R1*R1+R2*R2)	; 平方根位置 (R1 ² + R2 ²) に Z 軸を移動します

下記も参照

変数 (ページ 433)

4.1.1.3 予約ユーザー変数:グローバル算術変数(RG)

機能

チャンネル別 R 変数に加えて、グローバル R 変数にアクセスすることができます。これらはコントロールユニット内に存在し、すべてのチャンネルから読み取り、書き込むことができます。

グローバル R 変数は、たとえばチャンネル間で情報を転送するために使用されます。また、別の例として、主軸から張り出した未処理部分など、すべてのチャンネルに対して評価する必要のあるグローバルな設定に使用されます。

グローバル R 変数は、操作画面からまたは NC プログラムで先読み時に読み取りおよび書き込みされます。シンクロナイズドアクションおよびテクノロジサイクルは使用できません。

注記

グローバル R 変数の読み取りおよび書き込み時には、チャンネル間の同期はおこなわれません。

読み取りおよび書き込みは先読み時に実行されるため、1 つのチャンネルからの書き込み値が別のチャンネルで有効になるタイミングは定義されません。

例:

チャンネル 1 で、グローバル R 変数をループカウンタとしてループを実行します。チャンネル 2 がこのグローバル R 変数に値を書き込み、これによってチャンネル 1 でループが中止されます。ただし、チャンネル 1 で先読みで読み取ることのできるすべてのループは、引き続き実行されます。ループの数は定義されず、チャンネルの読み込みなどに依存します。

チャンネル間の同期は、WAIT フラグなどにより、アプリケーションとして実行してください。

4.1 フレキシブルな NC プログラミング

構文

NC プログラムでの書き込み

RG [<n>] =<値>

RG [<式>] =<値>

NC プログラムでの読み取り

R...=RG [<n>]

R...=RG [<式>]

意味

RG :	グローバル R 変数の NC アドレスのデフォルト名 注記: NC アドレスの名前は MD15800 \$MN_R_PARAM_NCK_NAME を使用して設定できます。	
<n>:	グローバル R 変数の番号	
	タイプ:	INT
	データ範囲:	0 ... MAX_INDEX 注: MAX_INDEX は、次のように、グローバル R 変数で設定した番号から計算します。 MAX_INDEX = (MD18156 \$MN_MM_NUM_R_PARAM_NCK) - 1
<式>:	式の結果が INT データタイプ(INT、REAL、BOOL、CHAR)に変換できるかぎり、任意の式を配列インデックスとして使用できます。	
<値>:	グローバル R 変数の値	
	タイプ:	REAL
	値の範囲:	非指数表記の場合: ± (0.000 0001 ... 9999 9999) 注: 小数点以下 8 桁までの数値を使用できます。 指数表記の場合: ± (1*10 ⁻³⁰⁰ ... 1*10 ⁺³⁰⁰) 注: - 表記: <仮数>EX<指数> 例: 8.2EX-3 - 符号と小数点を含め、10 文字までの文字を使用できます。

4.1.1.4 予約ユーザー変数リンク変数

ネットワークでリンクされた NCU 間で周期的にデータ交換をおこなう「NCU リンク」機能処理では、リンク変数を使用できます。この変数を使用すると、リンク変数メモリへのデータ形式別のアクセスが容易になります。リンク変数メモリは、ユーザー/工作機械メーカーがシステム用基準に、サイズとデータ構成の両方を定義します。

リンク通信を構成している場合、リンク変数は、リンクに関与するすべての NCU によりパートプログラムとサイクルで読み取りと書き込みが可能な、システムグローバルユーザー変数です。グローバルユーザー変数(GUD)とは異なり、リンク変数は、シンクロナイズドアクションでも使用できます。

動作中の NCU リンクのないシステムでは、リンク変数を制御装置でローカルに、グローバルユーザー変数(GUD)と一緒に追加のグローバルユーザー変数として使用できます。

構文

```
$A_DLB[<インデックス>]
$A_DLW[<インデックス>]
$A_DLD[<インデックス>]
$A_DLR[<インデックス>]
```

意味

\$A_DLB:	BYTE データ形式のリンク変数(1 バイト)	
	データタイプ:	UINT
	値の範囲:	0 ... 255
\$A_DLW:	WORD データ形式のリンク変数(2 バイト)	
	データタイプ:	INT
	値の範囲:	-32768 ... 32767
\$A_DLD:	DWORD データ形式のリンク変数(4 バイト)	
	データタイプ:	INT
	値の範囲:	-2147483648 ... 2147483647
\$A_DLR:	REAL データ形式のリンク変数(8 バイト)	
	データタイプ:	REAL
	値の範囲:	$\pm(2.2 \cdot 10^{-308} \dots 1.8 \cdot 10^{+308})$

4.1 フレキシブルなNCプログラミング

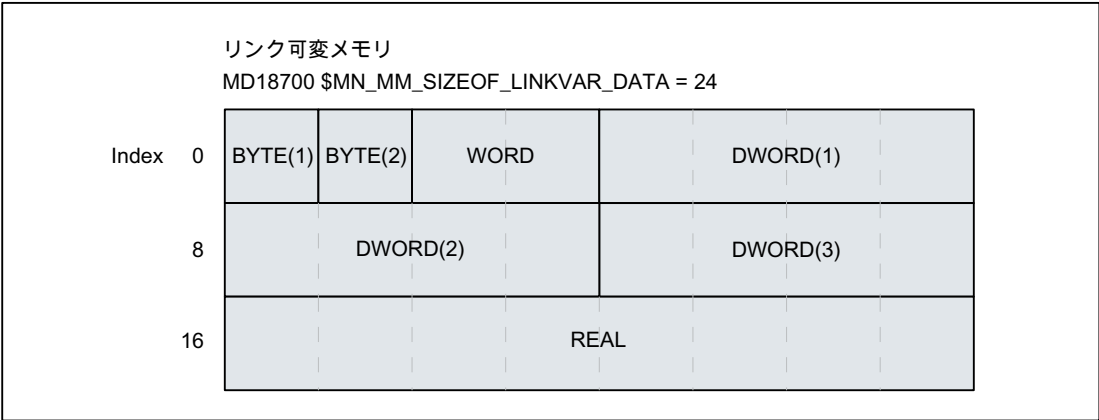
<インデックス>:	リンク変数メモリの先頭から数えたバイト単位のアドレスインデックス	
	データタイプ:	INT
	値の範囲:	0~MAX_INDEX 注 • MAX_INDEX は、次のように、パラメータ設定されたリンク変数メモリのサイズから計算します。 MAX_INDEX = (MD18700 \$MN_MM_SIZEOF_LINKVAR_DATA) - 1 • リンク変数メモリにアドレス指定したバイト数がデータ形式の制限範囲内に配置されるよう、インデックスをプログラム指令してください。つまり、インデックス = n * バイト(ここでは n = 0、1、2、...) になります。 – \$A_DLB[i]: i = 0、1、2、... – \$A_DLW[i]: i = 0、2、4、... – \$A_DLD[i]: i = 0、4、8、... – \$A_DLR[i]: i = 0、8、16、...

例

オートメーションシステムに 2 基の NCU (NCU1 と NCU2)が含まれます。機械軸 AX2 が NCU1 に接続されています。この軸は、NCU2 のリンク軸として移動します。

NCU1 は、軸 AX2 の電流フィードバック値(\$VA_CURR)を周期的にリンク変数メモリに書き込みます。NCU2 は、リンク通信で周期的に伝送される電流フィードバック値を読み取り、制限値を超えた場合はアラーム 61000 を表示します。

リンク変数メモリのデータ構成を、次の図に示します。電流フィードバック値は、REAL 値で伝送されます。



NCU1

NCU1 はリンク変数\$A_DLR[16]を使用して、軸 AX2 の電流フィードバック値を、内部的なシンクロナイズドアクションの補間サイクル中に周期的に、リンク変数メモリに書き込みます。

プログラムコード

```
N111 IDS=1 WHENEVER TRUE DO $A_DLR[16]=$VA_CURR[AX2]
```

NCU2

NCU2 はリンク変数\$A_DLR[16]を使用して、軸 AX2 の電流フィードバック値を、内部的なシンクロナイズドアクションの補間サイクル中に周期的に、リンク変数メモリに読み込みます。電流フィードバック値が 23.0 A を超える場合は、アラーム 61000 が表示されます。

プログラムコード

```
N222 IDS=1 WHEN $A_DLR[16] > 23.0 DO SETAL(61000)
```

4.1.1.5 ユーザー変数の定義(DEF)

DEF 命令を使用して、ユーザー固有の変数、すなわちユーザー変数(ユーザーデータ)を定義し、それに値を割り当てることができます。

4.1 フレキシブルな NC プログラミング

有効範囲(つまり、変数を表示できる範囲)に応じて、ユーザー変数には次のカテゴリがあります。

- ローカルユーザー変数(LUD)
ローカルユーザー変数(LUD)は、実行時にメインプログラム以外の NC プログラムで定義した変数です。この変数は、NC プログラムが呼び出されると作成され、プログラムリセットの終了時またはコントロールシステムの起動時に削除されます。ローカルユーザー変数には、この変数を定義した NC プログラム内でのみアクセスできます。
- プログラムユーザー変数(PUD)
プログラムユーザー変数(PUD)は、メインプログラムとして使用される NC プログラムのなかで定義したユーザー変数です。この変数は、NC プログラムが呼び出されると作成され、プログラムリセットの終了時またはコントロールシステムの起動時に削除されます。PUD は、メインプログラム、およびメインプログラムのすべてのサブプログラムのなかでアクセスできます。

注記

プログラムグローバルユーザー変数(PUD)の使用可能性

次のマシンデータを設定した場合は、メインプログラムで定義したプログラムグローバルユーザー変数(PUD)はサブプログラムにのみ表示されます。

MD11120 \$MN_LUD_EXTENDED_SCOPE = 1

MD11120=0 の場合、メインプログラムで定義したプログラムグローバルユーザー変数はメインプログラムでのみ使用できます。

- グローバルユーザー変数(GUD)
グローバルユーザー変数(GUD)は、データブロック(SGUD、MGUD、UGUD、GUD4～GUD9)で定義され、プログラムリセット終了時、またはコントロールシステムの次回起動の後も保持される NC グローバル変数またはチャンネルグローバル変数です。GUD は、すべての NC プログラムでアクセスできます。

ユーザー変数は、使用(読み取り/書き込み)前に定義してください。この場合は、以下の規則を守ってください。

- GUD は、定義ファイルで_N_DEF_DIR/_N_UGUD_DEF のように定義します。
- PUD と LUD は、NC プログラムの定義部で定義します。
- データは専用ブロックで定義してください。
- 各データ定義に使用できるのは、1 つのデータタイプのみです。
- 各データ定義で、複数の同じデータタイプの変数を定義できます。

構文

LUD と PUD

DEF <タイプ> <物理単位> <制限値> <名称> [<値 1>, <値 2>, <値 3>] = <初期値>

GUD

DEF <範囲> <先読み停止> <アクセス権> <データクラス> <タイプ> <物理単位> <制限値> <名称> [<値 1>, <値 2>, <値 3>] = <初期値>

意味

DEF:	GUD、PUD、LUD の各ユーザー変数の定義命令	
<範囲>:	適用範囲、GUD にのみ有効:	
	NC:	NC グローバルユーザー変数
	CHAN:	チャンネルグローバルユーザー変数
先読み停止>:	先読み停止、GUD にのみ有効(任意選択)	
	SYNR:	読み取り時の先読み停止
	SYNW:	書き込み時の先読み停止
	SYNRW:	読み取り/書き込み時の先読み停止
アクセス権>:	NC プログラムまたは OPI を介した GUD の読み取り/書き込み保護レベル(任意選択)	
	APRP <保護レベル>:	読み取り:NC プログラム
	APWP <保護レベル>:	書き込み:NC プログラム
	APRB <保護レベル>:	読み取り:OPI
	APWB <保護レベル>:	書き込み:OPI
	<保護レベル>:	値の範囲:0 ... 7
	「属性: アクセス権(APR、APW、APRP、APWP、APRB、APWB) (ページ 461)」を参照してください	
<データクラス>:	データクラスの割り当て(SINUMERIK 828D のみ)	
	DCM:	データクラス M (= Manufacturer)
	DCI:	データクラス I (= Individual)
	DCU:	データクラス U (= User)

4.1 フレキシブルな NC プログラミング

<タイプ>:	データタイプ:	
	INT:	符合付き整数
	REAL:	実数(IEEE では LONG REAL)
	BOOL:	真偽値 TRUE (1)/FALSE (0)
	CHAR:	ASCII 文字
	STRING [<最大長>]:	定義された長さの文字列
	AXIS:	軸/主軸の識別子
	FRAME:	内部的な座標変換のジオメトリデータ
	「データタイプ (ページ 474)」を参照してください	
<物理単位>:	物理単位(任意選択)	
	PHU <単位>:	物理単位
	「属性: 物理単位(PHU) (ページ 459)」を参照してください	
<制限値>:	下限値/上限値(任意選択)	
	LLI <制限値>:	下限値(下限)
	ULI <制限値>:	上限値(上限)
	「属性: 制限値(LLI、ULI) (ページ 457)」を参照してください	
<名称>:	変数の名称	
	注 - 最大 31 文字 - 先頭の 2 文字には、英字とアンダースコアのいずれかまたは両方を使用してください。 \$記号はシステム変数として予約されているため、使用できません。	
[<値 1>,<値 2>,<値 3>]:	1 次元から 3 次元までの配列変数の配列サイズの指定(任意選択)	
	配列変数の初期化については、「配列変数の定義と初期化 (DEF、SET、REP) (ページ 468)」を参照してください。	
<初期値>:	初期値(任意選択)	
	「属性:初期値 (ページ 454)」を参照してください 配列変数の初期化については、「配列変数の定義と初期化 (DEF、SET、REP) (ページ 468)」を参照してください。	

例

例 1: 工作機械メーカー用のデータブロックのユーザー変数の定義

プログラムコード	コメント
<pre> %_N_MGUD_DEF \$PATH=/_N_DEF_DIR DEF CHAN REAL PHU 24 LLI 0 ULI 10 CURRENT_1、CURRENT_2 ; 概要 ; 2つのGUD項目の定義:CURRENT_1、CURRENT_2 ; 適用範囲:チャネル全体 ; データタイプ:REAL 先読み停止:プログラム指令なし=>初期値=先読み停止なし ; 物理単位:24 = [A] ; 制限値:下限= 0.0、上限= 10.0 ; アクセス権:プログラム指令なし=>初期値= 7 =キー操作による切り替え位置 0 ; 初期値:プログラム指令なし=>初期値 = 0.0 DEF NCK REAL PHU 13 LLI 10 APWP 3 APRP 3 APWB 0 APRB 2 TIME_1=12、TIME_2=45 ; 概要 ; 2つのGUD項目の定義:TIME_1、TIME_2 ; 適用範囲:NC 全体 ; データタイプ:REAL 先読み停止:プログラム指令なし=>初期値=先読み停止なし ; 物理単位:13 = [s] ; 制限値:下限= 10.0、上限=プログラム指令なし=>制限定義範囲の上限 ; アクセス権: ; NC プログラム:書き込み/読み取り = 3 = ユーザー ; OPI:書き込み = 0 = 当社、読み取り = 3 = ユーザー ; 初期値:TIME_1 = 12.0、TIME_2 = 45.0 DEF NCK APWP 3 APRP 3 APWB 0 APRB 3 STRING[5] GUD5_NAME = "COUNTER" ; 概要 ; 1つのGUD項目の定義:GUD5_NAME ; 適用範囲:NC 全体 ; データタイプ:STRING、最大 5 文字 先読み停止:プログラム指令なし=>初期値=先読み停止なし ; 物理単位:プログラム指令なし=>初期値= 0 = 単位なし ; 制限値:プログラム指令なし=>制限定義範囲:下限= 0、上限= 255 ; アクセス権: ; NC プログラム:書き込み/読み取り = 3 = ユーザー ; OPI:書き込み = 0 = 当社、読み取り = 3 = ユーザー ; 初期値:"COUNTER" M30 </pre>	

4.1 フレキシブルなNC プログラミング

例 2:グローバルプログラムとローカルユーザー変数(PUD/LUD)

プログラムコード	コメント
PROC MAIN	; メインプログラム
DEF INT VAR1	; PUD 定義
...	
SUB2	; サブプログラム呼び出し
...	
M30	

プログラムコード	コメント
PROC SUB2	; サブプログラム SUB2
DEF INT VAR2	; LUD 定義
...	
IF (VAR1==1)	; PUD の読み取り
VAR1=VAR1+1	; PUD を読み取り、PUD に書き込みます
VAR2=1	; LUD に書き込みます
ENDIF	
SUB3	; サブプログラム呼び出し
...	
M17	

プログラムコード	コメント
PROC SUB3	; サブプログラム SUB3
...	
IF (VAR1==1)	; PUD の読み取り
VAR1=VAR1+1	; PUD を読み取り、PUD に書き込みます
VAR2=1	; エラー:SUB2 からは LUD は未知です
ENDIF	
...	
M17	

例 3:データタイプ AXIS のユーザー変数の定義と用途

プログラムコード	コメント
DEF AXIS ABSCISSA	; 1 番目のジオメトリ軸
DEF AXIS SPINDLE	; 主軸
...	
IF ISAXIS(1) == FALSE GOTOF CONTINUE	
ABSCISSA = \$P_AXN1	
CONTINUE:	
...	
SPINDLE=(S1)	; 1 番目の主軸

4.1 フレキシブルな NC プログラミング

プログラムコード	コメント
OVRA[SPINDLE]=80	; 主軸オーバライド = 80%
SPINDLE=(S3)	; 3 番目の主軸

必要条件

グローバルユーザー変数(GUD)

グローバルユーザー変数(GUD)を使用する場合は、次のマシンデータを考慮してください。

No.	識別子: \$MN_	意味
11140	GUD_AREA_SAVE_TAB	GUD ブロックの追加保存
18118 ¹⁾	MM_NUM_GUD_MODULES	アクティブファイルシステムの GUD ファイルの数
18120 ¹⁾	MM_NUM_GUD_NAMES_NCK	グローバル GUD の数
18130 ¹⁾	MM_NUM_GUD_NAMES_CHAN	チャンネル別 GUD の数
18150 ¹⁾	MM_GUD_VALUES_MEM	グローバル GUD 値のメモリロケーション
18660 ¹⁾	MM_NUM_SYNACT_GUD_REAL	REAL データタイプの設定可能な GUD の数
18661 ¹⁾	MM_NUM_SYNACT_GUD_INT	INT データタイプの設定可能な GUD の数
18662 ¹⁾	MM_NUM_SYNACT_GUD_BOOL	BOOL データタイプの設定可能な GUD の数
18663 ¹⁾	MM_NUM_SYNACT_GUD_AXIS	AXIS データタイプの設定可能な GUD の数
18664 ¹⁾	MM_NUM_SYNACT_GUD_CHAR	CHAR データタイプの設定可能な GUD の数
18665 ¹⁾	MM_NUM_SYNACT_GUD_STRING	STRING データタイプの設定可能な GUD の数

¹⁾ SINUMERIK 828D の場合は、MD は読み取りのみ可能です。

AXIS データタイプの NC グローバルユーザー変数のチャンネル相互間の用途

軸識別子を使用したデータブロック内での定義のときに初期化された AXIS データタイプの NC グローバルユーザー変数は、この軸がチャンネル間で同じチャンネル軸番号を持つ場合にのみ、他の NC チャンネルで使用できます。

4.1 フレキシブルな NC プログラミング

そうでない場合は、この変数を、NC プログラムの開始時にロードするか、または次の例のように、AXNAME(...)機能(「軸機能(AXNAME、AX、SPI、AXTOSPI、ISAXIS、AXSTRING、MODAXVAL) (ページ 1018)」を参照してください)を使用してください。

プログラムコード	コメント
DEF NCK STRING[5] ACHSE="X"	; データブロックで定義します。
...	
N100 AX[AXNAME(AXIS)]=111 G00	; NC プログラムでの使用

4.1.1.6 システムデータ、ユーザーデータ、NC 命令(REDEF)の再定義

REDEF 命令は、システムデータ、ユーザーデータ、および NC 命令の属性を変更します。再定義の基本状態は、対応する定義の日付けが新しくなることです。

再定義のときは、複数の属性を同時に変更できません。変更する属性毎に、個別の REDEF 命令をプログラム指令してください。

複数の属性の同時変更をプログラム指令した場合は、常に最後の変更属性が有効になります。

属性値のリセット

REDEF によるアクセス権と初期化タイミングの変更の属性は、変数名称または NC 言語命令が後に続く REDEF を再プログラム指令することで初期値にリセットできます。

- アクセス権:保護レベル 7
- 初期化タイミング:現在値の初期化または保持はおこなわれません。

再定義可能な属性

「定義と再定義が可能な属性の概要 (ページ 466)」を参照してください

ローカルユーザー変数(PUD/LUD)

ローカルユーザー変数(PUD/LUD)は再定義できません。

構文

REDEF <名称> <先読み停止>

REDEF <名称> <物理単位>

REDEF <名称> <制限値>

REDEF <名称> <アクセス権>

REDEF <名称> <初期化タイミング>

4.1 フレキシブルなNC プログラミング

REDEF <名称> <初期化タイミング> <初期値>

REDEF <名称> <データクラス>

REDEF <名称>

意味

REDEF:	特定の属性の再定義、または、システム変数、ユーザー変数、およびNC 言語命令の「アクセス権」/「初期化タイミング」をリセットするための命令	
<名称>:	定義済み変数またはNC 言語命令の名称	
<先読み停止>:	先読み停止	
	SYNR:	読み取り時の先読み停止
	SYNW:	書き込み時の先読み停止
	SYNRW:	読み取り/書き込み時の先読み停止
<物理単位>:	物理単位	
	PHU <単位>:	物理単位
	「属性: 物理単位(PHU) (ページ 459)」を参照してください 注: 以下については再定義できません。 - システム変数 - 以下のデータタイプのグローバルユーザーデータ (GUD): BOOL、AXIS、STRING、FRAME	
<制限値>:	下限/上限	
	LLI <制限値>:	下限値(下限)
	ULI <制限値>:	上限値(上限)
	「属性: 制限値(LLI、ULI) (ページ 457)」を参照してください 注: 以下については再定義できません。 - システム変数 - 以下のデータタイプのグローバルユーザーデータ (GUD): BOOL、AXIS、STRING、FRAME	

4.1 フレキシブルな NC プログラミング

<アクセス権>:	パートプログラムまたは OPI による読み取り/書き込みアクセス権	
	APX <保護レベル>:	実行:NC 言語要素
	APRP <保護レベル>:	読み取り:パートプログラム
	APWP <保護レベル>:	書き込み:パートプログラム
	APRB <保護レベル>:	読み取り:OPI
	APWB <保護レベル>:	書き込み:OPI
	<保護レベル>:	値の範囲:0 ... 7
	「属性: アクセス権(APR、APW、APRP、APWP、APRB、APWB) (ページ 461)」を参照してください	
<初期化タイミング>:	変数が再初期化されるタイミング	
	INIPO:	電源投入
	INIRE:	メインプログラムの終了時、NC リセット時、または電源投入時
	INICF:	NEWCONF またはメインプログラム終了、NC リセットまたは電源投入
	PRLOC:	メインプログラムの終了時、部分的な変更後の NC リセット時、または電源投入時
	「属性:初期値 (ページ 454)」を参照してください	
<初期値>:	初期値	
	初期値を再定義するときは、初期化タイミングも常に指定してください(<初期化タイミング>を参照してください)。 「属性:初期値 (ページ 454)」を参照してください 配列変数の初期化については、「配列変数の定義と初期化(DEF、SET、REP) (ページ 468)」を参照してください。 注 セッティングデータ以外のシステム変数に対しては再定義できません。	

4.1 フレキシブルな NC プログラミング

<データクラス>:	データクラスの割り当て(SINUMERIK 828D のみ)	
	DCM:	データクラス M (= Manufacturer)
	DCI:	データクラス I (= Individual)
	DCU:	データクラス U (= User)

例

工作機械メーカー用のデータブロックのシステム変数\$TC_DPCx の再定義

プログラムコード
%_N_MGUD_DEF ; GUD ブロック:工作機械メーカー
N100 REDEF \$TC_DPC1 APWB 2 APWP 3
N200 REDEF \$TC_DPC2 PHU 21
N300 REDEF \$TC_DPC3 LLI 0 ULI 200
N400 REDEF \$TC_DPC4 INIPO (100, 101, 102, 103)
N800 REDEF \$TC_DPC1
N900 REDEF \$TC_DPC4
M30

N100 書き込みアクセス権:OPI=保護レベル 2、パートプログラム=保護レベル 3
について
で:

N200 物理単位[%]
について
で:

N300 下限値 = 0、上限値 = 200
について
で:

N400 ; 配列変数は、POWER ON 時に 4 つの値で初期化されます。
について
で:

N800/ 「アクセス権」 / 「初期化タイミング」 属性値のリセット
N900
について
で:

4.1 フレキシブルな NC プログラミング

注記

ACCESS ファイルの使用

ACCESS ファイルを使用する場合、アクセス権の再定義を `_N_MGUD_DEF` から `_N_MACCESS_DEF` に再配置する必要があります。

必要条件

細分性

再定義は常に、名称で一義的に識別される変数全体に適用されます。たとえば、配列変数は、個々の配列要素への異なる属性の割り当てをサポートしていません。

4.1.1.7 属性:初期値

ユーザー変数の定義(DEF)

定義するときに、次のユーザー変数の初期値を事前に割り当てることができます。

- グローバルユーザー変数(GUD)
- プログラムユーザー変数(PUD)
- ローカルユーザー変数(LUD)

システム変数とユーザー変数の再定義(REDEF)

再定義するときに、次のユーザー変数の初期値を事前に割り当てることができます。

- システムデータ
 - セッティングデータ
- ユーザーデータ
 - R 変数
 - シンクロナイズドアクション変数(\$AC_MARKER、\$AC_PARAM、\$AC_TIMER)
 - シンクロナイズドアクション GUD (SYG_xy[]、ここで x=R、I、B、A、C、S および y=S、M、U、4～9)
 - EPS 変数
 - OEM 工具データ
 - OEM マガジンデータ
 - グローバルユーザー変数(GUD)

再初期化のタイミング

再定義のとき、変数を再初期化する(つまり、初期値にリセットする)タイミングを指定できます。

- INIPO (電源投入時)
変数は電源投入時に初期化されます。
- INIRE (リセット)
NC リセット時、モードグループリセット時、パートプログラム終了時(M02/M30)、または電源投入時に変数を最初期化します。
- INICF (NEWCONF)
「マシンデータ有効設定」機能については、パートプログラム命令 NEWCONF または NC リセット時、モードグループリセット時、パートプログラム終了時(M02/M30)、または電源投入時に HMI を介して変数が再初期化されます。
- PRLOC (プログラムの部分変更)
パートプログラム実行中に変数が変更された場合に、NC リセット時、モードグループリセット時、パートプログラム終了時(M02/M30)にのみ変数を再初期化します。
PRLOC 属性は、プログラム指令可能なセッティングデータと組み合わせた場合にのみ変更できます(以下の表を参照してください)。

表 4-1 プログラム指令可能なセッティングデータ

番号	識別子	G 命令 ¹⁾
42000	\$SC_THREAD_START_ANGLE	SF
42010	\$SC_THREAD_RAMP_DISP	DITS/DITE
42400	\$SA_PUNCH_DWELLTIME	PDELAYON
42800	\$SA_SPIND_ASSIGN_TAB	SETMS
43210	\$SA_SPIND_MIN_VELO_G25	G25
43220	\$SA_SPIND_MAX_VELO_G26	G26
43230	\$SA_SPIND_MAX_VELO_LIMS	LIMS
43300	\$SA_ASSIGN_FEED_PER_REV_SOURCE	FPRAON
43420	\$SA_WORKAREA_LIMIT_PLUS	G26
43430	\$SA_WORKAREA_LIMIT_MINUS	G25
43510	\$SA_FIXED_STOP_TORQUE	FXST
43520	\$SA_FIXED_STOP_WINDOW	FXSW
43700	\$SA_OSCILL_REVERSE_POS1	OSP1
43710	\$SA_OSCILL_REVERSE_POS2	OSP2

4.1 フレキシブルな NC プログラミング

番号	識別子	G 命令 ¹⁾
43720	\$SA_OSCILL_DWELL_TIME1	OST1
43730	\$SA_OSCILL_DWELL_TIME2	OST2
43740	\$SA_OSCILL_VELO	Fa
43750	\$SA_OSCILL_NUM_SPARK_CYCLES	OSNSC
43760	\$SA_OSCILL_END_POS	OSE
43770	\$SA_OSCILL_CTRL_MASK	OSCTRL
43780	\$SA_OSCILL_IS_ACTIVE	OS
43790	\$SA_OSCILL_START_POS	OSB
1)この G 命令は、セッティングデータをアドレス指定します。		

必要条件

初期値:グローバルユーザー変数(GUD)

- NC の適用範囲のグローバルユーザー変数(GUD)の初期化タイミングとして定義できるのは、INIPO (電源投入時)のみです。
- INIPO (電源投入時)に加えて、INIRE (リセット)または INICF (NEWCONF)を、CHAN の適用範囲のグローバルユーザー変数(GUD)の初期化タイミングとして定義できます。
- 適用範囲が CHAN で、初期化タイミングが INIRE(リセット)または INICF(NEWCONF)である、および NC リセット、モードグループリセット、および「マシンデータ有効化」のためのグローバルユーザー変数(GUD)では、名称を付けた事象が起動されたチャンネルでのみ、変数を再初期化できます。

初期値:FRAME データタイプ

FRAME データタイプの変数には初期値を指定できません。FRAME データタイプの変数は、常に既定のフレームで自動的に初期化されます。

初期値:CHAR データタイプ

CHAR データタイプの変数の場合、ASCII コード(0... 255)ではなく、対応する ASCII 文字を引用符で囲んでプログラム指令できます(例: "A")。

初期値:データタイプ STRING

STRING データタイプの変数の場合、文字列は引用符で囲んでください(例: ...="MACHINE_1")。

初期値:AXIS データタイプ

AXIS データタイプの変数の場合、拡張アドレス表記では、軸識別子を括弧で囲んでください(例: ...=(X3))。

初期値:システム変数

システム変数の場合、再定義を使用して、ユーザー用の初期値を定義することはできません。システム変数の初期値は、システムで指定されており、変更できません。ただし、再定義を使用して、システム変数を再初期化するタイミング(INIRE, INICE)を変更することはできます。

自動設定初期値:AXIS データタイプ

AXIS データタイプの変数の場合は、次の初期値を自動的に使用します。

- システムデータ:「1 番目のジオメトリ軸」
- シンクロナイズドアクション GUD(名称:SYG_A*)、PUD、LUD:
マシンデータからの軸名称:MD20082 \$MC_AXCONF_CHANAX_DEFAULT_NAME

自動設定初期値:工具とマガジンデータ

工具データとマガジンデータの初期値は、次のマシンデータを使用して定義できます。
MD17520 \$MN_TOOL_DEFAULT_DATA_MASK

注記**同期制御**

グローバル変数が異なる場所で読み取られる場合に、この変数の再初期化を起動する事象の同期制御については、そのユーザー/工作機械メーカーの自己責任となります。

下記も参照

変数 (ページ 433)

4.1.1.8 属性: 制限値(LLI、ULI)

定義範囲の上限と下限は、次のデータタイプの場合にのみ定義できます。

- INT
- REAL
- CHAR

4.1 フレキシブルなNCプログラミング

ユーザー変数の定義(DEF)制限値と自動設定初期値

上記のいずれかのデータタイプのユーザー変数を定義するときに、初期値を別途定義しない場合、変数はそのデータタイプの初期値へ自動的に設定されます。

- INT:0
- REAL:0.0
- CHAR:0

自動設定初期値が、プログラム指令した制限値で指定された定義範囲外にある場合は、自動設定初期値に最も近い制限値で変数が初期化されます。

- 自動設定初期値 < 下限値 (LLI) ⇒
初期値 = 下限値
- 自動設定初期値 > 上限値 (ULI) ⇒
初期値 = 上限値

例:

プログラムコード	コメント
DEF REAL GUD1	; 下限値=制限定義範囲 ; 上限値=制限定義範囲 ; プログラム指令された初期値がありません ; => 自動設定初期値 = 0.0
DEF REAL LLI 5.0 GUD2	; 下限値=5.0 ; 上限値=制限定義範囲 ; => 初期値 = 5.0
DEF REAL ULI -5 GUD3	; 下限値=制限定義範囲 ; 上限値= -5.0 ; => 初期値 = -5.0

ユーザー変数の再定義(REDEF)制限値と現在値

ユーザー変数の制限値の再定義による変更のために、現在値が新しい定義範囲の範囲外となった場合は、アラームが発生して、制限値は拒否されます。

注記

ユーザー変数の再定義(REDEF)

ユーザー変数の制限値を再定義する場合、次の値と矛盾しないように変更してください。

- 制限値
- 現在値
- 再定義の初期値、および INIPO、INIRE、または INICF に基づいて自動再定義

下記も参照

変数 (ページ 433)

4.1.1.9 属性: 物理単位(PHU)

物的単位は、次のデータタイプの変数についてのみ指定できます。

- INT
- REAL

プログラム指令可能な物理単位(PHU)

物理単位は、固定小数点数形式で指定します。PHU <単位>

次の物理単位をプログラム指令できます。

<単位>	意味	物理単位
0	物理単位はありません	-
1	直線位置または回転位置 ¹⁾²⁾	[mm]、[inch]、[°]
2	直線位置 ²⁾	[mm]、[inch]
3	回転位置	[°]
4	直線速度または回転速度 ¹⁾²⁾	[mm/min]、[inch/min]、[rpm]
5	直線速度 ²⁾	[mm/min]
6	回転速度	[rpm]
7	直線加減速度または回転加減速度 ¹⁾²⁾	[m/s ²]、[inch/s ²]、[rev/s ²]
8	直線加減速度 ²⁾	[m/s ²]、[inch/s ²]
9	回転加減速度	[rev/s ²]
10	直線加々速度または回転加々速度 ¹⁾²⁾	[m/s ³]、[inch/s ³]、[rev/s ³]
11	直線加々速度 ²⁾	[m/s ³]、[inch/s ³]
12	回転加々速度	[rev/s ³]
13	時間	[s]
14	位置制御ゲイン	[16.667/s]
15	毎回転送り速度 ²⁾	[mm/rev]、[inch/rev]
16	温度補正 ¹⁾²⁾	[mm]、[inch]
18	推力	[N]
19	質量	[kg]

4.1 フレキシブルな NC プログラミング

<単位>	意味	物理単位
20	慣性モーメント ³⁾	[kgm ²]
21	パーセント	[%]
22	周波数	[Hz]
23	電圧	[V]
24	電流	[A]
25	温度	[°C]
26	角度	[°]
27	Kv	[1000/min]
28	直線位置または回転位置 ³⁾	[mm]、[inch]、[°]
29	切削速度 ²⁾	[m/min]、[feet/min]
30	周速度 ²⁾	[m/s]、[feet/s]
31	抵抗	[Ω]
32	インダクタンス	[mH]
33	トルク ³⁾	[Nm]
34	トルク定数 ³⁾	[Nm/A]
35	電流制御ゲイン	[V/A]
36	速度制御ゲイン ³⁾	[Nm/(rad*s)]
37	速度	[rpm]
42	電力	[kW]
43	電流、低	[μA]
46	トルク、低 ³⁾	[μNm]
48	1000 分の 1	-
49	-	[Hz/s]
65	流量	[ℓ/min]
66	圧力	[bar]
67	体積 ³⁾	[cm ³]
68	制御対象ゲイン ³⁾	[mm/(V*min)]
69	トルク制限制御対象ゲイン	[N/V]
155	ねじリード ³⁾	[mm/rev]、[inch/rev]
156	ねじリードの可変量 ³⁾	[mm/rev/rev]、[inch/rev/rev]
1) 物理単位は軸タイプ(直線軸または回転軸)に依存します。		

<単位>	意味	物理単位
2) 単位系の切り替え		
G70/G71(inch/metric)		
長さに関するシステム変数とユーザー変数への読み取り/書き込み操作で、G70/G71により単位系(MD10240 \$MN_SCALING_SYSTEM_IS_METRIC)を変更した後は、その値は変換されません(現在値、初期値、および制限値)。		
G700/G710(inch/metric)		
長さに関するシステム変数とユーザー変数への読み取り/書き込み操作で、G700/G710により単位系(MD10240 \$MN_SCALING_SYSTEM_IS_METRIC)を変更した後に、その値が変換されます(現在値、初期値、および制限値)。		
3) 変数は、NC の現在の単位系(インチ/メトリック)に自動的に変換されません。変換については、ユーザー/工作機械メーカーの自己責任となります。		

注記**書式変換によるレベルオーバーフロー**

長さの物理単位を含むユーザー変数(GUD/PUD/LUD)の内部保存書式はすべてメトリックです。このようなタイプの変数を、シンクロナイズドアクションなどの NCK のメインランで過度に使用すると、単位系が切り替わるときに補間レベルで CPU 処理時間オーバーフローが発生し、アラーム 4240 が発生する場合があります。

注記**単位の互換性**

変数を使用する(割り当て、比較演算、計算など)ときに、関連する単位の互換性は確認されません。変換が必要な場合は、ユーザー/工作機械メーカーの自己責任となります。

下記も参照

変数 (ページ 433)

4.1.1.10 属性: アクセス権(APR、APW、APRP、APWP、APRB、APWB)**名称**

アクセス属性 AP...の名称には以下が含まれます。

1. A:Access (アクセス)
2. P:Protection (プロテクション)
3. R / W:Read / Write (読み取り/書き込み)
4. P / O:Program / BTSS (OPI)

4.1 フレキシブルな NC プログラミング

アクセス権/アクセスレベル

プログラミングのときに指定が必要な次のアクセスレベルは、アクセス権に対応しています。

アクセス権	保護レベル
システムパスワード	0
工作機械メーカーのパスワード	1
サービスパスワード	2
ユーザーパスワード	3
キー操作のスイッチ位置 3	4
キー操作のスイッチ位置 2	5
キー操作のスイッチ位置 1	6
キー操作のスイッチ位置 0	7

ユーザーデータの定義(DEF)

次のデータには、アクセス権(APR.../APW...)を定義できます。

- グローバルユーザーデータ(GUD)

システムデータとユーザーデータの再定義(REDEF)

次のデータには、アクセス権(APR.../APW...)を再定義できます。

- システムデータ
 - マシンデータ

注記

マシンデータの読み取り権限の再定義

マシンデータ読み取りの保護レベルは、パートプログラムと OPI のキーワード APR によってのみ設定できます。

キーワード APRP および APRB は読み取り権限の再定義によってサポートされていないため、アラーム 12490「アクセス権 APRP/APRB <保護レベル>が設定されていません」が表示されます。

- セッティングデータ
- システム変数
- プロセスデータ
- マガジンデータ
- 工具データ
- ユーザーデータ
 - R 変数
 - シンクロナイズドアクション変数(\$AC_MARKER、\$AC_PARAM、\$AC_TIMER)
 - シンクロナイズドアクション GUD (SYG_xy[], ここで x=R、I、B、A、C、S、および y=S、M、U、4~9)
 - EPS 変数
 - OEM 工具データ
 - OEM マガジンデータ
 - グローバルユーザー変数(GUD)

注記

再定義のときは、最低保護レベルの 7 から、1(工作機械メーカー)などの専用保護レベルの間で、変数に、アクセス権を自由に割り当てることができます。

NC 言語命令の再定義(REDEF)

次の NC 言語命令では、アクセス権または実行権(APX)を再定義できます。

- G 命令/準備機能 (ページ 190)
- 予約機能 (ページ 1562)

4.1 フレキシブルな NC プログラミング

- 予約サブプログラム呼び出し
- シンクロナイズドアクションによる DO 命令
- サイクルプログラム識別子
サイクルはサイクルディレクトリに格納し、PROC 命令を指令してください。

NC プログラムとサイクル関連のアクセス権(APRP、APWP)

さまざまなアクセス権により、NC プログラムまたはサイクルからのアクセスが次のように簡単になります。

- APRP 0/APWP 0
 - NC プログラムを処理するときに、システムパスワードを設定してください。
 - サイクルは、_N_CST_DIR ディレクトリ(システム)に格納してください。
 - システムに対して、_N_CST_DIR ディレクトリの実行権を MD11160 \$MN_ACCESS_EXEC_CST で設定してください。
- APRP 1/APWP 1 または APRP 2/APWP 2
 - NC プログラムを処理するときに、工作機械メーカまたはサービスのパスワードを設定してください。
 - サイクルは、_N_CMA_DIR (工作機械メーカ)または_N_CST_DIR ディレクトリに格納してください。
 - _N_CMA_DIR または_N_CST_DIR のディレクトリの実行権が工作機械メーカ以上になるように、マシンデータ MD11161 \$MN_ACCESS_EXEC_CMA または MD11160 \$MN_ACCESS_EXEC_CST にそれぞれ設定してください。
- APRP 3/APWP 3
 - NC プログラムを実行するときには、ユーザーパスワードを設定してください。
 - サイクルは、_N_CUS_DIR (エンドユーザー)、_N_CMA_DIR、または_N_CST_DIR ディレクトリに格納してください。
 - _N_CUS_DIR、_N_CMA_DIR または_N_CST_DIR のディレクトリの実行権がユーザー以上になるように、マシンデータ MD11162 \$MN_ACCESS_EXEC_CUS、MD11161 \$MN_ACCESS_EXEC_CMA または MD11160 \$MN_ACCESS_EXEC_CST にそれぞれ設定してください。
- APRP 4～7 / APWP 4～7
 - NC プログラムを処理するときに、キー操作スイッチを 3～0 に設定してください。
 - サイクルは、ディレクトリ_N_CUS_DIR、_N_CMA_DIR、またはディレクトリ_N_CST_DIR に格納してください。
 - _N_CUS_DIR、_N_CMA_DIR または_N_CST_DIR のディレクトリの実行権が該当するキー操作スイッチ位置以上になるように、マシンデータ MD11162 \$MN_ACCESS_EXEC_CUS、MD11161 \$MN_ACCESS_EXEC_CMA または MD11160 \$MN_ACCESS_EXEC_CST にそれぞれ設定してください。

OPI に関連するアクセス権(APRB、APWB)

アクセス権(APRB、APWB)によって、OPI を介したシステム変数とユーザー変数へのアクセスを、すべてのシステムコンポーネント(HMI、PLC、外部コンピュータ、EPS サービスなど)に対して一様に制限します。

注記

ローカル HMI アクセス権

システムデータへのアクセス権を変更するときは、変更したアクセス権が、HMI 操作で定義されたアクセス権と矛盾しないようにしてください。

APR/APW アクセス属性

互換性の理由から、属性 APR および APW は、属性 APRP/APRB および APWP/APWB に自動的に割り当てられます。

- APR x ⇒ APRP x APRB x
- APW y ⇒ APWP y APWB y

ACCESS ファイルを使用したアクセス権

ACCESS ファイルを使用してアクセス権を割り当てる場合は、システムデータ、ユーザーデータ、および NC 言語命令のアクセス権は ACCESS ファイルでのみ再定義してください。グローバルユーザーデータ(GUD)は例外です。このデータの場合は、アクセス権は対応する定義ファイル*_DEF で再定義してください。

アクセス保護が継続的におこなわれるように、実行権用のマシンデータ、および対応するディレクトリのアクセス保護を、矛盾がないように変更してください。

原則として、以下の手順を実行します。

1. 以下の、必要な定義ファイルの作成をおこないます。
 - _N_DEF_DIR/_N_SACCESS_DEF
 - _N_DEF_DIR/_N_MACCESS_DEF
 - _N_DEF_DIR/_N_UACCESS_DEF
2. 下記の定義ファイルの書き込み権限のマシンデータを、再定義のために必要な値へ設定します:
 - MD11170 \$MN_ACCESS_WRITE_SACCESS = <保護レベル>
 - MD11171 \$MN_ACCESS_WRITE_MACCESS = <保護レベル>
 - MD11172 \$MN_ACCESS_WRITE_UACCESS = <保護レベル>

4.1 フレキシブルな NC プログラミング

3. サイクルから保護要素にアクセスする場合は、サイクルディレクトリ `_N_CST_DIR`、`_N_CMA_DIR`、および `_N_CST_DIR` に対する実行権と書き込み権を変更してください。
実行権
 - MD11160 \$MN_ACCESS_EXEC_CST = <保護レベル>
 - MD11161 \$MN_ACCESS_EXEC_CMA = <保護レベル>
 - MD11162 \$MN_ACCESS_EXEC_CUS = <保護レベル>書き込み権
 - MD11165 \$MN_ACCESS_WRITE_CST = <保護レベル>
 - MD11166 \$MN_ACCESS_WRITE_CMA = <保護レベル>
 - MD11167 MN_ACCESS_WRITE_CUS = <保護レベル>実行権は、少なくとも、使用する要素の最高の保護レベルと同じ保護レベルに設定してください。
書き込み権は、少なくとも、実行権と同じ保護レベルに設定してください。
4. ローカル HMI サイクルディレクトリの書き込み権は、ローカル NC サイクルディレクトリと同じ保護レベルに設定してください。

ACCESS ファイルでのサブプログラム呼び出し

さらにアクセス保護をおこなうために、サブプログラム(SPF または MPF 識別子)を ACCESS ファイルで呼び出すことができます。サブプログラムは、呼び出しをおこなう ACCESS ファイルの実行権を継承します。

注記

ACCESS ファイルで再定義できるのは、アクセス権のみです。他のすべての属性のプログラム指令/再定義は、対応する定義ファイルで続行してください。

下記も参照

変数 (ページ 433)

4.1.1.11 定義と再定義が可能な属性の概要

次の表は、どのデータタイプに対して、どの属性が定義(DEF)、および/または再定義(REDEF)可能かを示します。

システムデータ

データタイプ	初期値	制限値	物理単位	アクセス権	データクラス (828D のみ)
マシンデータ	---	---	---	REDEF	REDEF
セッティングデータ	REDEF	---	---	REDEF	---
FRAME データ	---	---	---	REDEF	---
プロセスデータ	---	---	---	REDEF	---
ピッチ誤差補正(EEC)	---	---	---	REDEF	---
真直度補正(CEC)	---	---	---	REDEF	---
象限突起補償(QEC)	---	---	---	REDEF	---
マガジンデータ	---	---	---	REDEF	---
工具データ	---	---	---	REDEF	---
プロテクションゾーン	---	---	---	REDEF	---
工具ホルダ、旋回	---	---	---	REDEF	---
キネマティック結合	---	---	---	REDEF	---
3次元プロテクションゾーン	---	---	---	REDEF	---
作業領域リミット	---	---	---	REDEF	---

ユーザーデータ

データタイプ	初期値	制限値	物理単位	アクセス権	データクラス
R 変数	REDEF	REDEF	REDEF	REDEF	---
シンクロナイズドアクション変数(\$AC_...)	REDEF	REDEF	REDEF	REDEF	---
シンクロナイズドアクション GUD(SYG_...)	REDEF	REDEF	REDEF	REDEF	---
EPS 変数	REDEF	REDEF	REDEF	REDEF	---
OEM 工具データ	REDEF	REDEF	REDEF	REDEF	---
OEM マガジンデータ	REDEF	REDEF	REDEF	REDEF	---
グローバルユーザー変数(GUD)	DEF/REDEF	DEF	DEF	DEF/REDEF	DEF/REDEF
ローカルユーザー変数(PUD/ LUD)	DEF	DEF	DEF	---	---

4.1 フレキシブルなNCプログラミング

下記も参照

変数 (ページ 433)

4.1.1.12 配列変数の定義と初期化(DEF、SET、REP)

ユーザー変数を、1次元から3次元までの配列として定義できます。

- 1次元:DEF <データタイプ> <変数名称> [<n>]
- 2次元:DEF <データタイプ> <変数名称> [<n>,<m>]
- 3次元:DEF <データタイプ> <変数名称> [<n>,<m>,<o>]

注記

STRING データタイプのユーザー変数は、2次元までの配列として定義できます。

データタイプ

ユーザー変数を、次のデータタイプの配列として定義できます。BOOL、CHAR、INT、REAL、STRING、AXIS、FRAME

配列要素への値の割り当て

次のタイミングで、値を配列要素に割り当てることができます。

- 配列定義のとき(初期値)
- プログラム実行中

値は、次の方法で割り当てることができます。

- 配列要素の明示的な指定
- 開始要素としての配列要素の明示的な指定と値リストの指定(SET)
- 開始要素としての配列要素の明示的な指定、および値とその繰り返し頻度の指定(REP)

注記

FRAME データタイプのユーザー変数には、初期値を割り当ててはできません。

構文(DEF)

```
DEF <データタイプ> <変数名称> [<n>,<m>,<o>]  
DEF STRING [<文字列長>] <変数名称> [<n>,<m>]
```


構文(DEF...=SET...)

値リストの使用:

- 定義のとき:

DEF <データタイプ> <変数名称> [<n>, <m>, <o>] = SET (<値 1>, <値 2>, ...)

以下も同等です。

DEF <データタイプ> <変数名称> [<n>, <m>, <o>] = (<値 1>, <値 2>, ...)

注記

値リストによる初期化の場合は、SET を指定する必要はありません。

- 値の割り当てのとき:

<変数名称> [<n>, <m>, <o>] = SET (<値 1>, <値 2>, ...)

構文(DEF...=REP...)

繰り返し値の使用:

- 定義のとき:

DEF <データタイプ> <変数名称> [<n>, <m>, <o>] = REP (<値>)

DEF <データタイプ> <変数名称> [<n>, <m>, <o>] = REP (<値>, <配列要素数>)

- 値の割り当てのとき:

<変数名称> [<n>, <m>, <o>] = REP (<値>)

DEF <データタイプ> <変数名称> [<n>, <m>, <o>] = REP (<値>, <配列要素数>)

意味

DEF:	変数を定義する命令
<データタイプ>:	変数のデータタイプ
	値の範囲: • システム変数の場合: BOOL、CHAR、INT、REAL、STRING、AXIS • GUD 変数または LUD 変数の場合: BOOL、CHAR、INT、REAL、STRING、AXIS、FRAME
<文字列長>:	STRING データタイプの最大文字数
<変数名称>:	変数名称
[<n>, <m>, <o>]:	配列サイズまたは配列インデックス

4.1 フレキシブルな NC プログラミング

<n>:	1 次元の配列サイズまたは配列インデックス	
	タイプ:	INT (システム変数の場合は AXIS も)
	値の範囲:	最大配列サイズ:65535 配列インデックス: $0 \leq n \leq 65534$
<m>:	2 次元の配列サイズまたは配列インデックス	
	タイプ:	INT (システム変数の場合は AXIS も)
	値の範囲:	最大配列サイズ:65535 配列インデックス: $0 \leq m \leq 65534$
<o>:	3 次元の配列サイズまたは配列インデックス	
	タイプ:	INT (システム変数の場合は AXIS も)
	値の範囲:	最大配列サイズ:65535 配列インデックス: $0 \leq o \leq 65534$
SET:	指定した値リストによる値の割り当て	
(<値 1>,<値 2>, ...):	値リスト	
REP:	指定した<値>を使用した値割り当て	
<値>:	REP による初期化時に配列要素に書き込まれる値。	
<配列要素数>:	指定した<値>で書き込む配列要素数残りの配列要素には、そのタイミングに応じて、次の規則が適用されます。 - 配列定義時の初期化: →残りの配列要素にゼロが書き込まれます。 - プログラム実行中の割り当ての場合: →配列要素の現在値は変更されません。 このパラメータをプログラム指令しない場合は、すべての配列要素に<値>が書き込まれます。 このパラメータがゼロの場合は、そのタイミングに応じて、次の規則が適用されます。 - 配列定義時の初期化: →すべての要素にゼロが事前に割り当てられます。 - プログラム実行中の割り当ての場合: →配列要素の現在値は変更されません。	

配列インデックス

SET または REP による値の割り当ての場合などでは、配列要素の順序は自動的に、配列インデックスが右から左への繰り返しになります。

例:24 個の配列要素による 3 次元配列の初期化:

```
DEF INT FELD[2,3,4] = REP(1,24)

FELD[0,0,0] = 1    1 番目の配列エレメント
FELD[0,0,1] = 1    2 番目の配列エレメント
FELD[0,0,2] = 1    3 番目の配列エレメント
FELD[0,0,3] = 1    4 番目の配列エレメント
...
FELD[0,1,0] = 1    5 番目の配列エレメント
FELD[0,1,1] = 1    6 番目の配列エレメント
...
FELD[0,2,3] = 1    12 番目の配列エレメント
FELD[1,0,0] = 1    13 番目の配列エレメント
FELD[1,0,1] = 1    14 番目の配列エレメント
...
FELD[1,2,3] = 1    24 番目の配列エレメント
```

これは、以下に対応します。

```
FOR n=0 TO 1
  FOR m=0 TO 2
    FOR o=0 TO 3
      FELD[n,m,o] = 1
    ENDFOR
  ENDFOR
ENDFOR
```

例:配列変数全体の初期化

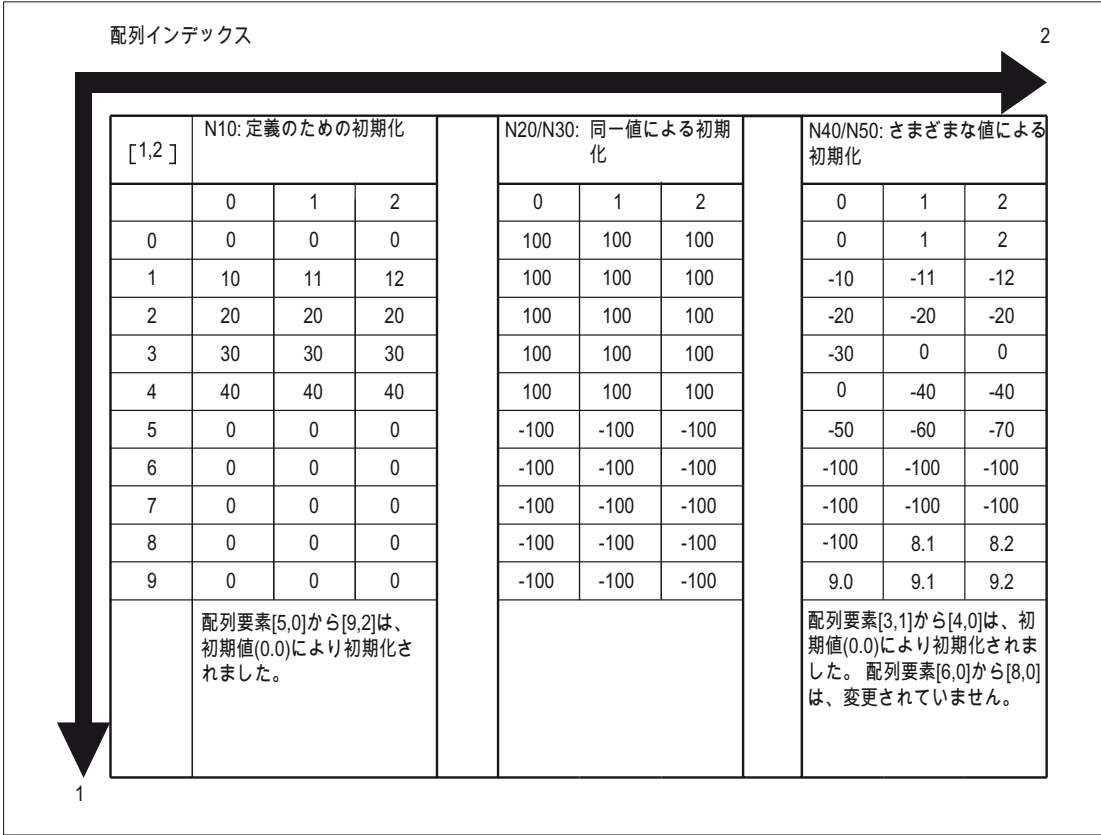
実際の割り当てについては、図を参照してください。

プログラムコード
N10 DEF REAL FELD1[10,3]=SET(0,0,0,10,11,12,20,20,20,30,30,30,40,40,40,)

4.1 フレキシブルなNCプログラミング

プログラムコード

N20 ARRAY1[0,0] = REP(100)
N30 ARRAY1[5,0] = REP(-100)
N40 FELDL[0,0]=SET(0,1,2,-10,-11,-12,-20,-20,-20,-30, , , -40,-40,-50,-60,-70)
N50 FELDL[8,1]=SET(8.1,8.2,9.0,9.1,9.2)



下記も参照

配列変数の定義と初期化(DEF、SET、REP) (ページ 468)

変数 (ページ 433)

追加情報(SET)

定義のための初期化

- 1 番目の配列要素から、値リストにプログラム指令された要素と同じ配列要素のすべてに、値リストの値が割り当てられます。
- 値 0 は、値リストで明示的に値が宣言されずに配列要素に割り当てられます(値リストの隙間)。

- AXIS データタイプの変数の場合は、値リストの隙間は使用できません。
- 値リストに、定義した配列要素より多くの値が含まれる場合は、アラームが表示されます。

プログラム実行による値の割り当て

プログラム実行で値を割り当てる場合は、上記の定義の規則が適用されます。以下のオプションもサポートされています。

- 値リストでは、式も要素として使用できます。
- 値の割り当ては、プログラム指令した配列インデックスから開始されます。値を選択してサブ配列に割り当てることができます。

例:

プログラムコード	コメント
DEF INT ARRAY[5,5]	; 配列を定義します。
ARRAY[0,0]=SET(1,2,3,4,5)	; 最初の5個の配列要素[0,0]～[0,4]への値の割り当て
ARRAY[0,0]=SET(1,2, , ,5)	; 最初の5個の配列要素[0,0]～[0,4]への隙間をもった値の割り当て、配列要素[0,2]、および[0,3] = 0
ARRAY[2,3]=SET(VARIABLE,4*5.6)	; 配列インデックス[2,3]から始まる、変数と式による値の割り当て: [2,3] = VARIABLE [2,4] = 4 * 5.6 = 22.4

詳細情報(REP)

定義のための初期化

- すべてまたは任意に選択した数の配列要素数を、指定した値(定数)で初期化します。
- FRAME データタイプの変数は初期化できません。

例:

プログラムコード	コメント
DEF REAL varName[10]=REP(3.5,4)	; 配列を定義して、配列要素[0]～[3]を値3.5で初期化します

プログラム実行による値の割り当て

プログラム実行で値を割り当てる場合は、上記の定義の規則が適用されます。以下のオプションもサポートされています。

- 値リストでは、式も要素として使用できます。
- 値の割り当ては、プログラム指令した配列インデックスから開始されます。値を選択してサブ配列に割り当てることができます。

4.1 フレキシブルなNC プログラミング

例:

プログラムコード	コメント
DEF REAL varName[10]	; 配列を定義します。
varName[5]=REP(4.5,3)	; 配列要素[5]～[7] = 4.5
R10=REP(2.4,3)	; R変数 R10～R12 = 2.4
DEF FRAME FRM[10]	; 配列を定義します。
FRM[5]=REP(CTTRANS(X,5))	; 配列要素[5]～[9] = CTRANS(X,5)

下記も参照

配列変数の定義と初期化(DEF、SET、REP) (ページ 468)

4.1.1.13 データタイプ

NC では、以下のデータタイプを使用できます。

データタイプ	意味	値の範囲
INT	符号付き整数	-2147483648 ... +2147483647
REAL	実数(IEEE では LONG REAL)	$\pm(\sim 2,2 \times 10^{-308} \dots \sim 1,8 \times 10^{+308})$
BOOL	真偽値 TRUE (1)と FALSE (0)	1, 0
CHAR	ASCII 文字	ASCII コード 0～255
STRING	定義された長さの文字列	最大 200 文字の文字(特殊文字は使用できません)
AXIS	軸/主軸の識別子	チャンネル軸識別子
FRAME	内部的な座標変換の形状パラメータ (平行移動、回転、スケーリング、ミラーリング)	---

自動データタイプ変換

割り当てのとき、およびパラメータ転送のときには、次のデータタイプ変換が可能で、自動的におこなわれます。

変換前↓/変換後 →	REAL	INT	BOOL
REAL	X	O	&
INT	X	X	&

変換前↓/変換後 →	REAL	INT	BOOL
BOOL	X	X	X
x:制限はなし O:; 変換後の値の範囲により、データ損失のおそれがあります⇒アラーム; 数値の四捨五入:小数点以下の値 $\geq 0.5 \Rightarrow$ 切り上げ、小数点以下の値 $< 0.5 \Rightarrow$ 切り捨て &;値 $\neq 0 \Rightarrow$ TRUE、値 $== 0 \Rightarrow$ FALSE			

下記も参照

変数 (ページ 433)

4.1.1.14 変数の最小値、最大値、および範囲(MINVAL、MAXVAL、および BOUND)

MINVAL および MAXVAL 命令は、2 つの変数の値を比較します。結果として、小さい方の値(MINVAL の場合)または大きい方の値(MAXVAL の場合)がそれぞれ得られます。

BOUND コマンドは、テスト変数の値が、定義した値の範囲内に入るかどうかをテストします。

構文

<小さい方の値>=MINVAL (<変数 1>,<変数 2>)
<大きい方の値>=MAXVAL (<変数 1>,<変数 2>)
<戻り値>=<BOUND> (<最小値>,<最大値>,<テスト変数>)

意味

MINVAL:	2 個の変数(<変数 1>、<変数 2>)のうち、 小さい方 の値が得られます。
<小さい方の値>:	MINVAL 命令の結果変数 小さい方の変数値に設定されます。
MAXVAL:	2 個の変数(<変数 1>、<変数 2>)のうち、 大きい方 の値が得られます。
<大きい方の値>:	MAXVAL 命令の結果変数 大きい方の変数値に設定されます。
BOUND:	変数(<テスト変数>)が、定義した値の範囲内にあるかどうかをテストします。
<最小値>:	値の範囲の最小値を定義する変数。

4.1 フレキシブルな NC プログラミング

<最大値>:	値の範囲の最大値を定義する変数。
<戻り値>:	BOUND 命令の結果変数 テスト変数の値が定義した値の範囲内にある場合は、結果変数がテスト変数の値に設定されます。 テスト変数の値が最大値より大きい場合は、結果変数が定義範囲の最大値に設定されます。 テスト変数の値が最小値より小さい場合は、結果変数が定義範囲の最小値に設定されます。

注記

MINVAL、MAXVAL、および BOUND は、シンクロナイズドアクションでもプログラム指令できます。

注記

両方の値が等しい場合の動作

両方の値が等しい場合は、MINVAL/MAXVAL では、この等しい値に設定されます。BOUND の場合は、テストされる変数の値がそのまま戻されます。

例

プログラムコード	コメント
DEF REAL rVar1=10.5, rVar2=33.7, rVar3, rVar4, rVar5, rValMin, rValMax, rRetVar	
rValMin=MINVAL(rVar1,rVar2)	; rValMin を値 10.5 に設定します。
rValMax=MAXVAL(rVar1,rVar2)	; rValMax を値 33.7 に設定します。
rVar3=19.7	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 は範囲内です。rRetVar を 19.7 に設定します。
rVar3=1.8	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 は最小値を下回っています。rRetVar を 10.5 に設定します。
rVar3=45.2	
rRetVar=BOUND(rVar1,rVar2,rVar3)	; rVar3 は最大値を上回っています。rRetVar を 33.7 に設定します。

4.1.1.15 変数の有効性確認(ISVAR)

事前定義された ISVAR 機能を使用して、システム変数/ユーザー変数(マシンデータ、セッティングデータ、システム変数、GUD などの一般変数)が NC で既知であるかどうかを確認できます。

変数

確認対象の変数の構造は以下のようになります。

無次元変数: <変数>
 配列インデックス無しの 1 次元変数: <変数>[]
 配列インデックス n 有りの 1 次元変数: <変数>[<n>]
 配列インデックス無しの 2 次元変数: <変数>[,]
 配列インデックス n および m 有りの 2 次元 <変数>[<n>,<m>]
 変数:

構文

<結果>=ISVAR (<変数> [<n>,<m>])

意味

<結果>:	戻り値		
	データタイプ:	BOOL	
	値の範囲:	1	使用可能な変数
		0	未知の変数
ISVAR:	システム変数/ユーザー変数が NC で既知かどうかを確認します		
<変数>:	システム変数/ユーザー変数の名称		
	データタイプ:	STRING	
<n>:	第 1 次元の配列インデックス(オプション)		
	データタイプ:	INT	
<m>:	第 2 次元の配列インデックス(オプション)		
	データタイプ:	INT	

以下の確認が、転送パラメータに従っておこなわれます。

- 名称は既知か
- 変数は配列か
- 1 次元配列か、2 次元配列か
- それぞれの配列インデックスは許容範囲内か

すべての確認結果が「はい」の場合のみ、TRUE (1)が返されます。

4.1 フレキシブルなNCプログラミング

確認結果が「いいえ」、またはシンタックスエラーが発生した場合、FALSE (0)が返されます。

例

プログラムコード	コメント
DEF INT VAR1	
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("VAR1")	; この場合、IS_VAR は TRUE です。

プログラムコード	コメント
DEF REAL VARARRAY[10,10]	
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("VARARRAY[,]")	; この場合、IS_VAR は TRUE で 2 次元配列です。
N20 IS_VAR=ISVAR("VARARRAY")	; IS_VAR は TRUE で、変数が存在します。
N30 IS_VAR=ISVAR("VARARRAY[8,11]")	; IS_VAR は FALSE で、配列インデックスは使用できません。
N40 IS_VAR=ISVAR("VARARRAY[8,8]")	; IS_VAR は FALSE で、「]」が抜けています (構文エラー)。
N50 IS_VAR=ISVAR("VARARRAY[,8]")	; IS_VAR は TRUE で、配列インデックスを使用できます
N60 IS_VAR=ISVAR("VARARRAY[8,]")	; ; IS_VAR は TRUE で、配列インデックスを使用できます

プログラムコード	コメント
DEF BOOL IS_VAR=FALSE	
N100 IS_VAR=ISVAR("\$MC_GCODE_RESET_VALUES[1]")	; 転送パラメータはマシンデータ項目で、IS_VAR は TRUE です。

プログラムコード	コメント
DEF BOOL IS_VAR=FALSE	
N10 IS_VAR=ISVAR("\$P_EP")	; この場合、IS_VAR は TRUE です。
N20 IS_VAR=ISVAR("\$P_EP[X]")	; この場合、IS_VAR は TRUE です。

4.1.1.16 属性値/データタイプ(GETVARPHU、GETVARAP、GETVARLIM、GETVARDIM、GETVARDFT、GETVARTYP)の読み取り

システム変数/ユーザー変数の属性値は、予約機能 GETVARPHU、GETVARAP、GETVARLIM、GETVARDIM、および GETVARDFT を使用して、システム変数/ユーザー変数のデータタイプは、GETVARTYP を使用して読み取ることができます。

物理単位の読み取り

構文:

<結果>=GETVARPHU (<名称>)

意味:

<結果>:	物理単位の数値		
	データタイプ:	INT	
	値の範囲:	「属性:物理単位(PHU) (ページ459)」の表を参照してください	
		障害時	
	- 2	指定した変数名が、システムパラメータまたはユーザー変数に割り当てられていません。	
GETVARPHU:	システム変数/ユーザー変数の物理単位の読み取り		
<名称>:	システム変数/ユーザー変数の名称		
	データタイプ:	STRING	

例:

NC は次の GUD 変数を含みます。

DEF CHAN REAL PHU 42 LLI 0 ULI 10000 electric

プログラムコード	コメント
DEF INT result=0	
result=GETVARPHU ("electric")	; GUD 変数の物理単位を特定します。
IF (result < 0) GOTO error	

結果として値 42 が返されます。これは物理単位[kW]に対応します。

注記

GETVARPHU を使用して、たとえば、2 つの変数を変数割り当て $a = b$ で期待される物理単位を持つかどうかを確認できます。

読み取りアクセス権

構文:

<結果>=GETVARAP (<名称>, <アクセス>)

4.1 フレキシブルな NC プログラミング

意味:

<結果>:	指定した<アクセス>の保護レベル			
	データタイプ:	INT		
	値の範囲:	0 ... 7	「属性: アクセス権(APR、APW、APRP、APWP、APRB、APWB) (ページ 461)」を参照してください	
		障害時		
		- 1	書き込みできません(アクセスタイプ「WP」および「WB」にのみ関連する)	
		- 2	指定した変数名が、システムパラメータまたはユーザー変数に割り当てられていません。	
		- 3	パラメータ<アクセス>の値が不正です	
GETVARAP:	システム変数/ユーザー変数へのアクセス権の読み取り			
<名称>:	システム変数/ユーザー変数の名称			
	データタイプ:	STRING		
<アクセス>:	アクセスのタイプ			
	データタイプ:	STRING		
	値の範囲:	"RP"	パートプログラムによる読み取り	
		"WP"	パートプログラムによる書き込み	
		"RB"	OPI による読み取り	
		"WB"	OPI による書き込み	

例:

プログラムコード	コメント
DEF INT result=0	
result=GETVARAP("\$TC_MAP8","WB")	; OPI による書き込みに対するシステムパラメータ「マガジン位置」のアクセス保護を特定します。
IF (result < 0) GOTO error	

結果として値 7 が返されます。これはキースイッチ位置 0(= アクセス保護なし)に対応します。

注記

GETVARAP を使用して、たとえば、アプリケーションで求められるアクセス権を確認する確認プログラムを実行できます。

制限値の読み取り

構文:

<ステータス>=GETVARLIM(<名称>,<制限値>,<結果>)

意味:

<ステータス>:	機能状態		
	データタイプ:	INT	
	値の範囲:	1	OK
		-1	制限値が定義されていません (AXIS、STRING、FRAME タイプの変数に対して)
		-2	指定した変数名が、システムパラメータまたはユーザー変数に割り当てられていません。
-3		パラメータ<制限値>の値が不正です	
GETVARLIM:	システム変数/ユーザー変数の下限値/上限値の読み取り		
<名称>:	システム変数/ユーザー変数の名称		
	データタイプ:	STRING	
<制限値>:	読み出す制限値を指定します		
	データタイプ:	CHAR	
	値の範囲:	"L "	下限値
		"U "	上限値
<結果>:	制限値を返します		
	データタイプ:	VAR REAL	

4.1 フレキシブルなNCプログラミング

例:

プログラムコード	コメント
DEF INT state=0	
DEF REAL result=0	
state=GETVARLIM("\$MA_MAX_AX_VELO","L",result)	; MD32000 \$MA_MAX_AX_VELO の下限値を特定します。
IF (result < 0) GOTOF error	

次元値の読み取り

構文:

<結果>=GETVARDIM(<名称>,<インデックス>)

意味:

<結果>:	パラメータ<インデックス>が指定する次元のサイズ			
	データタイプ:	INT		
GETVARDIM:	システム/ユーザー変数のフィールドの第 1、第 2 または第 3 次元のサイズの読み取り			
<名称>:	システム変数/ユーザー変数の名称			
	データタイプ:	STRING		
<インデックス>:	配列インデックス			
	データタイプ:	INT		
	値の範囲:	1 ... 3		
		1	フィールドの第 1 次元のインデックス	
		2	フィールドの第 2 次元のインデックス	
		3	フィールドの第 3 次元のインデックス	

例:

プログラムコード	コメント
N5 DEF REAL myReal[5,4]	
N10 R1=GETVARDIM("myReal",1)	; フィールドの第 1 次元のサイズを特定します。 ; 結果:R1 = 5
N15 R2=GETVARDIM("myReal",2)	; フィールドの第 2 次元のサイズを特定します。 ; 結果:R2 = 4

初期値の読み取り

構文:

<状態>=GETVARDFT (<名前>,<結果>[,<インデックス_1>,<インデックス_2>,<インデックス_3>])

意味:

<ステータス>:	機能状態		
	データタイプ:	INT	
	値の範囲:	1	OK
		-1	デフォルト値が存在しません (結果変数のデータタイプ定義がシステム/ユーザー変数のデータタイプと一致しないためなど)。
		-2	指定した変数名が、システムパラメータまたはユーザー変数に割り当てられていません。
		-3	パラメータ<インデックス_1>の値が不正です。1 より小さい次元です(⇒ フィールドはありませんがスカラー変数はあります)
		-4	パラメータ<インデックス_2>の値が不正です
		-5	パラメータ<インデックス_3>の値が不正です
GETVARDFT:	システム変数/ユーザー変数の初期値の読み取り		
<名称>:	システム変数/ユーザー変数の名称		
	データタイプ:	STRING	
<結果>:	初期値を戻します		
	データタイプ:	VAR REAL (INT、REAL、BOOL、AXIS タイプの変数の初期値を読み取る場合)	
		VAR STRING (STRING と CHAR タイプの変数の初期値を読み取る場合)	
		VAR FRAME (FRAME タイプの変数の初期値を読み取る場合)	

4.1 フレキシブルなNC プログラミング

<インデックス _1>:	フィールドの第 1 次元のインデックス(オプション)	
	データタイプ:	INT
	プログラム指令なし= 0	
<インデックス _2>:	フィールドの第 2 次元のインデックス(オプション)	
	データタイプ:	INT
	プログラム指令なし= 0	
<インデックス _3>:	フィールドの第 3 次元のインデックス(オプション)	
	データタイプ:	INT
	プログラム指令なし= 0	

例:

プログラムコード	コメント
DEF INT state=0	
DEF REAL resultR=0	; INT、REAL、BOOL、AXIS タイプの初期値を確定する変数
DEF FRAME resultF=0	; FRAME タイプの初期値を確定する変数
IF (GETVARTYP("\$MA_MAX_AX_VELO") <> 4) GOTO error	
state=GETVARDFT("\$MA_MAX_AX_VELO",resultR, AXTOINT(X))	; 「X」軸の初期値を特定します。
IF (result < 0) GOTO error	
IF (GETVARTYP("\$TC_TP8") <> 3) GOTO error	
state=GETVARDFT("\$TC_TP8", resultR)	
IF (GETVARTYP("\$P_UBFR") <> 7) GOTO error	
state=GETVARDFT("\$P_UBFR", resultF)	

データタイプの読み取り

構文:

<結果>=GETVARTYP(<名称>)

意味:

<結果>:	指定したシステム変数/ユーザー変数のデータタイプ		
	データタイプ:	INT	
	値の範囲:	1	= BOOL
		2	= CHAR
		3	= INT
		4	= REAL
		5	= STRING
		6	= AXIS
		7	= FRAME
		障害時	
<0	指定した変数名が、システムパラメータまたはユーザー変数に割り当てられていません。		
GETVARTYP:	システム変数/ユーザー変数のデータタイプの読み取り		
<名称>:	システム変数/ユーザー変数の名称		
	データタイプ:	STRING	

例:

プログラムコード	コメント
DEF INT result=0	
DEF STRING name="R"	
result=GETVARTYP(name)	; R 変数のタイプを特定します。
IF (result < 0) GOTO error	

結果として値 4 が返されます。これは REAL データタイプに対応します。

4.1.1.17 実行可能なタイプ変換

変数に割り当てた定数値、変数、または式は、変数タイプを互換にしてください。この場合は、値を割り当てると、自動的にタイプが変換されます。

4.1 フレキシブルな NC プログラミング

実行可能なタイプ変換

変換後	REAL	INT	BOOL	CHAR	STRING	AXIS	FRAME
変換前							
REAL	あり	可*	可 ¹⁾	可*	—	—	—
INT	あり	あり	可 ¹⁾	可 ²⁾	—	—	—
BOOL	あり	あり	あり	あり	あり	—	—
CHAR	あり	あり	可 ¹⁾	あり	あり	—	—
STRING	—	—	可 ⁴⁾	可 ³⁾	あり	—	—
AXIS	—	—	—	—	—	あり	—
FRAME	—	—	—	—	—	—	あり

意味

* REAL から INT へのタイプ変換では、0.5 以上の端数値は切り上げられ、それ以外の値は切り捨てられます(ROUND 機能を参照ください)。

1) 値が 0 以外の場合は TRUE、0 の場合は FALSE です。

2) 値が許容範囲内である場合です。

3) 1 文字のみの場合です。

4) 文字列長が 0 以下の場合は FALSE、それ以外の場合は TRUE です。

注記

変換により、値が目標範囲を超える場合は、エラーメッセージが出力されます。

式に複数のタイプが混合している場合は、タイプが自動変換されます。タイプ変換はシンクロナイズドアクションでもできます。「シンクロナイズドアクション、自動的なタイプ変換」の章を参照してください。

4.1.2 間接プログラミング

4.1.2.1 アドレスの間接プログラミング

アドレス (ページ 1485) を間接的にプログラム指令すると、拡張アドレス (<インデックス>) が、適切なタイプの変数に置き換えられます。

注記

次のアドレスは間接的にはプログラム指令できません。

- N(ブロック番号)
- L(サブプログラム)
- 設定可能アドレス
(例えば、X[1]の代わりに X1 は使用できません)

構文

<ADDRESS> [<インデックス>]

意味

<ADDRESS> [...]:	拡張子付き固定アドレス(インデックス)
<インデックス>:	変数、たとえば主軸番号、軸の変数など

例

例 1: 主軸番号の間接プログラミング

直接プログラミング:

プログラムコード	コメント
S1=300	; 主軸番号 1 の速度 300 rpm。

間接プログラミング:

プログラムコード	コメント
DEF INT SPINU=1	; INT タイプの変数と値の割り当てを定義しています。
S[SPINU]=300	; SPINU 変数に保存された番号の主軸の速度が 300 rpm です (この例 1 では、番号が 1 の主軸です)。

4.1 フレキシブルな NC プログラミング

例 2:軸の間接プログラミング

直接プログラミング:

プログラムコード	コメント
FA[U]=300	; 軸「U」の送り速度 300。

間接プログラミング:

プログラムコード	コメント
DEF AXIS AXVAR2=U	; AXIS タイプの変数と値の割り当てを定義しています。
FA[AXVAR2]=300	; AXVAR2 という名称の変数に保存されたアドレス名称の軸の送り速度が、300 です。

例 3:軸の間接プログラミング

直接プログラミング:

プログラムコード	コメント
\$AA_MM[x]	; 軸「X」のプロープ計測値 (MCS) を読み取ります。

間接プログラミング:

プログラムコード	コメント
DEF AXIS AXVAR3=X	; AXIS タイプの変数と値の割り当てを定義しています。
\$AA_MM[AXVAR3]	; 変数 AXVAR3 に保存された名称のプロープ計測値 (MCS) を読み取ります。

例 4:軸の間接プログラミング

直接プログラミング:

プログラムコード	コメント
X1=100 X2=200	

間接プログラミング:

プログラムコード	コメント
DEF AXIS AXVAR1 AXVAR2	; AXIS タイプの変数 2 個を定義しています。
AXVAR1=(X1) AXVAR2=(X2)	; 軸名称を割り当てています。
AX[AXVAR1]=100 AX[AXVAR2]=200	; AXVAR1 と AXVAR2 という名称の変数に保存されたアドレス名称の軸を移動しています。

例 5:軸の間接プログラミング

直接プログラミング:

プログラムコード	コメント
G2 X100 I20	

4.1 フレキシブルな NC プログラミング

間接プログラミング:

プログラムコード	コメント
DEF AXIS AXVAR1=X	; AXIS タイプの変数と値の割り当てを定義しています。
G2 X100 IP[AXVAR1]=20	; 軸の中心点データの間接プログラミング。この軸のアドレス名称は AXVAR1 という名称の変数に保存されています。

例 6 :配列要素の間接プログラミング

直接プログラミング:

プログラムコード	コメント
DEF INT ARRAY1[4,5]	; 配列 1 を定義しています

間接プログラミング:

プログラムコード	コメント
DEFINE DIM1 AS 4	; 配列次元の場合は、配列サイズを固定値で指定してください。
DEFINE DIM2 AS 5	
DEF INT ARRAY[DIM1,DIM2]	
ARRAY[DIM1-1,DIM2-1]=5	

例 7 :間接サブプログラム呼び出し

プログラムコード	コメント
CALL "L" << R10	; R10 に置かれている番号のプログラムを呼び出します (文字列のカスケード化)。

4.1.2.2 G 命令の間接プログラム

G 命令 (ページ 1500)の間接プログラムにより、サイクルを効率的にプログラム指令できます。

構文

G[<グループ>]=<番号>

意味

G[...]:	拡張子付き G 命令(インデックス)	
<グループ>:	インデックスパラメータ:G グループ	
	タイプ:	INT

4.1 フレキシブルな NC プログラミング

<番号>:	G 命令番号の変数	
	タイプ:	INT または REAL

注記

通常、間接プログラム指令ができるのは、構文を特定しない G 命令のみです。
 構文を特定する G 命令では、G グループ 1 のみが可能です。
 構文が特定される G グループ 2、3、4 の G 命令は使用できません。

注記

間接 G 命令プログラム指令では、算術機能は使用できません。G 命令番号を計算する必要がある場合は、間接 G 命令プログラムの前に別のパートプログラム行で計算してください。

例**例 1:設定できるゼロオフセット(G グループ 8)**

プログラムコード	コメント
N1010 DEF INT INT_VAR	
N1020 INT_VAR=2	
...	
N1090 G[8]=INT_VAR G1 X0 Y0	;G54
N1100 INT_VAR=INT_VAR+1	; G 命令演算
N1110 G[8]=INT_VAR G1 X0 Y0	;G55

例 2:レベル選択(G グループ 6)

プログラムコード	コメント
N2010 R10=\$P_GG[6]	; G グループ 6 の有効な G 命令を読み取り
...	
N2090 G[6]=R10	

4.1.2.3 位置属性の間接プログラミング(GP)

軸位置のインクレメンタル指令やアブソリュート指令などの位置属性は、キーワード BP と組み合わせて、変数として間接的にプログラム指令できます。

用途

位置属性の間接プログラミングは**置換サイクル**で使います。この場合、位置属性のキーワード(IC、AC、...)としてのプログラミングに、次の特長があります。

変数による間接プログラミングでは、CASE 命令が**不要**となります。間接プログラミング以外の場合は、この命令が、考えられるすべての位置属性への分岐のために必要になります。

構文

```
<POSITIONING COMMAND> [<軸/主軸>]=
BP (<位置>, <位置属性>)
<軸/主軸>=BP (<位置>, <位置属性>)
```

意味

<POSITIONING COMMAND> []:	次の位置決め命令を、キーワード BP とともにプログラム指令できます。 POS、POSA、SPOS、SPOSA 以下もプログラムできます。 - チャンネルに存在するすべての軸と主軸の識別子: <軸/主軸> - 可変軸/主軸識別子 AX
<軸/主軸>:	位置決めされる軸/主軸
GP ():	位置決めのキーワード
<位置>:	パラメータ 1 軸/主軸位置を表わす定数または変数
<位置属性>:	パラメータ 2 位置属性(位置アプローチモードなど)を表わす変数 (\$P_SUB_SPOSMODE など)やキーワード(IC、AC、...) など

変数で返される値には、次の意味があります。

値	意味	使用できる命令、他
0	位置属性への変更なし	
1	AC	POS、POSA、SPOS、SPOSA、AX、軸アドレス

4.1 フレキシブルなNCプログラミング

値	意味	使用できる命令、他
2	IC	POS、POSA、SPOS、SPOSA、AX、軸アドレス
3	DC	POS、POSA、SPOS、SPOSA、AX、軸アドレス
4	ACP	POS、POSA、SPOS、SPOSA、AX、軸アドレス
5	ACN	POS、POSA、SPOS、SPOSA、AX、軸アドレス
6	OC	-
7	PC	-
8	DAC	POS、POSA、AX、軸アドレス
9	DIC	POS、POSA、AX、軸アドレス
10	RAC	POS、POSA、AX、軸アドレス
11	RIC	POS、POSA、AX、軸アドレス
12	CAC	POS、POSA
13	CIC	POS、POSA
14	CDC	POS、POSA
15	CACP	POS、POSA
16	CACN	POS、POSA

例

マスタ主軸 S1 とスレーブ主軸 S2 の間で動作中の主軸同期連結に対して、主軸を位置決めするために、次の置換サイクルがメインプログラムで SPOS 命令を使用して呼び出されます。

位置決めは、N2230 の命令を使用して実現されます。

```
SPOS[1]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)
```

```
SPOS[2]=GP($P_SUB_SPOSIT,$P_SUB_SPOSMODE)
```

アプローチする位置をシステム変数\$P_SUB_SPOSIT から読み取ります。位置アプローチモードをシステム変数\$P_SUB_SPOSMODE から読み取ります。

プログラムコード	コメント
N1000 PROC LANG_SUB DISPLOF SBLOF	
...	
N2100 IF(\$P_SUB_AXFCT==2)	

4.1 フレキシブルなNC プログラミング

プログラムコード	コメント
N2110	; 動作中の主軸同期連結に対する SPOS/SPOSA/M19 命令の置換
N2185 DELAYFSTON	; 停止遅延領域の始まり
N2190 COUPOF(S2,S1)	; 主軸同期連結の解除
N2200	; マスタ主軸とスレーブ主軸の位置決め
N2210 IF(\$P_SUB_SPOS==TRUE) OR (\$P_SUB_SPOSA==TRUE)	
N2220	; 主軸を、次のように SPOS で位置決めします
N2230 SPOS[1]=GP(\$P_SUB_SPOSIT, \$P_SUB_SPOSMODE)	
SPOS[2]=GP(\$P_SUB_SPOSIT, \$P_SUB_SPOSMODE)	
N2250 ELSE	
N2260	; 主軸を M19 で位置決めします
N2270 M1=19 M2=19	; マスタ主軸とスレーブ主軸の位置決め
N2280 ENDIF	
N2285 DELAYFSTOF	; 停止遅延領域の終わり
N2290 COUPON(S2,S1)	; 主軸同期連結を起動します
N2410 ELSE	
N2420	; さらに置換をおこなうか確認します
...	
N3300 ENDIF	
...	
N9999 RET	

必要条件

シンクロナイズドアクションでは、位置属性の間接プログラミングは使用できません。

4.1.2.4 パートプログラム行の間接プログラミング(EXECSTRING)

パートプログラム命令 EXECSTRING を使用すると、事前に生成した文字列変数をパートプログラム行として実行できます。

構文

EXECSTRING は、次のように個別のパートプログラム行にプログラム指令します。
EXECSTRING (<文字列変数>)

4.1 フレキシブルな NC プログラミング

意味

EXECSTRING:	文字列変数をパートプログラム行として実行する命令
<文字列変数>:	実際に実行するパートプログラム行を含んだ STRING タイプの変数

注記

EXECSTRING を使用すると、パートプログラムのプログラム区間にプログラム指令できるパートプログラム構成は、フロー制御 (ページ 538) を除き、すべて抽出できます。これは、PROC と DEF 命令が、INI ファイルと DEF ファイルでの一般的な使用法と同じように除外されることを意味します。

例

プログラムコード	コメント
N100 DEF STRING[100] MY_BLOCK	; 実行するパートプログラム行を受け取る文字列変数の定義です。
N110 DEF STRING[10] MFCT1="M7"	
...	
N200 EXECSTRING(MFCT1 << "M4711")	; パートプログラム行「M7 M4711」を実行します。
...	
N300 R10=1	
N310 MY_BLOCK="M3"	
N320 IF(R10)	
N330 MY_BLOCK = MY_BLOCK << MFCT1	
N340 ENDIF	
N350 EXECSTRING(MY_BLOCK)	; パートプログラム行「M7 M3」を実行します。

4.1.3 命令

4.1.3.1 算術機能

演算子/算術機能	意味
+	加算
-	減算
*	乗算

4.1 フレキシブルな NC プログラミング

/ ¹⁾	除算 ¹⁾
DIV ¹⁾	整数除算 ¹⁾
MOD ¹⁾	モジュロ除算(整数除算の余りを返します) ¹⁾
:	FRAME 変数用結合演算子
Sin ()	サイン
COS ()	コサイン
TAN ()	タンジェント
ASIN ()	アークサイン
ACOS ()	アークコサイン
ATAN2 (,) ¹⁾	アークタンジェント 2 ¹⁾
SQRT ()	平方根
ABS ()	絶対値
POT ()	2 乗(平方)
TRUNC ()	整数要素 比較演算命令の精度は TRUNC で設定できます(「比較演算誤差の精度補正(TRUNC) (ページ 500)」を参照してください)
ROUND ()	整数になるように四捨五入
LN ()	自然対数
EXP ()	指数関数
MINVAL ()	2 つの変数の小さい方の値 (「変数の最小値、最大値、および範囲(MINVAL、MAXVAL、および BOUND) (ページ 475)」を参照してください)
MAXVAL ()	2 つの変数の大きい方の値 (「変数の最小値、最大値、および範囲(MINVAL、MAXVAL、および BOUND) (ページ 475)」を参照してください)
BOUND ()	定義した数値の範囲内の変数値 (「変数の最小値、最大値、および範囲(MINVAL、MAXVAL、および BOUND) (ページ 475)」を参照してください)
CTrans ()	オフセット
CROT ()	回転
CSCALE ()	スケーリングの変更

4.1 フレキシブルな NC プログラミング

CMIRROR ()	ミラーリング
1) 「例」の節を参照してください	

プログラミング

算術機能には、通常の数学表記を使用します。実行の優先度を括弧で示します。角度は、三角関数と、その逆三角関数で指定します(直角= 90°)。

例

除算: /

(REAL タイプ) = (INT タイプまたは REAL タイプ) / (INT タイプまたは REAL タイプ);

例: $3 / 4 = 0.75$

整数除算: DIV

(INT タイプ) = (INT タイプまたは REAL タイプ) / (INT タイプまたは REAL タイプ);

例: $7 \text{ DIV } 4.1 = 1$

モジュロ除算(整数除算の余りを返します): MOD

(REAL タイプ) = (INT タイプまたは REAL タイプ) MOD (INT タイプまたは REAL タイプ);

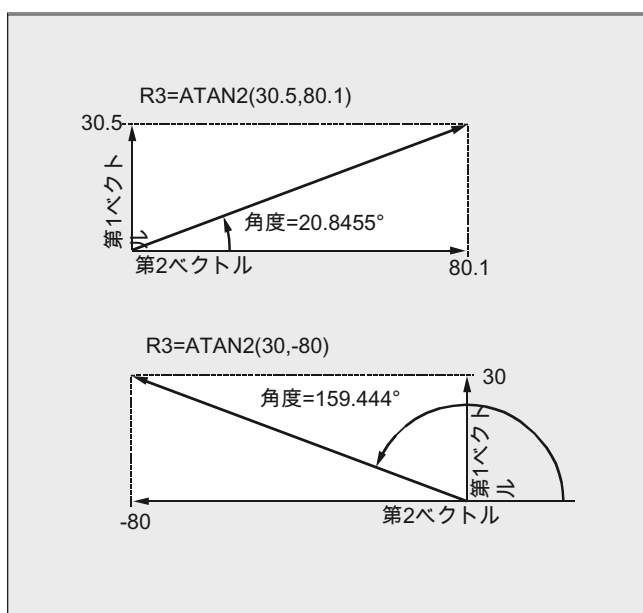
例: $7 \text{ MOD } 4.1 = 2.9$

アークタンジェント 2: ATAN2

この関数は、2つの相互に直交するベクトルの合計ベクトルの角度を計算します。

結果は、4つの象限($-180^\circ < 0 < +180^\circ$)のいずれかに該当します。

基準角度は常に、2番目のベクトル値の正方向を基準にした角度です。



プログラミング例

プログラムコード	コメント
R1=R1+1	; 新しい R1 = 古い R1 + 1
R1=R2+R3 R4=R5-R6 R7=R8*R9	
R10=R11/R12 R13=SIN(25.3)	
R14=R1*R2+R3	; 乗算または除算は、加算または減算より先に実行します。
R14=(R1+R2)*R3	; 括弧内の式を最初に計算します。
R15=SQRT(POT(R1)+POT(R2))	; 内側の括弧内を最初に計算します。
	R15 = ((R1^2 + R2^2)) の平方根
RESFRAME= FRAME1:FRAME2	; 結合演算子による FRAME 論理演算
FRAME3=CTTRANS (...):CROT (...)	FRAME 成分での値の割り当て

4.1.3.2 比較演算と論理演算

比較演算子を使用すると、ジャンプの条件などを定式化できます。複雑な式も比較できます。

比較演算子は、CHAR、INT、REAL、および BOOL タイプの変数に適用されます。コード値は CHAR タイプで比較されます。

タイプが STRING、AXIS、および FRAME の場合、STRING タイプの演算子に使用できる == と <> を、シンクロナイズドアクションでも使用できます。

比較演算の結果は常に BOOL タイプです。

真の値をリンクするには、**論理演算子**を使用します。

4.1 フレキシブルな NC プログラミング

論理演算子は、BOOL タイプの変数にのみ使用できます。ただし、内部タイプ変換によって CHAR、INT、および REAL データタイプにも使用できます。

論理(ブール)演算の場合は、BOOL、CHAR、INT、および REAL データタイプには以下が適用されます。

- 0 の場合:FALSE
- 0 以外の場合:TRUE

ビット単位の論理演算子

論理演算子は、CHAR と INT タイプの単独ビットにも適用できます。タイプは自動的に変換されます。

プログラミング

関係演算子	意味
==	等しい
<>	等しくない
>	より大きい
<	より小さい
>=	等しいかそれより大きい
<=	等しいかそれより小さい

論理演算子	意味
および	論理積
OR	論理和
NOT	否定
XOR	排他的論理和

ビット単位の論理演算子	意味
B_AND	ビット毎の論理積
B_OR	ビット毎の論理和
B_NOT	ビット毎の否定
B_XOR	ビット毎の排他的論理和

注記

算術式では、通常の優先規則にかかわらず、括弧を使用して、すべての演算子の実行順序を指定できます。

注記

BOOLEAN オペランドと演算子の間はスペースを空けてください。

注記

演算子 B_NOT は 1 つのオペランドのみ参照します。このオペランドは、演算子の後に置かれます。

例**例 1:比較演算子**

```
IF R10>=100 GOTOF DEST
```

または

```
R11=R10>=100
```

```
IF R11 GOTOF DEST
```

R10>=100 比較演算の結果は、まず、R11 に保持されます。

例 2:論理演算子

```
IF (R10<50) AND ($AA_IM[X]>=17.5) GOTOF DESTINATION
```

または

```
IF NOT R10 GOTOB START
```

NOT は 1 つのオペランドのみ参照します。

例 3:ビット単位の論理演算子

```
IF $MC_RESET_MODE_MASK B_AND 'B10000' GOTOF ACT_PLANE
```

4.1.3.3 演算子の優先度

各演算子には優先度が割り当てられます。式を解析するときには、常に最も優先度が高い演算子が最初に使用されます。複数の演算子の優先度が同じである場合は、左から右へと解析されます。

算術式では、通常の優先規則にかかわらず、括弧を使用して、すべての演算子の実行順序を指定できます。

4.1 フレキシブルな NC プログラミング

演算の順序

優先度の高い方から低い方へ

1.	NOT, B_NOT	否定、ビット毎の否定
2.	*, /, DIV, MOD	乗算、除算
3.	+, -	加算、減算
4.	B_AND	ビット毎の論理積
5.	B_XOR	ビット毎の排他的論理和
6.	B_OR	ビット毎の論理和
7.	および	論理積
8.	XOR	排他的論理和
9.	OR	論理和
10.	<<	文字列の結合、結果は STRING タイプ
11.	==, <>, >, <, >=, <=	比較演算子

注記

フレーム用の結合演算子「:」は、他の演算子と同じような式で使用できません。したがって、この演算子には優先度はありません。

例:IF 命令

```
If (otto==10) and (anna==20) goto f end
```

4.1.3.4 比較演算誤差の精度補正(TRUNC)

TRUNC 命令は、精度係数で乗算されたオペランドの切り捨てを実行します。

設定可能な比較演算命令の精度

REAL タイプのプログラムデータは、内部では IEEE 形式の 64 ビットで表わされます。この表示形式により、10 進数が不正確に表現され、最適な計算値と比較したときに予期しない結果となる場合があります。

相対等式値

この表示形式が原因による不正確さによってプログラムの流れが中断されないように、比較演算命令では、絶対等式値ではなく相対等式値でチェックします。

構文

比較演算誤差の精度補正

TRUNC (R1*1000)

意味

TRUNC:	小数点以下を切り捨て
--------	------------

10⁻¹² の精度の相対等式値が考慮される演算子

- 等しい: (==)
- 等しくない: (<>)
- 以上: (>=)
- 以下: (<=)
- より大きい/より小さい: (><) (絶対等式値を使用して比較します)
- より大きい: (>)
- より小さい: (<)

互換性

互換性の理由により、(>)と(<)の相対等式のチェックは、マシンデータを MD10280 \$MN_PROG_FUNCTION_MASK Bit0 = 1 に設定して無効にできます。

注記

REAL タイプのデータとの比較演算は、上記の理由により、不正確な結果となることがあります。誤差が大きすぎる場合は、オペランドを精度係数で乗算して、TRUNC で切り捨てをおこなう INTEGER 計算を使用してください。

シンクロナイズドアクション

比較演算命令の動作説明は、シンクロナイズドアクションにも該当します。

例

例 1: 精度の配慮

プログラムコード	コメント
N40 R1=61.01 R2=61.02 R3=0.01	; 初期値の割り当て
N41 IF ABS(R2-R1) > R3 GOTOF ERROR	; 今まではジャンプが実行されていたはずです。

4.1 フレキシブルなNCプログラミング

プログラムコード	コメント
N42 M30	; プログラム終了
N43 ERROR:SETAL(66000)	
R1=61.01 R2=61.02 R3=0.01	; 初期値の割り当て
R11=TRUNC(R1*1000) R12=TRUNC(R2*1000)	; 精度補正
R13=TRUNC(R3*1000)	
IF ABS(R12-R11) > R13 GOTOF ERROR	; ジャンプは実行されません
M30	; プログラム終了
ERROR:SETAL(66000)	

例 2 :両方のオペランドの商の計算と評価

プログラムコード	コメント
R1=61.01 R2=61.02 R3=0.01	; 初期値の割り当て
IF ABS((R2-R1)/R3)-1 > 10EX-5 GOTOF ERROR	; ジャンプは実行されません
M30	; プログラム終了
ERROR:SETAL(66000)	

4.1.3.5 切り上げ(ROUNDUP)

入力値、REAL タイプ(小数点有りの有理数)は、「ROUNDUP」機能を使用して、次に大きい整数に切り上げることができます。

構文

ROUNDUP (<値>)

意味

ROUNDUP:	入力値を切り上げる命令
<値>:	入力値、REAL タイプ

注記

入力値が INTEGER タイプ(整数)の場合は、変更されずに返されます。

例

例 1:さまざまな入力値とその切り上げ結果

例	切り上げ結果
ROUNDUP (3.1)	4.0
ROUNDUP (3.6)	4.0
ROUNDUP (-3.1)	-3.0
ROUNDUP (-3.6)	-3.0
ROUNDUP (3.0)	3.0
ROUNDUP (3)	3.0

例 2:NC プログラムの ROUNDUP

プログラムコード
N10 X=ROUNDUP (3.5) Y=ROUNDUP (R2+2)
N15 R2=ROUNDUP (\$AA_IM[Y])
N20 WHEN X=100 DO Y=ROUNDUP (\$AA_IM[X])
...

4.1.4 文字列演算子

文字列演算子

従来の演算子の他に、次の「割り当て」と「比較演算」の文字列演算子が実行できます。

- STRING へのタイプ変換(AXSTRING) (ページ 504)
- STRING からのタイプ変換(NUMBER、ISNUMBER、AXNAME) (ページ 505)
- 文字列の結合(<<) (ページ 506)
- 小文字/大文字への変換(TOLOWER、TOUPPER) (ページ 508)
- 文字列の長さの特定(STRLEN) (ページ 508)
- 文字列中の文字/文字列の検索(INDEX、RINDEX、MINDEX、MATCH) (ページ 509)
- 抽出文字列の選択(SUBSTR) (ページ 510)

4.1 フレキシブルなNC プログラミング

- 個々の文字の読み取りと書き込み (ページ 511)
- 文字列のフォーマット (SPRINT) (ページ 513)

0 文字の特別な意味

内部では、0 文字は、文字列の終了識別子として解釈されます。文字を 0 文字に置き換えると、それ以降の文字列は切り捨てられます。

例:

プログラムコード	コメント
DEF STRING[20] STRG="axis . stationary"	
STRG[6]="X"	
MSG (STRG)	; メッセージ「axis X stationary」を返します。
STRG[6]=0	
MSG (STRG)	; メッセージ「axis」を返します。

4.1.4.1 STRING へのタイプ変換 (AXSTRING)

「STRING へのタイプ変換」機能により、さまざまなタイプの変数を、メッセージの成分として使用できます (MSG)。

<< 演算子を使用すると、データタイプ INT、REAL、CHAR、および BOOL については、この機能が自動的に実行されます (「文字列の結合 (<<) (ページ 506)」を参照してください)。

INT 値は、通常の読み取り可能な書式に変換されます。REAL 値は小数点以下 10 桁までの値に変換されます。

AXIS タイプの変数は、AXSTRING 命令を使用して STRING に変換できます。

構文

<STRING_RES> = << <任意のタイプ>
<STRING_RES> = AXSTRING (<軸識別子>)

意味

<STRING_RES>:	タイプ変換の結果の変数	
	タイプ:	STRING
<任意のタイプ>:	変数のタイプ INT、REAL、CHAR、STRING、および BOOL	

AXSTRING:	AXSTRING 命令は、指定した軸識別子を文字列として返します。	
<軸識別子>:	軸識別子の変数	
	タイプ:	AXIS

注記

FRAME 変数は変換できません。

4.1.4.2 STRING からのタイプ変換(NUMBER、ISNUMBER、AXNAME)

NUMBER 命令を使用して、STRING から REAL に変換します。変換できるかどうかは、ISNUMBER 命令を使用して確認できます。

文字列は、AXNAME 命令を使用して軸データタイプに変換します。

構文

```
<REAL_RES>=NUMBER("<文字列>")  
<BOOL_RES>=ISNUMBER("<文字列>")  
<AXIS_RES>=AXNAME("<文字列>")
```

意味

NUMBER:	NUMBER 命令は、<文字列>で示された数値を REAL 値として返します。	
<文字列>:	変換する STRING タイプの変数	
<REAL_RES>:	NUMBER によるタイプ変換の結果の変数	
	タイプ:	REAL
ISNUMBER:	ISNUMBER 命令は、<文字列>を有効な数値に変換できるかどうかを確認します。	

4.1 フレキシブルなNC プログラミング

<BOOL_RES>:	ISNUMBER による問い合わせ結果の変数		
	タイプ:	BOOL	
	値:	TRUE	ISNUMBER は、<文字列>が、言語規則に準じた有効な REAL 数値を表わす場合は、値 TRUE を返します。
		FALSE	NUMBER を同じ<文字列>で呼び出したときに、ISNUMBER が値 FALSE を返す場合は、アラームが発生します。
AXNAME:	AXNAME 命令は、指定した<文字列>を軸識別子へ変換します。 注: 設定した軸識別子に<文字列>が割り当てられない場合は、アラームが発生します。		
<AXIS_RES>:	AXNAME によるタイプ変換の結果の変数		
	タイプ:	AXIS	

例

プログラムコード	コメント
DEF BOOL BOOL_RES	
DEF REAL REAL_RES	
DEF AXIS AXIS_RES	
REAL_RES == 1234.9876Ex-7	; BOOL_RES == TRUE
BOOL_RES=ISNUMBER("1234XYZ")	; BOOL_RES == FALSE
REAL_RES=NUMBER("1234.9876Ex-7")	; REAL_RES == 1234.9876Ex-7
AXIS_RES=AXNAME("X")	; AXIS_RES == X

4.1.4.3 文字列の結合(<<)

「文字列の結合」機能を使用すると、個々の成分で文字列を設定できます。

結合を実行するには、演算子「<<」を使用します。この演算子では、基本タイプである CHAR、BOOL、INT、REAL、および STRING のすべての組み合わせの目標タイプが STRING です。必要な変換はいずれも、既存の規則に従って実行されます。

構文

<任意のタイプ> << <任意のタイプ>

意味

<任意のタイプ>:	CHAR、BOOL、INT、REAL、または STRING タイプの変数
<< :	変数(<任意のタイプ>)を結合して、文字列(STRING タイプ)を作成する演算子。 この演算子はいわゆる「単一」タイプとして、単独でも使用できます。これは、STRING への明示的なタイプ変換(FRAME と AXIS は変換できません)に使用できます。 << <任意のタイプ>

たとえば、次のように、メッセージまたは命令をテキストリストから設定後に、パラメータを挿入できます(ブロック名称など)。

```
MSG (STRG_TAB[LOAD_IDX]<<BLOCK_NAME)
```

注記

文字列結合の中間の結果が、最大文字列長を超えないようにしてください。

注記

FRAME と AXIS タイプは、演算子「<<」と組み合わせて使用することはできません。

例

例 1:文字列の結合

プログラムコード	コメント
DEF INT IDX=2 DEF REAL VALUE=9.654 DEF STRING[20] STRG="INDEX:2" IF STRG=="Index:"<<IDX GOTOF NO_MSG MSG("Index:"<<IDX<<"/value:"<<VALUE) NO_MSG:	; 表示: "Index:2/value:9.654"

例 2 :<<による明示的なタイプ変換

プログラムコード	コメント
DEF REAL VALUE=3.5 <<VALUE	; 指定した REAL タイプの変数を STRING タイプに変換します。

4.1 フレキシブルなNCプログラミング

4.1.4.4 小文字/大文字への変換(TOLOWER、TOUPPER)

「小文字/大文字の英字への変換」機能を使用すると、文字列のすべての英字を標準表記に変換できます。

構文

```
<STRING_RES>=TOUPPER("<文字列>")
<STRING_RES>=TOLOWER("<文字列>")
```

意味

TOUPPER:	TOUPPER 命令を使用すると、文字列のすべての英字が 大文字 の英字に変換されます。	
TOLOWER:	TOLOWER 命令を使用すると、文字列のすべての英字が 小文字 の英字に変換されます。	
<文字列>:	変換する文字列	
	タイプ:	STRING
<STRING_RES>:	変換結果の変数	
	タイプ:	STRING

例

ユーザーは操作画面で入力を開始できるため、標準の文字表記(大文字または小文字)を使用できます。

```
プログラムコード
DEF STRING [29] STRG
...
IF "LEARN.CNC"==TOUPPER(STRG) GOTOF LOAD_LEARN
```

4.1.4.5 文字列の長さの特定(STRLEN)

STRLEN 命令は、文字列の長さを特定します。

構文

```
<INT_RES>=STRLEN("<STRING>")
```


意味

STRLEN:	STRLEN 命令は、指定した文字列の長さを特定します。 文字列の先頭から数えた、0 文字でない文字数が返されます。	
<文字列>:	文字列の長さを特定します。	
	タイプ:	STRING
<INT_RES>:	特定結果の変数	
	タイプ:	INT

例

この機能を、単一文字へのアクセスと組み合わせて使用すると、文字列の終了を特定できます。

プログラムコード

```
IF (STRLEN(BLOCK_NAME)>10) GOTO ERROR
```

4.1.4.6 文字列中の文字/文字列の検索(INDEX、RINDEX、MINDEX、MATCH)

この機能は、単一文字または文字列を文字列で検索します。この機能の結果は、検索した文字列中の文字/文字列の位置を指定します。

構文

INT_RES=INDEX (STRING, CHAR) ; 結果のタイプ:INT

INT_RES=RINDEX (STRING, CHAR) ; 結果のタイプ:INT

INT_RES=MINDEX (STRING, STRING) ; 結果のタイプ:INT

INT_RES=MINDEX (STRING, STRING) ; 結果のタイプ:INT

意味

検索機能:検索が正常に終了した場合、文字列の位置(1 番目のパラメータ)を返します。文字/文字列が見つからない場合は、値-1 が返されます。最初の文字の位置は 0 です。

4.1 フレキシブルなNC プログラミング

意味

INDEX:	2 番目のパラメータとして指定した文字を、(先頭から)1 番目のパラメータで検索します。
RINDEX:	2 番目のパラメータとして指定した文字を、(末尾から)1 番目のパラメータで検索します。
MINDEX:	INDEX 機能に対応します。ただし、最初に見つかった文字のインデックスを返す文字のリストが(文字列として)渡された場合を除きます。
MATCH:	文字列中で文字列を検索します。

これを使用すると、ブランクまたはパス区切り文字(「|」)の位置など、特定の条件に従って文字列が分割されます。

例

パス名称とブロック名称へ入力文字列を分割

プログラムコード	コメント
DEF INT PFADIDX, PROGIDX DEF STRING[26] INPUT DEF INT LISTIDX INPUT = "/_N_MPF_DIR/_N_EXECUTE_MPF" LISTIDX = MINDEX (INPUT, "M,N,O,P") + 1 PFADIDX = INDEX (INPUT, "/") +1 PROGIDX = RINDEX (INPUT, "/") +1 VARIABLE = SUBSTR (INPUT, PFADIDX, PROGIDX-PFADIDX-1) VARIABLE = SUBSTR (INPUT, PROGIDX)	 ; LISTIDX で返される値は 3 です。これは、選択リストのパラメータ INPUT の「N」が先頭から開始して最初の文字のためです。 ; この結果、以下が適用されます。PFADIDX = 1 ; この結果、以下が適用されます。PROGIDX = 12 ; 次の章で説明する SUBSTR 機能を使用すると、変数 INPUT を「パス」と「モジュール」成分に分割できます。 ; この時に、"_N_MPF_DIR"を返します。 ; この時に、"_N_EXECUTE_MPF"を返します。

4.1.4.7 抽出文字列の選択(SUBSTR)

SUBSTR 命令で文字列内の任意の部分を読み取ることができます。

構文

```
<STRING_RES>=SUBSTR(<文字列>,<インデックス>,<長さ>)  
  
<STRING_RES>=SUBSTR(<文字列>,<インデックス>)
```

意味

SUBSTR:	この命令は、<文字列>の<インデックス>位置から指定した<長さ>の抽出文字列を返します。 パラメータ<長さ>を指定しない場合は、<インデックス>位置から文字列の末尾までの抽出文字列が返されます。
<インデックス>:	文字列内の抽出文字列の開始位置開始位置が文字列の末尾の後の場合は、空の文字列("")が返されます。文字列の最初の文字:インデックス = 0 値の範囲:0 ... (文字列長 - 1)
<長さ>:	抽出文字列の長さ。指定した長さが長すぎる場合は、文字列の末尾までの抽出文字列が返されます。 値の範囲:1 ... (文字列長 - 1)

例

プログラムコード	コメント
DEF STRING[29] RES	
;	1
;	0123456789012345678
RES = SUBSTR("QUITTTUNG:10 to 99", 10, 2)	; RES == "10"
RES = SUBSTR("QUITTTUNG:10 to 99", 10)	; RES == "10 to 99"

4.1.4.8 個々の文字の読み取りと書き込み

文字列内の個々の文字を読み取りと書き込みできます。

次の必要条件を遵守してください。

- ユーザー定義変数でのみ可能で、システム変数ではおこなえません。
- 文字列の個々の文字は、サブプログラム呼び出しで「値渡し」のみを転送します。

構文

<文字>=<文字列>[<インデックス>]
 <文字>=<文字列配列>[<配列インデックス>,<インデックス>]
 <文字列>[<インデックス>]=<文字>
 <文字列配列>[<配列インデックス>,<インデックス>]=<文字>

4.1 フレキシブルなNCプログラミング

意味

<文字列>:	任意の文字列
<文字>:	CHAR タイプの変数
<インデックス>:	文字列内の文字の位置 文字列の最初の文字:インデックス = 0 値の範囲:0 ... (文字列長 - 1)

例

例 1: 変数メッセージ

プログラムコード	コメント
<pre>; 0123456789 DEF STRING [50] MESSAGE = "Axis n has reached position" MESSAGE [6] = "X" MSG (MESSAGE)</pre>	<pre>; "軸 x は位置に到達しました"</pre>

例 2: システム変数の評価

プログラムコード	コメント
<pre>DEF STRING [50] STRG ... STRG = \$P_MMCA IF STRG[0] == "E" GOTO ...</pre>	<pre>; システム変数用バッファ ; システム変数のロード ; システム変数の評価</pre>

例 3: パラメータ転送「値渡し」と「参照渡し」

プログラムコード	コメント
<pre>; 0123456 DEF STRING[50] STRG = "Axis X" DEF CHAR CHR ... EXTERN UP_VAL (ACHSE) EXTERN UP_REF (VAR ACHSE) ... UP_VAL (STRG[6]) ... CHR = STRG[6] UP_REF (CHR)</pre>	<pre>; 「値渡し」 パラメータを含むサブプログラムの定義 ; 「参照渡し」 パラメータを含むサブプログラムの定義 ; パラメータ転送「値渡し」 ; バッファ ; パラメータ転送「参照渡し」</pre>

4.1.4.9 文字列のフォーマット(SPRINT)

定義済みの SPRINT 機能を使用して、文字列をフォーマットし、たとえば外部機器に出力するための準備を整えることができます(「Process DataShare - 外部機器/ファイルへの出力 (EXTOPEN、WRITE、EXTCLOSE) (ページ 1163)」も参照してください)。

構文

"<結果文字列>"=SPRINT("<フォーマット文字列>",<値_1>,<値_2>,...,<値_n>)

意味

SPRINT:	STRING タイプの値を提供する予約機能のための識別子。
"<フォーマット文字列>":	固定要素と可変要素を含む文字列。可変要素は、フォーマット制御文字%とそれに続くフォーマット記述を使用して定義します。
<値_1>,<値_2>,...,<値_n>:	定数形式または NC 変数形式の値。<フォーマット文字列>のフォーマット記述に対応して、n 番目のフォーマット制御文字%の位置する場所に挿入されます。
"<結果文字列>":	フォーマットされた文字列(最大 400 バイト)

4.1 フレキシブルなNCプログラミング

使用可能なフォーマット記述

%B:	変換する値が以下の場合に、「TRUE」文字列に変換します。 • 0 と等しくない。 • 空の文字列でない(文字列値の場合)。 変換する値が以下の場合に、「FALSE」文字列に変換します。 • 0 と等しい。 • 空の文字列。 例: <pre>N10 DEF BOOL BOOL_VAR=1 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF BOOL_VAR:%B", BOOL_VAR)</pre> 結果文字列「CONTENT OF BOOL_VAR:TRUE」が文字列変数 RESULT に書き込まれます。
%C:	ASCII 文字に変換します。 例: <pre>N10 DEF CHAR CHAR_VAR="X" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF CHAR_VAR:%C", CHAR_VAR)</pre> 結果文字列「CONTENT OF CHAR_VAR:X」が文字列変数 RESULT に書き込まれます。
%D:	整数値の文字列(INTEGER)に変換します。 例: <pre>N10 DEF INT INT_VAR=123 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF INT_VAR:%D", INT_VAR)</pre> 結果文字列「CONTENT OF INT_VAR:123」が文字列変数 RESULT に書き込まれます。
%<m>D:	整数値の文字列(INTEGER)に変換します。文字列は最低、<m>文字長となります。文字が欠落している場所には、スペースが左揃えで埋められます。 例: <pre>N10 DEF INT INT_VAR=-123 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF INT_VAR:%6D", INT_VAR)</pre> 結果文字列「CONTENT OF INT_VAR:xx-123」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。

4.1 フレキシブルなNC プログラミング

%F:	小数点以下 6 桁の 10 進数の文字列に変換します。必要に応じて、小数位が四捨五入されるか 0 で埋められます。 例: <pre>N10 DEF REAL REAL_VAR=-1.2341234EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%F",REAL_VAR)</pre> 結果文字列変数 RESULT に文字列「CONTENT OF REAL_VAR:-1234.123400」が書き込まれます。
%<m>F:	小数点以下 6 桁で全体の長さが最低<m>文字の 10 進数の文字列に変換します。必要に応じて、小数位が四捨五入されるか 0 で埋められます。文字が欠落している場所には、全体の長さ<m>まで左詰でスペースが埋められます。 例: <pre>N10 DEF REAL REAL_VAR=-1.23412345678EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15F",REAL_VAR)</pre> 結果文字列変数 RESULT に文字列「CONTENT OF REAL_VAR:xxx-1234.123457」が書き込まれます(「x」はスペース用のプレースホルダーです)。
%.<n>F:	小数点以下<n>桁の 10 進数の文字列に変換します。必要に応じて、小数位が四捨五入されるか 0 で埋められます。 例: <pre>N10 DEF REAL REAL_VAR=-1.2345678EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3F",REAL_VAR)</pre> 結果文字列「CONTENT OF REAL_VAR:-1234.568」が文字列変数 RESULT に書き込まれます。
%<m>.<n>F:	小数点以下<n>桁で全体の長さが最低<m>文字の 10 進数の文字列に変換します。必要に応じて、小数位が四捨五入されるか 0 で埋められます。文字が欠落している場所には、全体の長さ<m>まで左詰でスペースが埋められます。 例: <pre>N10 DEF REAL REAL_VAR=-1.2341234567890EX+03 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%10.2F",REAL_VAR)</pre> 結果文字列「CONTENT OF REAL_VAR:xx-1234.12」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。

4.1 フレキシブルな NC プログラミング

%E:	指数表記で表した 10 進数の文字列に変換します。仮数は保存され、小数点前の 1 桁と小数点以下 6 桁で正規化されます。必要に応じて、小数位が四捨五入されるか 0 で埋められます。指数はキーワード「EX」で始まります。その後に符号(「+」または「-」)と 2 桁または 3 桁の数字が続きます。 例: <pre>N10 DEF REAL REAL_VAR=-1234.567890 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%E",REAL_VAR)</pre> 結果文字列「CONTENT OF REAL_VAR:-1.234568EX+03」が文字列変数 RESULT に書き込まれます。
%<m>E:	指数表記で表した、全体の長さが最低<m>文字の 10 進数の文字列に変換します。文字が欠落している場所には、スペースが左揃えで埋められます。仮数は保存され、小数点前の 1 桁と小数点以下 6 桁で正規化されます。必要に応じて、小数位が四捨五入されるか 0 で埋められます。指数はキーワード「EX」で始まります。その後に符号(「+」または「-」)と 2 桁または 3 桁の数字が続きます。 例: <pre>N10 DEF REAL REAL_VAR=-1234.5 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%20E",REAL_VAR)</pre> 結果文字列「CONTENT OF REAL_VAR:xxxxxx-1.234500EX+03」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。
%.<n>E:	指数表記で表した 10 進数の文字列に変換します。仮数は保存され、小数点前の 1 桁と小数点以下<n>桁で正規化されます。必要に応じて、小数位が四捨五入されるか 0 で埋められます。指数はキーワード「EX」で始まります。その後に符号(「+」または「-」)と 2 桁または 3 桁の数字が続きます。 例: <pre>N10 DEF REAL REAL_VAR=-1234.5678 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.2E",REAL_VAR)</pre> 結果文字列「CONTENT OF REAL_VAR:-1.23EX+03」が文字列変数 RESULT に書き込まれます。

%<m>.<n>E:	指数表記で表した、全体の長さが最低<m>文字の 10 進数の文字列に変換します。文字が欠落している場所には、スペースが左揃えで埋められます。仮数は保存され、小数点前の 1 桁と小数点以下<n>桁で正規化されます。必要に応じて、小数位が四捨五入されるか 0 で埋められます。指数はキーワード「EX」で始まります。その後に符号(「+」または「-」)と 2 桁または 3 桁の数字が続きます。 例: <pre>N10 DEF REAL REAL_VAR=-1234.5678 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.2E", REAL_VAR)</pre> 結果文字列「CONTENT OF REAL_VAR:xx-1.23EX+03」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。
%G:	10 進数表記または指数表記で表した 10 進数 - 数値の範囲によって異なる - の文字列に変換します。表記する絶対値が 1.0EX-04 未満であるか 1.0EX+06 より大きい等しい場合は、指数表記が選択されます。それ以外の場合は、10 進数表記となります。最大で上位 6 桁が表示されるか、必要に応じて四捨五入されます。 10 進数表記の場合の例: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX-04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G", REAL_VAR)</pre> 結果文字列「CONTENT OF REAL_VAR:0.000123457」が文字列変数 RESULT に書き込まれます。 指数表記の場合の例: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX+06 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%G", REAL_VAR)</pre> 結果文字列「CONTENT OF REAL_VAR:1.23457EX+06」が文字列変数 RESULT に書き込まれます。

4.1 フレキシブルな NC プログラミング

%<m>G:	10 進数表記または指数表記で表した 10 進数 - 数値の範囲によって異なる - の文字列への変換(%G と同様)。文字列の全体の長さは最低、<m>文字となります。文字が欠落している場所には、スペースが左揃えで埋められます。 10 進数表記の場合の例: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX-04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)</pre> 結果文字列「CONTENT OF REAL_VAR:xxxx0.000123457」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。 指数表記の場合の例: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX+06 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%15G",REAL_VAR)</pre> 結果文字列「CONTENT OF REAL_VAR:xxx1.23457EX+06」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。
%.<n>G:	10 進数表記または指数表記で表した 10 進数 - 数値の範囲によって異なる - の文字列に変換します。最大で上位<n>桁が表示されるか、必要に応じて四捨五入されます。表記する絶対値が 1.0EX-04 未満であるか 1.0EX(+<n>)より大きい/等しい場合は、指数表記が選択されます。それ以外の場合は、10 進数表記となります。 10 進数表記の場合の例: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX-04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)</pre> 結果文字列「CONTENT OF REAL_VAR:0.000123」が文字列変数 RESULT に書き込まれます。 指数表記の場合の例: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX+03 N20 DEF STRING[80] RESULT N30 RESULT = SPRINT("CONTENT OF REAL_VAR:%.3G",REAL_VAR)</pre> 結果文字列「CONTENT OF REAL_VAR:1.23EX+03」が文字列変数 RESULT に書き込まれます。

%<m>.<n>G:	10 進数表記または指数表記で表した 10 進数 - 数値の範囲によって異なる - の文字列への変換(%.<n>G と同様)。文字列の全体の長さは最低、<m>文字となります。文字が欠落している場所には、スペースが左揃えで埋められます。 10 進数表記の場合の例: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX-04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)</pre> 結果文字列「CONTENT OF REAL_VAR:xxx0.0001235」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。 指数表記の場合の例: <pre>N10 DEF REAL REAL_VAR=1.234567890123456EX+04 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF REAL_VAR:%12.4G",REAL_VAR)</pre> 結果文字列「CONTENT OF REAL_VAR:xx1.235EX+06」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。
------------	---

4.1 フレキシブルな NC プログラミング

%.<n>P:	小数点以下<n>桁を考慮して、REAL 値を INTEGER 値に変換。INTEGER 値は 32 ビットのバイナリ値として出力されます。変換する値が 32 ビットで表せない場合、処理はアラームで中断されます。 フォーマット命令%.<n>P を使用して生成されるバイトシーケンスにもバイナリゼロを入れることができるため、この方法で生成された文字列全体が NC データタイプ STRING の規約に対応しなくなります。その結果、STRING タイプの変数に保存することも、NC 言語の文字列命令を使用してさらに処理することもできません。唯一可能なのは、パラメータを該当する外部機器の出力で WRITE 命令に転送するのに使用することです(以下の例を参照してください)。 <フォーマット文字列>にフォーマット記述、タイプ%P が入ると同時に、%.<n>P で生成された 2 進数を除く文字列全体が MD10750 \$MN_SPRINT_FORMAT_P_CODE に対応して、ASCII 文字コード、ISO (DIN6024)、または EIA (RS244)で出力されます。変換できない文字がプログラム指令されている場合、処理はアラームで中断されます。 例: <pre> N10 DEF REAL REAL_VAR=123.45 N20 DEF INT ERROR N30 DEF STRING[20] EXT_DEVICE="/ext/dev/1" ... N100 EXTOPEN(ERROR,EXT_DEVICE) N110 IF ERROR <> 0 ... ; エラーの処理 N200 WRITE(ERROR,EXT_DEVICE,SPRINT("INTEGER BINARY CODED:%.3P",REAL_VAR) N210 IF ERROR <> 0 ... ; エラーの処理 </pre> 結果文字列「INTEGER BINARY CODED:'H0001E23A」が出力機器/ext/dev/1 に転送されます。16 進数値 0x0001E23A は、10 進数値 123450 に対応しています。
---------	---

%<m>.<n>P:	マシンデータ MD10751 \$MN_SPRINT_FORMAT_P_DECIMAL の設定に対応する REAL 値を以下の文字列に変換します。 • <m> + <n>桁の整数、または • 小数点前最大<m>桁および正確な小数点以下<n>桁の 10 進数 フォーマット記述%.<n>P と同様に、文字列全体が MD10750 \$MN_SPRINT_FORMAT_P_CODE で定義された文字コードで保存されます。 MD10751 = 0 の場合の変換: REAL 値が<m> + <n>桁の整数の文字列に変換されます。必要に応じて、小数位が<n>桁に四捨五入されるか 0 で埋められます。小数点より前の桁の欠落している場所は、スペースで埋められます。-符号が左詰めで付けられます。+符号の代わりにスペースが入力されます。 例: <pre>N10 DEF REAL REAL_VAR=-123.45 N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P",REAL_VAR)</pre> 結果文字列「PUNCHED TAPE FORMAT:-xx123450」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。 MD10751 = 1 の場合の変換: REAL 値が、小数点前最大<m>桁および正確な小数点以下<n>桁の 10 進数の文字列に変換されます。必要に応じて、小数点の前の桁が切り捨てられ、小数位が四捨五入されるか 0 で埋められます。<n>が 0 の場合、小数点も省略されます。 例: <pre>N10 DEF REAL REAL_VAR1=-123.45 N20 DEF REAL REAL_VAR2=123.45 N30 DEF STRING[80] RESULT N40 RESULT=SPRINT("PUNCHED TAPE FORMAT:%5.3P VAR2:%2.0P", REAL_VAR1,REAL_VAR2)</pre> 結果文字列「PUNCHED TAPE FORMAT:-123.450 VAR2:23」が文字列変数 RESULT に書き込まれます。
%S:	文字列を挿入。 例: <pre>N10 DEF STRING[16] STRING_VAR="ABCDEFGH" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%S",STRING_VAR)</pre> 結果文字列「CONTENT OF STRING_VAR:ABCDEFGH」が文字列変数 RESULT に書き込まれます。

4.1 フレキシブルな NC プログラミング

%<m>S:	最低<m>文字長の文字列を挿入。欠落している桁はスペースで埋められます。 例: <pre>N10 DEF STRING[16] STRING_VAR="ABCDEFGH" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10S",STRING_VAR)</pre> 結果文字列「CONTENT OF STRING_VAR:xxxABCDEFGH」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。
%.<n>S:	文字列の<n>個の文字(最初の文字から始まる)を挿入。 例: <pre>N10 DEF STRING[16] STRING_VAR="ABCDEFGH" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%.3S",STRING_VAR)</pre> 結果文字列「CONTENT OF STRING_VAR:ABC」が文字列変数 RESULT に書き込まれます。
%<m>.<n>S:	文字列の<n>個の文字(最初の文字から始まる)を挿入。生成された文字列の全体長は最低、<m>文字となります。欠落している桁はスペースで埋められます。 例: <pre>N10 DEF STRING[16] STRING_VAR="ABCDEFGH" N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("CONTENT OF STRING_VAR:%10.5S", STRING_VAR)</pre> 結果文字列「CONTENT OF STRING_VAR:xxxxxABCDE」が文字列変数 RESULT に書き込まれます(例中の「x」はスペースを表します)。
%X:	INTEGER 値を 16 進数表記の文字列に変換。 例: <pre>N10 DEF INT INT_VAR='HA5B8' N20 DEF STRING[80] RESULT N30 RESULT=SPRINT("INTEGER HEXADECIMAL:%X",INT_VAR)</pre> 結果文字列「INTEGER HEXADECIMAL:A5B8」が文字列変数 RESULT に書き込まれます。

注記

識別子とキーワードの大文字と小文字を区別しないという NC 言語の特性が、フォーマット記述にも適用されます。その結果、大文字と小文字のどちらを使用してプログラム指令しても、機能的な違いはありません。

組み合わせのオプション

下の表に、どの NC データタイプをどのフォーマット記述と組み合わせられるかについての情報を記載します。自動的なデータタイプ変換に関する規則が適用されます(「データタイプ (ページ 474)」を参照してください)。

	NC データタイプ						
	BOOL	CHAR	INT	REAL	STRING	AXIS	FRAME
%B	+	+	+	+	+	-	-
%C	-	+	-	-	+	-	-
%D	+	+	+	+	-	-	-
%F	-	-	+	+	-	-	-
%E	-	-	+	+	-	-	-
%G	-	-	+	+	-	-	-
%S	-	+	-	-	+	-	-
%X	+	+	+	-	-	-	-
%P	-	-	+	+	-	-	-

注記

表では、NC データタイプ **AXIS** および **FRAME** を **SPRINT** 機能で直接使用できないことが示されています。ただし、以下は可能です。

- **AXSTRING** 機能を使用して **AXIS** データタイプを文字列に変換 – その後、**SPRINT** で処理することができます。
- フレーム成分アクセス毎に、**FRAME** データタイプの個々の値を読み取り。その結果、**REAL** データタイプを入手して **SPRINT** で処理することができます。

4.1.5 プログラムのジャンプと分岐

4.1.5.1 プログラムの先頭への復帰ジャンプ(GOTOS)

GOTOS 命令を使用すると、プログラムを繰り返すために、メインプログラムまたはサブプログラムの先頭にジャンプして戻ることができます。

4.1 フレキシブルな NC プログラミング

マシンデータを使用して、プログラムの先頭への全ての復帰ジャンプに以下を設定できます。

- プログラムの実行時間を「0」に設定します。
- ワーク数のカウントが値「1」ずつ増加します。

構文

GOTOS

意味

GOTOS:	プログラムの先頭にジャンプするジャンプ命令。	
	実行は、次の NC/PLC インタフェース信号によって制御されます。 DB21 の DBX384.0(プログラム分岐の制御)	
	値:	意味
	0	プログラムの先頭へ復帰ジャンプしません。プログラムの実行は、GOTOS の後の次のパートプログラムブロックで再開します。
	1	プログラムの先頭へ復帰ジャンプします。パートプログラムは繰り返されます。

制約事項

- GOTOS は内部で STOPRE (先読み停止)を開始します。
- データ定義(ローカルユーザー変数(LUD))および GOTOS を含むパートプログラムでは、定義区間後の最初のプログラムブロックにジャンプします。データ定義は再実行されません。定義した変数は、GOTOS ブロックで達した値に残り、定義区間でプログラムしたデフォルト値にリセットされません。

4.1 フレキシブルなNC プログラミング

- ジャンプ命令があるプログラムが、「外部メモリからの実行」機能によって外部プログラムメモリから実行される場合は、ジャンプ先を再ロードメモリ内で特定してください。

この条件は、プログラムの先頭(GOTOS)へのジャンプ命令に関して特に問題となります。プログラムは一般的に大きすぎて、全体が再ロードメモリに適合しないためです。初めて再ロードすると、プログラムの先頭は再ロードメモリから削除されます。プログラムの先頭へのジャンプ命令を実行すると、この機能はジャンプ先を見つけられなくなります。プログラムは中止され、アラーム 14000 が出力されます。

注記

プログラムしたジャンプ命令に関して、制限なく外部プログラムを実行できるようにするには、機能「外部メモリからの実行」の代わりにオプション「外部メモリからの実行(EES)」の使用を推奨します。

- GOTOS 命令は、シンクロナイズドアクションとテクノロジーサイクルでは使用できません。

例

プログラムコード	コメント
N10 ...	; プログラムの開始
...	
N90 GOTOS	; プログラムの先頭へジャンプします
...	

4.1.5.2 ジャンプマークへのプログラムのジャンプ(GOTOB、GOTOF、GOTO、GOTOC)

ジャンプラベルをプログラムに設定し、同じプログラム内の別の場所からそこに、GOTOF、GOTOB、GOTO、または GOTOC 命令を使用してジャンプできます。プログラムの実行はジャンプラベルの直後にある命令で再開されます。これは、プログラム内で分岐を実行できることを意味します。

ジャンプラベルの他に、メインブロック番号とサブブロック番号をジャンプ先にすることができます。

ジャンプ命令の前にジャンプ条件(IF ...)の式を指定した場合は、その条件を満たす場合にのみ、プログラムジャンプが実行されます。

構文

```
GOTOB <ジャンプ先>
IF <ジャンプ条件> == TRUE GOTOB <ジャンプ先>
```

4.1 フレキシブルなNCプログラミング

```

GOTOF <ジャンプ先>
IF <ジャンプ条件> == TRUE GOTOF <ジャンプ先>

GOTO <ジャンプ先>
IF <ジャンプ条件> == TRUE GOTO <ジャンプ先>

GOTOC <ジャンプ先>
IF <ジャンプ条件> == TRUE GOTOC <ジャンプ先>

```

意味

GOTOB:	プログラムの先頭方向へジャンプするジャンプ命令。	
GOTOF:	プログラムの末尾方向へジャンプするジャンプ命令。	
GOTO:	ジャンプ先を検索するジャンプ命令。検索は、最初にプログラムの末尾方向へ、次にプログラムの先頭方向へ実行されます。	
GOTOC:	GOTO と同じ結果となりますが、この場合は、アラーム 14080 「Jump destination not found」 (ジャンプ先が見つかりません) がマスクされます。 つまり、ジャンプ先の検索が正常に終了しなかった場合でも、プログラムの実行が中断されず、GOTOC 命令の後のプログラム行が続行されます。	
<ジャンプ先>:	ジャンプ先パラメータ 指定できるデータには以下のものがあります。	
	<ジャンプラベル>:	ジャンプ先は、次のように、ユーザーが定義した名称でプログラムに設定されたジャンプラベルです。<ジャンプラベル>:
	<ブロック番号>:	ジャンプ先は、メインブロック番号またはサブブロック番号です(例:200, N300)。
	STRING タイプ変数:	ジャンプ先となる変数です。変数は、ジャンプラベルまたはブロック番号を表わします。
IF:	条件ジャンプの式に使用するキーワード。 条件ジャンプには、すべての比較演算子と論理演算子を使用できます(結果:TRUE または FALSE)。プログラムジャンプは、この演算の結果が TRUE である場合に実行されます。	

注記**ジャンプラベル**

ジャンプラベルは常に、ブロックの先頭に配置します。プログラム番号がある場合は、ブロック番号の直後にジャンプラベルを配置します。

ジャンプラベルの名称を付けるときは、次の規則が適用されます。

- 文字数:
 - 最小 2
 - 最大 32
 - 次の文字を使用できます。
 - 英字
 - 数字
 - アンダースコア
 - 最初の 2 文字は、英字またはアンダースコアにしてください。
 - ジャンプラベルの名称の後にはコロン(:)を付けます。
-

制約事項

- ジャンプ先となるのは、プログラムの中にある、ジャンプラベルまたはブロック番号が付いているブロックのみです。
 - ジャンプ条件のないジャンプ命令は、個別のブロックにプログラム指令してください。この制限は、ジャンプ条件のあるジャンプ命令には適用されません。この場合は、複数のジャンプ命令を同じブロックの式に使用できます。
 - ジャンプ条件のないジャンプ命令を含むプログラムの場合、プログラムの終了 M2/M30 は必ずしも、プログラムの末尾にある必要はありません。
 - ジャンプ命令があるプログラムが、「外部メモリからの実行」機能によって外部プログラムメモリから実行される場合は、ジャンプ先を再ロードメモリ内で特定してください。その他の場合、ジャンプ先は見つからず、プログラムは中止されてアラーム 14000 が出力されます。
-

注記

プログラムしたジャンプ命令に関して、制限なく外部プログラムを実行できるようにするには、機能「外部メモリからの実行」の代わりにオプション「外部メモリからの実行(EES)」の使用を推奨します。

4.1 フレキシブルなNCプログラミング

例

例 1:ジャンプラベルへのジャンプ

プログラムコード	コメント
N10 ...	
N20 GOTOF Label_1	; プログラムの末尾方向の、 ; ジャンプラベル「Label_1」にジャンプします。
N30 ...	
N40 Label_0:R1=R2+R3	; ジャンプラベル「Label_0」が設定されています。
N50 ...	
N60 Label_1:	; ジャンプラベル「Label_1」が設定されています。
N70 ...	
N80 GOTOB Label_0	; プログラム先頭方向の、 ; ジャンプラベル「Label_0」にジャンプします。
N90 ...	

例 2:ブロック番号へ間接ジャンプ

プログラムコード	コメント
IF <条件> == TRUE	
R10=100	; ジャンプ先を割り当てます
ELSE	
R10=110	; ジャンプ先を割り当てます
ENDIF	
; プログラム末尾方向の、ブロック番号が R10 にあるブロックへジャンプします	
N10 GOTOF "N"<<R10	
...	
N90 ...	
N100 ...	; ジャンプ先
N110 ...	
...	

例 3:ジャンプ先変数へジャンプ

プログラムコード	コメント
DEF STRING[20] DESTINATION	
IF <条件> == TRUE	
DESTINATION = "Label1"	; ジャンプ先を割り当てます
ELSE	
DESTINATION = "Label2"	; ジャンプ先を割り当てます
ENDIF	
; プログラムの末尾方向の、ジャンプ先となる変数「DESTINATION の内容」にジャンプします。	
GOTOF DESTINATION	
Label1:T="Drill1"	; ジャンプ先 1
...	

4.1 フレキシブルなNC プログラミング

プログラムコード	コメント
Label2:T="Drill2"	; ジャンプ先 2
...	

例 4:ジャンプ条件によるジャンプ

プログラムコード	コメント
N40 R1=30 R2=60 R3=10 R4=11 R5=50 R6=20	; 初期値を割り当てます。
N41 LA1:G0 X=R2*COS(R1)+R5 Y=R2*SIN(R1)+R6	; ジャンプラベル LA1
N42 R1=R1+R3 R4=R4-1	
; IF ジャンプ条件 == TRUE	
; THEN プログラム先頭方向のジャンプラベル LA1 にジャンプします	
N43 IF R4>0 GOTOB LA1	
N44 M30	; プログラム終了

4.1.5.3 プログラム分岐(CASE ... OF ... DEFAULT ...)

CASE 機能を使用すると、変数または算術機能の現在値(タイプ:INT)を確認して、結果によっては、プログラムの別の位置へジャンプできます。

構文

```
CASE(<式>) OF <定数_1> GOTOF <ジャンプ先_1> <定数_2> GOTOF <ジャンプ先_2> ... DEFAULT GOTOF <ジャンプ先_n>
```

意味

CASE:	ジャンプ命令
<式>:	変数または算術機能です。
OF:	条件付きプログラム分岐の式を作成するためのキーワードです。
<定数_1>:	変数または算術機能に対して、1 番目に指定する定数値です。
	タイプ: INT
<定数_2>:	変数または算術機能に対して、2 番目に指定する定数値です。
	タイプ: INT
DEFAULT:	変数または算術機能が、指定した定数値を使用しない場合は、DEFAULT 命令を使用してジャンプ先を特定できます。 注: DEFAULT 命令をプログラム指令しない場合は、CASE 命令の次のブロックがジャンプ先になります。

4.1 フレキシブルな NC プログラミング

GOTOF:	プログラムの末尾方向へジャンプするジャンプ命令。 GOTOF の代わりに、すべての他の GOTO 命令をプログラム指令できます(「ジャンプマークへのプログラムジャンプ」を参照してください)。	
<ジャンプ先_1>:	変数または算術機能の値が、指定した 1 番目の定数に対応する場合は、分岐はこのジャンプ先となります。 ジャンプ先は、次のように指定できます。	
	<ジャンプラベル>:	ジャンプ先は、次のように、ユーザーが定義した名称でプログラムに設定されたジャンプラベルです。<ジャンプラベル>:
	<ブロック番号>:	ジャンプ先は、メインブロック番号またはサブブロック番号です(例:200, N300)。
	STRING タイプ変数:	ジャンプ先となる変数です。変数は、ジャンプラベルまたはブロック番号を表わします。
<ジャンプ先_2>:	変数または算術機能の値が、指定した 2 番目の定数に対応する場合は、分岐はこのジャンプ先となります。	
<ジャンプ先_n>:	変数の値が、指定した定数値でない場合は、分岐はこのジャンプ先となります。	

制約事項

外部メモリからのプロセス

CASE 機能があるプログラムが、「外部メモリからの実行」機能によって外部プログラムメモリから実行される場合は、ジャンプ先を再ロードメモリ内で特定してください。その他の場合、ジャンプ先は見つからず、プログラムは中止されてアラーム 14000 が出力されます。

注記

プログラムしたジャンプ命令に関して、制限なく外部プログラムを実行できるようにするには、機能「外部メモリからの実行」の代わりにオプション「外部メモリからの実行(EES)」の使用を推奨します。

例

プログラムコード
...

プログラムコード

```
N20 DEF INT VAR1 VAR2 VAR3
N30 CASE (VAR1+VAR2-VAR3) OF 7 GOTOF Label_1 9 GOTOF
Label_2 DEFAULT GOTOF Label_3
N40 Label_1:G0 X1 Y1
N50 Label_2:G0 X2 Y2
N60 Label_3:G0 X3 Y3
...
```

N30 の CASE 命令は、次のプログラム分岐の方法を定義します。

1. 算術機能 $VAR1+VAR2-VAR3$ の値が 7 の場合は、ジャンプマーク定義「Label_1」のあるブロックにジャンプします(→ N40)。
2. 算術機能 $VAR1+VAR2-VAR3$ の値が 9 の場合は、ジャンプマーク定義「Label_2」のあるブロックにジャンプします(→ N50)。
3. 算術機能 $VAR1+VAR2-VAR3$ の値が 7 でも 9 でもない場合は、ジャンプマーク定義「Label_3」のあるブロックにジャンプします(→ N60)。

4.1.6 プログラム区間の繰り返し(REPEAT、REPEATB、ENDLABEL、P)

プログラム区間の繰り返しを使用すると、1 つのプログラム内で既存のプログラム区間を任意の順序で繰り返すことができます。

繰り返すプログラム行またはプログラム区間は、ジャンプマーク(ラベル)が付けられます。

注記**ジャンプマーク(ラベル)**

ジャンプマークは常に、ブロックの先頭に配置します。プログラム番号がある場合は、ブロック番号の直後にジャンプマークを配置します。

ジャンプマークの名称を付けるときは、次の規則が適用されます。

- 文字数:
 - 最小 2
 - 最大 32
- 次の文字を使用できます。
 - 英字
 - 数字
 - アンダースコア
- 最初の 2 文字は、英字またはアンダースコアにしてください。
- ジャンプマークの名称の後にはコロン(「:」)を付けます。

4.1 フレキシブルな NC プログラミング

構文

1. 個々のプログラム行の繰り返し:

```
<ジャンプマーク>: ...  
...  
REPEATB <ジャンプマーク> P=<n>  
...
```

2. ジャンプマークと REPEAT 命令の間のプログラム区間の繰り返し:

```
<ジャンプマーク>: ...  
...  
REPEAT <ジャンプマーク> P=<n>  
...
```

3. 2 個のジャンプマーク間の区間の繰り返し:

```
<開始ジャンプマーク>: ...  
...  
<終了ジャンプマーク>: ...  
...  
REPEAT <開始ジャンプマーク> <終了ジャンプマーク> P=<n>  
...
```

注記

括弧で 2 個のジャンプマークを含む REPEAT 命令を入れ子にすることはできません。<開始ジャンプマーク>が REPEAT 命令の前にあり、<終了ジャンプマーク>に到達する前に REPEAT 命令がある場合は、<開始ジャンプマーク>と REPEAT 命令の間の区間が繰り返されます。

4. ジャンプマークと ENDLABEL の間の区間の繰り返し:

```
<ジャンプマーク>: ...  
...  
ENDLABEL: ...  
...  
REPEAT <ジャンプマーク> P=<n>  
...
```


注記

括弧で<ジャンプマーク>と ENDLABEL を含む REPEAT 命令を入れ子にすることはできません。<ジャンプマーク>が REPEAT 命令の前にあり、ENDLABEL に到達する前に REPEAT 命令がある場合は、<ジャンプマーク>と REPEAT 命令の間の区間が繰り返されます。

意味

REPEATB:	プログラム行を繰り返す命令です。
REPEAT:	プログラム区間を繰り返す命令です。
<ジャンプマーク>:	<ジャンプマーク>は、以下を示します。 - 繰り返すプログラム行(REPEATB の場合) または - 繰り返すプログラム区間の先頭(REPEAT の場合) <ジャンプマーク>で示されたプログラム行は、REPEAT/REPEATB 命令の前にある場合も、後にある場合もあります。検索はまず、プログラムの先頭方向へ開始されます。この方向にジャンプマークが見つからない場合は、検索が、プログラムの末尾方向へ続行されます。 例外: ジャンプマークと REPEAT 命令の間のプログラム区間を繰り返す(「構文」の 2 を参照してください)場合は、<ジャンプマーク>で示されたプログラム行が REPEAT 命令の前にあるように作成してください。これは、この場合には、プログラムの先頭方向のみ検索が実行されるためです。 <ジャンプマーク>を含む行に、さらに命令が含まれる場合は、繰り返し毎にそれらの命令が実行されます。
ENDLABEL:	繰り返すプログラム区間の末尾をマークするキーワードです。 ENDLABEL を含む行に、さらに命令が含まれる場合は、繰り返し毎にそれらの命令が実行されます。 ENDLABEL はプログラムで複数回、使用できます。
P:	繰り返し回数を指定するアドレスです。

4.1 フレキシブルな NC プログラミング

<n>:	プログラム区間の繰り返し回数です。	
	タイプ:	INT
	繰り返すプログラム区間を<n>回、繰り返します。最後の繰り返しの後に、REPEAT/REPEATB 行の次の行からプログラムを再開します。 注: 回数を P=<n>で指定しない場合は、プログラム区間を 1 回だけ繰り返します。	

例

例 1:個々のプログラム行の繰り返し

プログラムコード	コメント
N10 POSITION1:X10 Y20	
N20 POSITION2:CYCLE(0,,9,8)	; サイクルを位置決めます。
N30 ...	
N40 REPEATB POSITION1 P=5	; BLOCK N10 を 5 回実行します。
N50 REPEATB POSITION2	; N20 ブロックを 1 回実行します。
N60 ...	
N70 M30	

例 2 :ジャンプマークと REPEAT 命令の間のプログラム区間の繰り返し:

プログラムコード	コメント
N5 R10=15	
N10 Begin:R10=R10+1	; 幅
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y7=R10	
N50 X=-R10	
N60 Y=-R10	
N70 Z=10+R10	
N80 REPEAT BEGIN P=4	; N10～N70 の区間を 4 回実行します。
N90 Z10	
N100 M30	

例 3 :2 個のジャンプマーク間の区間の繰り返し

プログラムコード	コメント
N5 R10=15	
N10 Begin:R10=R10+1	; 幅

4.1 フレキシブルな NC プログラミング

プログラムコード	コメント
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y7=R10	
N50 X=-R10	
N60 Y=-R10	
N70 END:Z = 10	
N80 Z10	
N90 CYCLE(10,20,30)	
N100 REPEAT BEGIN END P=3	; N10～N70 の区間を 3 回実行します。
N110 Z10	
N120 M30	

例 4 :ジャンプマークと ENDLABEL の間の区間の繰り返し

プログラムコード	コメント
N10 G1 F300 Z-10	
N20 BEGIN1:	
N30 X10	
N40 Y10	
N50 BEGIN2:	
N60 X20	
N70 Y30	
N80 ENDLABEL:Z10	
N90 X0 Y0 Z0	
N100 Z-10	
N110 BEGIN3:X20	
N120 Y30	
N130 REPEAT BEGIN3 P=3	; N110～N120 の区間を 3 回実行します。
N140 REPEAT BEGIN2 P=2	; N50～N80 の区間を 2 回実行します。
N150 M100	
N160 REPEAT BEGIN1 P=2	; N20～N80 の区間を 2 回実行します。
N170 Z10	
N180 X0 Y0	
N190 M30	

例 5 :フライス加工、さまざまなテクノロジーによるドリル位置決め加工

プログラムコード	コメント
N10 CENTER DRILL()	; センタードリルを取り付けます。
N20 POS_1:	; 穴あけ位置 1
N30 X1 Y1	
N40 X2	
N50 Y2	
N60 X3 Y3	
N70 ENDLABEL:	

4.1 フレキシブルなNCプログラミング

プログラムコード	コメント
N80 POS_2:	; 穴あけ位置 2
N90 X10 Y5	
N100 X9 Y-5	
N110 X3 Y3	
N120 ENDLABEL:	
N130 DRILL()	; ドリルと穴あけサイクルを変更します。
N140 THREAD(6)	; タップ M6 とねじ切りサイクルをロードします。
N150 REPEAT POS_1	; POS_1 から ENDLABEL のプログラム区間を 1 回、繰り返します。
N160 DRILL()	; ドリルと穴あけサイクルを変更します。
N170 THREAD(8)	; タップ M8 とねじ切りサイクルをロードします。
N180 REPEAT POS_2	; POS_2 から ENDLABEL のプログラム区間を 1 回、繰り返します。
N190 M30	

詳細

ネスト

プログラム区間繰り返しは、入れ子にすることができます。各呼び出しは、サブプログラムレベルを使用します。

M17 / RET

プログラム区間繰り返しのときに M17 または RET をプログラム指令した場合は、繰り返しがキャンセルされます。プログラムは、REPEAT 行の次のブロックから再開します。

プログラム表示

現在のプログラム表示では、プログラム区間繰り返しが別のサブプログラムレベルとして表示されます。

キャンセルレベル

プログラム区間繰り返しのときにレベルがキャンセルされた場合は、プログラム区間繰り返しを呼び出した後の地点からプログラムを再開します。

例:

プログラムコード	コメント
N5 R10=15	
N10 Begin:R10=R10+1	; 幅
N20 Z=10-R10	
N30 G1 X=R10 F200	
N40 Y=R10	; 繰り返しレベル中断
N50 X=-R10	
N60 Y=-R10	
N70 END:Z10	

4.1 フレキシブルなNC プログラミング

プログラムコード	コメント
N80 Z10	
N90 CYCLE(10,20,30)	
N100 REPEAT BEGIN END P=3	
N120 Z10	; プログラムの実行を再開します。
N130 M30	

制御系およびプログラム区間の繰り返し

チェック命令とプログラム区間繰り返しは、組み合わせて使用できます。ただし、この2つが重複しないようにしてください。プログラム区間繰り返しがチェック命令分岐内にあるか、またはチェック命令がプログラム区間繰り返しの中にあるように作成してください。

ジャンプとプログラム区間の繰り返し

ジャンプとプログラム区間繰り返しが混在している場合は、ブロックは単純に順番どおりに実行されます。たとえば、ジャンプがプログラム区間繰り返し部から実行された場合は、プログラム指令したプログラム区間終了が見つかるまで、処理がおこなわれます。

例:

プログラムコード
N10 G1 F300 Z-10
N20 BEGIN1:
N30 X=10
N40 Y=10
N50 GOTOF BEGIN2
N60 ENDLABEL:
N70 BEGIN2:
N80 X20
N90 Y30
N100 ENDLABEL:Z10
N110 X0 Y0 Z0
N120 Z-10
N130 REPEAT BEGIN1 P=2
N140 Z10
N150 X0 Y0
N160 M30

注記

REPEAT 命令は、移動ブロックの後に指令してください。

外部ソースからの実行とプログラム区間の繰り返し

プログラム区間の繰り返しを含む外部プログラムの場合は、繰り返すプログラム行(REPEATB)または繰り返すプログラム区間の先頭(REPEAT)を再ロードメモリ内に置いてく

4.1 フレキシブルなNC プログラミング

ださい。その他の場合、ジャンプ先は見つからず、プログラムは中止されてアラーム 14000 が出力されます。

注記

プログラム区間の繰り返しに関して、制限なく外部プログラムを実行できるようにするには、機能「外部メモリからの実行」の代わりに機能「外部メモリからの実行(EES)」の使用を推奨します。

4.1.7 チェック命令

コントローラは、標準では、プログラム指令順に処理します。

この順序は、プログラムブロック選択とプログラムループをプログラム指令して、可変にすることができます。これらのチェック命令は、キーワード IF、ELSE、ENDIF、LOOP、FOR、WHILE、REPEAT を使用してプログラム指令します。

通知

プログラム指令エラー

チェック命令は、プログラムの命令区間にのみに挿入できます。プログラムヘッダーの定義は、条件付きで実行することも、繰り返し実行することもできません。

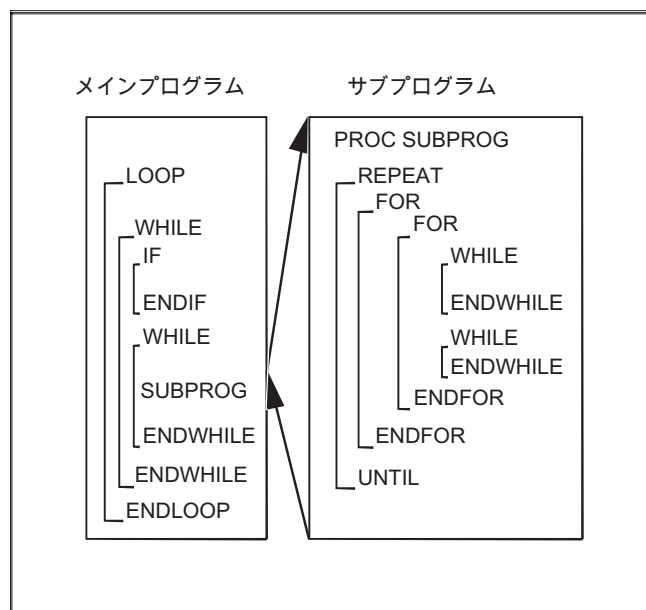
制御系のキーワードにもジャンプ先にも、マクロを重ね合わせることはできません。マクロを定義した場合は、このチェックはおこなわれません。

有効性

チェック命令はプログラム全体で使用することはできません。

ネストレベル

各サブプログラムレベルには、入れ子の深さが 16 層までのチェック命令を設定できます。



動作の実行時間

インタプリタモード(標準で有効)では、プログラム分岐を使用すると、チェック命令より効率よく、プログラムの処理時間を短縮できます。

事前にコンパイルされたサイクルでは、プログラム分岐とチェック命令の間の差はありません。

プログラムループの実行中のブロックの表示

選択されたブロックだけがプログラムループ内で実行されている場合、プログラムループの前の最後のメインランブロックが実行中のブロックの表示に表示されます。

したがって、処理済みの選択されたブロックも実行中のブロックの表示に表示されます。たとえば、診断のために、デコードシングルブロック SBL2 を有効にする必要があります。

メインラインブロックなしでの研削

プログラムループ内で、メインランブロックがプログラムされている場合、ループ条件が満たされるまでループが先読みされます。

その結果、負荷率が高くなり、表示に悪影響が生じる場合があります。

4.1 フレキシブルな NC プログラミング

対策として、STOPRE 命令、または 0 秒のドウェル時間 G04 を、ループに**挿入**できます。

制約事項

ブロック表示

チェック命令要素を含むブロックはマスクできません。

制御系とプログラムジャンプ

ジャンプマーク(ラベル)は、チェック命令要素を含むブロックでは使用できません。

注記

一般的に、チェック命令とプログラム分岐を混ぜて使用することは望ましくありません。

外部メモリからのプロセス

制御系を含む外部プログラムでは、ループの先頭を再ロードメモリ内に置いてください。その他の場合、ジャンプ先は見つからず、プログラムは中止されてアラーム 14000 が出力されます。

注記

制御系に関して、制限なく外部プログラムを実行できるようにするには、機能「外部メモリからの実行」の代わりに機能「外部メモリからの実行(EES)」の使用を推奨します。

インタプリタモード

チェック命令はインタプリタ的に処理されます。ループ終了を検出すると、ループ先頭の検索がおこなわれ、処理のなかに見つけたチェック命令を可能にします。このため、プログラムのブロック構成は、インタプリタモードで完全にはチェックされません。

サイクルの先読み

サイクルの解析時に、チェック命令が正しく入れ子になっていることを確認できます。

4.1.7.1 条件付き命令と分岐(IF、ELSE、ENDIF)

条件付き命令 IF - プログラムブロック - ENDIF

条件付き命令では、条件が満たされたときにだけ、IF と ENDIF の間のプログラムブロックが実行されます。

分岐:IF - プログラムブロック 1 - ELSE - プログラムブロック 2 - ENDIF

分岐では、2 つのプログラムブロックのうちの 1 つが常に実行されます。

条件が満たされた場合、IF と ELSE の間のプログラムブロック 1 が実行されます。

条件が満たされなかった場合、ELSE と ENDIF の間のプログラムブロック 2 が実行されます。

注記

シンクロナイズドアクションの ELSE

キーワード ELSE は、シンクロナイズドアクションにもプログラム指令できます。したがって、条件が満たされなかった場合に実行されるアクションで、シンクロナイズドアクションを拡張できます。

構文

条件付き命令

```
IF <条件>
    プログラムブロック                ; 次で実行: <条件> == TRUE
ENDIF
```

分岐

```
IF <条件>
    プログラムブロック 1                ; 次で実行: <条件> == TRUE
ELSE
    プログラムブロック 2                ; 次で実行: <条件> == FALSE
ENDIF
```

意味

IF:	条件付き命令または分岐を通知します。
ELSE:	プログラムブロック選択を開始します。
ENDIF:	条件付き命令または分岐の終了をマークします。
<条件>:	TRUE または FALSE として評価される論理式です。

例:工具交換サブプログラム

プログラムコード	コメント
PROC L6	工具交換プログラムです。
N500 DEF INT TNR_AKTUELL	動作中の T 番号の変数です。

4.1 フレキシブルなNC プログラミング

プログラムコード	コメント
N510 DEF INT TNR_VORWAHL	事前選択された T 番号用の変数
	現在の工具を特定します。
N520 STOPRE	
N530 IF \$P_ISTEST	プログラムテストモードの場合...
N540 TNR_AKTUELL = \$P_TOOLNO	...「現在の」工具がプログラム指令結果から読み取られます。
N550 ELSE	そうでない場合は...
N560 TNR_AKTUELL = \$TC_MPP6[9998,1]	... 主軸の工具が読み出されます。
N570 ENDIF	
N580 GETSELT(TNR_VORWAHL)	主軸で事前に選択した工具の T 番号を読み取ります。
N590 IF TNR_AKTUELL <> TNR_VORWAHL	事前に選択した工具がまだ、現在の工具ではない場合は、...
N600 G0 G40 G60 G90 SUPA X450 Y300 Z300 D0	...工具交換位置へアプローチして、...
N610 M206	... 工具交換を実行します。
N620 ENDIF	
N630 M17	

4.1.7.2 連続プログラムループ(LOOP、ENDLOOP)

無限ループが無限プログラムで使用されます。ループの終了には常に、先頭へ戻る分岐があります。

構文

```
LOOP
...
ENDLOOP
```

意味

LOOP:	無限ループを開始します。
ENDLOOP:	ループの終了を示し、そして、ループの先頭へ復帰ジャンプします。

例

```
プログラムコード
...
LOOP
MSG ("no tool cutting edge active")
```

プログラムコード

```
M0
Stopre
ENDLOOP
...
```

4.1.7.3 カウンترلープ(FOR ... TO ..., ENDFOR)

カウンترلープは、運転を固定の実行回数だけ繰り返して実行する場合に使用します。

構文

```
FOR <変数> = <開始値> TO <終了値>
...
ENDFOR
```

意味

FOR:	カウンترلープを開始します。	
ENDFOR:	ループの終了を示し、そして、カウントの終了値に達していないかぎり、ループの先頭へ復帰ジャンプします。	
<変数>:	開始値から終了値へと、実行のたびに値「1」ずつ増えていくカウント変数です。	
	タイプ	INT または REAL 注: R 変数などをカウンترلープにプログラム指令している場合は、REAL タイプを使用します。カウント変数が REAL タイプの場合は、その値が整数に丸められます。
<開始値>:	カウントの開始値 条件:この開始値は、終了値より小さい数にしてください。	
<終了値>:	カウントの終了値	

4.1 フレキシブルなNCプログラミング

例

例 1:INTEGER 変数、または R 変数のカウント変数

INTEGER 変数のカウント変数:

プログラムコード	コメント
DEF INT iVARIABLE1	
R10=R12-R20*R1 R11=6	
FOR iVARIABLE1 = R10 TO R11	; カウント変数= INTEGER 変数
R20=R21*R22+R33	
ENDFOR	
M30	

R 変数のカウント変数:

プログラムコード	コメント
R11=6	
FOR R10=R12-R20*R1 TO R11	; カウント変数= R 変数 (実数タイプの変数)
R20=R21*R22+R33	
ENDFOR	
M30	

例 2 :固定数の部品の製造

プログラムコード	コメント
DEF INT WKPCCOUNT	; INT タイプの変数を「WKPCCOUNT」という名称で定義します。
FOR WKPCCOUNT = 0 TO 100	; カウントループを開始します。「WKPCCOUNT」変数は、開始値「0」から終了値「100」まで増えていきます。
G01 ...	
ENDFOR	; カウントループの終了
M30	

4.1.7.4 ループの先頭に条件があるプログラムループ(WHILE、ENDWHILE)

WHILE ループの場合は、ループの先頭に条件があります。WHILE ループは、条件を満たしているかぎり、実行されます。

構文

WHILE <条件>

4.1 フレキシブルな NC プログラミング

```
...  
ENDWHILE
```

意味

WHILE:	プログラムループを開始します。
ENDWHILE:	ループの終了を示し、そして、ループの先頭へ復帰ジャンプします。
<条件>:	WHILE ループが実行されるように、条件を満たしてください。

例

プログラムコード	コメント
... WHILE \$AA_IW[DRILL_AXIS] > -10 G1 G91 F250 AX[DRILL_AXIS] = -1 ENDWHILE ...	； 次の条件下で、WHILE ループを呼び出します。「穴あけ軸 の実際の WCS 指令値が-10 より大きい」

4.1.7.5 ループの終了に条件があるプログラムループ(REPEAT、UNTIL)

REPEAT ループの場合は、ループの終了で条件があります。REPEAT ループを実行したあと、条件が満たされるまで継続的に繰り返されます。

構文

```
REPEAT  
...  
UNTIL <条件>
```

意味

REPEAT:	プログラムループを開始します。
UNTIL:	ループの終了を示し、そして、ループの先頭へ復帰ジャンプします。
<条件>:	REPEAT ループが実行されないように、条件を満たします。

4.1 フレキシブルなNC プログラミング

例

プログラムコード	コメント
...	
REPEAT	; REPEAT ループを呼び出します。
...	
UNTIL ...	; 条件が満たされているかどうかを確認します。
...	

4.1.7.6 入れ子のチェック命令を含むプログラム例

プログラムコード	コメント
LOOP	
IF NOT \$P_SEARCH	; ブロック検索がおこなわれない場合
G1 G90 X0 Z10 F1000	
WHILE \$AA_IM[X] <= 100	; WHILE (指令値 X 軸 <= 100)
G1 G91 X10 F500	; 穴あけパターン
Z-5 F100	
Z5	
ENDWHILE	
ELSE	; ELSE ブロック検索
MSG("No drilling during block search")	
ENDIF	; ENDIF
\$A_OUT[1] = 1	; 次のプレートの穴あけ
G4 F2	
ENDLOOP	
M30	

4.1.8 チャンネル間プログラム協調(INIT、START、WAITM、WAITMC、WAITE、SETM、CLEARM)

原則として、NC のチャンネルは、その中で開始されたプログラムを、そのモードグループ内の他のチャンネルとは無関係に処理できます。ただし、モードグループの複数のチャンネルにある複数のプログラムがワークの加工に関与している場合は、異なるチャンネルのプログラム処理を以下の協調コマンドによって協調させる必要があります。

必要条件

プログラム協調に使用するすべてのチャンネルは、**同じ**モードグループに属します。

MD10010 \$MC_ASSIGN_CHAN_TO_MODE_GROUP[<チャンネル>] = <モードグループ番号>

チャンネル番号ではなくチャンネル名

チャンネル番号の代わりに MD20000 \$MC_CHAN_NAME[<チャンネルインデックス>]に入力されたチャンネル名は、プログラム協調のための予約処理のパラメータとして使用できます。まず NC プログラムでチャンネル名の使用を有効にする必要があります。

MD10280 \$MN_PROG_FUNCTION_MASK, bit 1 = TRUE

注記**命令間の最小距離**

命令 INIT、START、WAITE、WAITM、SETM、CLEARM と命令 WAITMC との間の距離は、最低でも 2 つの移動ブロックとしてください。WAITMC は実行可能ブロックですが、最適化のために前のブロックに移され、ブロックとして削除されます。たとえば、SETM は実行可能ブロックではなく、次のブロックに移されるため、2 つの命令間の距離が 1 ブロックの場合、両方の命令が中間ブロックに入ります。1 つのブロックのみが有効であるため、1 ブロックの距離では WAITMC の最適化が行なわれません。

これにより、プログラムが停止し、処理がしばらく中断されます。

構文

```
INIT(<ChanNo>, <Prog>, <AckMode>)
START(<ChanNo>, <ChanNo>, ...)
WAITM(<MarkNo>, <ChanNo>, <ChanNo>, ...)
WAITE(<ChanNo>, <ChanNo>, ...)
WAITMC(<MarkNo>, <ChanNo>, <ChanNo>, ...)
SETM(<MarkNo>, <MarkNo>, ...)
CLEARM(<MarkNo>, <MarkNo>, ...)
```

意味

INIT():	指定したチャンネルで処理する NC プログラムを選択する予約処理
START():	それぞれのチャンネルで選択したプログラムを開始するための予約機能
WAITM():	指定したチャンネルで待機マークに到達するまで待機する予約処理 指定した待機マークは同じチャンネルの WAITM で設定します。その前のブロックをイグザクトストップで終了します。待機マークは、同期制御が終わると解除されます。 チャンネルごとに最大 10 個のマークを同時に設定できます。
WAITE():	複数の他のチャンネルのプログラムの終了まで待機する予約処理

4.1 フレキシブルな NC プログラミング

WAITMC () : 1)	指定したチャンネルで待機マークに到達するまで待機する予約処理 WAITMとは異なり、イグザクトストップの減速は他のチャンネルが待機マークに到達していない場合のみ開始します。		
SETM () : 1)	チャンネル協調用の複数の待機マークを設定する予約処理 これはそのチャンネル内の処理には影響しません SETMはチャンネルリセットと NC スタート後も有効です。		
CLEARM () : 1)	チャンネル協調用の複数の待機マークを削除する予約処理 これはそのチャンネル内の処理には影響しません CLEARM () はチャンネル内のすべての待機マークを削除します。 CLEARM (0) は待機マーク「0」のみを削除します。 CLEARMはチャンネルリセットと NC スタート後も有効です。		
<ChanNo>:	チャンネル番号 自身のチャンネルの番号を指定する必要はありません。		
	タイプ:	INT	
<Prog>:	絶対または相対パス指定(オプション)+プログラム名称		
	タイプ:	STRING	
	軌跡データについては、『プログラミングマニュアル NC プログラミング』の「プログラムメモリファイルのアドレス指定」(ページ 632)の章を参照してください。		
<AckMode>:	応答モード(オプション)		
	タイプ:	CHAR	
	規格値:	"n"	応答なし 命令の送信後にプログラムの実行を続行します。命令を正常に実行できない場合でも、送信側には通知されません。
		"S"	同期確認応答 プログラムの実行は、受信コンポーネントが命令を応答するまで中止します。確認の応答があった場合は、次の命令を実行します。確認の応答がない場合は、故障メッセージを出力します。

<MarkNo>:	待機マークの番号
	注 マルチチャネルシステムでは、最大 100 個の待機マーク(待機マーク 0~99)を使用できます。 シングルチャネルシステムで利用できるのは、待機マーク 0 のみです。
1) ユーザー別の通信/チャネル協調では、SETM/CLEARM を使用して待機マークを配置できます。また、条件付き待機指令 WAITMC を使用する必要もありません。チャネルのリセットおよび NC スタート後も、待機マークは値を保持します。	

例

MD20000 からチャネル名の使用を開始する

- パラメータの割り付け

```
MD10280 $MN_PROG_FUNCTION_MASK, bit 1 = TRUE
$MC_CHAN_NAME[ 0 ] = "MACHINING"; チャネル 1 の名前
$MC_CHAN_NAME[ 1 ] = "INFEED"; チャネル 2 の名前
```
- プログラミング

プログラムコード	コメント
START (MACHINING)	; チャネル 1 の開始
START (INFEED)	; チャネル 2 の開始

START でローカルの「チャネル名称」と「ユーザー変数」を使用

プログラムコード	コメント
DEF INT MACHINE = 1	; チャネル 1 のユーザー変数の定義
DEF INT LOADER = 2	; チャネル 2 のユーザー変数の定義
...	
START (MACHINE)	; チャネル 1 の開始
START (LOADER)	; チャネル 2 の開始

START でローカルの「チャネル名称」、ユーザー変数およびパラメータ設定したチャネル名称を使用

プログラムコード	コメント
DEF INT chanNo1	; チャネル 1 のユーザー変数の定義
DEF INT chanNo2	; チャネル 2 のユーザー変数の定義
chanNo1 = CHAN_1	; パラメータ設定したチャネル名称チャネル 1 の割り当て
chanNo2 = CHAN_2	; パラメータ設定したチャネル名称チャネル 2 の割り当て
...	
START (ChanNo1)	; チャネル 1 の開始

4.1 フレキシブルな NC プログラミング

プログラムコード	コメント
START (ChanNo2)	; チャンネル 2 の開始

絶対パス指定による INIT 命令

チャンネル 2 でのプログラム/_N_MPF_DIR/_N_ABSPAN1_MPF の選択

プログラムコード

```
INIT(2, "/_N_WCS_DIR/_N_SHAFT1_WPD/_N_CUT1_MPF")
```

プログラム名称を使用した INIT 命令

名称"MYPROG"のプログラムの選択検索パスを使用してプログラムが検索されます。

プログラムコード

```
INIT(2, "MYPROG")
```

4.1 フレキシブルな NC プログラミング

WAITM によるプログラム協調

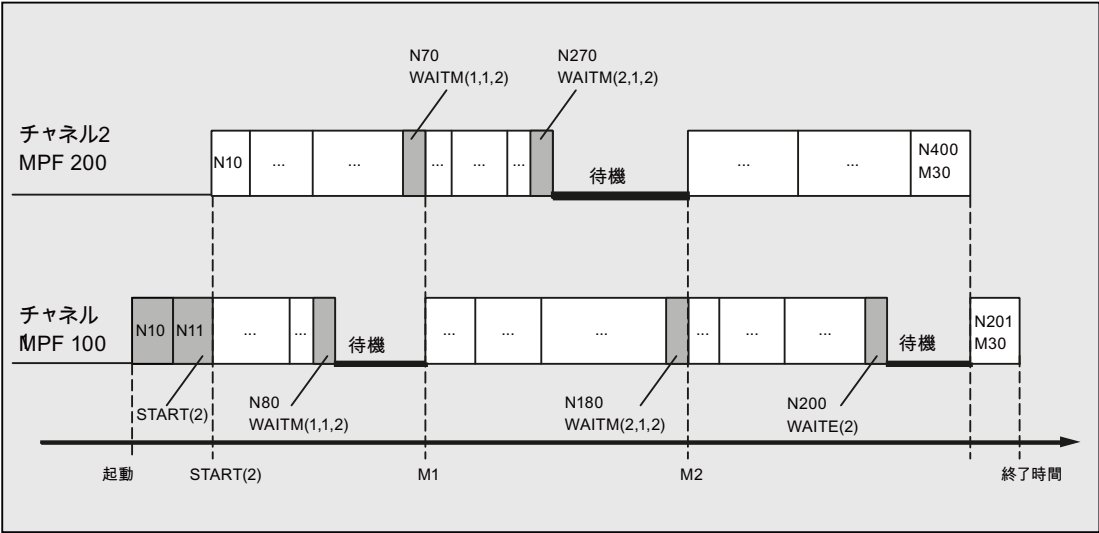
- チャンネル 1:プログラム/_N_MPF_DIR/_N_MPF100_MPF はすでに選択され、開始されています。

プログラムコード	コメント
	; プログラム MPF100
N10 INIT(2,"MPF200","N")	; プログラム MPF200 の選択、チャンネル 2
N11 START(2)	; チャンネル 2 の開始
...	
N80 WAITM(1,1,2)	; チャンネル 1 と 2 で WAIT マーク 1 を待機
N81 ...	; チャンネル 1、N81 とチャンネル 2、N71 は ; 同時に開始されます
...	
N180 WAITM(2,1,2)	; チャンネル 1 と 2 で WAIT マーク 2 を待機
N181 ...	; チャンネル 1、N181 とチャンネル 2、N271 は ; 同時に開始されます
...	
N200 WAITE(2)	; チャンネル 2 のプログラムの終了を待機
N201 ...	; N201 はプログラム終了まで開始されません ; MPF200 はチャンネル 2 で起動
N201 M30	; チャンネル 1 のプログラム終了

- チャンネル 2:チャンネル 1 で、ブロック N10 と N20 を使用してチャンネル 2 のためにプログラム MPF200_MPF が選択され、開始されます。

プログラムコード	コメント
;\$PATH=_N_MPF_DIR	; プログラム MPF200
...	
N70 WAITM(1,1,2)	チャンネル 1 と 2 で WAIT マーク 1 を待機
N71 ...	; チャンネル 1、N81 とチャンネル 2、N71 は ; 同時に開始されます
...	
N270 WAITM(2,1,2)	チャンネル 1 と 2 で WAIT マーク 2 を待機
N271 ...	; チャンネル 1、N181 とチャンネル 2、N271 は ; 同時に開始されます
...	
N400 M30	チャンネル 2 のプログラム終了

4.1 フレキシブルなNC プログラミング



必要条件

待機マーク後の後続ブロック実行の非同期開始

WAIT マークを使用したチャンネル協調の場合、次のブロックの実行を非同期で開始できます。この動作は、共通の待機マークに到達する直前に、同期化されるチャンネルのいずれかで操作がトリガされた場合に発生します。その結果、この残移動距離削除で自動的に再位置決め (REPOSA)が行われます。

前提条件:チャンネル 1 と 2 での現在の軸割り当て

- チャンネル 1:軸 X1 と軸 U
- チャンネル 2:軸 X2

表 4-2 チャンネル 1 と 2 の処理の流れ

チャンネル 1	チャンネル 2	説明
...	...	チャンネル 1 と 2 で任意の処理
N100 WAITM(20,1,2)		チャンネル 1:WAIT マークに到達し、チャンネル 2 との同期のために待機

4.1 フレキシブルなNCプログラミング

チャンネル 1	チャンネル 2	説明
GETD(U)処理の開始: • 軸入れ替え • 残移動距離削除 • REPOSA 終了	N200 GETD (U)	チャンネル 2:チャンネル 1 に軸 U を要求 チャンネル 1:バックグラウンドで GET (U) の処理
	N210 WAITM(20,1,2)	チャンネル 2:WAIT マークに到達⇒ これによって チャンネル 1 と 2 の同期が完了
	N220 G0	チャンネル 2:N220 の処理を開始
N110 G0 X1=100	X2=100	チャンネル 1:N110 の処理をずらして開始

4.1.9 マクロ機能(DEFINE ... AS)

通知
マクロ機能によるプログラミングの複雑化 マクロにより、制御装置のプログラミング言語が大幅に変わる場合があります。マクロ機能を使用する場合は、細心の注意が必要です。

マクロとは、マクロ名称と一緒に、個々の命令処理を割り当てたものです。プログラム運転中にマクロを呼び出すと、プログラム名称で指令された命令群が連続的に実行されます。

4.1 フレキシブルな NC プログラミング

有効範囲(つまり、マクロ定義が有効な範囲)に応じて、以下のマクロカテゴリがあります。

- ローカルマクロ

ローカルマクロは、実行時にメインプログラム以外の NC プログラムの先頭で定義した変数です。この変数は、NC プログラムが呼び出されると作成され、プログラムリセットの終了時またはコントロールシステムの起動時に削除されます。ローカルマクロには、この変数を定義した NC プログラム内でのみアクセスできます。

- プログラムグローバルマクロ

プログラムグローバルマクロは、実行時にメインプログラム以外の NC プログラムの開始場所で定義した変数です。この変数は、NC プログラムが呼び出されると作成され、プログラムリセットの終了時またはコントロールシステムの起動時に削除されます。プログラムグローバルマクロは、メインプログラムと、すべてのサブプログラム内でアクセスできます。

注記

プログラムグローバルマクロの使用可能性

次のマシンデータを設定した場合は、メインプログラムで定義したプログラムグローバルマクロはサブプログラムにのみ表示されます。

MD11120 \$MN_LUD_EXTENDED_SCOPE = 1

MD11120 = 0 の場合、メインプログラムで定義したプログラムグローバルマクロはメインプログラムでのみ使用できます。

- グローバルマクロ

グローバルマクロは、定義ファイル(マクロファイル)に定義され、プログラムリセット終了時、またはコントロールシステムの次回起動の後も保持される、NC グローバルマクロまたはチャネルグローバルマクロです。グローバルマクロは、任意のメインプログラムまたはサブプログラムで呼び出して実行できます。

注記

NC プログラムの**外部**マクロファイルのマクロを使用するには、マクロファイルを NC にダウンロードしてください。

マクロを使用する前に定義してください。この場合は、以下の規則を守ってください。

- あらゆる識別子、G、M、H の各機能、および L サブプログラムの名称は、マクロに定義できます。
- マクロは、プログラムの先頭、または専用の定義ファイル(マクロファイル)に定義できます。
- ローカルマクロとプログラムグローバルマクロは、プログラムの先頭に定義します。

4.1 フレキシブルな NC プログラミング

- グローバルファイルは、_N_DEF_DIR/_N_UMAC_DEF などのマクロファイルに定義してください。
- G 命令マクロは、グローバルマクロとしてしか定義できません。
- H 機能と L 機能は 2 桁でプログラム指令できます。
- M 命令と G 命令は 3 桁でプログラム指令できます。

注記

キーワードと予約名称は、マクロを使用して上書きすることはできません。これは、GOTO 命令内のすべてのジャンプの宛先と、FOR、WHILE、LOOP、REPEAT などのプログラムループ内ののキーワードにも当てはまります。

構文

マクロ定義:

DEFINE <マクロ名> AS <操作 1> <操作 2> ...

NC プログラムでの呼び出し:

<マクロ名>

意味

DEFINE ... AS:	マクロを定義するキーワードの組み合わせです。
<マクロ名>:	マクロ名称です。 マクロ名称として使用できるのは識別子のみです。 マクロは、NC プログラムからマクロ名称で呼び出します。
<操作 1>:	マクロ内の最初のプログラム命令
<操作 2>:	マクロ内の 2 番目のプログラム命令

例

例 1:プログラムの先頭のマクロ定義

プログラムコード	コメント
DEFINE LINE AS G1 G94 F300	; マクロ定義
...	
N70 LINE X10 Y20	; マクロの呼び出し
...	

4.2 サブプログラム機能

例 2:マクロファイルでのマクロ定義

プログラムコード	コメント
DEFINE M6 AS L6	; サブプログラムを工具交換時に呼び出し、必要なデータ送受信を処理します。実際の工具交換 M 機能はサブプログラムのなかで出力されます (M106 など)。
DEFINE G81 AS DRILL(81)	; DIN G 命令のエミュレーション
DEFINE G33 AS M333 G333	; ねじ切りのときに PLC によって同期を要求します。ユーザーのプログラムが同一のままになるよう、オリジナルの G 命令の G33 の名称が、マシンデータによって G333 へ変更されました。

例 3:外部マクロファイル

外部マクロファイルを制御装置に読み込んだ後で、マクロファイルを NC にダウンロードしてください。その後だけに、マクロを NC プログラムで使用できます。

プログラムコード	コメント
%_N_UMAC_DEF	
;\$PATH=/_N_DEF_DIR	; ユーザー用マクロ
DEFINE PI AS 3.14	
DEFINE TC1 AS M3 S1000	
DEFINE M13 AS M3 M7	; 主軸は右回り、クーラントはオン
DEFINE M14 AS M4 M7	; 主軸は左回り、クーラントはオン
DEFINE M15 AS M5 M9	; 主軸は停止、クーラントはオフ
DEFINE M6 AS L6	; 工具交換プログラムの呼び出し
DEFINE G80 AS MCALL	; 穴あけサイクルの選択解除
M30	

4.2 サブプログラム機能

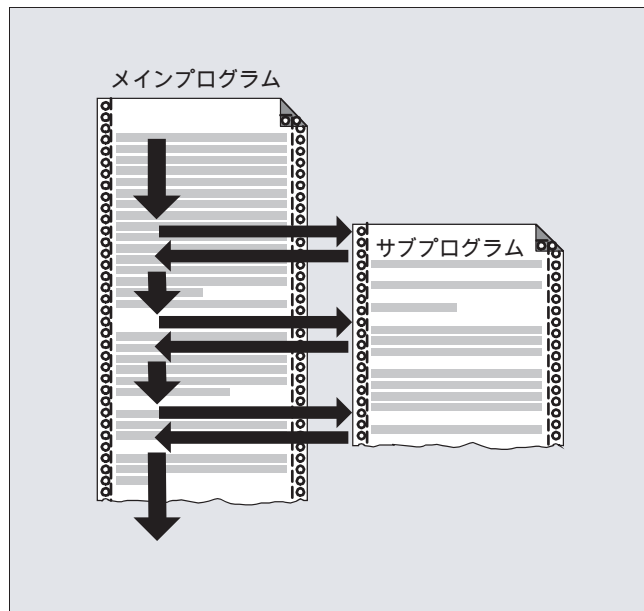
4.2.1 概要

4.2.1.1 サブプログラム

用語「サブプログラム」は、パートプログラムが、厳密にメインプログラムとサブプログラムに分類されたときから使用されるようになりました。メインプログラムとは、制御装置で処理をおこなうために選択され、起動されるパートプログラムでした。サブプログラムとは、メインプログラム内から呼び出されるパートプログラムでした。

今日の SINUMERIK NC 言語では、このように厳密には分類されていません。原則として、各パートプログラムは、メインプログラムとして選択して起動したり、サブプログラムとして別のパートプログラムから呼び出したりできます。

したがって、サブプログラムを使用すると、別のパートプログラム内から呼び出したパートプログラムを参照できます。



用途

すべての高レベルのプログラミング言語と同様に、NC 言語では、サブプログラムは、複数回使用されるプログラム区間を、独立した完全なプログラムに置き換えます。

サブプログラムには次の特長があります。

- プログラムをさらに使い易く、分かり易くする
- テスト済みのプログラム要素を再利用して品質を高める
- 特定の加工ライブラリの作成方法を提供する
- メモリ空間を節約する

4.2 サブプログラム機能

4.2.1.2 サブプログラム名称

命名規則

サブプログラム名称は、以下の規則に従って自由に選択できます。

- 使用可能な文字:
 - 英字:A ... Z、a ... z
 - 数字:0 ... 9
 - アンダースコア:_
- 先頭の 2 文字は、英字 2 文字またはアンダースコアと英字 1 文字にしてください。

注記

この条件を満たす場合は、プログラム名称を指定するだけで、別のプログラムからサブプログラムとして NC プログラムを呼び出すことができます。ただし、プログラム名称の先頭が数字である場合、サブプログラム呼び出しは CALL 命令でのみ呼び出すことができます。

- 最大長さ:24 文字

注記

大文字/小文字

SINUMERIK NC 言語では、大文字と小文字は **区別されません**。

注記

使用できないプログラム名称

Windows アプリケーションで問題が生じることのないように、次のプログラム名の使用は **避けて** ください。

- CON、PRN、AUX、NUL
 - COM1、COM2、COM3、COM4、COM5、COM6、COM7、COM8、COM9
 - LPT1、LPT2、LPT3、LPT4、LPT5、LPT6、LPT7、LPT8、LPT9
-

制御装置内部の拡張

サブプログラムの作成時に割り当てたプログラム名称には、制御装置内で接頭語と接尾語が追加され、拡張されます。

- 接頭語: _N_
- 接尾語: _SPF

プログラム名称の使用

サブプログラム呼び出し処理などでプログラム名称を使用するときは、接頭語、プログラム名称、および接尾語のすべての組み合わせを使用できます。

例:

プログラム名称が「SUB_PROG」のサブプログラムは、次の識別子で起動できます。

1. SUB_PROG
2. _N_SUB_PROG
3. SUB_PROG_SPF
4. _N_SUB_PROG_SPF

同じ名称のメインプログラムとサブプログラム

同じ名称のメインプログラム(.MPF)とサブプログラム(.SPF)が存在する場合は、NC プログラムでその名称を使用するときに、プログラムが一義的に識別されるように、適切なファイル拡張子を指定してください。そうしない場合、検索パスで最初に見つかった、指定した名称のプログラムが使用されます。

ファイル名とプログラム名

このファイルに含まれるファイル名とプログラム名は必ず同一でなければなりません。

そのため、1つのファイルは必ず1つのプログラムを含みます。

4.2.1.3 サブプログラムの入れ子

メインプログラムはサブプログラムを呼び出すことができ、そのサブプログラムはさらにサブプログラムを呼び出すことができます。このようにして、プログラムの順序はそれぞれに入れ子となります。各プログラムは、専用のプログラムレベルで実行されます。

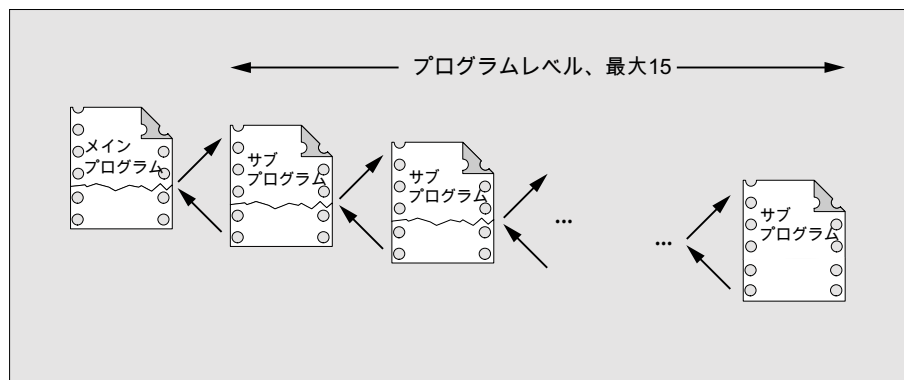
ネストレベル

NC 言語では現在、16 個のプログラムレベルが使用できます。メインプログラムは常に、最上位のプログラムレベル、0 で実行します。サブプログラムは常に、呼び出しレベルの次に高いプログラムレベルで実行します。したがって、プログラムレベル 1 が最初のサブプログラムレベルです。

4.2 サブプログラム機能

プログラムレベルの分類:

- プログラムレベル 0: メインプログラムのレベル
- プログラムレベル 1～15: サブプログラムレベル 1～15:



割り込みルーチン(ASUB)

サブプログラムを割り込みルーチン処理で呼び出した場合は、チャンネル(n)で現在有効なプログラムレベルではなく、その次に低いプログラムレベル(n+1)でサブプログラムが実行されます。現在のレベルが、最低のプログラムレベルでも可能となるよう、割り込みルーチンと組み合わせると、2個の追加プログラムレベル(16と17)を使用できます。

2個を超えるプログラムレベルが必要な場合は、そのチャンネルで実行するパートプログラムの構築時に、これを別途考慮してください。つまり、割り込み処理に必要な、十分な数のプログラムレベルの最大数まで、使用する場合があります。

たとえば、割り込み処理に4個のプログラムレベルが必要な場合、13までのプログラムレベルを使用するパートプログラムを構成してください。これで、割り込みが生じた場合に、必要な4個のプログラムレベル(14～17)を使用できます。

SIEMENS サイクル

SIEMENS サイクルには3個のプログラムレベルが必要です。したがって、SIEMENS サイクルは、少なくとも次のプログラムレベルで呼び出してください。

- パートプログラムの処理: プログラムレベル 12
- 割り込みルーチン: プログラムレベル 14

4.2.1.4 検索パス

パスを指定せずにサブプログラムを呼び出すと、コントロールシステムは事前定義された検索順序を使用して使用可能なプログラムメモリを検索します(「サブプログラム呼び出しの検索パス (ページ 637)」を参照してください)。

4.2.1.5 仮パラメータと実パラメータ

仮パラメータと実パラメータは、定義と、パラメータ転送を含むサブプログラム呼び出しを組み合わせると発生します。

仮パラメータ

サブプログラムを定義すると、サブプログラムに転送されるパラメータ(仮パラメータと呼ばれます)を、タイプとパラメータ名称で定義する必要があります。

したがって、仮パラメータはサブプログラムのインタフェースです。

例:

プログラムコード	コメント
PROC CONTOUR (REAL X, REAL Y)	; 仮オペランド:X と Y、ともに REAL タイプです。
N20 X1=X Y1=Y	; 軸 X1 が位置 X へ、および軸 Y1 が位置 Y へ移動
...	
N100 RET	

実パラメータ

サブプログラムを呼び出すときは、それに絶対値または変数(実パラメータと呼ばれます)を転送する必要があります。

そして、実パラメータは、サブプログラム呼び出し時に、そのインタフェースに最新値を割り当てます。

例:

プログラムコード	コメント
N10 DEF REAL WIDTH	;変数の定義
N20 WIDTH=20.0	; 変数割り当て
N30 CONTOUR(5.5, WIDTH)	; 実パラメータによるサブプログラム呼び出し:5.5 と WIDTH
...	
N100 M30	

4.2 サブプログラム機能

4.2.1.6 パラメータ転送

パラメータ転送を含むサブプログラムの定義

パラメータ転送を含むサブプログラムは、PROC キーワード、とサブプログラムが予期しているすべてのパラメータのリスト全体を使用して定義されます。

不完全なパラメータ転送

サブプログラムの呼び出し時に、サブプログラムインタフェースで定義されたすべてのパラメータを明示的に転送する必要があるわけではありません。パラメータを省略すると、初期値「0」が転送されます。

ただし、パラメータの順序が一義的に識別されるよう、パラメータ区切り文字として、コンマを含めてください。最後のパラメータは例外です。呼び出しで最後のパラメータを省略した場合は、最後のコンマも省略できます。

例:
サブプログラム:

プログラムコード	コメント
PROC SUB_PROG (REAL X, REAL Y, REAL Z)	; 仮オペランド:X、Y、および Z
...	
N100 RET	

メインプログラム

プログラムコード	コメント
PROC MAIN_PROG	
...	
N30 SUB_PROG(1.0,2.0,3.0)	; 完全なパラメータの転送によるサブプログラム呼び出し: X=1.0、Y=2.0、Z=3.0
...	
N100 M30	

N30 に、不完全なパラメータ転送によるサブプログラム呼び出しの例:

N30 SUB_PROG(,2.0,3.0)	; X=0.0、Y=2.0、Z=3.0
N30 SUB_PROG(1.0, ,3.0)	; X=1.0、Y=0.0、Z=3.0
N30 SUB_PROG(1.0,2.0)	; X=1.0、Y=2.0、Z=0.0
N30 SUB_PROG(, ,3.0)	; X=0.0、Y=0.0、Z=3.0
N30 SUB_PROG(, ,)	; X=0.0、Y=0.0、Z=0.0

通知
参照渡しパラメータ転送 参照渡しで転送されたパラメータは、サブプログラム呼び出しでは省略しないでください。
通知
AXIS データタイプ AXIS データタイプのパラメータは、サブプログラム呼び出しでは省略しないでください。

転送パラメータのチェック

システム変数\$P_SUBPAR[n](ここでn=1,2,...)を使用すると、パラメータが明示的に転送されたか、サブプログラムで省略されたかをチェックできます。インデックスnは、仮パラメータの順序を表わします。インデックスn=1は、1番目の仮パラメータ、インデックスn=2は2番目の仮パラメータを表わし、以降も同様です。

次のプログラム引用部分は、1番目の仮パラメータに基づいてチェックがおこなわれる例を示します。

プログラミング	コメント
PROC SUB_PROG (REAL X, REAL Y, REAL Z)	; 仮オペランド:X、Y、およびZ
N20 IF \$P_SUBPAR[1]==TRUE	; 1番目の仮オペランドXのチェック。
...	; この部分の動作は、仮オペランドXが明示的に転送された場合に実行されます。
N40 ELSE	
...	; この部分の動作は、仮オペランドXが転送されなかった場合に実行されます。
N60 ENDIF	
...	; 一般動作
N100 RET	

4.2.2 サブプログラムの定義

4.2.2.1 パラメータ転送を含まないサブプログラム

パラメータ転送を含まないサブプログラムを定義するときは、プログラムの先頭の定義行を省略できます。

4.2 サブプログラム機能

構文

```
[PROC <プログラム名称>]  
...
```

意味

PROC:	プログラムの先頭の定義演算
<プログラム名称>:	プログラムの名前

例

例 1:PROC 命令のあるサブプログラム

プログラムコード	コメント
PROC SUB_PROG	; 定義行
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; サブプログラム戻り

例 2 :PROC 命令のないサブプログラム

プログラムコード	コメント
N10 G01 G90 G64 F1000	
N20 X10 Y20	
...	
N100 RET	; サブプログラム戻り

下記も参照

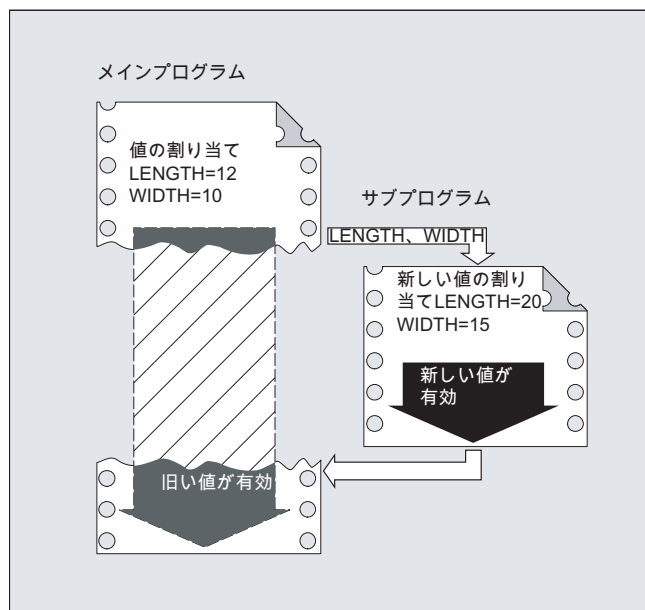
パラメータ転送を含まないサブプログラム呼び出し (ページ 595)

4.2.2.2 値渡しパラメータ転送を含むサブプログラム(PROC)

値渡しパラメータ転送を含むサブプログラムは、PROC キーワードと、その次にプログラム名称、その次にすべてのパラメータのタイプと名称の完全なリストを使用して定義します。定義演算は、最初のプログラム行に入れてください。

値渡し

呼び出しプログラムは、値渡しパラメータ転送時に変数の値をサブプログラムに転送するだけです。そのため、サブプログラムには変数の直接アクセスは与えられません。このように、パラメータ値が変更されるときに、サブプログラム内に表示される値だけが変更されます。呼び出しプログラムで定義された変数の値は変更されないままです。結果として、値渡しパラメータ転送は、呼び出しプログラムには影響しません。



構文

```
PROC <プログラム名称> (<パラメータタイプ> <パラメータ名称>=<初期値>, ...)
```

注記

127 個までのパラメータを転送できます。

意味

PROC:	プログラムの先頭の定義演算
<プログラム名称>:	プログラムの名称
<パラメータタイプ>:	パラメータのデータタイプ (REAL、INT、BOOL など)

4.2 サブプログラム機能

<パラメータ名称>:	パラメータの名称
<初期値>:	パラメータの初期値の選択(オプション) サブプログラムを呼び出したときにパラメータが指定されていない場合は、パラメータに初期値が割り当てられます。

例

例 1

初期値を持つ REAL タイプの 3 つのパラメータによるサブプログラム SUB_PROG の定義

プログラムコード

```
PROC SUB_PROG (REAL LENGTH=10.0, REAL WIDTH=20.0, REAL HIGHT=30.0)
```

例 2

さまざまな呼び出しバージョン

プログラムコード

```
PROC MAIN_PROG
  REAL PAR_1 = 100
  REAL PAR_2 = 200
  REAL PAR_3 = 300
  ; 呼び出し変数
  SUB_PROG
  SUB_PROG (PAR_1, PAR_2, PAR_3)
  SUB_PROG (PAR_1)
  SUB_PROG (PAR_1, , PAR_3)
  SUB_PROG ( , , PAR_3)
N100 RET
```

下記も参照

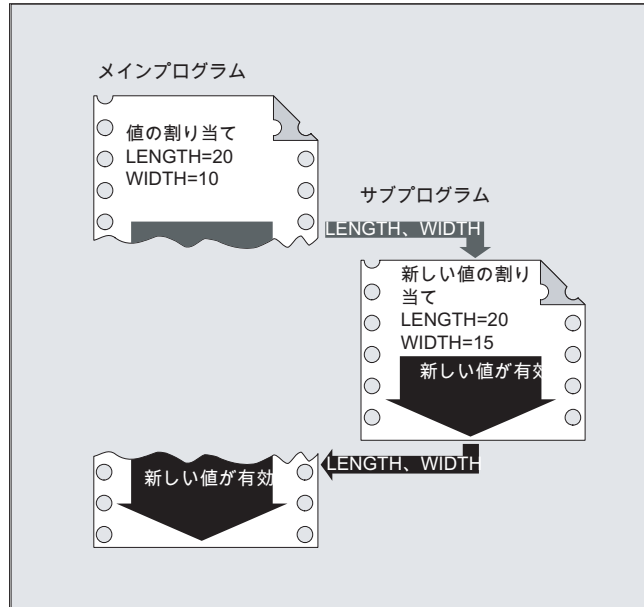
パラメータ転送を含むサブプログラム呼び出し(EXTERN) (ページ 597)

4.2.2.3 参照渡しパラメータ転送を含むサブプログラム(PROC、VAR)

参照渡しパラメータ転送を含むサブプログラムは、PROC キーワードと、その次にプログラム名称、その次にすべてのパラメータの VAR キーワード、タイプ、および名称のリスト全体を使用して定義します。定義演算は、最初のプログラム行に入れてください。パラメータとして、配列に対する参照先も転送できます。

参照渡し

呼び出しプログラムは、参照渡しパラメータ転送時に変数の値をサブプログラムに転送するだけでなく、変数への参照(ポインタ)も転送します。これにより、サブプログラムに変数への直接アクセスが与えられます。このように、パラメータ値が変更されるときに、サブプログラム内に表示される値が変更されるだけでなく、呼び出しプログラムで定義された変数の値も変更されます。したがって、参照渡しパラメータ転送は、サブプログラムの終了後も呼び出しプログラムに影響を与えます。



注記

参照渡しパラメータ転送が必要となるのは、転送した変数を呼び出しプログラムで(LUD)でローカルに定義している場合のみです。チャンネルグローバル変数または NC グローバル変数は、サブプログラム内から直接アクセスできないため、これらの変数を転送しないでください。

構文

```

PROC <プログラム名称> (VAR <パラメータタイプ> <パラメータ名称>, ...)
PROC <プログラム名称> (VAR <配列タイプ><配列名称>, [<m>,<n>,<o>], ...)
  
```

注記

127 個までのパラメータを転送できます。

4.2 サブプログラム機能

意味

PROC:	プログラムの先頭の定義演算
VAR:	参照によるパラメータ転送のキーワード
<プログラム名称>:	プログラムの名称
<パラメータタイプ>:	パラメータのデータタイプ (REAL、INT、BOOL など)
<パラメータ名称>:	パラメータの名称
<配列タイプ>:	配列要素のデータタイプ (REAL、INT、BOOL など)
<配列名称>:	配列の名称
[<m>, <n>, <o>]:	配列サイズ 現在は、3次元までの配列を使用できます。
	<m>: 1次元の配列サイズ
	<n>: 2次元の配列サイズ
	<o>: 3次元の配列サイズ

注記

- PROC キーワードの後に指定するプログラム名称は、操作画面で割り当てたプログラム名称と一致させてください。
- 配列長を定義していない配列を使用すると、サブプログラムは、可変長の配列を仮パラメータとして処理できます。たとえば、2次元の配列を仮パラメータとして定義すると、1次元の長さは指定されません。ただし、コンマは入れてください。
例: PROC <プログラム名称> (VAR REAL ARRAY[, 5])

例

2 個のパラメータを含むサブプログラムを、REAL タイプの参照として定義:

プログラムコード

```
; パラメータ 1: タイプの参照: REAL、名称: LENGTH
; パラメータ 2: タイプの参照: REAL、名称: WIDTH
PROC SUB_PROG (VAR REAL LENGTH, VAR REAL WIDTH)
```

下記も参照

パラメータ転送を含むサブプログラム呼び出し (EXTERN) (ページ 597)

4.2.2.4 モーダル G 機能の保存(SAVE)

SAVE 属性は、サブプログラム呼び出しの前に、動作中のモーダル G 命令が保存され、サブプログラムの終了後に再度有効になることを意味します。

通知

連続軌跡モードの中断

連続軌跡モードが動作中の場合に、サブプログラムが SAVE 属性で呼び出されると、サブプログラムの終了時に連続軌跡モードが中断します(復帰ジャンプ)。

構文

PROC <サブプログラム名称> SAVE

意味

SAVE:	サブプログラム呼び出しの前にモーダル G 命令を保存し、サブプログラムの終了後に復帰します。
-------	--

例

CONTOUR サブプログラムには、モーダル G 命令 G91 が適用されます(インクリメンタル指令)。モーダル G 命令 G90 は、メインプログラムで有効です(アブソリュート指令)。G90 は、SAVE によるサブプログラム定義により、サブプログラムの終了後に再度メインプログラムで有効になります。

サブプログラム定義:

プログラムコード	コメント
PROC CONTOUR (REAL VALUE1) SAVE	; SAVE パラメータによるサブプログラムの定義
N10 G91 ...	; モーダル G 命令 G91:インクレメンタル指令
N100 M17	; サブプログラム終了

メインプログラム

プログラムコード	コメント
N10 G0 X...Y...G90	; モーダル G 命令 G90:アブソリュート指令
N20 ...	
...	
N50 CONTOUR (12.4)	; サブプログラム呼び出し
N60 X...Y...	; SAVE を使用してモーダル G 命令 G90 を再度有効にします

4.2 サブプログラム機能

必要条件

フレーム

SAVE 属性を含むサブプログラムに関するフレームの動作は、フレームタイプによって異なり、マシンデータで設定できます。

4.2.2.5 シングルブロック実行のマスク(SBLOF、SBLON)

シングルブロック加工が有効な場合でも、中断せずに NC プログラムの全体または一部を処理することができます。シングルブロック加工は SBLOF 命令でマスクし、SBLON 命令で再度有効にします。

NC プログラム全体に対するシングルブロック加工のマスク

メインプログラムの第 1 行(PROC)でシングルブロック加工(SBLOF)の解除がプログラムされている場合、NC プログラムの終了時、または NC プログラムがキャンセルされるまで有効なままです。この場合、NC プログラムは、シングルブロックモード内では停止せずに実行されます。

サブプログラムの第 1 行(PROC)でシングルブロック加工(SBLOF)の解除がプログラムされている場合、NC プログラムの終了時、または NC プログラムがキャンセルされるまで有効なままです。プログラムされたリターン命令によって、サブプログラムの最後で停止するかどうかが決まります。

- M17 で戻ります。サブプログラムの終了で停止します
- RET で戻ります。サブプログラムの終了で停止しません

NC プログラム内でのシングルブロック加工のマスク

NC プログラム内のブロックでシングルブロック加工の解除(SBLOF)がプログラムされている場合、このブロック以降、シングルブロック加工の次にプログラムされた起動(SBLON)または有効なサブプログラムレベルの終了まで、シングルブロック加工が解除されます。

構文

NC プログラム全体に対するシングルブロック加工のマスク:

```
PROC ... SBLOF  
...
```

NC プログラム内でのシングルブロック加工のマスク:

```
...  
SBLOF
```

```

...
SBLON
...

```

意味

PROC:	プログラムの最初の命令	
SBLOF:	シングルブロック加工を解除するための事前定義された手順	
	単独ブロック指令:	はい、PROC ブロックで可能
	有効性:	モーダル
SBLON:	シングルブロック加工を起動するための事前定義された手順	
	単独ブロック指令:	あり
	有効性:	モーダル

特殊な側面

- シングルブロック加工をマスクしたブロック表示**
 DISPLOF を使用して、実行中のブロック表示をサブプログラムで抑制できます。
 DISPLOF が SBLOF と組み合わせてプログラムされている場合、サブプログラム内のシングルブロック停止について、サブプログラムの呼び出しが表示されます。
- 非同期サブプログラム(ASUB)によるシングルブロック加工のマスク**
 シングルブロック加工が有効な ASUB を 1 ステップで実行するには、PROC 命令を SBLOF を含む ASUB でプログラム指令してください。これは、機能「編集可能なシステム ASUB」(MD11610 \$MN_ASUP_EDITABLE)の場合にも該当します。
 システムまたはユーザー ASUB のシングルブロック停止を、PROC 行で SBLOF をプログラム指令するか、マシンデータ MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK (bit0 = 1 または bit1 = 1) の設定内容を使用してマスクした場合は、ASUB で SBLON をプログラム指令することでシングルブロック停止を再度有効にできます。
 ユーザー ASUB でのシングルブロック停止を、マシンデータ MD20117 \$MC_IGNORE_SINGLEBLOCK_ASUP での設定を使用してマスクした場合は、ASUB で SBLON をプログラム指令してシングルブロック停止を再度有効にすることはできません。

4.2 サブプログラム機能

- 各種シングルブロック加工タイプの特記事項
 - 「SB2:算術ブロック」および MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK、ビット 12=1:
→ SBLON ブロックではプログラムは停止されません。
 - 「SB3:精密シングルブロック」:
→ → SBLOF 命令がマスクされます。
- ネスト化されているプログラムでのシングルブロック加工のマスク
SBLOF をサブプログラムの PROC 命令でプログラム指令した場合は、M17 のサブプログラム復帰で実行が停止します。これにより、呼び出しプログラムの次のブロックが先に実行されないようにします。SBLOF を含むが PROC ステートメント内には SBLOF がないサブプログラムで、シングルブロック加工のマスクを有効にした場合は、呼び出しプログラムの次の運転機能ブロックの後にのみ、実行が停止します。この動作を望まない場合は、サブプログラムで戻りジャンプ(M17)の前に SBLON をプログラム指令してください。RET では、実行が停止して、より高レベルのプログラムへ復帰することはありません。

例

例 1:NC プログラム内でのシングルブロック加工のマスク

初期状態シングルブロック加工が有効です。

プログラムコード	コメント
N10 G1 X100 F1000	
N20 SBLOF	; シングルブロック加工のオフ
N30 Y20	
N40 M100	
N50 R10=90	
N60 SBLON	; シングルブロック加工の再有効化
N70 M110	
N80 ...	

N20 と N60 間の領域は、シングルブロックモードでは 1 ステップとして実行されます。

例 2:ユーザー命令と同じように動作するサイクル

初期状態シングルブロック加工が有効です。

メインプログラム

プログラムコード
...
N100 G1 X10 G90 F200
N120 X-4 Y6

プログラムコード

```

N130 CYCLE1
N140 G1 X0
N150 M30

```

サイクル CYCLE1:

プログラムコード

コメント

プログラムコード	コメント
N100 PROC CYCLE1 DISPLOF SBLOF	; プログラム全体のシングルブロック加工のマスク。
N110 R10=3*SIN(R20)+5	
N120 IF (R11 <= 0)	
N130 SETAL(61000)	
N140 ENDIF	
N150 G1 G91 Z=R10 F=R11	
N160 M17	

サイクル CYCLE1 はシングルブロック加工が有効な状態で処理されます。シングルブロック加工が有効なときに CYCLE1 が処理されます。つまり、CYCLE1 を処理するには、起動キーを 1 回だけ押さなければなりません。

例 3:変更されたゼロオフセットと工具オフセットを有効にするために、PLC が起動した ASUB が、表示されずに実行されます。

プログラムコード

コメント

プログラムコード	コメント
N100 PROC NV SBLOF DISPLOF	; シングルブロック加工とブロック表示のマスク。
N110 CASE \$P_UIFRNUM OF	
0 GOTOF _G500	
1 GOTOF _G54	
2 GOTOF _G55	
3 GOTOF _G56	
4 GOTOF _G57	
DEFAULT GOTOF END	
N120 _G54:G54 D=\$P_TOOL T=\$P_TOOLNO	
N130 RET	
N140 _G54:G55 D=\$P_TOOL T=\$P_TOOLNO	
N150 RET	
N160 _G56:G56 D=\$P_TOOL T=\$P_TOOLNO	
N170 RET	
N180 _G57:G57 D=\$P_TOOL T=\$P_TOOLNO	
N190 RET	
N200 END:D=\$P_TOOL T=\$P_TOOLNO	
N210 RET	

4.2 サブプログラム機能

例 4:サブプログラムでの個別の停止

初期状態

- シングルブロック加工が有効です。
- MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK, Bit12 = 1

メインプログラム

プログラムコード	コメント
N10 G0 X0	; シングルブロック停止
N20 X10	; シングルブロック停止
N30 CYCLE	; サイクルが生成した移動ブロックです。
N50 G90 X20	; シングルブロック停止
M30	

サイクル CYCLE:

プログラムコード	コメント
PROC CYCLE SBLOF	; シングルブロック加工のマスク
N100 R0=1	
N110 SBLON	; MD10702、ビット 12=1 によりシングルブロック停止は発生しません
N120 X1	; シングルブロック停止
N140 SBLOF	
N150 R0=2	
RET	

例 5:プログラムのネスト化によるシングルブロック加工のマスク

初期状態

- シングルブロック加工タイプ 2 が有効です。
- プログラム実行は SBLON ブロックで中断されません
(MD10702 \$MN_IGNORE_SINGLEBLOCK_MASK, Bit12 = 1)

プログラムコード	コメント
N10 X0 F1000	; シングルブロック停止
N20 UP1 (0)	
PROC UP1 (INT _NR) SBLOF	; UP1 のシングルブロック加工の解除
N100 X10	
N110 UP2 (0)	
PROC UP2 (INT _NR)	
N200 X20	
N210 SBLON	; シングルブロック加工の有効化
N220 X22	; シングルブロック停止
N230 UP3 (0)	

プログラムコード	コメント
PROC UP3 (INT _NR)	
N300 SBLOF	; シングルブロック加工のオフ
N305 X30	
N310 SBLON	; シングルブロック加工の有効化
N320 X32	; シングルブロック停止
N330 SBLOF	; シングルブロック加工のオフ
N340 X34	
N350 M17	; シングルブロック停止 (M17)
N240 X24	; シングルブロック停止 (N210)
N250 M17	; シングルブロック停止 (M17)
N120 X12	
N130 M17	; シングルブロック停止 (M17)
N30 X0	; シングルブロック停止
N40 M30	; シングルブロック停止

4.2.2.6 実行中のブロック表示のマスク (DISPLOF、DISPLON、ACTBLOCNO)

通常では、実行中のプログラムブロックがブロック表示に表示されます。DISPLOF 命令を使用すると、実行中のサイクルとサブプログラムのブロック表示をマスクできます。実行中のブロックではなく、サイクルまたはサブプログラムの呼び出しが表示されます。DISPLON 命令は、ブロック表示のマスクを無効にします。

DISPLOF と DISPLON は、PROC 命令でプログラム行にプログラム指令され、サブプログラム全体に対して有効です。また、このサブプログラムが呼び出した、DISPLON 命令も DISPLOF 命令も含まないすべてのサブプログラムにも自動的に有効になります。これは、すべての ASUB に当てはまります。

構文

```
PROC ... DISPLOF
PROC ... DISPLOF ACTBLOCNO
PROC ... DISPLON
```

4.2 サブプログラム機能

意味

DISPLOF:	実行中のブロック表示をマスクする命令です。	
	場所:	PROC 命令を含むプログラム行の末尾
	効果:	サブプログラムから、またはプログラムの終点からの復帰ジャンプまでです。
	注: サブプログラムから DISPLOF 命令を使用して、さらにサブプログラムを呼び出した場合は、実行中のブロック表示は、これらのサブプログラムでもマスクされます。ただし、これらのサブプログラムで DISPLON が明示的にプログラム指令されている場合はマスクされません。	
DISPLON:	実行中のブロック表示のマスクを無効にする命令です。	
	場所:	PROC 命令を含むプログラム行の末尾
	効果:	サブプログラムから、またはプログラムの終了からの復帰ジャンプまで。
	注: サブプログラムから DISPLON 命令で、さらにサブプログラムを呼び出した場合は、実行中のブロックが、これらのサブプログラムでも表示されます。ただし、これらのサブプログラムで DISPLOF が明示的にプログラム指令されている場合は表示されません。	
ACTBLOCNO:	DISPLOF を ACTBLOCNO 属性と併用すると、アラームが発生する場合に、アラームが発生した実際のブロック番号が出力されます。これは、より低いプログラムレベルで DISPLOF のみをプログラム指令した場合にも当てはまります。 また、DISPLOF を ACTBLOCNO と併用しない場合は、DISPLOF で指定されていない最後のプログラムレベルのサイクル呼び出しまたはサブプログラム呼び出しのブロック番号が表示されます。	

例

例 1:サイクルの実行中のブロック表示をマスク

プログラムコード	コメント
PROC CYCLE (AXIS TOMOV, REAL POSITION)	; 実行中のブロック表示をマスクし、代わりに、サイクル呼び出しを表示します。例:CYCLE (X,100.0)
SAVE DISPLOF	
DEF REAL DIFF	; サイクルの内容
G01 ...	

プログラムコード	コメント
...	
RET	; サブプログラム復帰ジャンプ。サイクル呼び出しの次のブロックがブロック表示に表示されます。

例 2: アラーム発生ブロックの表示

サブプログラム SUBPROG1 (ACTBLOCNO を含む):

プログラムコード	コメント
PROC SUBPROG1 DISPLOF	
ACTBLOCNO	
N8000 R10 = R33 + R44	
...	
N9040 R10 = 66 X100	; アラーム 12080 のトリガ
...	
N10000 M17	

サブプログラム SUBPROG2 (ACTBLOCNO を含まない):

プログラムコード	コメント
PROC SUBPROG2 DISPLOF	
N5000 R10 = R33 + R44	
...	
N6040 R10 = 66 X100	; アラーム 12080 のトリガ
...	
N7000 M17	

メインプログラム

プログラムコード	コメント
N1000 G0 X0 Y0 Z0	
N1010 ...	
...	
N2050 SUBPROG1	; アラーム出力=「12080 チャネル K1 ブロック N9040 テキスト R10=の構文エラー」
N2060 ...	
N2350 SUBPROG2	; アラーム出力=「12080 チャネル K1 ブロック N2350 テキスト R10=の構文エラー」
...	
N3000 M30	

4.2 サブプログラム機能

例 3 :実行中のブロック表示のマスクは無効です

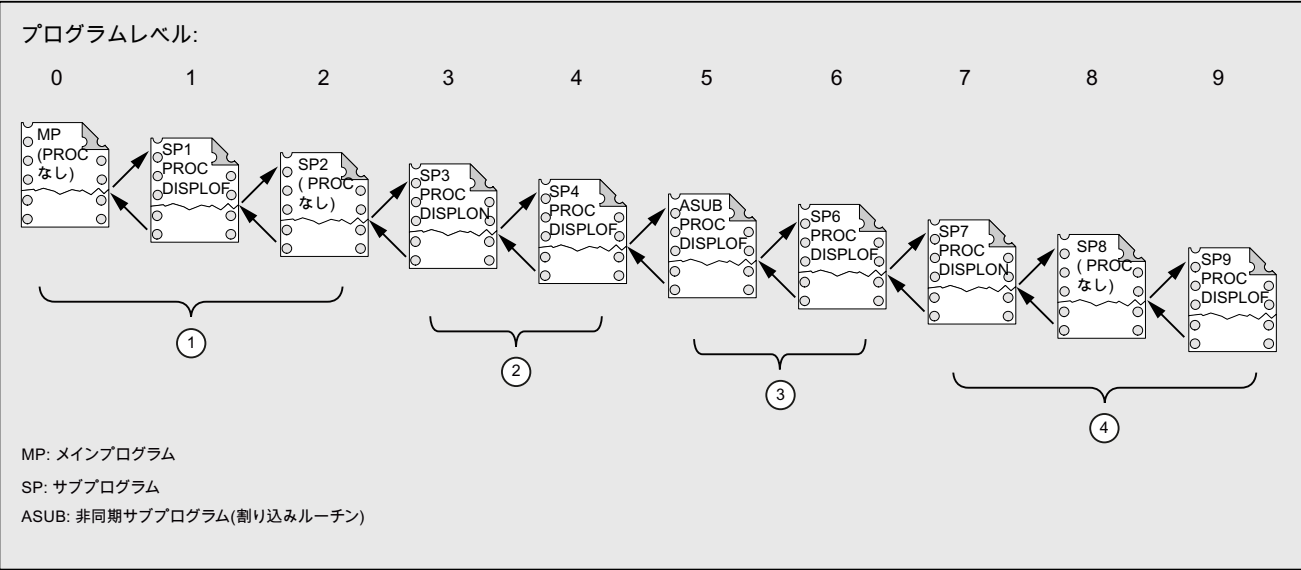
マスクを含むサブプログラム SUB1:

プログラムコード	コメント
PROC SUB1 DISPLOF	; サブプログラム SUB1 で実行中のブロック表示をマスクします。代わりに、ブロックが SUB1 呼び出しで表示されます。
...	
N300 SUB2	; サブプログラム SUB2 を呼び出します。
...	
N500 M17	

マスクを含まないサブプログラム SUB2:

プログラムコード	コメント
PROC SUB2 DISPLON	; サブプログラム SUB2 で実行中のブロック表示のマスクを無効にします。
...	
N200 M17	; サブプログラム SUB1 に戻ります。実行中のブロック表示のマスクを SUB1 で復帰します。

例 4 :DISPLON/DISPLOF の組み合わせが異なる場合の表示動作



- ① プログラムレベル 0 のパートプログラム行が、実行中のブロック表示に表示されます。
- ② プログラムレベル 3 のパートプログラム行が、実行中のブロック表示に表示されます。
- ③ プログラムレベル 3 のパートプログラム行が、実行中のブロック表示に表示されます。
- ④ プログラムレベル 7/8 のパートプログラム行が、実行中のブロック表示に表示されます。

4.2.2.7 解析によるサブプログラムの識別(PREPRO)

すべてのファイルは電源投入時に、PROC 命令行の末尾の PREPRO キーワードで識別されます。

注記

この NC プログラム事前処理のタイプは、マシンデータ MD10700 \$MN_PREPROCESSING_LEVEL の設定によって異なります。工作機械メーカーの説明書に従ってください。

構文

```
PROC ... PREPRO
```

意味

PREPRO:	電源投入時に解析される(サイクルディレクトリに格納された NC プログラムの)全ファイル識別用キーワードです。
---------	---

解析されたサブプログラムの読み取り、およびサブプログラムの呼び出し

サイクルディレクトリは、パラメータにより、サブプログラムが電源投入時に先読みされる場合も、サブプログラム呼び出しのときに先読みされる場合も、同じ順序で処理されます。

1. _N_CUS_DIR ユーザーのサイクル
2. _N_CMA_DIR メーカーサイクル
3. _N_CST_DIR 標準サイクル

同名称でも内容が異なる NC プログラムの場合は、最初に見つかった PROC 命令が有効になり、それ以外の PROC 命令はアラームメッセージなしで無視されます。

4.2 サブプログラム機能

4.2.2.8 サブプログラム戻り M17

復帰命令 M17 (またはパートプログラム終了命令 M30)は、サブプログラムの末尾にあります。この命令は、サブプログラム呼び出しの次のパートプログラムブロックの位置で、呼び出しプログラムに戻るよう指示します。

注記

M17 と M30 は、NC 言語では同等に扱われます。

構文

```
PROC <プログラム名称>  
...  
M17/M30
```

必要条件

連続軌跡モードのサブプログラム戻りの動作

M17 (または M30)が単独でパートプログラムブロックに含まれる場合は、チャンネルで動作中の連続軌跡モードが中断されます。

連続軌跡モードの中断を回避するには、M17 (または M30)を最後の移動ブロックに含めてください。さらに、次のマシンデータを「0」に設定してください。

MD20800 \$MC_SPF_END_TO_VDI = 0 (NC/PLC インタフェースへの M30/M17 出力なし)

例

1. 個別のブロックの M17 のサブプログラム

プログラムコード	コメント
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10	
N30 M17	; 連続軌跡モードを中断して復帰ジャンプします。

2. 最後の移動ブロックに M17 を含むサブプログラム

プログラムコード	コメント
N10 G64 F2000 G91 X10 Y10	
N20 X10 Z10 M17	; 連続軌跡モードを中断しない復帰ジャンプです。

4.2.2.9 RET サブプログラム戻り

RET 命令も、M17 復帰ジャンプ命令の代わりとしてサブプログラムで使用できます。RET は、個別のパートプログラムブロックでプログラム指令してください。M17 と同様に、RET は、サブプログラム呼び出しの次のパートプログラムブロックで呼び出しプログラムに戻るよう指示します。

注記

RET の復帰ジャンプ動作は、パラメータをプログラム指令して変更することができます(「設定可能なサブプログラム復帰ジャンプ(RET ...) (ページ 582)」を参照してください)。

用途

G64 連続軌跡モード(G641～G645)が復帰ジャンプで中断されないようにする場合は、RET 命令を使用してください。

必要条件

RET 命令は、SAVE 属性で定義されていないサブプログラムでのみ使用できます。

構文

```
PROC <プログラム名称>  
...  
RET
```

例

メインプログラム

プログラムコード	コメント
PROC MAIN_PROGRAM	; プログラムの開始
...	
N50 SUB_PROG	; サブプログラム呼び出し: SUB_PROG
N60 ...	
...	
N100 M30	; プログラム終了

4.2 サブプログラム機能

サブプログラム:

プログラムコード	コメント
PROC SUB_PROG	
...	
N100 RET	; メインプログラムの N60 ブロックへ復帰ジャンプ

4.2.2.10 設定可能なサブプログラム復帰ジャンプ(RET ...)

通常、復帰ジャンプは、RET 命令を使用してサブプログラムから呼び出しプログラムに戻ります。そして、サブプログラム呼び出しの次のプログラム行から処理が継続されます。ただし、次の例のように、プログラムの処理を別の位置から継続することもできます。

- (輪郭の記述後に)ISO 系言語モードの旋削サイクルの呼び出し後に、プログラムの実行を再開します。
- エラー処理のために、(ASUB 後でも)任意のサブプログラムレベルからメインプログラムに戻ります。
- コンパイルサイクルと ISO 系言語モードで、特別な用途のために複数のプログラムレベルをジャンプして復帰します。

これを達成するには、RET 命令を追加のパラメータを付けてプログラム指令します。

検索方向

パラメータ<ターゲットブロック>を指定した場合、最初に呼び出しブロックの後のブロックに復帰ジャンプします。次に、復帰ジャンプ先のプログラムの末尾方向にターゲットの検索がおこなわれます。見つからなかった場合は、プログラムの先頭方向に検索がおこなわれます。

構文

```
RET("<ターゲットブロック>")
RET("<ターゲットブロック>",<ターゲットブロックの後のブロック>)
RET("<ターゲットブロック>",<ターゲットブロックの後のブロック>,<復帰ジャンプレベルの数>)
RET("<ターゲットブロック>",<復帰ジャンプレベルの数>)
RET("<ターゲットブロック>",<ターゲットブロックの後のブロック>,<復帰ジャンプレベルの数>,<プログラムの先頭への復帰ジャンプ>)
RET( , ,復帰ジャンプレベルの数,<プログラムの先頭への復帰ジャンプ>)
```

意味

RET:	サブプログラム終了	
<ターゲットブロック>:	パラメータにおいて、プログラムの実行を再開するブロックをジャンプ先として指定してください。 パラメータ<復帰ジャンプレベルの数>をプログラム指令していない場合は、ジャンプ先は実行中のサブプログラムを呼び出したプログラムになります。 指定できるデータには以下のものがあります。	
	<ブロック番号> >	ターゲットブロックの番号です。 ブロック番号の検索は、復帰ジャンプ先プログラムで、最初にプログラムの末尾方向に実行されます。
	<ジャンプマーク>	復帰ジャンプ先プログラムに存在するジャンプマークにしてください。 ジャンプマーカの検索は、復帰ジャンプ先プログラムで、最初にプログラムの末尾方向に実行されます。
	<文字列>	復帰ジャンプ先プログラムに存在する文字列にしてください(プログラム名称や変数名称など)。 文字列の検索は、復帰ジャンプ先プログラムで、最初にプログラムの末尾方向に実行されます。 文字列をプログラム指令するときは、以下の規則が適用されます。 • 末尾にブランクを付けます(末尾に「:」を付けるジャンプマークとは異なります)。 • 文字列の前には、1つのブロック番号やジャンプマークのいずれか、または両方のみを設定できます。プログラム命令は設定できません。

4.2 サブプログラム機能

<ターゲットブロックの後のブロック>:	プログラム処理を、パラメータ<ターゲットブロック>で指定したブロックから続行するか、その次のブロックから続行するかを指定します。	
	タイプ:	INT
	値:	0 復帰ジャンプのジャンプ先は、パラメータ<ターゲットブロック>で指定したブロックです。
		> 0 復帰ジャンプのジャンプ先は、パラメータ<ターゲットブロック>で指定したブロックの次のブロックです。
<復帰ジャンプレベルの数>:	ターゲットブロックを検索してプログラムの処理を続行するために、ジャンプして戻るべき(復帰ジャンプ)プログラムレベルの数を指定します。	
	タイプ:	INT
	値:	1 プログラムが「現在のプログラムレベル-1」で再開されます(パラメータなしの RET と同様)。
		2 プログラムが「現在のプログラムレベル-2」で再開されます。つまり、レベルを1つスキップします。
		3 プログラムが「現在のプログラムレベル-3」で再開されます。つまり、レベルを2つスキップします。
		...
<プログラムの先頭への復帰ジャンプ>:	値の範囲:	1 ... 15
	メインプログラムへの復帰ジャンプの場合、有効な ISO 系言語モード でプログラムをプログラムの先頭から続行するかどうかを指定します。	
	タイプ:	BOOL
	値:	1 復帰ジャンプのジャンプ先がメインプログラムの中にあり、そこで ISO 系言語モード が有効である場合は、プログラムはプログラムの先頭に分岐します。

注記

ターゲットブロック検索を指定する文字列を含むサブプログラム復帰ジャンプの場合は、最初に、呼び出しプログラムのジャンプマークの検索が必ず実行されます。

文字列を使用してジャンプ先を一義的に定義する場合は、ジャンプマーク名称と一致する文字列は使用できません。これは、一致する文字列を使用すると、サブプログラム復帰ジャンプのジャンプ先が常に文字列でなく、ジャンプマークとなるためです(例 2 を参照してください)。

制約事項**複数のプログラムレベルでの復帰ジャンプ**

複数のプログラムレベルにわたる復帰ジャンプをおこなうときは、個々のプログラムレベルの SAVE 命令が使用されます。

複数のプログラムレベルの復帰ジャンプのときに、モーダルサブプログラムが有効な場合、およびスキップするプログラムのいずれかで選択解除命令 MCALL をモーダルサブプログラムに対してプログラム指令している場合は、モーダルサブプログラムがそのまま有効になります。

通知
プログラム指令エラー 複数のプログラムレベルでの復帰ジャンプの場合、処理が必要なモーダル設定で続行されることを必ず確認してください。これは、適切なメインブロックをプログラム指令するなどの方法でおこなうことができます。

外部メモリからのプロセス

「外部メモリからの実行」機能を介して外部プログラムメモリからメインプログラムとサブプログラムを実行する場合は、RET(...)でプログラムした復帰先を再ロードメモリ内に置いてください。その他の場合、ジャンプ先は見つからず、プログラムは中止されてアラーム 14000 が出力されます。

注記

プログラムしたジャンプ命令に関して、制限なく外部プログラムを実行できるようにするには、機能「外部メモリからの実行」の代わりにオプション「外部メモリからの実行(EES)」の使用を推奨します。

4.2 サブプログラム機能

例

例 1:ASUB 実行後のメインプログラムの再開

プログラミング	コメント
N10010 CALL "UP1"	; プログラムレベル 0 (メインプログラム)
N11000 PROC UP1	; プログラムレベル 1
N11010 CALL "UP2"	
N12000 PROC UP2	; プログラムレベル 2
...	
N19000 PROC ASUP	; プログラムレベル 3 (ASUB の実行)
...	
N19100 RET("N10900", , \$P_STACK)	; メインプログラムへサブプログラム復帰ジャンプ
	; \$P_STACK:実際のプログラムレベル
N10900	; メインプログラムのターゲットブロック
N10910 MCALL	; モーダルサブプログラム呼び出しを解除します
N10920 G0 G60 G40 M5	; 追加モーダル設定を初期化します

例 2 :ターゲットブロック検索を指定する文字列(string>)

メインプログラム

プログラムコード	コメント
PROC MAIN_PROGRAM	
N1000 DEF INT iVar1=1, iVar2=4	
N1010 ...	
N1200 subProg1	; サブプログラム「subProg1」を呼び出します。
N1210 M2 S1000 X10 F1000	
N1220	
N1400 subProg2	; サブプログラム「subProg2」を呼び出します。
N1410 M3 S500 Y20	
N1420 ..	
N1500 lab1:iVar1=R10*44	
N1510 F500 X5	
N1520 ...	
N1550 subprog1:G1 X30	; ここで、「subProg1」がジャンプマークとして定義されます。
N1560 ...	
N1600 subProg3	; サブプログラム「subProg3」を呼び出します。
N1610 ...	
N1900 M30	

サブプログラム subProg1:

プログラムコード	コメント
PROC subProg1	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg2")	; メインプログラムの N1400 ブロックへ復帰ジャンプ

サブプログラム subProg2:

プログラムコード	コメント
PROC subProg2	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("iVar1")	; メインプログラムの N1500 ブロックへ復帰ジャンプ

サブプログラム subProg3:

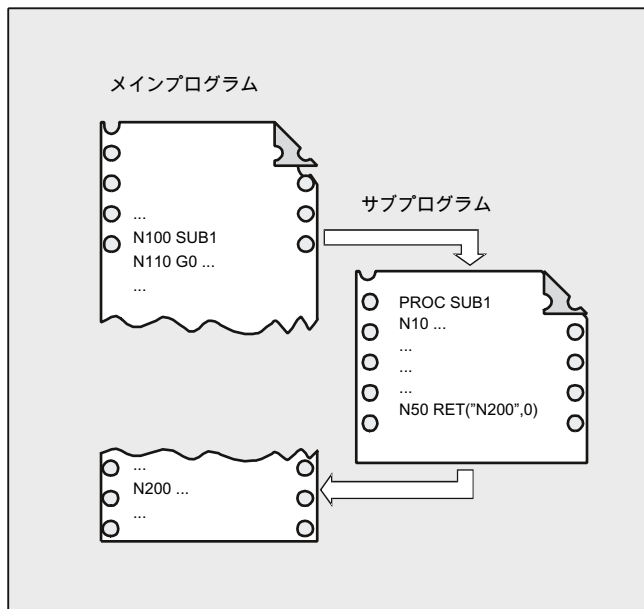
プログラムコード	コメント
PROC subProg3	
N2000 R10=R20+100	
N2010 ...	
N2200 RET("subProg1")	; メインプログラムの N1550 ブロックへ復帰ジャンプ

関連情報

次の図は、復帰ジャンプパラメータのさまざまな働きを示しています。

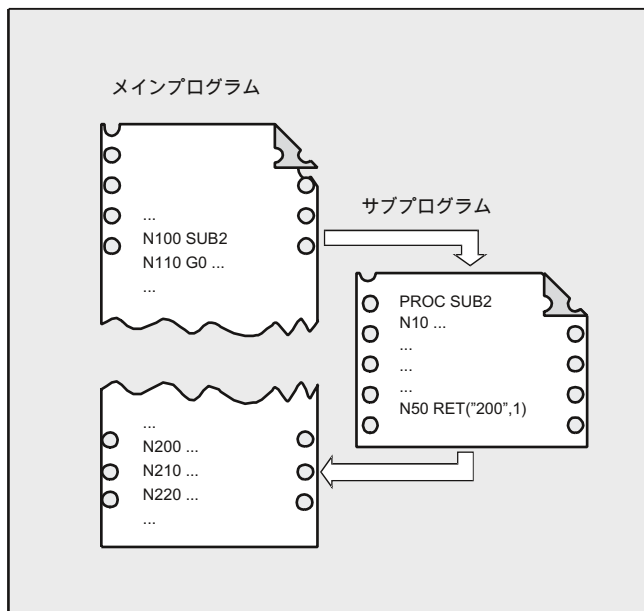
4.2 サブプログラム機能

1. <ターゲットブロック> = "N200"、<ターゲットブロックの後のブロック> = 0



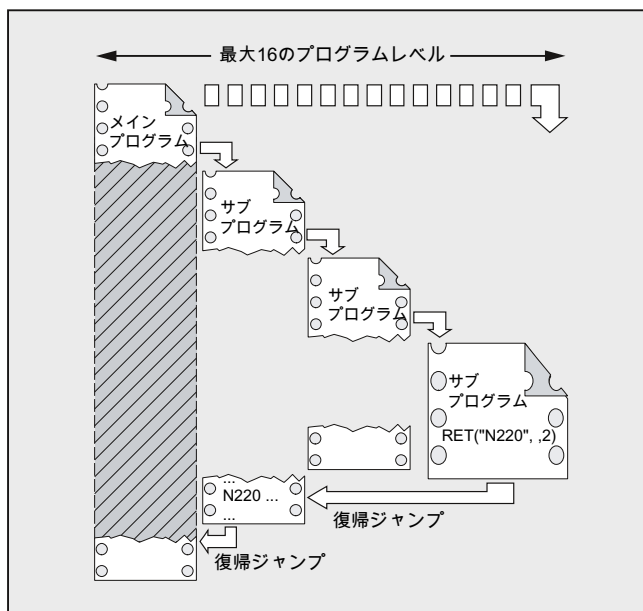
RET 命令の後に、プログラムの実行をメインプログラムの N200 ブロックで続行します。

2. <ターゲットブロック> = "N200"、<ターゲットブロックの後のブロック> = 1



RET 命令の後に、プログラムの実行を、メインプログラムの N200 ブロックの次の(N210)ブロックで続行します。

3. <ターゲットブロック> = "N220"、<復帰ジャンプレベルの数> = 2



RET 命令の後に、2つのプログラムレベルをジャンプし、プログラムの実行を N220 ブロックで続行します。

4.2.2.11 パラメータ設定可能なサブプログラム復帰ジャンプ(RETB ...)

通常、復帰ジャンプは、RETB 命令を使用してサブプログラムから呼び出しプログラムに戻ります。そして、サブプログラム呼び出しの次のプログラム行から処理が続行されます。ただし、次の例のように、プログラムの処理を別の位置から続行することもできます。

- (輪郭の記述後に)ISO 系言語モードの旋削サイクルの呼び出し後に、プログラムの実行を再開します。
- エラー処理のために、(ASUB 後でも)任意のサブプログラムレベルからメインプログラムに戻ります。
- コンパイルサイクルと ISO 系言語モードで、特別な用途のために複数のプログラムレベルをジャンプして復帰します。

これを達成するには、RETB 命令を追加のパラメータを付けてプログラム指令します。

検索方向

パラメータ<ターゲットブロック>を指定した場合、最初に呼び出しブロックの後のブロックに復帰ジャンプします。次に、復帰ジャンプ先のプログラムの先頭方向にターゲットの検索がおこなわれます。見つからなかった場合は、プログラムの末尾方向に検索がおこなわれます。

4.2 サブプログラム機能

構文

```
RETB("<ターゲットブロック>")  
RETB("<ターゲットブロック>",<ターゲットブロックの後のブロック>)  
RETB("<ターゲットブロック>",<ターゲットブロックの後のブロック> <復帰ジャンプ  
レベルの数>)  
RETB("<ターゲットブロック>",<復帰ジャンプレベルの数>)  
RETB("<ターゲットブロック>",<ターゲットブロックの後のブロック>,<復帰ジャンプ  
レベルの数>,  
<プログラムの先頭への復帰ジャンプ>)  
RETB( , ,<復帰ジャンプレベルの数>,<プログラムの先頭への復帰ジャンプ>)
```

意味

RETB:	サブプログラム終了	
<ターゲットブロック>:	パラメータにおいて、プログラムの実行を再開するブロックをジャンプ先として指定してください。 パラメータ<復帰ジャンプレベルの数>をプログラム指令していない場合は、ジャンプ先は実行中のサブプログラムを呼び出したプログラムになります。 指定できるデータには以下のものがあります。	
	<ブロック番号> >	ターゲットブロックの番号です。 ブロック番号の検索は、復帰ジャンプ先プログラムで、最初にプログラムの先頭方向に実行されます。
	<ジャンプマーク>	復帰ジャンプ先プログラムに存在するジャンプマークにしてください。 ジャンプマークの検索は、復帰ジャンプ先プログラムで、最初にプログラムの先頭方向に実行されます。
	<文字列>	復帰ジャンプ先プログラムに存在する文字列にしてください(プログラム名称や変数名称など)。 文字列の検索は、復帰ジャンプ先プログラムで、最初にプログラムの先頭方向に実行されます。 文字列をプログラム指令するときは、以下の規則が適用されます。 • 末尾にブランクを付けます(末尾に「:」を付けるジャンプマークとは異なります)。 • 文字列の前には、1つのブロック番号やジャンプマークのいずれか、または両方のみを設定できます。プログラム命令は設定できません。

4.2 サブプログラム機能

<ターゲットブロックの後のブロック>:	プログラム処理を、パラメータ<ターゲットブロック>で指定したブロックから続行するか、その次のブロックから続行するかを指定します。		
	タイプ:	INT	
	値:	0	復帰ジャンプのジャンプ先は、パラメータ<ターゲットブロック>で指定したブロックです。
		> 0	復帰ジャンプのジャンプ先は、パラメータ<ターゲットブロック>で指定したブロックの次のブロックです。
<復帰ジャンプレベルの数>:	ターゲットブロックを検索してプログラムの処理を続行するために、ジャンプして戻るべき(復帰ジャンプ)プログラムレベルの数を指定します。		
	タイプ:	INT	
	値:	1	プログラムが「現在のプログラムレベル-1」で再開されます(パラメータなしの RET と同様)。
		2	プログラムが「現在のプログラムレベル-2」で再開されます。つまり、レベルを1つスキップします。
		3	プログラムが「現在のプログラムレベル-3」で再開されます。つまり、レベルを2つスキップします。
		...	
値の範囲:	1 ... 15		
<プログラムの先頭への復帰ジャンプ>:	メインプログラムへの復帰ジャンプの場合、有効な ISO 系言語モード でプログラムをプログラムの先頭から続行するかどうかを指定します。		
	タイプ:	BOOL	
	値:	1	復帰ジャンプのジャンプ先がメインプログラムの中にあり、そこで ISO 系言語モード が有効である場合は、プログラムはプログラムの先頭に分岐します。

注記

ターゲットブロック検索を指定する文字列を含むサブプログラム復帰ジャンプの場合は、最初に、呼び出しプログラムのジャンプマークの検索が必ず実行されます。

文字列を使用してジャンプ先を一義的に定義する場合は、ジャンプマーク名称と一致する文字列は使用できません。これは、一致する文字列を使用すると、サブプログラム復帰ジャンプのジャンプ先が常に文字列でなく、ジャンプマークとなるためです(例 2 を参照してください)。

制約事項**複数のプログラムレベルでの復帰ジャンプ**

複数のプログラムレベルにわたる復帰ジャンプをおこなうときは、個々のプログラムレベルの SAVE 命令が使用されます。

複数のプログラムレベルの復帰ジャンプのときに、モーダルサブプログラムが有効な場合、およびスキップするプログラムのいずれかで選択解除命令 MCALL をモーダルサブプログラムに対してプログラム指令している場合は、モーダルサブプログラムがそのまま有効になります。

通知
プログラム指令エラー 複数のプログラムレベルでの復帰ジャンプの場合、処理が必要なモーダル設定で続行されることを必ず確認してください。これは、適切なメインブロックをプログラム指令するなどの方法でおこなうことができます。

外部メモリからのプロセス

「外部メモリからの実行」機能を介して外部プログラムメモリからメインプログラムとサブプログラムを実行する場合は、RETB(...)でプログラムした復帰先を再ロードメモリ内に置いてください。その他の場合、ジャンプ先は見つからず、プログラムは中止されてアラーム 14000 が出力されます。

注記

プログラムしたジャンプ命令に関して、制限なく外部プログラムを実行できるようにするには、機能「外部メモリからの実行」の代わりにオプション「外部メモリからの実行(EES)」の使用を推奨します。

4.2 サブプログラム機能

例

プログラムコード	コメント
EXAMPLE.MPF	
...	
N3000 START_CYC (パラメータ 1, パラメータ 2,	
...)	
N3010 TECH_CYC1 (パラメータ 1, パラメータ 2,	
...)	
N3020 TECH_CYC2 (パラメータ 1, パラメータ 2,	
...)	
N3030 TECH_CYC3 (パラメータ 1, パラメータ 2,	
...)	
N3040 END_CYC (パラメータ 1, パラメータ 2, ...)	
N3040 END_CYC (パラメータ 1, パラメータ 2, ...)	
N3050 ...	
N4500 START_CYC (パラメータ 11, パラメータ	
12, ...)	
N4510 ...	
N4590 END_CYC (パラメータ 11, パラメータ	
12, ..)	
N5000 ...	
...	
N6000 M30	

プログラムコード	コメント
PROC END_CYC (...)	; メインプログラム、N3040 行で呼び出し
N10000 ...	
N15000 if status == 1	
N15010 RETB ("START_CYC")	; 呼び出しプログラム EXAMPLE.MPF へ復帰ジャンプ
	; 文字列「START_CYC」を検索
	; 検索方向: プログラムの先頭方向へ
	; プログラム処理を N3000 行から続行します
N15020 ENDIF	
N15030 if status == 0	
N15040 RET	; 呼び出しプログラム EXAMPLE.MPF へ復帰ジャンプ
	; プログラム処理を N3050 行から続行します
N15050 ENDIF	
N16000 RET ("START_CYC")	; 呼び出しプログラム EXAMPLE.MPF へ復帰ジャンプ
	; 文字列「START_CYC」を検索
	; 検索方向: プログラムの末尾方向へ
	; プログラム処理を N4500 行から続行します
N17060 RETB	; 呼び出しプログラム EXAMPLE.MPF へ復帰ジャンプ
	; プログラム処理を N3050 行から続行します
	; パラメータなしの RETB は RET と同じです

4.2.3 サブプログラムの呼び出し

4.2.3.1 パラメータ転送を含まないサブプログラム呼び出し

サブプログラムは、アドレス L とサブプログラム番号で、またはプログラム名称を指定して呼び出します。

メインプログラムもサブプログラムとして呼び出すことができます。この場合は、メインプログラムに設定したプログラムの終点 M2 または M30 を M17(呼び出しプログラムへの戻りがあるプログラムの終了)として使用します。

注記

したがって、サブプログラムをメインプログラムとして起動することもできます。

制御装置での検索方法:

*_MPF があるかどうか

*_SPF があるかどうか

つまり、呼び出すサブプログラムの名称がメインプログラムの名称と同じである場合は、呼び出しを発行したメインプログラムが再度呼び出されます。これは、一般的には不必要な動作のため、サブプログラムとメインプログラムに固有の名称を割り当てて回避してください。

注記

パラメータ転送が不要なサブプログラムは、初期化ファイルから呼び出すこともできます。

構文

L<番号>/<プログラム名称>

注記

サブプログラム呼び出しは常に、個別の NC ブロックでプログラム指令してください。

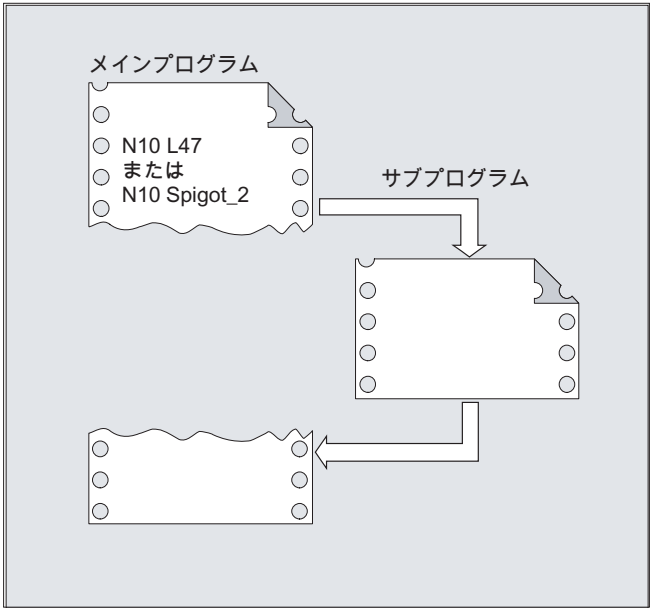
4.2 サブプログラム機能

意味

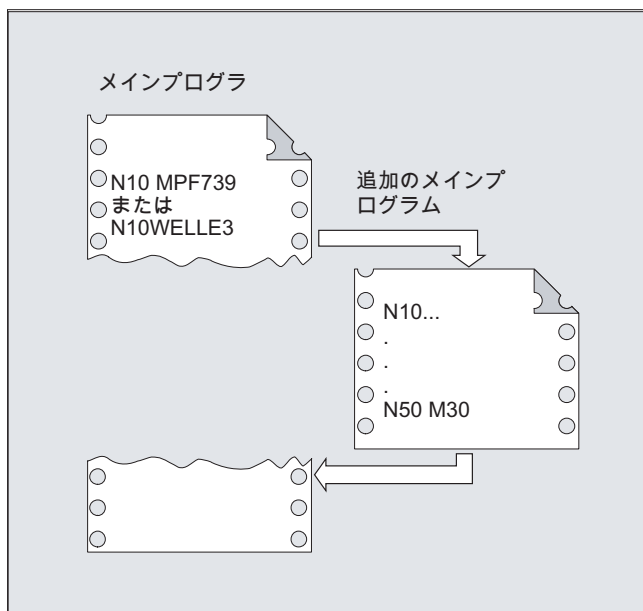
L:	サブプログラム呼び出しのアドレス	
<番号>:	サブプログラムの名称	
	タイプ:	INT
	値:	7桁までの数値 重要: 名称は、先頭のゼロに意味があります(⇒ L123、L0123、および L00123 は 3 個の異なるサブプログラムです)。
<プログラム名称>:	サブプログラム(またはメインプログラム)の名称	

例

例 1:パラメータ転送を含まないサブプログラム呼び出し



例 2:メインプログラムをサブプログラムとして呼び出し



下記も参照

パラメータ転送を含まないサブプログラム (ページ 563)

4.2.3.2 パラメータ転送を含むサブプログラム呼び出し(EXTERN)

パラメータ転送を含むサブプログラム呼び出しの場合は、変数または値を直接転送できます (ただし、VAR パラメータを除きます)。

パラメータ転送を含むサブプログラムは、メインプログラムのなかで呼び出す前に、メインプログラム(プログラムの先頭など)に EXTERNAL で宣言してください。したがって、サブプログラムの名称と変数タイプは、転送される順に指定します。

通知

混乱の可能性

変数タイプとその転送順序は、サブプログラムの PROC で宣言した定義に一致させてください。メインプログラムとサブプログラムで、異なるパラメータ名称を使用できます。

構文

```
EXTERNAL <プログラム名称>(<タイプ_パラメータ 1>,<タイプ_パラメータ 2>,<タイプ_パラメータ 3>)
...
```

4.2 サブプログラム機能

<プログラム名称> (<値_パラメータ 1>,<値_パラメータ 2>,<値_パラメータ 3>)

注記
サブプログラム呼び出しは常に、個別の NC ブロックでプログラム指令してください。

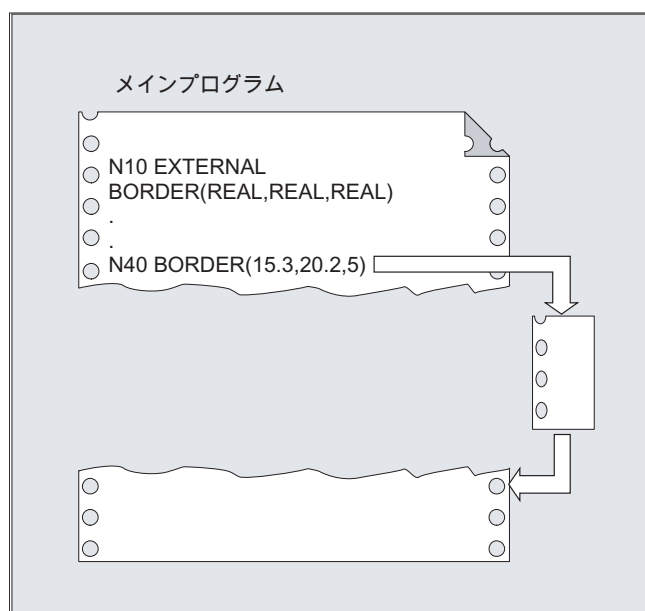
意味

<プログラム名称>:	サブプログラムの名称です。
EXTERNAL:	パラメータ転送を含むサブプログラムを宣言するキーワードです。 注: EXTERNAL は、サブプログラムがワーク、またはグローバルサブプログラムディレクトリにある場合にのみ指定してください。 サイクルは、EXTERNAL として宣言する必要はありません。
<タイプ_パラメータ 1>,<タイプ_パラメータ 2>,<タイプ_パラメータ 3>:	転送順に転送されるパラメータの変数タイプです。
<値_パラメータ 1>,<値_パラメータ 2>,<値_パラメータ 3>):	転送されるパラメータの変数値です。

例

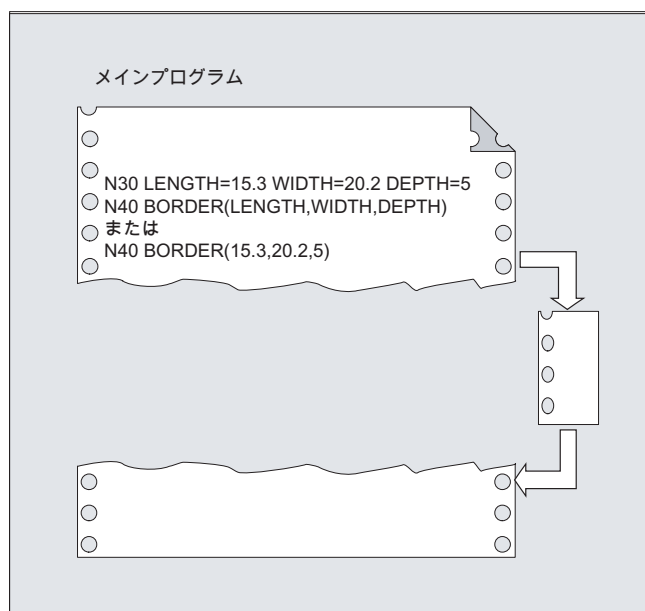
例 1:宣言があるサブプログラム呼び出し

プログラムコード	コメント
N10 EXTERNAL BORDERS (REAL,REAL,REAL) ... N40 BORDER (15.3,20.2,5)	; サブプログラムを指定します。 ; パラメータ転送を含むサブプログラムを呼び出します。



例 2 :宣言のないサブプログラム呼び出し

プログラムコード	コメント
N10 DEF REAL LENGTH, WIDTH, DEPTH	
N20...	
N30 LENGTH=15.3 WIDTH=20.2 DEPTH=5	
N40 BORDER(LENGTH,WIDTH,DEPTH)	; または:N40 BORDER(15.3,20.2,5)



4.2 サブプログラム機能

下記も参照

値渡しパラメータ転送を含むサブプログラム(PROC) (ページ 564)

参照渡しパラメータ転送を含むサブプログラム(PROC、VAR) (ページ 566)

4.2.3.3 プログラム繰り返し回数(P)

サブプログラムを複数回連続して実行する場合は、サブプログラム呼び出しと一緒にプログラム繰り返し回数を、このブロックのアドレス P へ入力できます。

 注意

プログラム繰り返しとパラメータ転送を含むサブプログラム呼び出し
パラメータは、プログラムが呼び出されたとき、つまり、1 回目の実行時にのみ転送されます。その後のプログラム繰り返しでは、同じパラメータがそのまま有効です。プログラム繰り返し中にパラメータを変更する場合は、サブプログラムのなかに、変更の準備をしてください。

構文

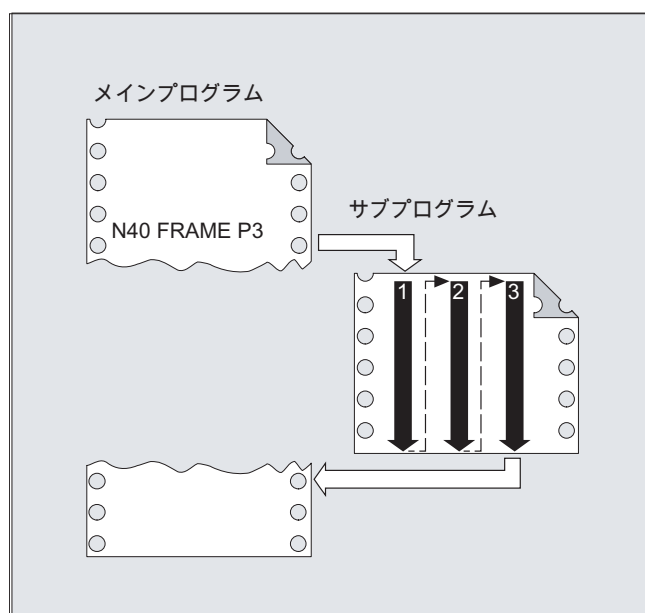
<プログラム名称> P<値>

意味

<プログラム名称>:	サブプログラム呼び出し	
P:	プログラム繰り返しを設定するアドレスです。	
<値>:	プログラム繰り返し回数です。	
	タイプ:	INT
	値の範囲:	1 ... 9999 (符号なし)

例

プログラムコード	コメント
...	
N40 FRAME P3	; FRAME のサブプログラムが 3 回、連続して実行されます。
...	



4.2.3.4 モーダルサブプログラム呼び出し(MCALL)

モーダルサブプログラム呼び出し `MCALL<プログラム名称>` では、指定したサブプログラムはすぐには呼び出されません。そうではなく、軌跡移動を含む各移動ブロック後にパートプログラムで呼び出しが実行されます。プログラムレベル全体で実行することもできます。

注記

プログラムが実行される場合、最後のモーダルサブプログラム呼び出し `MCALL<プログラム名称>` のみが常に有効になります。最新のモーダルサブプログラム呼び出しが、それまでに有効であった呼び出しと置き換わります。

パラメータをサブプログラムに転送する場合、パラメータは呼び出し `MCALL(<プログラム名称>(Par1, Par2, ...))` でのみ転送されます。

注記

サブプログラムの最初のデータ定義領域は、プログラム指令された `MCALL` 呼び出しでブロックを実行するときに 1 回だけ実行されます。それ以降のサブプログラム呼び出しでは、移動ブロックの後、データ定義領域は実行されなくなります。つまり、ローカルユーザー変数(LUD)の定義で割り当てられた値は最初の呼び出し後は使用できなくなりますが、サイクルで最後書き込まれた値は使用できます。変数は最初の呼び出し後に再作成されませんが、前の呼び出しの最後の値を維持します。この動作を回避するために、LUD の定義と値の割り付けを切り離し、データ定義領域の後で専用のブロックに値の割り付けをプログラム指令することをお勧めします。

4.2 サブプログラム機能

通知
軌跡移動なしでのモーダルサブプログラム呼び出し 以下の事例では、モーダルサブプログラムも、軌跡移動のプログラム指令なしで呼び出されます。 • G0 または G1 が有効な場合に、アドレス S または F をプログラム指令した場合。 • G0 または G1 を、このブロックに単独で、または別の G 命令と一緒にプログラムした場合。

構文

MCALL <プログラム名称>
...
MCALL

意味

MCALL <プログラム名称>:	「モーダルサブプログラム呼び出し」機能を有効にします。
<プログラム名称>:	サブプログラムの名称です。
MCALL:	「モーダルサブプログラム呼び出し」機能を解除するには、プログラム名称の指定を含まない MCALL を使用します。

境界条件

ASUB

パートプログラムの処理が ASUB によって中断される場合(「割り込みルーチン(ASUB) (ページ 615)」の章を参照してください)、この ASUB でモーダルサブプログラム呼び出しは実行されません。

ただし、ASUB が「リセット」チャネル状態で開始された場合は、モーダルサブプログラム呼び出しでの動作は、通常のパートプログラムと同じになります。

工具交換サイクル

「モーダルサブプログラム呼び出し」機能が工具交換サイクル中に選択解除された場合、検索 ASUB により、ブロック検索後であっても自動的に、またはオーバストアにより手動で、工具交換サイクルが呼び出されることに注意してください。この場合、「モーダルサブプログラム呼び出し」機能は選択解除しないでください。選択解除すると、正しい検索結果が

得られません。そのため、工具交換サイクルでの「モーダルサブプログラム呼び出し」機能の選択解除は、以下のようにプログラム指令することをお勧めします。

プログラムコード	コメント
...	
IF \$AC_ASUB == 0	; 呼び出しは検索 ASUB またはオーバストアにより実行されません。
MCALL	; 「モーダルサブプログラム呼び出し」機能を解除します。
ENDIF	
...	

例

例 1

プログラムコード	コメント
N10 G0 X0 Y0	
N20 MCALL L70	; L70 のモーダルサブプログラム呼び出しを有効にします。
N30 X10 Y10	; X10 Y10 へのアプローチ後に、L70 が呼び出されます。
N40 X20 Y20	; X20 Y20 へのアプローチ後に、L70 が呼び出されます。
...	
N100 MCALL	; 「モーダルサブプログラム呼び出し」機能を解除します。
N110 X0 Y0	; X0 Y0 にアプローチし、L70 は呼び出され ません 。

例 2

プログラムコード
N10 G0 X0 Y0
N20 MCALL L70
N30 L80

この例では、プログラム指令された軌跡軸がある次の NC ブロックがサブプログラム L80 にあります。L70 は L80 で呼び出されます。

4.2 サブプログラム機能

4.2.3.5 間接サブプログラム呼び出し(CALL)

プログラムの特定の位置での主な条件によっては、異なる複数のサブプログラムの呼び出しができます。サブプログラムの名称は、**STRING** タイプの変数に保存されます。サブプログラム呼び出しは、**CALL** と変数名称を使用して実行されます。

注記

間接サブプログラム呼び出しは、パラメータ転送を含まないサブプログラムに対してのみ使用できます。直接サブプログラム呼び出しをおこなうためには、**STRING** 定数に名称を保存してください。

構文

CALL <プログラム名称>

意味

CALL:	間接サブプログラム呼び出しの命令です。	
<プログラム名称>:	サブプログラムの名称(変数または定数)です。	
	タイプ:	STRING

例

STRING 定数による直接呼び出し:

プログラムコード	コメント
...	
CALL "/_N_WKS_DIR/_N_SUBPROG_WPD/_N_PART1_SPF"	; CALL によりサブプログラム PART1 を直接呼び出します。
...	

変数による間接呼び出し:

プログラムコード	コメント
...	
DEF STRING[100] PROGNAME	:変数を定義します。
PROGNAME="/_N_WKS_DIR/_N_SUBPROG_WPD/_N_PART1_SPF"	; サブプログラム PART1 を PROGNAME 変数に割り当てます。

プログラムコード	コメント
CALL PROGNAME	; CALL と PROGNAME 変数によりサブプログラム PART1 の間接呼び出しをします。
...	

4.2.3.6 呼び出しプログラム要素を指定した間接サブプログラム呼び出し(CALL BLOCK ... TO ...)

CALL とキーワードの組み合わせ BLOCK ... TO を使用して、サブプログラムを間接的に呼び出し、開始ラベルと終了ラベルで指定されたプログラム区間を実行します。

構文

```
CALL <プログラム名称> BLOCK <開始ラベル> TO <終了ラベル>
CALL BLOCK <開始ラベル> TO <終了ラベル>
```

意味

CALL:	間接サブプログラム呼び出しの命令です。	
<プログラム名称>:	実行するプログラム区間を含むサブプログラムの名称(変数または定数) (指定はオプション)。	
	タイプ:	STRING
	注: <プログラム名称>をプログラム指令していない場合は、<開始ラベル>と<終了ラベル>が示すプログラム区間は、実行中のプログラムで検索され、実行されます。	
BLOCK ... TO ... :	プログラム区間を間接的に実行するためのキーワードの組み合わせ	
<開始ラベル>:	実行するプログラム区間の先頭を参照する変数	
	タイプ:	STRING
<終了ラベル>:	実行するプログラム区間の末尾を参照する変数	
	タイプ:	STRING

4.2 サブプログラム機能

例

メインプログラム

プログラムコード	コメント
...	
DEF STRING[20] STARTLABEL, ENDLABEL	; 開始ラベルと終了ラベルの変数を定義します。
STARTLABEL="LABEL_1"	
ENDLABEL="LABEL_2"	
...	
CALL "CONTUR_1" BLOCK STARTLABEL TO ENDLABEL	; 間接サブプログラム呼び出し、および呼び出しプログラム区間に関連する識別子です。
...	

サブプログラム:

プログラムコード	コメント
PROC CONTUR_1 ...	
LABEL_1	; 開始ラベル:実行するプログラム区間の開始
N1000 G1 ...	
...	
LABEL_2	; 終了ラベル:実行するプログラム区間の終了
...	

4.2.3.7 ISO 言語で作成したプログラムの間接呼び出し(ISOCALL)

ISO 言語で作成したプログラムは、間接プログラム呼び出し ISOCALL を使用して呼び出すことができます。この場合、マシンデータで設定した ISO モードが有効です。オリジナルの実行モードは、プログラムの終了時に再び有効になります。ISO モードをマシンデータで設定していない場合は、サブプログラムを SIEMENS モードで呼び出します。

ISO モードに関する追加情報:機能マニュアル G コード言語

構文

ISOCALL <プログラム名称>

意味

ISOCALL:	マシンデータで設定した ISO モードを有効にする、間接サブプログラム呼び出しのキーワードです。
<プログラム名称>:	ISO 言語で作成したプログラムの名称(変数または定数、STRING タイプ)です。

例:サイクルプログラミングによる ISO モードからの輪郭の呼び出し

プログラムコード	コメント
0122_SPF	; ISO モードの輪郭記述
N1010 G1 X10 Z20	
N1020 X30 R5	
N1030 Z50 C10	
N1040 X50	
N1050 M99	
N0010 DEF STRING[5] PROGNAME = "0122"	; SIEMENS パートプログラム (サイクル)
...	
N2000 R11 = \$AA_IW[X]	
N2010 ISOCALL PROGNAME	
N2020 R10 = R10+1	; ISO モードでプログラム 0122.spf を実行します
...	
N2400 M30	

4.2.3.8 パス指定とパラメータによるサブプログラム呼び出し(PCALL)

PCALL を使用すると、絶対パスとパラメータ転送によりサブプログラムを呼び出すことができます。

構文

PCALL <パス/プログラム名称> (<パラメータ 1>, ..., <パラメータ n>)

4.2 サブプログラム機能

意味

PCALL:	絶対パス名称によるサブプログラム呼び出しのキーワードです。
<パス/プログラム名称>:	サブプログラム名称を含む絶対パスデータ パスデータに関する規則については、「プログラムメモリファイルのアドレス指定 (ページ 632)」を参照してください。 絶対パス名称を指定しない場合は、PCALL が、プログラム識別子による標準のサブプログラム呼び出しと同様に動作します。 プログラム名称は、接頭語を付けずにファイル識別子なしで指定します。接頭語やファイル識別子を使用してプログラム名称をプログラム指令する場合は、EXTERN 命令を使用して、接頭語とファイル識別子を明示的に宣言してください。
<パラメータ 1>, ...:	サブプログラムの PROC 命令に従った実パラメータです。

例

プログラムコード

```
PCALL/_N_WKS_DIR/_N_SHAFT_WPD/SHAFT (パラメータ 1, パラメータ 2, ...)
```

4.2.3.9 サブプログラム呼び出しの拡張検索パス(CALLPATH)

サブプログラム呼び出しの検索パスを、CALLPATH 命令を使用して拡張できます。これは、サブプログラムの完全な絶対パス名称を指定しなくても、選択していないワークディレクトリから直接、サブプログラムも呼び出すことができるということです。

グローバルなサブプログラムを保存するために外部プログラムメモリ上の別のディレクトリを使用する場合、EES モードでは別のアプリケーションオプション「EES without GDIR」(GDIR を含まない EES)を使用できます。この場合、CALLPATH を使用して、検索パスをこのサブプログラムディレクトリによって拡張できます。

拡張検索パスは、ユーザーサイクル(_N_CUS_DIR)の項目の前に作成されます。

拡張検索パスは、以下の事象が発生すると、再び選択解除されます。

- ブランクを含む CALLPATH
- パラメータのない CALLPATH

- パートプログラムの終了
- リセット

構文

CALLPATH("<パス名称>")

意味

CALLPATH:	プログラム指令可能な拡張検索パスのキーワードです。 個別のパートプログラム行にプログラム指令します。
<パス名称>:	定数または変数、 STRING タイプ。 検索パスの拡張に使用するディレクトリの絶対パス名称を含みます。 パスデータに関する規則については、「プログラムメモリファイルのアドレス指定 (ページ 632)」を参照してください。

例

検索パスを特定のワークディレクトリによって拡張します。

```
プログラムコード
...
CALLPATH ("/_N_WKS_DIR/_N_MYWPD_WPD")
...
```

これは、次の検索パスが設定されることを意味します(項番 5 が新しい設定です)。

1. 現在のディレクトリ/名称
2. 現在のディレクトリ/名称_SPF
3. 現在のディレクトリ/名称_MPF
4. //NC:/_N_SPF_DIR/名称_SPF
5. /_N_WKS_DIR/_N_MYWPD_WPD/名称_SPF
6. /N_CUS_DIR/名称_SPF
7. /_N_CMA_DIR/名称_SPF
8. /_N_CST_DIR/名称_SPF

4.2 サブプログラム機能

必要条件

- CALLPATH は、プログラム指令パス名称が実在するかどうかをチェックします。エラーが発生した場合は、パートプログラムの実行がブロック修正アラーム 14009 により中断されます。
- CALLPATH は、INI ファイルにプログラム指令することもできます。この場合は、INI ファイルの処理時のみ有効となります(WPD-INI ファイル、または 1 番目のチャンネル _N_CH1_UFR_INI のフレームなど、NC の有効データの初期化プログラム)。検索パスは再度リセットされます。

4.2.3.10 外部サブプログラムの実行(EXTCALL)

パートプログラムを外部メモリからロードして EXTCALL 命令で実行できます。

外部メモリとしては、以下が使用できます。

- ローカルドライブ
- ネットワークドライブ
- USB ドライブ

注記

USB ドライブ上の外部プログラムの処理には、操作パネルまたは TCU の USB インターフェースのみインターフェースとして使用できます。

通知

USB フラッシュメモリが原因の工具/ワークの損傷

外部サブプログラムの実行には USB フラッシュメモリを使用しないことを推奨します。サブプログラムの実行時に、接触異常、故障、トリガによる中止、または意図しないプラグ接続解除による USB フラッシュメモリへの通信割り込みが発生すると、直ちに加工が停止します。工具かワークまたは両方が損傷する恐れがあります。

外部プログラムのパスの初期設定

外部プログラムディレクトリのパスは、次のセッティングデータを使用して設定しておくことができます。

SD42700 \$SC_EXT_PROG_PATH

このパスと、EXTCALL 呼び出しで指定したプログラムのパスと識別子で、呼び出すプログラムの完全なパスが形成されます。

注記

プログラムのパスを EXTCALL 呼び出しのみで指定する場合は、SD42700 を空にしてください。

注記**パラメータ**

外部プログラムが呼び出されても、パラメータをそこに転送することはできません。

構文

EXTCALL ("<パス/><プログラム名称>")

意味

EXTCALL:	外部サブプログラムを呼び出す命令	
"<パス/><プログラム名称>":	STRING タイプの定数/変数です。	
	<パス/>:	絶対パスまたは相対パスの指定 (任意選択)
	<プログラム名称>:	プログラム名称は、接頭語「_N_」を付けずに指定します。 文字「_」または「.」を使用して、プログラム名称にファイル拡張子(「MPF」、「SPF」)を付けることができます(任意選択)。 例: "SHAFT" "SHAFT_SPF" "SHAFT.SPF"

4.2 サブプログラム機能

パスの指定:省略名称

次のような短縮名称でパスを指定できます。

- ローカルドライブ:"LOCAL_DRIVE:"
- メモリカード:"CF_CARD:"
- USB ドライブ(操作パネル):"USB:"

または、短縮名称「CF_CARD:」と「LOCAL_DRIVE:」を使用可能です。

例

ローカルドライブからの実行

「MAIN.MPF」メインプログラムが NC メモリに格納され、実行するために選択されます。

サブプログラム「SP_1」

外部サブプログラム「SP_1.SPF」または「SP_1.MPF」は、ローカルドライブのディレクトリ「/user/sinumerik/data/prog/WKS.DIR/WST1.WPD」にあります。

外部プログラムディレクトリのパスは、以下を使用して設定されます。

SD42700 \$SC_EXT_PROG_PATH = LOCAL_DRIVE:WKS.DIR/WST1.WPD

注記

外部サブプログラム呼び出し用のパスの指定

- 初期設定なし:「LOCAL_DRIVE:WKS.DIR/WST1.WPD/SP_1」
- 初期設定あり:「SP_1」

サブプログラム「SP_2」

外部サブプログラム「SP_2.SPF」または「SP_2.MPF」は、USB ドライブの WKS.DIR / WST1.WPD ディレクトリにあります。外部プログラムディレクトリへの初期パス設定がサブプログラム「SP_1」のパスで使用され、メインプログラムでは再作成もされません。そのため、サブプログラム「SP_2」を呼び出すときには完全なパスを指定する必要があります。

メインプログラム「MAIN」

プログラムコード
N010 PROC MAIN
N020 ...
N030 EXTCALL("SP_1")
N030 EXTCALL("USB:WKS.DIR/WST1.WPD/SP_2")
N050 ...

プログラムコード

N060 M30

詳細

絶対パス名での EXTCALL 呼び出し

指定したパスにサブプログラムがある場合は、EXTCALL 呼び出しで実行されます。指定したパスにサブプログラムがない場合は、EXTCALL 呼び出しによるプログラムの実行は中止されます。

相対パス名付き/パス名なしでの EXTCALL 呼び出し

相対パス名付きまたはパス名なしで EXTCALL が呼び出された場合、使用可能なプログラムメモリが次のように検索されます。

1. SD42700 \$SC_EXT_PROG_PATH にパス名が設定されている場合は、EXTCALL 呼び出しで指定されたデータ(プログラム名称または相対パス名による)の検索が最初におこなわれ、このパスから開始されます。絶対パスは、次の文字を結合して得られます。
 - SD42700 \$SC_EXT_PROG_PATH の初期パス指定
 - 区切り文字「/」
 - EXTCALL 命令でのパス指定とサブプログラム名称
2. サブプログラムが 1 で見つからなかった場合は、ユーザーメモリのディレクトリが検索されます。

サブプログラムが最初に検出されたとき、検索が終了します。サブプログラムが見つからない場合は、EXTCALL 呼び出しによるプログラムの実行は中止されます。

4.2 サブプログラム機能

調整可能な再ロードメモリ(FIFO バッファ)

外部サブプログラムの実行には、再ロードメモリが必要です。再ロードメモリのサイズは 30KB に設定済みで、(MD18360 MM_EXT_PROG_BUFFER_SIZE を使用して)工作機械メーカーだけが変更できます。

注記

ジャンプ命令を含むサブプログラム

ジャンプ命令(GOTO、GOTOB、CASE、FOR、LOOP、WHILE、REPEAT、IF、ELSE、ENDIF など)を含む外部サブプログラムの場合、ジャンプ先は再ロードメモリ内になければなりません。

この条件は、プログラムの先頭(GOTOS)へのジャンプ命令に関して特に問題となります。プログラムは一般的に大きすぎて、全体が再ロードメモリに適合しないためです。初めて再ロードすると、プログラムの先頭は再ロードメモリから削除されます。プログラムの先頭へのジャンプ命令を実行すると、この機能はジャンプ先を見つけられなくなります。プログラムは中止され、アラーム 14000 が出力されます。

プログラムしたジャンプ命令に関して、制限なく外部サブプログラムを実行できるようにするには、機能「外部サブプログラムからの実行(EXTCALL)」の代わりに機能「外部メモリからの実行(EES)」の使用を推奨します。

注記

ShopMill/ShopTurn プログラム

ファイルの最後に輪郭の説明を追加した場合、ShopMill および ShopTurn プログラムをすべて読み取り専用メモリに保存する必要があります。

外部サブプログラムを並列して実行は、別の再メモリロードが必要です。

リセット/エンドオブプログラム/電源投入

リセットおよび電源投入をおこなうと、外部サブプログラム呼び出しが中断され、関連するロードメモリが消去されます。

「外部メモリからの実行」で選択されたプログラムは、リセット/エンドオブプログラムの後も、「外部メモリからの実行」で選択されたままです。外部プログラムメモリがまだ使用可能であれば、動作は内部で選択したプログラムと同じです。

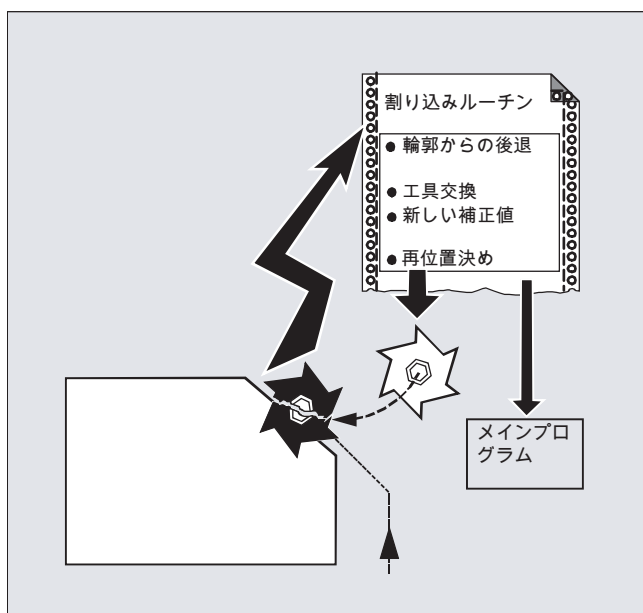
4.3 割り込みルーチン(ASUB)

4.3.1 割り込みルーチンの機能

注記

用語「非同期サブプログラム(ASUB)」と「割り込みルーチン」は、以下の説明で同義的に使用され、同じ機能を表わします。

次の一般的な例は、割り込みルーチンの機能をわかりやすく説明しています。



工具が加工中に破損します。これにより、現在の加工処理を停止する信号が出力され、同時に、割り込みルーチンと呼ばれるサブプログラムが起動されます。割り込みルーチンには、この場合に実行するすべての命令が含まれます。

割り込みルーチンの実行が終了し、機械が動作を続行できるようになると、制御装置がメインプログラムに戻り、割り込み発生点から加工を続行します。ただし、これは、REPOS 命令の場合は異なります(「輪郭での再位置決め (ページ 995)」を参照してください)。

⚠ 注意

干渉の可能性

REPOS 命令をサブプログラムにプログラム指令していない場合は、制御は、割り込まれたブロックの次のブロックの終点に移動します。

4.3 割り込みルーチン(ASUB)

4.3.2 割り込みルーチンの作成

サブプログラムとしての割り込みルーチンの作成

割り込みルーチンは定義ではサブプログラムとして認識されます。

例:

プログラムコード	コメント
PROC LIFT_Z	; プログラム名称「ABHEB_Z」
N10 ...	; この後に NC ブロックが続きます。
...	
N50 M17	; 最後に、プログラムを終了し、メインプログラムに戻ります。

モーダル G 命令の保存(SAVE)

割り込みルーチンは、SAVE で定義して指定できます。

SAVE 属性は、動作中のモーダル G 命令が割り込みルーチンの前に保存され、割り込みルーチンの後に再度有効にされることを意味します(「SAVE メカニズムを持つサブプログラム(SAVE) (ページ 569)」を参照してください)。

つまり、中断点の処理は、割り込みルーチンが完了後に再開できます。

例:

プログラムコード
PROC LIFT_Z SAVE
N10 ...
...
N50 M17

追加の割り込みルーチンの割り当て(SETINT)

SETINT 命令を割り込みルーチン内にプログラム指令すると(「割り込みルーチンの割り当てと起動(SETINT)」 (ページ 617)を参照してください)、追加の割り込みルーチンを起動できます。これらの割り込みルーチンは入力で起動されます。

4.3.3 割り込みルーチンの割り当てと起動(SETINT、PRIO、BLSYNC)

制御装置には複数の高速入力(入力 1 ... 8)があり、これが割り込み(1 ... 8)を開始します。SETINT 命令を使用して、各割り込みに優先度と割り込みルーチンを割り当てることができます。高速入力を設定して割り込みを開始する場合、チャンネルでの処理が中断され、割り込みルーチンが開始されます。

割り込みの優先度

パートプログラムで複数の入力が割り込みに割り当てられている場合、割り込みにそれぞれ異なる優先度を割り当ててください。

割り込みには、優先度の値 1 ... 128 を割り当てることができます。優先度の値 1 は最高優先度で、128 は最低優先度です。

構文

```
SETINT (<n>) <名称>
SETINT (<n>) PRIO=<値> <名称>
SETINT (<n>) PRIO=<値> <名称> BLSYNC
SETINT (<n>) PRIO=<値> <名称> LIFTFAST
```

意味

SETINT (<n>) :	入力<n>は、割り込みルーチン<名称>に割り当てられます。割り当てられた割り込みルーチンは、入力<n> == 1 が検出されると直ちに起動されます。 注: すでにプログラム指令された入力<n>が別の割り込みルーチンに割り当てられた場合、以前の割り当ては無効になります。	
<n>:	入力番号	
	タイプ:	INT
	値の範囲:	1 ... 8
PRIO= :	割り込みの優先度 (任意選択)	
<値>:	優先度の値 (任意選択)	
	タイプ:	INT
	値の範囲:	1 ... 128 (1 ⇒ 最高優先度)
<名称>:	割り込みルーチン(サブプログラム)の名称	

4.3 割り込みルーチン(ASUB)

BLSYNC:	BLSYNC は、割り込みの開始後、まず現在のブロックが完了するまでシステムが待機するようにします。これが完了するまで割り込みルーチンは実行されません。 (任意選択)
LIFTFAST:	LIFTFAST は、割り込みの開始後、まず高速リトラクトをおこなうようにします(「輪郭からの高速リトラクト(SETINT LIFTFAST、ALF) (ページ 621)」の章を参照してください)。これが完了するまで割り込みルーチンは実行されません。 (任意選択)

必要条件

割り込み規則

- 1. すぐに実行できない、または現在すでに処理されているすべての割り込みに対して、追加の割り込み要求が保存されます。この割り込みに対するその他の割り込み要求はすべて失われます。
- 2. 割り込みが現在処理されていて、これよりも優先度の高い追加の割り込みが開始された場合、優先度の低い割り込みが中断されます。優先度の低い割り込みは、優先度の高い割り込みの完了後に続行されます。優先度の高い割り込みが処理されている間、優先度の低い割り込みに対する追加の要求を受信した場合、その要求は保存されます。その他すべての要求は失われます。
- 3. 割り込みが現在処理されていて、優先度の高い追加の割り込みが開始された場合、優先度の低い割り込みは中断されます。優先度の高い割り込みが処理されます。優先度の高い割り込みが開始された場合、現在の割り込みは中断され、優先度の高い割り込みが処理されます。6 つまでの有効な割り込みレベルを使用できます。1 つの割り込みレベルが現在処理され、5 つの割り込みレベルが待機します。有効な割り込みレベルそれぞれに対して、最大 1 つの追加の割り込み要求が保存されます。その他の割り込み要求はすべて失われます。その他の割り込みレベル(割り込みレベル ≥ 7)に対して割り込みが要求された場合も、割り込み要求は失われます。

例

例 1:割り込みルーチンの割り当てと優先度の定義

プログラムコード	コメント
...	
N20 SETINT (3) PRIO=1 ABHEB_Z	; 入力 3 == 1 の場合、割り込みルーチン「ABHEB_Z」を起動します。
N30 SETINT (2) PRIO=2 ABHEB_X	; 入力 2 == 1 の場合、割り込みルーチン「ABHEB_X」を起動します。
...	

4.3 割り込みルーチン(ASUB)

入力が同時に使用可能になった(同時に通電された)場合は、割り込みルーチンが優先度の値の順に実行されます。つまり、最初に「ABHEB_Z」、次に「ABHEB_X」が実行されます。

例 2:割り込みルーチンの新規割り当て

プログラムコード	コメント
...	
N20 SETINT(3) PRIO=2 ABHEB_Z	; 入力 3 == 1 の場合、割り込みルーチン「ABHEB_Z」を起動します。
...	
N80 SETINT(3) PRIO=1 ABHEB_X	; 入力 3 == 1 の場合、割り込みルーチン「ABHEB_X」を起動します。
...	

4.3.4 割り込みルーチンの割り当ての解除/再起動(DISABLE、ENABLE)

SETINT 命令は、DISABLE で無効に、ENABLE で、入力(割り込みルーチンの割り当て)をそのまま保持して、再度有効にできます。

構文

DISABLE (<n>)
ENABLE (<n>)

意味

DISABLE (<n>):	命令:割り込みルーチンの入力<n>の割り当ての解除	
ENABLE (<n>):	命令:割り込みルーチンの入力<n>の割り当ての再設定	
<n>:	パラメータ:割り込み信号番号	
	タイプ:	INT
	値の範囲:	1 ... 32

例

プログラムコード	コメント
N20 SETINT(3) PRIO=1 ABHEB_Z	; 入力 3 が切り替わると、割り込み ; ルーチン「ABHEB_Z」が起動します。
...	
N90 DISABLE(3)	; N20 からの SETINT 命令を無効にします。

4.3 割り込みルーチン(ASUB)

プログラムコード	コメント
...	
N130 ENABLE(3)	; N20 からの SETINT 命令を再度有効にします。
...	

4.3.5 割り込みルーチンの割り当ての解除(CLRINT)

NC プログラム(ASUP)に対して SETINT で定義された割り込み信号の割り当てを CLRINT で解除することができます。

構文

CLRINT (<n>)

意味

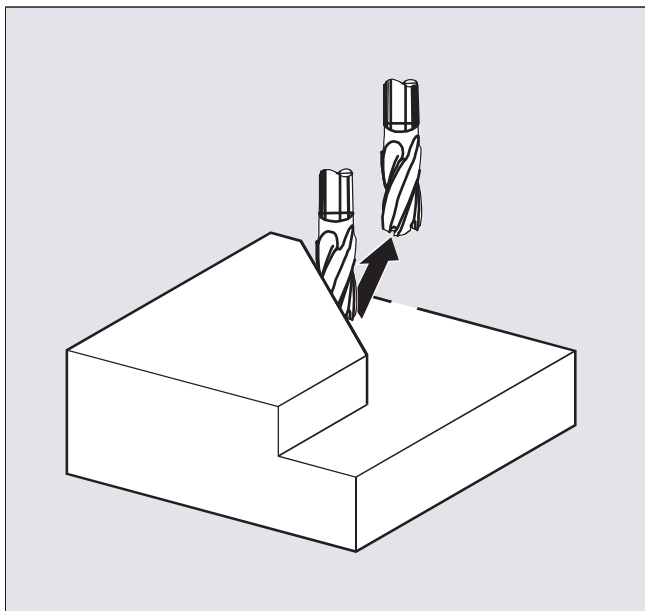
CLRINT (<n>):	命令:SETINT <n>で定義した NC プログラム(ASUP)への割り込み信号 <n>の割り当てを解除します。	
<n>:	パラメータ:割り込み信号番号	
	タイプ:	INT
	値の範囲:	1 ... 32

例

プログラムコード	コメント
N20 SETINT(3) PRIO=2 ABHEB_Z	
...	
N50 CLRINT(3)	; 入力「3」と割り込みルーチン「ABHEB_Z」の割り当てを解除します。

4.3.6 輪郭からの高速リトラクト(SETINT LIFTFAST、ALF)

SETINT 命令が LIFTFAST と一緒に指令された場合は、入力に変化すると、工具がワーク輪郭から高速リトラクト機能を使って離れます。



それ以降の順序は、SETINT 命令が LIFTFAST の他に割り込みルーチンを含むかどうかにより変わります。

割り込みルーチンを含む 高速リトラクトの**後**に割り込みルーチンが実行されます。

場合:

割り込みルーチンを含み 高速リトラクトの後に加工が停止し、アラームが発生します。

ない場合:

構文

```
SETINT (<n>) PRIO=1 LIFTFAST  
SETINT (<n>) PRIO=1 <NAME> LIFTFAST
```

4.3 割り込みルーチン(ASUB)

意味

SETINT (<n>) :	命令:入力<n>を割り込みルーチンに割り当てます。割り当てた割り込みルーチンは、入力<n>が変化すると起動します。	
<n>:	パラメータ:入力番号	
	タイプ:	INT
	値の範囲:	1 ... 8
PRIO= :	優先度の定義	
<値>:	優先度の値	
	値の範囲:	1 ... 128
	優先度 1 が最高の優先度です。	
<NAME>:	実行するサブプログラム(割り込みルーチン)の名称です。	
LIFTFAST:	命令:輪郭からの高速リトラクト	
ALF=... :	命令:プログラム指令可能な(移動ブロックの)移動方向 ALF によるプログラミング方法については、「輪郭からの高速リトラクトの移動方向 (ページ 623)」の章を参照してください。	

必要条件

ミラーリングを使用した動作中のフレームの動作

後退方向を特定するときは、ミラーリングを使用したフレームが有効かどうかの確認をおこないます。この場合の後退方向は、接線方向を基準にして左右が入れ替わります。工具方向の方向成分はミラーリングされません。この動作は、次のマシンデータの設定により有効になります。

MD21202 \$MC_LIFTFAST_WITH_MIRROR = TRUE

例

破損した工具は、自動的に予備工具へ交換されます。その後、新しい工具で加工が続行されます。

メインプログラム

メインプログラム	コメント
N10 SETINT (1) PRIO=1 W_WECHS LIFTFAST	; 入力 1 が変化すると、工具は直ちに輪郭から高速リトラクトで後退します (工具径補正 G41 のコード番号 7 の場合です)。そして、割り込みルーチン「W_WECHS」が実行されます。

4.3 割り込みルーチン(ASUB)

メインプログラム	コメント
N20 G0 Z100 G17 T1 ALF=7 D1	
N30 G0 X-5 Y-22 Z2 M3 S300	
N40 Z-7	
N50 G41 G1 X16 Y16 F200	
N60 Y35	
N70 X53 Y65	
N90 X71.5 Y16	
N100 X16	
N110 G40 G0 Z100 M30	

サブプログラム:

サブプログラム	コメント
PROC W_CHANGE SAVE	; 実際の動作状態が保存されるサブプログラム
N10 G0 Z100 M5	; 工具交換位置、主軸停止
N20 T11 M6 D1 G41	; 工具交換
N30 REPOS L RMBBL M3	; 輪郭に再位置決め、その後メインプログラムへ復帰 ジャンプします (これを 1 ブロックにプログラム指令 します)

4.3.7 輪郭からの高速リトラクトの移動方向

後退移動

次の G 命令は、後退移動平面を定義します。

- LFTXT
後退移動平面は、軌跡タンジェントと工具方向(初期設定)で定義されます。
- LFWP
後退移動平面は、G 命令 G17、G18 または G19 により選択された有効な作業平面です。
後退移動の方向は、軌跡タンジェントには影響されません。このため、高速リトラクトは、軸と平行にプログラム指令できます。
- LFPOS
POLFMASK/POLFMLIN を使用して宣言された軸が、POLF でプログラム指令されたアブソリュート軸位置へ後退します。
ALF は、直線系の複数軸だけでなく、複数軸の後退方向にも影響を与えません。

4.3 割り込みルーチン(ASUB)

プログラム指令可能な移動方向(ALF=...)

方向は、後退移動の平面で ALF により、45°の分割単位でプログラム指令します。

可能な移動方向は、コントローラに特殊コード番号で格納され、これらの番号を使用して呼び出すことができます。

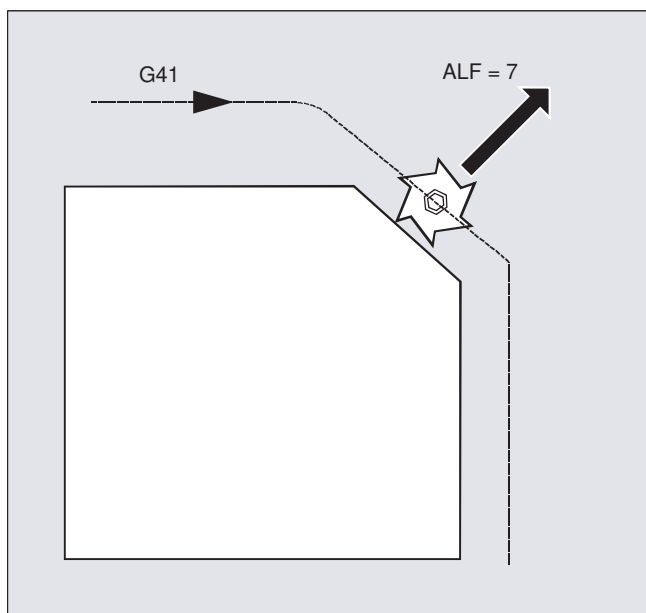
例:

プログラムコード

```
N10 SETINT(2) PRIO=1 ABHEB_Z LIFTFAST
```

```
ALF=7
```

G41 を有効(加工方向は輪郭の左側)にした場合は、工具が輪郭から垂直方向に離れます。

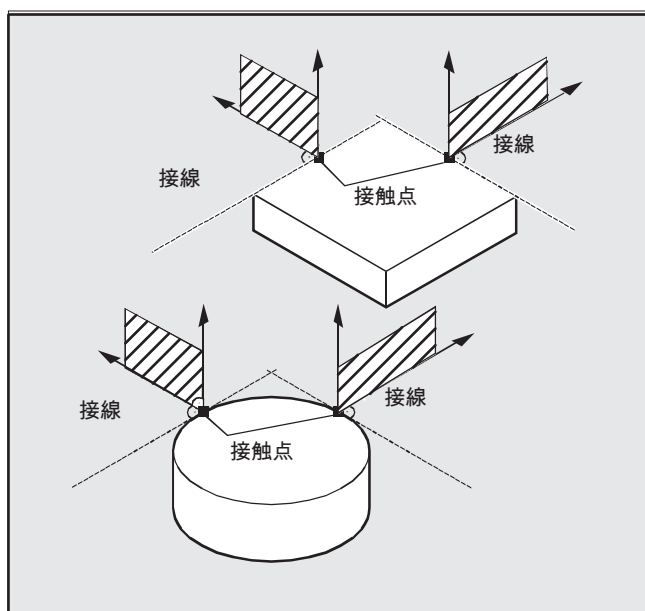


LFTXT の移動方向を定義するための基準平面

プログラム指令輪郭へ工具を適用するときに、工具は、対応するコード番号によって、戻り移動を指定するための基準として使用される平面で固定されます。

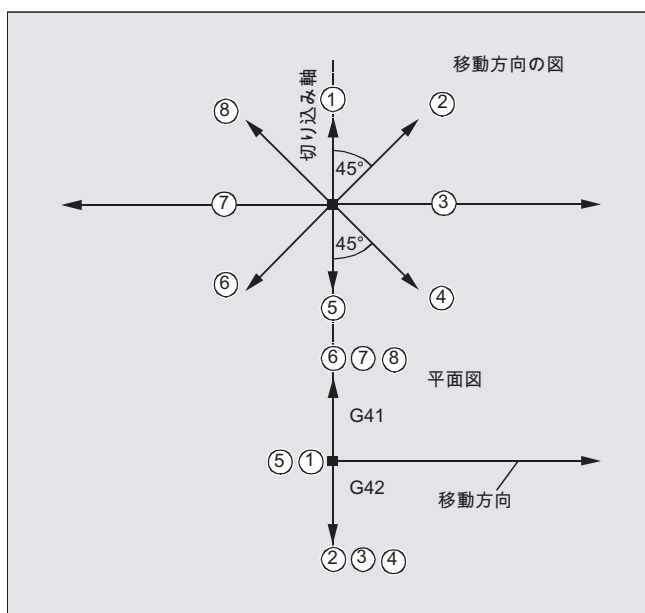
工具を適用時には、基準平面は、長手工具軸(切り込み方向)、この軸に垂直に位置するベクトル、およびこの接線に垂直に位置するベクトルから求められます。

4.3 割り込みルーチン(ASUB)



LFTXT の移動方向を表すコード番号

基準平面から開始するために、次の図に移動方向を表すコード番号があります。



工具方向の後退は ALF=1 で定義されます。

4.3 割り込みルーチン(ASUB)

「高速リトラクト」機能は ALF=0 で無効にします。



注意

干渉の可能性

工具径補正を有効にした場合、次の規則を遵守してください。

- G41 の場合はコード 2、3、4
- G42 の場合はコード 6、7、8

工具が輪郭へ移動してワークに干渉するような可能性がある場合は、上記のコード番号は使用しないでください。

LFWP の移動方向を表すコード番号

LFWP の場合、加工平面内の方向が、次の割り当てから導出されます。

- G17: X/Y 平面
ALF=1:X 方向の後退
ALF=3:Y 方向の後退
- G18: Z/X 平面
ALF=1:Z 方向の後退
ALF=3:X 方向の後退
- G19: Y/Z 平面
ALF=1:Y 方向の後退
ALF=3:Z 方向の後退

下記も参照

ねじ切り時の高速リトラクト(LFON、LFOF、DILF、ALF、LFTXT、LFWP、LFPOS、POLF、POLFMASK、POLFMLIN) (ページ 259)

4.3.8 割り込みルーチンの動作順序

LIFTFAST を使用しない割り込みルーチン

軸移動は軌跡に沿って、停止状態まで減速します(ゼロ速度)。その後、割り込みルーチンが開始されます。

停止状態の位置は、中断位置として保存され、RMIBL を使用した REPOS に対して、割り込みルーチンの終点でアプローチされます。

LIFTFAST を使用する割り込みルーチン

軸移動は、軌跡に沿って減速します。LIFTFAST 移動は、重畳移動で同時に実行されます。軌跡移動と LIFTFAST 移動が停止状態(ゼロ速度)となった場合に、割り込みルーチンが開始されます。

輪郭上の位置は中断位置として保存され、ここから LIFTFAST 移動が開始されて、軌跡から離れます。

LIFTFAST と ALF=0 を使用する割り込みルーチンは、LIFTFAST を使用しない割り込みルーチンと全く同じ方法で動作します。

注記

輪郭からの高速後退時にジオメトリ軸が移動するアブソリュート値は、マシンデータを使用して設定できます。

4.4 ファイルとプログラムの管理

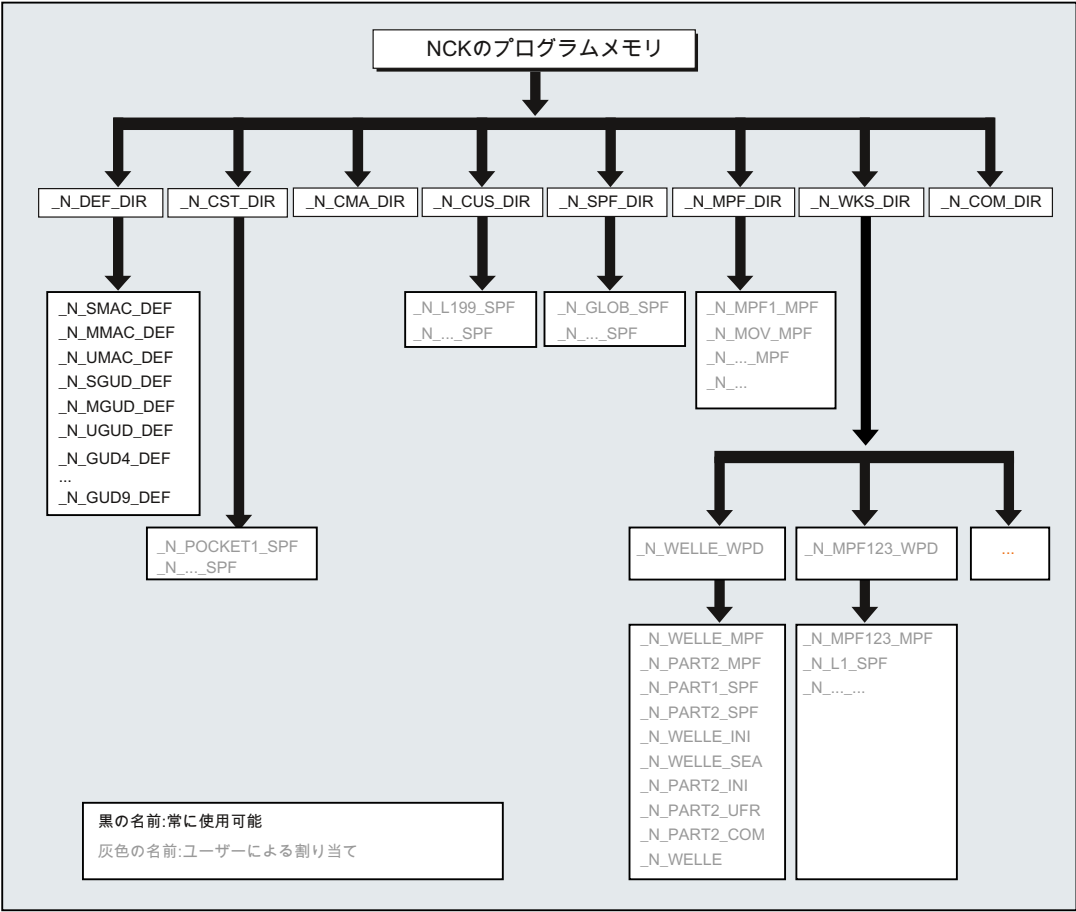
4.4.1 プログラムメモリ

4.4.1.1 NCK のプログラムメモリ

ファイルとプログラム(メインプログラムとサブプログラム、マクロ定義)は、不揮発性のプログラムメモリに保存されます(→パッシブファイルシステム)。

多くのファイルタイプもここに一時的に格納されます。これらのファイルは必要に応じて(特定のワークの加工時の初期化のためなど)、作業メモリに転送されます。

4.4 ファイルとプログラムの管理



標準ディレクトリ

以下の標準ディレクトリを使用できます。

ディレクトリ	内容
_N_DEF_DIR	データモジュールとマクロモジュール
_N_CST_DIR	標準サイクル
_N_CMA_DIR	メーカーサイクル
_N_CUS_DIR	ユーザーサイクル
_N_WKS_DIR	ワーク
_N_SPF_DIR	グローバルサブプログラム
_N_MPF_DIR	メインプログラム
_N_COM_DIR	コメント

ファイルタイプ

以下のファイルタイプをメインメモリへ格納できます。

ファイルタイプ	説明
<名称>_MPF	メインプログラム
<名称>_SPF	サブプログラム
<名称>_TEA	マシンデータ
<名称>_SEA	セッティングデータ
<名称>_TOA	工具オフセット
<名称>_UFR	ゼロオフセット/FRAME
<名称>_INI	初期化ファイル
<名称>_GUD	グローバルユーザーデータ
<名称>_RPA	R 変数
<名称>_COM	コメント
<名称>_DEF	グローバルユーザーデータとマクロの定義

ワークメインディレクトリ(_N_WKS_DIR)

ワークメインディレクトリは、プログラムメモリの標準セットアップのなかにあり、_N_WKS_DIR という名称です。ワークメインディレクトリには、プログラム指令したワークのすべてのワークディレクトリが含まれます。

ワークディレクトリ(..._WPD)

ワークディレクトリには、ワークの加工に必要なすべてのファイルが含まれます。これらのファイルは、メインプログラム、サブプログラム、あらゆる初期化プログラム、およびコメントファイルなどです。

パートプログラムの最初の起動時には、選択したプログラムに応じて初期化プログラムが 1 回だけ実行されます(マシンデータ MD11280 \$MN_WPD_INI_MODE に従います)。

4.4 ファイルとプログラムの管理

例:

SHAFT ワーク用に作成されたワークディレクトリ `_N_SHAFT_WPD` には、次のファイルが含まれます。

ファイル	説明
<code>_N_SHAFT_MPF</code>	メインプログラム
<code>_N_PART2_MPF</code>	メインプログラム
<code>_N_PART1_SPF</code>	サブプログラム
<code>_N_PART2_SPF</code>	サブプログラム
<code>_N_SHAFT_INI</code>	ワークのデータの一般初期化プログラム
<code>_N_SHAFT_SEA</code>	セッティングデータ初期化プログラム
<code>_N_PART2_INI</code>	Part 2 プログラムのデータの一般初期化プログラム
<code>_N_PART2_UFR</code>	Part 2 プログラムのフレームデータの初期化プログラム
<code>_N_SHAFT_COM</code>	コメントファイル

ワークディレクトリに、NC が加工に直接必要としないデータを保存することもできます。ASCII ファイルの他に、JPG 形式の画像や PDF 形式の記述など、バイナリファイルも使用できます。これらのファイルを NC がバイナリファイルとして解釈できるように、ファイル拡張子を NCK で既知にしておいてください(セットアップ時に `MD17000 $MN_EXTENSIONS_OF_BIN_FILES` で設定。初期設定で次のファイル拡張子が設定されています:JPG、GIF、PNG、BMP、PDF、ICO、HTM)。

加工ワークの選択

チャンネルで実行するワークディレクトリを選択できます。同じ名称のメインプログラム、または 1 つのメインプログラム(`_MPF`)のみがこのディレクトリに格納されている場合は、それが自動的に選択され、実行されます。

4.4.1.2 外部プログラムメモリ

NC のパッシブファイルシステムに加えて、(たとえば、ローカルドライブやネットワークドライブ上の)外部プログラムメモリも機械で使用できます。

「外部から実行」機能や「EES (外部記憶からの実行)」機能を使用して、外部プログラムメモリからパートプログラムを直接実行できます。

詳細情報:機能マニュアル、基本機能

グローバルパートプログラムメモリ(GDIR)

ドライブを宣言するときに、いずれかのドライブをグローバルパートプログラムメモリ (GDIR)に指定することができます。

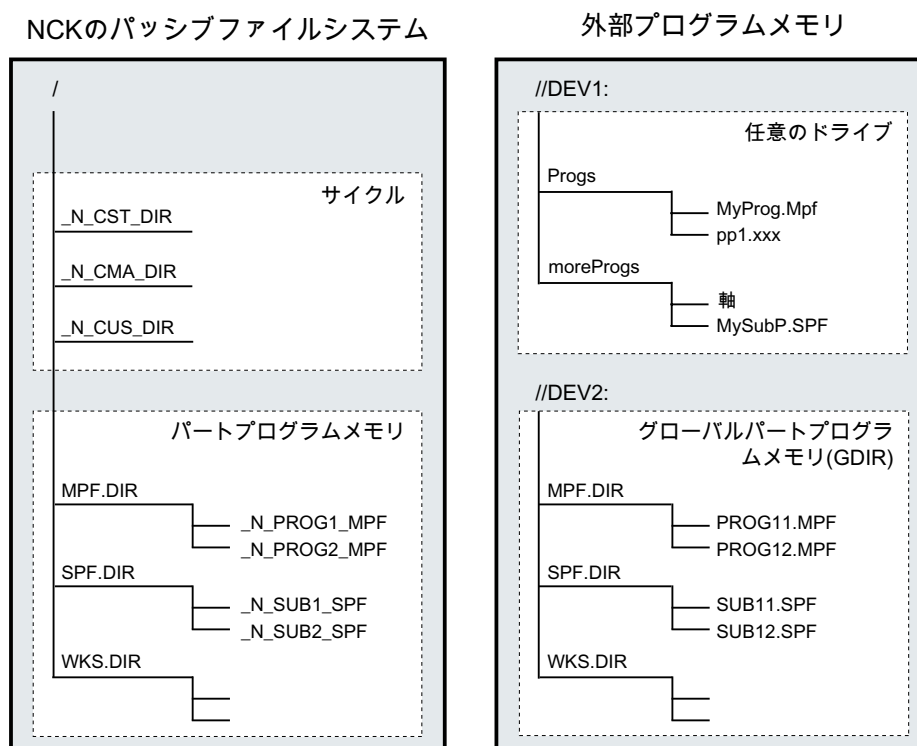
システムはドライブ上に MPF.DIR、SPF.DIR、WKS.DIR の各ディレクトリを自動的に作成します。これら 3 つのディレクトリで GDIR が形成されます。

GDIR は EES 機能のための役割を担うだけです。GDIR は、ドライブ設定に応じて、NC パートプログラムメモリを置き換えるか拡張します。ただし、GDIR の作成は、EES 操作に必須なわけではありません。

GDIR のディレクトリとファイルは、パートプログラムでパッシブファイルシステムと同じ方法でアドレス指定できます。このため、パスの詳細を含む NC プログラムを、パッシブファイルシステムから GDIR へ互換性を保って転送することができます。GDIR の SPF.DIR ディレクトリは、サブプログラムの検索パスに含められます。

プログラム構成

外部プログラムメモリでのプログラム構成を次の図に示します。



4.4 ファイルとプログラムの管理

大文字小文字を区別しないファイルシステム

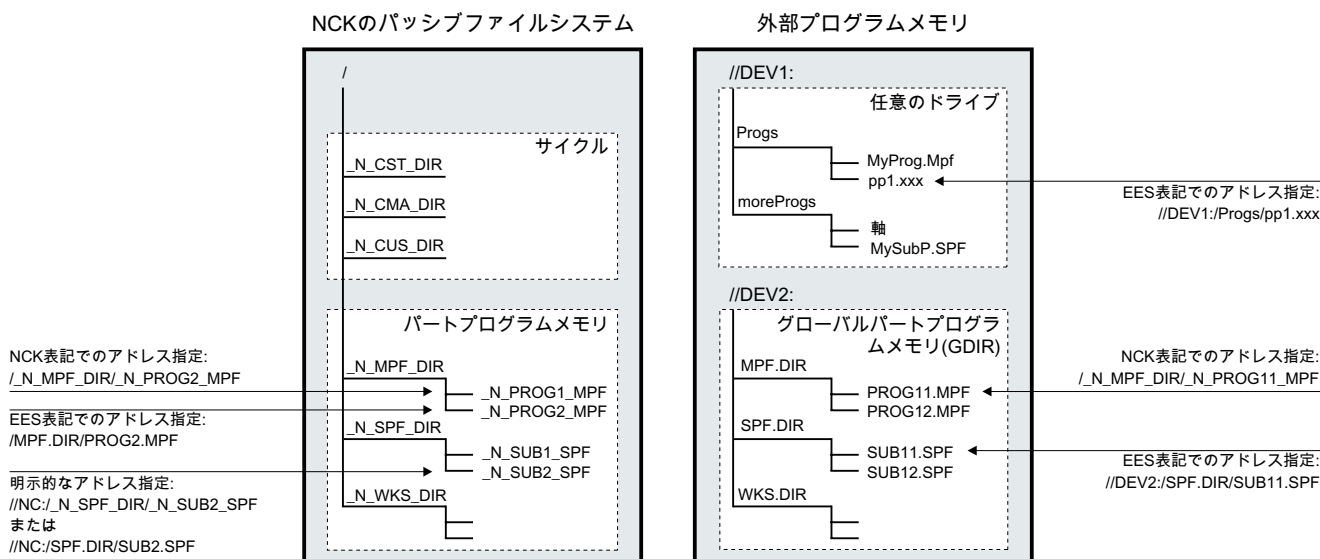
注記

ファイルのアドレス指定(「プログラムメモリファイルのアドレス指定 (ページ 632)」を参照してください)での大文字小文字の区別による問題を回避するために、外部プログラムメモリには大文字小文字を区別しないファイルシステムを使用してください。

4.4.1.3 プログラムメモリファイルのアドレス指定

プログラムメモリ内のファイルは、ファイル処理命令(WRITE、DELETE、READ、ISFILE、FILEDATE、FILETIME、FILESIZE、FILESTAT、FILEINFO など)でアドレス指定され、絶対パスとファイル名、またはファイル名のみで参照されます。ファイル名のみで参照する場合、選択したプログラムのパスがファイルパスとして使用されます。

NC/EES 表記でのアドレス指定:



パッシブファイルシステムのファイルのアドレス指定

パッシブファイルシステムのファイルは通常、ドライブ名を指定せずに **NC 表記** (ディレクトリ名とファイル名の先頭がドメイン識別子「`_N`」で、ファイル識別子の区切り文字が「`_`」)でアドレス指定されます。ただし、**EES 表記** (ドメイン識別子「`_N`」なし、ディレクトリ/ファイル拡張子の区切り文字が「`.`」)でのアドレス指定も使用できます。

例:

- NC 表記: "/_N_SPF_DIR/_N_SUB1_SPF"
- EES 表記: "/SPF.DIR/SUB1.SPF"

注記

EES 表記でのパッシブファイルシステムのファイルのアドレス指定は、以下の規則に従って内部で NC 表記に変換されます。

- ディレクトリ名とファイル名に、ドメイン識別子「_N_」を追加します。
 - ディレクトリ名またはファイル名の末尾から 4 番目の文字がピリオド(「.」)の場合、これをアンダースコア(「_」)に変換します。
-

事前定義されたドライブ名「//NC:」を使用して、パッシブファイルシステムを明確にアドレス指定することもできます。

例:

- NC 表記: "//NC:/_N_SPF_DIR/_N_SUB1_SPF"
- EES 表記: "//NC:/SPF.DIR/SUB1.SPF"

外部プログラムメモリのファイルのアドレス指定

GDIR として記録されていない外部プログラムメモリのファイルは、EES 表記でアドレス指定してください。アドレス指定パスの先頭で、ドライブ名(例:「//DEV1:」)を指定してください。/user/sinumerik/hmi/cfg/logdrive.ini で設定されているすべてのシンボリックデバイス名を使用できます。

例:

- EES 表記: "//DEV1:/MyProgDir/pp1.xxx"
- NC 表記不可

グローバルパートプログラムメモリ(GDIR)のファイルのアドレス指定

GDIR のファイルをアドレス指定する場合、EES 表記でのパス指定に加えて、NC 表記でのパス指定も使用できます。

例:

- EES 表記: "//DEV2:/MPF.DIR/PROG11.MPF"
- NC 表記: "/_N_MPF_DIR/_N_PROG11_MPF"

4.4 ファイルとプログラムの管理

注記

NC 表記での GDIR のファイルのアドレス指定は、以下の規則に従って内部で EES 表記に変換されます。

- ディレクトリ名およびファイル名のドメイン識別子「_N_」が削除されます。
 - ディレクトリ名またはファイル名の最後から 4 番目の文字がアンダースコア(「_」)の場合、これがピリオド(「.」)に変換されます。
-

パス指定の規則

完全なパス指定は、ドライブ名、ディレクトリパス、およびファイル名で構成されます。

ドライブ名

ドライブ名の指定には次の規則が適用されます。

- /user/sinumerik/hmi/cfg/logdrive.ini で設定されているすべてのシンボリックデバイス名を使用できます。
- 先頭は記号「//」で、その後に 1 つ以上の文字または数字が続きます。
- 後続の文字は、英文字、数字、「_」および空白文字の任意の組み合わせにすることができます。
- 名前の最後は文字または数字で、その後に「:」を付けます。
- その他の特殊文字は使用できません。

注記

パッシブファイルシステムにはデバイス名「//NC:」が事前定義されています。

例:

- 外部プログラムメモリ:
 - //Drive1:
 - //Drive_1:
 - //Drive 1:
 - //A B:
 - //1 B C 2:

ディレクトリパス

ディレクトリパスの指定には次の規則が適用されます。

- 「/」はディレクトリパスの最初と最後に付けられ、また個々のパス区分の区切り文字として使用されます。

注記

二重スラッシュ(「//」)は、ディレクトリパス内では使用 **できません**。

- ディレクトリ名:
 - ディレクトリ名の先頭は英文字または数字にしてください。NC 表記でのアドレス指定の場合のみ、ディレクトリ名の先頭にドメイン識別子「_N_」を付けます。
 - 後続の文字は英文字、数字、および「_」の任意の組み合わせにすることができます。

注記

外部プログラムメモリの場合は、ディレクトリ名に空白文字も使用できます。ただし、外部プログラムメモリをグローバルパートプログラムメモリ(GDIR)として作成する場合は例外です。

- その他の特殊文字は使用できません。
- ディレクトリの拡張子:
 - ディレクトリの拡張子は英文字/数字 3 文字にしてください。
 - 拡張子とディレクトリ名を「_」(NC 表記)または「.」(EES 表記)で区切ります。

注記

パッシブファイルシステムのディレクトリ拡張子は_DIR と_WPD のみです。

例:

- パッシブファイルシステムまたは GDIR:
 - NC 表記: _N_WKS_DIR/_N_MYNCPROGS_WPD/...
 - EES 表記: WKS.DIR/MYPROGS.WPD/...
- 外部プログラムメモリ:
 - /abc
 - /ab_c.def
 - /ab c1.def
 - /a b c .d11
 - /abc.def/ghi.klm

4.4 ファイルとプログラムの管理

ファイル名

ファイル名には次の規則が適用されます。

- NC 表記でのアドレス指定の場合のみ、先頭にドメイン識別子「_N_」を付けます。
- 次の 2 文字は、英字 2 文字またはアンダースコアと英字 1 文字にしてください。

注記

この条件を満たす場合は、プログラム名称を指定するだけで、別のプログラムからサブプログラムとして NC プログラムを呼び出すことができます。ただし、プログラム名称の先頭が数字である場合、サブプログラム呼び出しは CALL 命令でのみ呼び出すことができます。

-
- 後続の文字は英文字、数字、および「_」の任意の組み合わせにすることができます。
 - ファイル拡張子:
 - ファイル拡張子は英文字/数字 3 文字にしてください。

注記

パッシブファイルシステムで利用できるファイル拡張子については、「NCK のプログラムメモリ (ページ 627)」を参照してください。

-
- 拡張子とファイル名を「_」(NC 表記)または「.»(EES 表記)で区切ります。

例:

- パッシブファイルシステムまたは GDIR:
 - NC 表記: _N_SUB1_SPF
 - EES 表記: SUB1.SPF
- 外部プログラムメモリ:
 - Part 1
 - _Part1
 - Part_1.spf
 - Part1.mpf

DIN サブプログラム名称

DIN サブプログラム名称には以下の規則が適用されます。

- 先頭文字は英字「L」にしてください。
- 後続の文字は数字です(1 つ以上)。
- ファイル拡張子:
 - ファイル拡張子は英字 3 文字にしてください。
 - 拡張子とファイル名を「_」(NC 表記)または「.»(EES 表記)で区切ります。

例:

- L123
- L1_SPF (NC 表記)または L1.SPF (EES 表記)

最大軌跡長

ドライブ名およびディレクトリパスの指定には最大 128 バイトを使用できます。ファイル名の最大長は 31 バイトです。完全パスの最大長は 159 バイトです。

4.4.1.4 サブプログラム呼び出しの検索パス

パスデータを使用しないサブプログラム呼び出しの場合、固定の検索パスを処理することで絶対パスが決定されます。

その後、次の順番でプログラムメモリが検索されます。

		ディレクトリ	説明
1		現在のディレクトリ/名称	現在のディレクトリはプログラムを選択するディレクトリです。 これは以下のディレクトリにすることができます。 • NC パートプログラムメモリまたはグローバルパートプログラムメモリ内のワークディレクトリまたは標準ディレクトリ _N_MPF_DIR または • 外部プログラムメモリの任意のディレクトリ
2		現在のディレクトリ/名称_SPF	
3		現在のディレクトリ/名称_MPF	
4	A	//NC:/_N_SPF_DIR / 名称_SPF	NC パートプログラムメモリ内のサブプログラムディレクトリ
	B	//DEV2:/_N_SPF_DIR/名称_SPF ¹⁾	グローバルパートプログラムメモリ内のサブプログラムディレクトリ 注: この検索ステップは、グローバルパートプログラムメモリが作成されていない場合、または NC パートプログラムメモリでプログラムが選択されていない場合は実行されません。

4.4 ファイルとプログラムの管理

	ディレクトリ	説明
5	CALLPATH でプログラム指令された拡張検索パス(「サブプログラム呼び出しの拡張検索パス(CALLPATH) (ページ 608)」を参照してください)。 注: この検索ステップは、CALLPATH がプログラム指令されていない場合は実行されません。	
6	I_N_CUS_DIR/名称_SPF	ユーザーサイクルディレクトリ
7	I_N_CMA_DIR/名称_SPF	メーカーサイクルディレクトリ
8	I_N_CST_DIR/名称_SPF	標準サイクルディレクトリ

¹⁾ //DEV2:"たとえば、グローバルパートプログラムメモリが作成されているドライブを表します。

検索には次の規則が適用されます。

- 個々のサブプログラム呼び出しごとに検索パスが使用されます通過します。つまり、上位レベルのプログラムが配置されている場所には無関係です。
- ディレクトリに応じて、さまざまなファイルタイプが考慮されます。
- 検索はディレクトリ内で実行され、下位の、つまりネストされたディレクトリ内では実行されません。

4.4.1.5 パスとファイル名の問い合わせ

以下のシステム変数は、パートプログラムで読み込むことができ、NC プログラムのパスとファイル名を問い合わせるために使用できます。

システム変数	タイプ	意味
\$P_STACK	INT	現在の NC プログラムを実行するプログラムレベルを返します。
\$P_PATH[<n>]	STRING	フィールドインデックス<n>を使用して選択したプログラムレベルで処理する NC プログラムのパスを返します。 例: \$P_PATH[0]は、メインプログラムのパス、たとえば「/_N_WKS_DIR/_N_WELLE_WPD/」を返します。 \$P_PATH[\$P_STACK - 1]は、呼び出しプログラムのパスを返します。 パスが NC のパッシブファイルシステムまたはグローバルパートプログラムメモリ (GDIR) に保存されている NC プログラムを参照する場合、そのパスは NC 表記で返されます。 パスがグローバルパートプログラムメモリ以外の外部プログラムメモリで実行される NC プログラムを参照する場合、\$P_PATH はパスを EES 表記で返します。
\$P_PROG[<n>]	STRING	フィールドインデックス<n>を使用して選択したプログラムレベルで処理する NC プログラムの名称を返します。 NC プログラムが NC のパッシブファイルシステムまたはグローバルパートプログラムメモリに保存されている場合、プログラム名称は NC 表記で返されます。 NC プログラムがグローバルパートプログラムメモリ以外の外部ドライブで実行される場合、\$P_PROG はプログラム名称を EES 表記で返します。
\$P_PROGPATH	STRING	現在処理中の NC プログラムのパスを返します。 \$P_PROGPATH の呼び出しは、\$P_PATH[\$P_STACK]と同じです。

4.4 ファイルとプログラムの管理

システム変数	タイプ	意味
\$P_IS_EES_PATH[<n>]	BOOL	\$P_PATH[<n>]で返されるパスまたは\$P_PROG[<n>]で返されるプログラム名称が、NC 表記と EES 表記のどちらに対応しているかを問い合わせます。
		= FALSE \$P_PATH[<n>]と\$P_PROG[<n>]は、NC 表記を返します。これは、各識別子に接頭語「_N_」があることを意味します。ファイル識別子の区切り文字は「_」です。 例: • NC 表記のパス: "/_N_WKS_DIR/_N_MYWPD_WPD/" • NC 表記のプログラム名称: "_N_MYPROG_MPF" NC 表記のパスは、NC のパッシブファイルシステムだけでなく、グローバルパートプログラムメモリも参照できます。
		= TRUE \$P_PATH[<n>]と\$P_PROG[<n>]は、EES 表記を返します。これは、各識別子に接頭語「_N_」がないことを意味します。ファイル識別子の区切り文字は「.」です。 例: • EES 表記のパス: "//DEV1:/WKS.DIR/MYWPD.WPD/" • EES 表記のプログラム名称: "MYPROG.MPF"

<n>:インデックス<n>は、パス情報を読み込むプログラムレベルを定義します(数値の範囲:0 ... 17)。

注記

EES モードでは、グローバルパートプログラムメモリ (GDIR)以外、システム変数\$P_PROG、\$P_PATH、\$P_PROGPATH のパス名は EES 表記です。EES モードでは、これらのパス名を評価および処理するユーザープログラムは、EES 表記でのパス名も処理できるように拡張してください。

4.4.2 作業メモリ(CHANDATA、COMPLETE、INITIAL)

機能

作業メモリには、現在のシステム、およびコントローラの運転に使用するユーザデータ(アクティブファイルシステム)が含まれます。データの例を以下に示します。

- 動作中のマシンデータ
- 工具オフセットデータ
- ゼロオフセット
- ...

初期化プログラム

これは、作業メモリデータの初期化に使用するプログラムです。これには、次のファイルタイプを使用できます。

ファイルタイプ	説明
name_TEA	マシンデータ
name_SEA	セッティングデータ
name_TOA	工具オフセット
name_UFR	ゼロオフセット/フレーム
name_INI	初期化ファイル
name_GUD	グローバルユーザデータ
name_RPA	R 変数

データ領域

データは、その適用先のさまざまな領域で処理できます。たとえば、コントローラは複数のチャンネルを持つことができます。または、通常の場合、複数の軸を自由に配置できます。

以下のものがあります。

識別子	データ領域
NC	NC 別データ
CH<n>	チャンネル別データ(<n>でチャンネル名称を指定します)
AX<n>	軸別データ(<n>で機械軸の番号を指定します)

4.4 ファイルとプログラムの管理

識別子	データ領域
TO	工具データ
COMPLETE	全データ

外部 PC による初期化プログラムの作成

データ領域識別子とデータタイプ識別子を使用すると、データの保存時に一つの単位として扱われる領域を特定できます。

<code>_N_AX5_TEA_INI</code>	5 軸のマシンデータ
<code>_N_CH2_UFR_INI</code>	チャンネル 2 のフレーム
<code>_N_COMPLETE_TEA_INI</code>	すべてのマシンデータ

コントローラの最初の起動時に、コントローラが正常に動作するよう、一組のデータのセットが自動的にロードされます。

マルチチャンネル制御の手順(CHANDATA)

複数のチャンネル用の CHANDATA (<チャンネル番号) は、`_N_INITIAL_INI` ファイルのみで使用できます。これは、コントローラのすべてのデータの初期化に使用するセットアップファイルです。

プログラムコード	コメント
<code>%_N_INITIAL_INI</code>	
<code>CHANDATA (1)</code>	
<code>\$MC_AXCONF_MACHAX_USED[0]=1</code>	
<code>\$MC_AXCONF_MACHAX_USED[1]=2</code>	
<code>\$MC_AXCONF_MACHAX_USED[2]=3</code>	
<code>CHANDATA (2)</code>	
<code>\$MC_AXCONF_MACHAX_USED[0]=4</code>	
<code>\$MC_AXCONF_MACHAX_USED[1]=5</code>	
<code>CHANDATA (1)</code>	
<code>\$MA_STOP_LIMIT_COARSE[AX1]=0.2</code>	
<code>\$MA_STOP_LIMIT_COARSE[AX2]=0.2</code>	
<code>\$MA_STOP_LIMIT_FINE[AX1]=0.01</code>	
<code>\$MA_STOP_LIMIT_FINE[AX1]=0.01</code>	

; 機械軸の割り当て、チャンネル 1:

; 機械軸の割り当て、チャンネル 2:

; 軸マシンデータ:

; 汎用イグザクトストップ範囲:

; 軸 1

; 軸 2

; 精密イグザクトストップ範囲

; 軸 1

; 軸 2

通知**CHANDATA 命令**

パートプログラムでは、CHANDATA 命令は NC プログラムが実行されるチャンネルに対してのみ設定できます。つまり、この命令は NC プログラムが間違ったチャンネルで実行されないよう、NC プログラムを保護するのに使用されます。

プログラムの処理は、エラーが発生すると中止されます。

注記

CHANDATA 命令は、ジョブリストの INI ファイルは含まれません。

初期化プログラムの保存(COMPLETE、INITIAL)

作業メモリのファイルは、外部 PC に保存して、そこから再度読み込むことができます。

- ファイルは、COMPLETE で保存します。
- INITIAL を使用して、すべての領域にわたる INI ファイル(_N_INITIAL_INI)を作成します。

初期化プログラムの読み込み**通知****データの消失**

ファイルを「INITIAL_INI」という名称で読み込んだ場合は、ファイルで提供されないデータがすべて、標準データを使用して初期化されます。例外はマシンデータのみです。つまり、**セッティングデータ、工具データ、ZO、GUD 値、...**が標準データ(通常は「ゼロ」)で提供されます。

たとえば、ファイル COMPLETE_TEA_INI は、個々のマシンデータの読み込みに対応しています。コントローラでは、このファイルにあるのはマシンデータのみであると想定されます。このため、この場合は、他のデータ領域は影響されずにそのままです。

初期化プログラムのロード

1 つのチャンネルのデータのみを使用する場合は、INI プログラムを選択し、パートプログラムとして呼び出すこともできます。これは、プログラム制御データの初期化も可能であるという意味です。

4.5 ファイル処理

4.5 ファイル処理

4.5.1 ファイルの書き込み(WRITE)

WRITE 命令は、ブロック/データを NC プログラムからパッシブファイルシステム内のファイル(ログファイル)の末尾に、または外部のプログラムメモリに書き込みます。これは、現在実行中のプログラムでもかまいません。

注記

プログラムメモリに該当ファイルが存在しない場合は、1 つのファイルが作成され、WRITE 命令による書き込みが可能となります。

必要条件

現在設定されている保護レベルを、ファイルの WRITE 権のレベル以上にしてください。そうでない場合は、アクセスが拒否されてエラーメッセージが表示されます(エラー変数の戻り値= 13)。

構文

```
DEF INT <エラー>
...
WRITE (<エラー>,"<ファイル名称>"/"<ExtG>","<ブロック/データ>")
```


意味

WRITE:	指定したファイルの末尾にブロックまたはデータを付加する命令です。		
<エラー>:	パラメータ 1:エラー値を返すための変数		
	タイプ:	INT	
	値:	0	エラーなし
		1	パスは許容されません
		2	パスが見つかりません
		3	ファイルが見つかりません
		4	ファイルタイプが不正です
		10	ファイルがいっぱいです
		11	ファイルが使用中です
		12	使用可能なリソースがありません
		13	アクセス権がありません
		14	出力機器の EXTOPEN が欠落しているか正常に終了しませんでした
		15	外部機器への書き込み時にエラーが発生しました
		16	無効な外部パスがプログラム指令されています。
<ファイル名称>:	パラメータ 2:指定したブロックまたはデータを追加するファイルの名称です。		
	タイプ:	STRING	
	実際のファイル名の前に絶対パスを指定できます。パスを指定しない場合は、現在のディレクトリ(選択したプログラムのディレクトリ)でファイルを検索します。 パスデータに関する規則については、「プログラムメモリファイルのアドレス指定 (ページ 632)」を参照してください。		

4.5 ファイル処理

<ExtG>:	「Process DataShare」機能を使用してデータを外部機器/ファイルに出力する場合は、ファイル名称ではなく、開こうとしている外部機器/ファイルのシンボルの識別子を指定してください。	
	タイプ:	STRING
	詳細については、「Process DataShare - 外部機器/ファイルへの出力 (EXTOPEN、WRITE、EXTCLOSE) (ページ 1163)」を参照してください。 注: 識別子は EXTOPEN 命令で指定した識別子と同一のものにしてください。	
<ブロック/データ>:	パラメータ 3:指定したファイルに追加されるブロックまたはデータです。	
	タイプ:	STRING

注記

パッシブファイルシステムまたは外部プログラムメモリに書き込むときに、WRITE 命令により自動的に「LF」文字(LINE FEED = 新しい行)が出力文字列の末尾に挿入されます。この動作は、「Process DataShare」機能を使用した外部機器/ファイルへの出力には適用されません。「LF」も出力する場合は、出力文字列でこれを明示的に指定してください。
 → 例 3 も参照してください:自動的/明示的「LF」!

必要条件

- 最大ファイルサイズ(→ 工作機械メーカー)

パッシブファイルシステム内のログファイルの最大可能ファイルサイズは、次のマシンデータで設定されます。

MD11420 \$MN_LEN_PROTOCOL_FILE

最大ファイル長は、WRITE 命令を使用して作成したパッシブファイルシステム内のすべてのファイルに適用されます。この値を超えた場合は、エラーメッセージが出力され、ブロックまたはデータは保存されません。十分な空きメモリがある場合は、新しいファイルを作成できます。

例

例 1:絶対パスデータを使用しないパッシブファイルシステムへの WRITE 命令

プログラムコード	コメント
N10 DEF INT ERROR	; エラー変数を定義します。
N20 WRITE(ERROR,"PROT","LOG FROM 7.2.97")	; 「LOG FROM 7.2.97」からファイル _N_PROT_MPF にテキストを書き込みます。
N30 IF ERROR	; エラーを評価します。
N40 MSG ("Error with WRITE command:" <<ERROR)	
N50 M0	
N60 ENDIF	
...	

例 2:絶対パスデータを使用したパッシブファイルシステムへの WRITE 命令

プログラムコード
...
WRITE(ERROR,"/_N_WKS_DIR/_N_PROT_WPD/_N_PROT_MPF","LOG FROM 7.2.97")
...

例 3:自動的/明示的「LF」

a) 明示的に生成された「LF」ありのパッシブファイルシステムへの書き込み

プログラムコード
...
N110 DEF INT ERROR
N120 WRITE(ERROR,"/_N_MPF_DIR/_N_MYPROTFILE_MPF","MY_STRING")
N130 WRITE(ERROR,"/_N_MPF_DIR/_N_MYPROTFILE_MPF","MY_STRING")
N140 M30

出力結果:

MY_STRING

MY_STRING

b) 自動的に生成された「LF」なしの外部ファイルへの書き込み

プログラムコード
...
N200 DEF STRING[30] DEV_1

4.5 ファイル処理

プログラムコード

```

N210 DEF INT ERROR
N220 DEV_1="LOCAL_DRIVE/myprotfile.mpf"
N230 EXTOPEN(ERROR,DEV_1)
N240 WRITE(ERROR,DEV_1,"MY_STRING")
N250 WRITE(ERROR,DEV_1,"MY_STRING")
N260 EXTCLOSE(ERROR,DEV_1)
N270 M30

```

出力結果:

MY_STRINGMY_STRING

c) 明示的に生成された「LF」ありの外部ファイルへの書き込み

a の場合と同じ結果を達成するには、以下をプログラム指令してください。

プログラムコード

```

...
N200 DEF STRING[30] DEV_1
N210 DEF INT ERROR
N220 DEV_1="LOCAL_DRIVE/myprotfile.mpf"
N230 EXTOPEN(ERROR,DEV_1)
N240 WRITE(ERROR,DEV_1,"MY_STRING'H0A'")
N250 WRITE(ERROR,DEV_1,"MY_STRING'H0A'")
N260 EXTCLOSE(ERROR,DEV_1)
N270 M30

```

出力結果:

MY_STRING

MY_STRING

4.5.2 ファイルの削除(DELETE)

DELETE 命令は、WRITE 命令で作成したかどうかにかかわらず、すべてのファイルを削除します。より高いアクセス権限で作成したファイルも、DELETE で削除できます。

構文

```

DEF INT <エラー>
DELETE (<エラー>,"<ファイル名称>")

```

意味

DELETE:	指定したファイルを削除する命令です。		
<エラー>:	エラー値を返す変数です。		
	タイプ:	INT	
	値:	0	エラーなし
		1	パスは使用できません
		2	パスが見つかりません
		3	ファイルが見つかりません
		4	ファイルタイプが不正です
		11	ファイルが使用中です
		12	使用可能なリソースがありません
		20	その他のエラー
<ファイル名称>:	削除するファイルの名称		
	タイプ:	STRING	
	実際のファイル名の前に絶対パスを指定できます。パスを指定しない場合は、現在のディレクトリ(選択したプログラムのディレクトリ)でファイルを検索します。 パスデータに関する規則については、「プログラムメモリファイルのアドレス指定 (ページ 632)」を参照してください。		

例

プログラムコード	コメント
N10 DEF INT ERROR	; エラー変数を定義します。
N15 STOPRE	; 先読み停止
N20 DELETE(ERROR, "/_N_SPF_DIR/_N_TEST1_SPF")	; サブプログラムのディレクトリのファイル TEST1 を削除します。
N30 IF ERROR	; エラーを評価します。
N40 MSG("error for DELETE command:" <<ERROR)	
N50 M0	
N60 ENDIF	

4.5 ファイル処理

4.5.3 ファイルの行の読み取り(READ)

READ 命令は、指定したファイルの 1 行以上の行を読み取り、読み取った情報を **STRING** タイプの配列で保存します。この配列では、読み取った各行が 1 つの配列要素を使用します。

必要条件

現在設定されている保護レベルを、ファイルの **READ** 権レベル以上にしてください。そうでない場合は、アクセスが拒否されてエラーメッセージが表示されます(エラー変数の戻り値= 13)。

構文

```
DEF INT <エラー>  
DEF STRING[<文字列長>] <結果>[<n>,<m>]  
READ(<エラー>,"<ファイル名>",<開始行>,<行数>,<結果>)
```

意味

READ:	指定したファイルから行を読み取り、それらを配列変数に保存する命令です。		
<エラー>:	エラー値を返す変数です(参照渡しパラメータ)。		
	タイプ:	INT	
	値:	0	エラーなし
		1	パスは使用できません
		2	パスが見つかりません
		3	ファイルが見つかりません
		4	ファイルタイプが不正です
		11	ファイルが使用中です
		13	アクセス権が不十分です
		21	行が存在しません(<開始行>または<行数>パラメータが指定したファイルの行数を超えています)。
		22	結果変数(結果>)のフィールド長が短すぎます。
		23	行範囲が大きすぎます(選択した<行数>パラメータが大きすぎるため、ファイルの最後を超えて読み取られます)。

<ファイル名称>:	読み取るファイルの名称です(値渡しパラメータ)。		
	タイプ:	STRING	
	実際のファイル名の前に絶対パスを指定できます。パスを指定しない場合は、現在のディレクトリ(選択したプログラムのディレクトリ)でファイルを検索します。 パスデータに関する規則については、「プログラムメモリファイルのアドレス指定 (ページ 632)」を参照してください。		
<開始行>:	読み取るファイル区間の開始行(値渡しパラメータ)		
	タイプ:	INT	
	値:	0	<行数>パラメータで指定した行数をファイルの最後に到達する前まで読み取ります。
		1～n	読み取る最初の行の番号です。
<行数>:	読み取る行数です(値渡しパラメータ)。		
	タイプ:	INT	
<結果>:	結果変数です(参照渡しパラメータ)。 読み取ったテキストの保存先の配列変数です。		
	タイプ:	STRING (最大長:255)	
	<行数>パラメータで指定した行数が、結果変数の配列サイズ[<n>,<m>]より少ない場合、残りの配列要素は変更されません。 制御文字「LF」(改行)または「CR LF」(復帰改行)による行の終了は、結果変数に 保存されません 。 読み取った行が、指定した文字列長より長い場合は、切り捨てられます。エラーメッセージは出力されません。		

注記

バイナリファイルは読み込むことができません。「不正なデータタイプ」エラーが出力されます(エラー変数の戻り値= 4)。次のタイプのファイルは読み取ることができません:
_BIN、_EXE、_OBJ、_LIB、_BOT、_TRC、_ACC、_CYC、_NCK。

例

プログラムコード	コメント
N10 DEF INT ERROR	; エラー変数を定義します。
N20 DEF STRING[255] RESULT[5]	; 結果変数を定義します。

4.5 ファイル処理

プログラムコード	コメント
N30 READ (ERROR,"/_N_CST_DIR/ _N_TESTFILE_MPF",1,5,RESULT)	; ドメインとファイル識別子、およびパス名称の付いた ファイル名称
N40 IF ERROR <>0	; エラーを評価します。
N50 MSG ("ERROR"<<ERROR<<"ON READ COMMAND")	
N60 M0	
N70 ENDIF	
...	

4.5.4 ファイルの存在の確認(ISFILE)

ISFILE 命令は、ファイルがプログラムメモリに存在するかどうかをチェックします。

構文

<結果>=ISFILE("<ファイル名>")

意味

ISFILE:	ファイルの使用可能性をチェックするコマンド		
<ファイル名>:	使用可能性をチェックするファイルの名称。		
	タイプ:	STRING	
	実際のファイル名の前に絶対パスを指定できます。パスを指定しない場合は、現在のディレクトリ(選択したプログラムのディレクトリ)でファイルを検索します。		
	パスデータに関する規則については、「プログラムメモリファイルのアドレス指定 (ページ 632)」を参照してください。		
<結果>:	確認結果の割り当て先となる結果変数です。		
	タイプ:	BOOL	
	値:	TRUE	ファイルが存在します
		FALSE	ファイルが存在しません

例

例 1

プログラムコード	コメント
N10 DEF BOOL RESULT	; 結果変数を定義します。

プログラムコード	コメント
N20 RESULT=ISFILE("TESTFILE")	
N30 IF (RESULT==FALSE)	
N40 MSG("ファイルが存在しません")	
N50 M0	
N60 ENDIF	
...	

例 2

プログラムコード	コメント
N10 DEF BOOL RESULT	; 結果変数を定義します。
N20 RESULT=ISFILE("TESTFILE")	
N30 IF (NOT ISFILE("TESTFILE"))	
N40 MSG("ファイルが存在しません")	
N50 M0	
N60 ENDIF	
...	

4.5.5 ファイル情報の読み出し(FILEDATE、FILETIME、FILESIZE、FILESTAT、FILEINFO)

FILEDATE、FILETIME、FILESIZE、FILESTAT、および FILEINFO 命令は、最後の書き込みアクセスの日付/時刻、現在のファイルサイズ、ファイル状態、またはこれらの情報などすべて、特定のファイル情報を読み出します。

必要条件

現在設定されている保護レベルを、上位のディレクトリの表示権レベル以上にしてください。そうでない場合は、アクセスが拒否されてエラーメッセージが表示されます(エラー変数の戻り値= 13)。

構文

FILE.... (<エラー>,"<ファイル名>",<結果>)

意味

FILEDATE:	ファイルへの最後の書き込みアクセスの 日付 を返します。
FILETIME:	ファイルへの最後の書き込みアクセスの 時刻 を返します。
FILESIZE:	ファイルの 現在のサイズ を返します。

4.5 ファイル処理

FILESTAT:	次の 権限 の中からファイルの 状態 を返します。 - 読み取り(r: read) - 書き込み(w: write) - 実行(x: execute) - 表示(s: show) - 削除(d: delete) 注: これらの保護レベルはパッシブファイルシステムに固有の特性です。そのため、外部のプログラムメモリにアクセスする場合、FILESTAT は初期設定のアクセス権(77777)を返します。		
FILEINFO:	ファイルに対して、FILEDATE、FILETIME、FILESIZE、FILESTAT から読み出せる 情報の合計 を返します。		
<Error>:	エラー値を返す変数です(参照渡しパラメータ)。		
	タイプ:	VAR INT	
	値:	0	エラーなし
		1	パスは使用できません
		2	パスが見つかりません
		3	ファイルが見つかりません
		4	ファイルタイプが不正です
		13	アクセス権が不十分です
	22	結果変数の文字列長(<結果>)が短すぎます。	
<ファイル名>:	ファイル情報の読み出し元のファイルの名称です。		
	タイプ:	CHAR[160]	
	実際のファイル名の前に絶対パスを指定できます。パスを指定しない場合は、現在のディレクトリ(選択したプログラムのディレクトリ)でファイルを検索します。 パスデータに関する規則については、「プログラムメモリファイルのアドレス指定 (ページ 632)」を参照してください。		

<結果>:	結果変数(Call-By-Reference パラメータ) 要求したファイル情報の保存先となる変数です。			
	タイプ:	VAR CHAR[8]	情報	FILEDATE 形式: 「dd.mm.yy」
		VAR CHAR[8]	情報	FILETIME 形式: 「hh.mm.ss」
		VAR INT	情報	FILESIZE ファイルサイズはバイト単位で出力されます。
		VAR CHAR[5]	情報	FILESTAT 形式: 「rwxsd」 (r: read, w: write, x: execute, s: show, d: delete)
		VAR CHAR[32]	情報	FILEINFO 形式: 「rwxsd nnnnnnnn dd.mm.yy hh:mm:ss」

例

プログラムコード	コメント
N10 DEF INT ERROR	; エラー変数を定義します。
N20 STRING[32] RESULT	; 結果変数を定義します。
N30 FILEINFO(ERROR,"/_N_MPF_DIR/ _N_TESTFILE_MPF",RESULT)	; ドメイン、ファイル識別子、およびパス指定を含むファイル名称です。
N40 IF ERROR <> 0	; エラー分析
N50 MSG("ERROR"<<ERROR<<"FOR FILE INFORMATION COMMAND")	
N60 M0	
N70 ENDIF	
...	

結果変数 RESULT では、例によって以下の結果が得られる場合があります。

"77777 12345678 26.05.00 13:51:30"

4.6 保護領域

4.6.1 プロテクションゾーンの定義(CPROTDEF、NPROTDEF)

衝突から機械要素を保護するプロテクションゾーンは、ブロック内のパートプログラムで定義します。これには、次の要素が含まれます。

- 1. 加工平面の定義
実プロテクションゾーン定義の前に、プロテクションゾーンの輪郭の記述で参照する加工平面を選択する必要があります。
- 2. 定義の開始
特定の NC 命令に応じて、チャンネル別または機械別のプロテクションゾーンを作成します。
- 3. プロテクションゾーンの輪郭の記述
プロテクションゾーンの輪郭は、移動動作によって定義されます。これらは実行されず、以前または後続のジオメトリ軸指令との関連はありません。プロテクションゾーンが定義されるだけです。
- 4. 定義終了

構文

```
DEF INT <Var>
G17/G18/G19
CPROTDEF/NPROTDEF (<n>,<t>,<AppLim>,<AppPlus>,<AppMinus>)
G0/G1/...X/Y/Z...
...
EXECUTE (<Var>)
```

意味

DEF INT <Var>:	ローカルヘルプ変数定義、INTEGER データタイプ
<Var>:	ヘルプ変数の名称
G17/G18/G19:	加工平面 注: 定義を終了する前に加工平面を変更することはできません。定義の開始から終了までの間には、垂直軸をプログラムできません。
CPROTDEF ():	チャンネル別のプロテクションゾーンを定義する事前定義された手順

NPROTDEF ():	機械別のプロテクションゾーンを定義する事前定義された手順		
<n>:	プロテクションゾーン定義番号		
	データタイプ:	INT	
<t>:	プロテクションゾーンのタイプ		
	データタイプ:	BOOL	
	値:	TRUE	工具関連プロテクションゾーン
		FALSE	ワーク関連プロテクションゾーン
<applim>:	第 3 次元での制限のタイプ		
	データタイプ:	INT	
	値:	0	制限なし
		1	正方向の制限
		2	負方向の制限
		3	正および負方向の制限
<AppPlus>:	3 次元での正方向の制限値		
	データタイプ:	REAL	
<appminus>:	3 次元での負方向の制限値		
	データタイプ:	REAL	

4.6 保護領域

G0/G1/...X/Y/Z... ..:	プロテクションゾーンの輪郭は、選択加工平面で、最大 11 回の移動動作を行って指定します。最初の移動は、輪郭への移動です。輪郭記述の最後の点は、常に輪郭記述の最初の点に一致させてください。 以下のように、有効なプロテクションゾーンは、輪郭の左側の領域です。 • 内側プロテクションゾーン 内側プロテクションゾーンの輪郭は、反時計方向で記述してください。 • 外側プロテクションゾーン(ワーク対応プロテクションゾーンでのみ許可) 外側プロテクションゾーンの輪郭は、時計方向で記述してください。 次の輪郭要素が使用可能です。 • 直線の輪郭要素のための G0、G1 • 右回り方向の円弧部分の場合、G2 ワーク対応プロテクションゾーンでのみ使用可能です。工具対応プロテクションゾーンは凸型形状が必要なため、使用できません。 • 左回り方向の円弧部分の場合、G3 注: プロテクションゾーンでは一周円を記述できません。一周円を 2 つの半円に分割してください。 注: シーケンス G2 → G3 または G3 → G2 は使用できません! 2 つの円弧ブロックの間には、短い G1 ブロックを挿入してください。
EXECUTE (<Var>):	定義の終了を特定する事前定義された手順 EXECUTE により、通常のプログラム処理に戻ります。

例

「プロテクションゾーンの起動/解除(CPROT、NPROT) (ページ 660)」の例を参照してください。

他の情報

機械別保護領域

機械別プロテクションゾーンやその輪郭は、ジオメトリ軸を使用して、つまり、チャンネルの基本座標系(BCS)を基準にして定義されます。機械別プロテクションゾーンが有効なすべてのチャンネル内で正しいプロテクションゾーン監視が行われるようにするには、対応するすべてのチャンネルの基本座標系(BCS)を同じにしてください。

- 機械原点を基準にした座標原点の位置
- 座標軸の向き

輪郭の記述の基準点

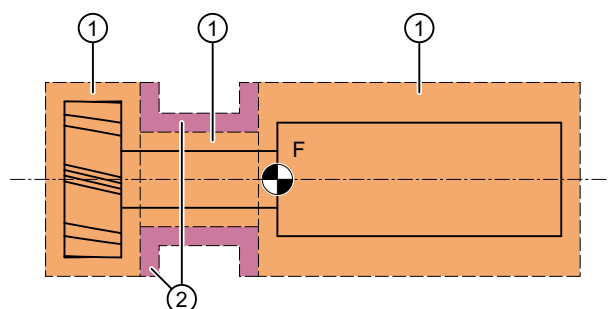
- 工具基準プロテクションゾーン
工具基準プロテクションゾーンの座標値は、**工具ホルダレファレンス点 F** を基準とする絶対値として指定する必要があります。
- ワーク基準プロテクションゾーン
ワーク基準プロテクションゾーンの座標値は、**基本座標系(BCS)**のゼロ点を基準とする絶対値として指定する必要があります。

回転対称なプロテクションゾーン

回転対称なプロテクションゾーン(主軸のチャックなど)の場合は、回転中心までの輪郭のみではなく、輪郭全体を記述してください。

工具基準プロテクションゾーン

工具関連プロテクションゾーンは常に凸型にしてください。凹型プロテクションゾーンが望ましい場合は、複数の凸型プロテクションゾーンに分割してください。



- ① 凸型プロテクションゾーン
- ② 凹型プロテクションゾーン(使用できません)
- F 工具ホルダの基準点

4.6 保護領域

一般条件

プロテクションゾーンの定義時には、次の機能を有効にしたり使用したりすることはできません。

- 工具径補正(フライス工具径補正、ノーズ R 補正)
- 座標変換
- レファレンス点復帰(G74)
- 固定点アプローチ(G75)
- ドウェル時間(G4)
- ブロック検索停止(STOPRE)
- プログラム終了(M17、M30)
- M 機能:M0、M1、M2

4.6.2 プロテクションゾーンの起動/解除(CPROT、NPROT)

以前にパートプログラムで定義したプロテクションゾーンはいつでも起動できます。また、PLC ユーザープログラムによる以降の起動に対して起動待ちにできます。有効なプロテクションゾーンはいつでも解除できます。

起動時または起動待ち時に、プロテクションゾーンのレファレンス点を相対的にシフトすることもできます。

注記

プロテクションゾーンは、起動されたチャネルのすべてのジオメトリ軸の原点確立後にのみ考慮されます。

注記

プロテクションゾーンの監視

動作中の工具関連プロテクションゾーンがない場合、工具軌跡がワーク関連プロテクションゾーンと照合されます。

ワーク方向のプロテクションゾーンが有効でない場合、プロテクションゾーンは監視されません。

構文

CPROT (<n>,<Status>,<XMov>,<YMov>,<ZMov>)

NPROT (<n>,<Status>,<XMov>,<YMov>,<ZMov>)

意味

CPROT:	チャンネル別のプロテクションゾーンを有効にする事前定義された手順		
NPROT:	機械別のプロテクションゾーンを有効にする事前定義された手順		
<n>:	プロテクションゾーンの番号		
	データタイプ:	INT	
<状態>:	チャンネル別の起動状態は、このパラメータを使用して設定します		
	データタイプ:	INT	
	値:	0	プロテクションゾーンの解除
		1	プロテクションゾーンの起動待ち
		2	プロテクションゾーンの起動
		3	オプションナルストップありのプロテクションゾーンの起動待ち
<移動>,<y 移動>,<z 移動>:	X/Y/Z 方向の追加オフセット値		
	オフセットは 1、2、または 3 次元で実行できます。オフセット値は以下を示します。 - ワーク関連プロテクションゾーンに対する機械原点 - 工具別プロテクションゾーンの工具ホルダのレファレンス点 F		
	データタイプ:	REAL	

例

フライスカッターと計測プローブの衝突の発生の可能性を、フライス盤で監視します。機能が有効なときは、オフセットにより計測プローブの位置が定義されます。

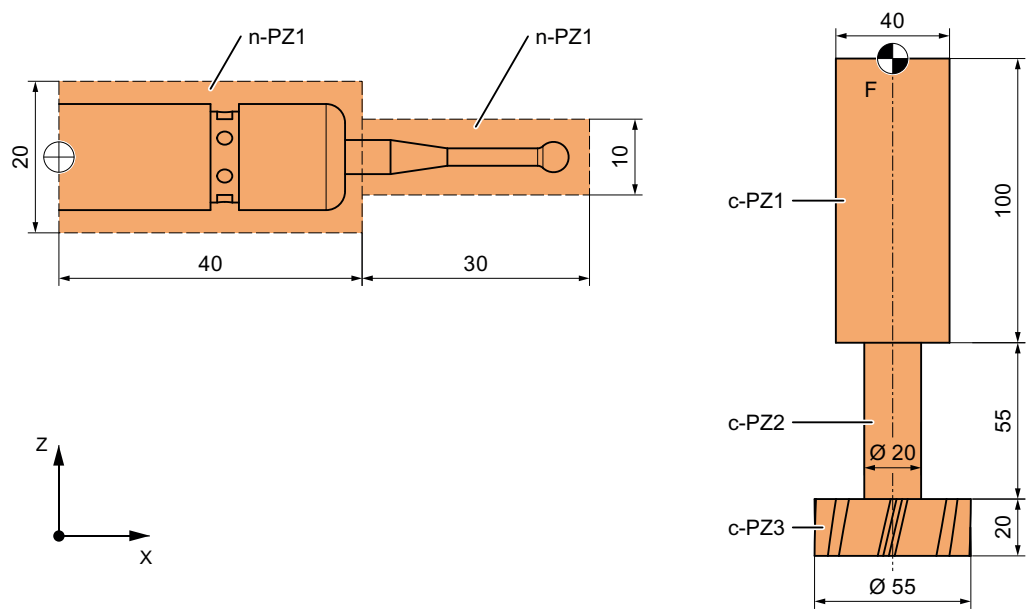
4.6 保護領域

このために、以下のプロテクションゾーンが定義されます。

- 計測プローブホルダ(n-PZ1)と計測プローブ(n-PZ2)両方の、機械別プロテクションゾーンとワーク関連プロテクションゾーン
- フライスカッターホルダ(c-PZ1)、カッターシャンク(c-PZ2)、およびフライスカッター(c-PZ3)のチャネル別プロテクションゾーンと工具関連プロテクションゾーン

すべてのプロテクションゾーンの向きはZ方向です。

機能の起動時の計測プローブの基準点の位置は、X = -120、Y = 60、Z = 80 とします。



① プローブのプロテクションゾーンの名前

F 工具ホルダの基準点

プログラムコード	コメント
DEF INT PROTZONE	; Help 変数の定義
G17	; 加工平面 XY
; プロテクションゾーンの定義	
NPROTDEF (1,FALSE,3,10,-10)	; プロテクションゾーン n-PZ1
G01 X0 Y-10	
X40	
Y10	
X0	
Y-10	
EXECUTE (PROTZONE)	
NPROTDEF (2,FALSE,3,5,-5)	; プロテクションゾーン n-PZ2

プログラムコード	コメント
G01 X40 Y-5	
X70	
Y5	
X40	
Y-5	
EXECUTE (PROTZONE)	
CPROTDEF (1, TRUE, 3, 0, -100)	; プロテクションゾーン c-PZ1
G01 X-20 Y-20	
X20	
Y20	
X-20	
Y-20	
EXECUTE (PROTZONE)	
CPROTDEF (2, TRUE, 3, -100, -150)	; プロテクションゾーン c-PZ2
G01 X0 Y-10	
G03 X0 Y10 J10	
X0 Y-10 J-10	
EXECUTE (PROTZONE)	
CPROTDEF (3, TRUE, 3, -150, -170)	; プロテクションゾーン c-PZ3
G01 X0 Y-27.5	
G03 X0 Y27.5 J27.5	
X0 Y27.5 J-27.5	
EXECUTE (PROTZONE)	
; プロテクションゾーンの起動:	
NPROT (1, 2, -120, 60, 80)	; オフセットを使用してプロテクションゾーン n-PZ1 を起動
NPROT (2.2, -120, 60, 80)	; オフセットを使用してプロテクションゾーン n-PZ2 を起動
CPROT (1, 2, 0, 0, 0)	; プロテクションゾーン c-PZ1 を起動
CPROT (2, 2, 0, 0, 0)	; プロテクションゾーン c-PZ2 を起動
CPROT (3, 2, 0, 0, 0)	; プロテクションゾーン c-PZ3 を起動

詳細情報

コントローラの電源投入後の状態の起動

コントロールシステムの電源を投入し、軸が原点確立した後に、プロテクションゾーンはすでに有効になっている場合があります。これは、プロテクションゾーンで、次のシステム変数が TRUE に設定されている場合に該当します。

- \$SN_PA_ACTIV_IMMED[<n>] (機械別プロテクションゾーン)または
- \$SC_PA_ACTIV_IMMED[<n>] (チャネル別プロテクションゾーン)
インデックス"<n>"はプロテクションゾーンの数に対応します:0 = 1 プロテクションゾーン

4.6 保護領域

プロテクションゾーンは状態=2、オフセットなしで起動します。

プロテクションゾーンの複数起動

機械別プロテクションゾーンは、同時に複数のチャンネルで有効にできます(2面が対向する心押し台のプロテクションゾーンなど)。プロテクションゾーンは、すべてのジオメトリ軸が原点確立済みの場合にのみ監視されます。

単一のチャンネル内で1つのプロテクションゾーンを異なるオフセットで同時に有効にすることはできません。

工具径補正が動作中の場合のプロテクションゾーンの監視

工具径補正の動作中には、工具径補正の平面がプロテクションゾーン定義の平面と同じである場合にかぎり、プロテクションゾーンの監視機能を使用できます。

4.6.3 プロテクションゾーン違反、作業領域リミット、およびソフトウェアリミットスイッチのチェック(CALCPOSI)

機能

ワーク座標系(WCS)で CALCPOSI 機能は、有効なリミットに違反することなく指定した距離、開始位置からジオメトリ軸を移動できるかどうかをチェックします。リミットが原因で距離を完全に移動できない場合、正の10進数コードの状態値と最大許容移動距離が返されます。

定義

```
INT CALCPOSI (VAR REAL[3] <Start>, VAR REAL[3] <Dist>, VAR REAL[5]  
<Limit>, VAR REAL[3] <MaxDist>, BOOL <MeasSys>, INT <TestLim>)
```

構文

```
<Status> = CALCPOSI (VAR <Start>, VAR <Dist>, VAR <Limit>, VAR  
<MaxDist>, <MeasSys>, <TestLim>)
```

意味

CALCPOSI (.. .):		ジオメトリ軸に関する限界値違反をテストするための事前定義された機能	
		先読み停止:	いいえ
		単独ブロック指令:	はい
<status>: (パート 1)		機能の戻り値。負の値は異常状態を示します。	
		データタイプ:	INT
		値の範囲:	$-8 \leq x \leq 100000$
		値:	0 距離をすべて移動できます。
			-1 少なくとも 1 つの成分が<リミット>に違反しています。
			-2 座標変換計算の異常。 例:移動距離が特異点を通過するため、軸位置を定義できません。
			-3 指定された移動距離<距離>と最大許容移動距離<最大距離>は、線形従属です。 注 <リミットのテスト>、ビット 4 == 1 と組み合わせた場合にのみ発生します。
			-4 リミット面までの<距離>に含まれている移動方向の突起がゼロベクトルであるか、移動方向が違反されたリミット面に垂直です。 注 <リミットのテスト>、ビット 5 == 1 と組み合わせた場合にのみ発生します。
			-5 <リミットのテスト>、ビット 4 == 1 およびビット 5 == 1 の場合
			-6 移動リミットのチェックのために考慮が必要な少なくとも 1 つの機械軸が、原点確立されていません。
			-7 干渉の回避機能:キネマティック結合またはプロテクションゾーンの定義が無効です。
			-8 干渉の回避機能:メモリが不足しているため、この命令を実行できません。

4.6 保護領域

<status>: (パート 2)	1 の桁		
	注 複数のリミットに同時に違反している場合、指定された移動距離に最も大きな制限を課すリミットが通知されます。		
	値:	1	ソフトウェアリミットスイッチにより、移動距離が制限されています。
		2	作業領域リミットにより、移動距離が制限されています。
		3	プロテクションゾーンにより、移動距離が制限されています。
		4	干渉の回避機能:プロテクションゾーンにより、移動軌跡が制限されています。
<status>: (パート 3)	10 の桁		
	値:	1x	初期値が制限に違反しています
		2x	指定した直線が制限に違反しています。 終点はどの制限にも違反していないが、始点から終点までの軌跡により制限値の違反が発生する(極座標補間などの非線形座標変換に対して、WCS のソフトウェアリミットスイッチ曲線のプロテクションゾーンを通過するなど)場合でも、この値が返されます。
<status>: (パート 3)	100 の桁		
	値:	1xx	および 1 の桁== 1 または 2 正の制限値に違反しています。
			および 1 の桁== 3 ¹⁾ NC 別のプロテクションゾーンに違反しています。
		2xx	および 1 の桁== 1 または 2 負の制限値に違反しています。
			および 1 の桁== 3 ¹⁾ チャンネル別プロテクションゾーンに違反しています。

<status>: (パート 4)	1000 の桁		
	値:	1xxx	および 1 の桁 == 1 または 2 軸番号に乗算する係数がリミットに違反しています。軸のナンバリングは 1 から始まります。 基準 - ソフトウェアリミットスイッチ:機械軸 - 作業領域リミット:ジオメトリ軸 および 1 の桁 == 3¹⁾ 違反されたプロテクションゾーンの数に乗算する係数。
<status>: (パート 5)	100000 の桁		
	値:	0xxxxx	100000 の桁 == 0: <距離>はそのまま変更されません。
		1xxxxx	方向ベクトルが、リミット面での詳細な移動方向を定義する<距離>で返されます。 以下の補足条件でのみ発生します。 - ソフトウェアリミットスイッチまたは作業領域リミットに違反した(始点以外で)。 - 座標変換が有効でない。 <TestID>、ビット 4 またはビット 5 == 1
<開始>:	開始位置のベクトルの参照		
	<開始> [0]:1 番目のジオメトリ軸 <開始> [1]:2 番目のジオメトリ軸 <開始> [2]:3 番目のジオメトリ軸		
	パラメータタイプ:	入力	
	データタイプ:	VAR REAL [3]	
	値の範囲:	-最大 REAL 値 ≤ x[<n>] ≤ +最大 REAL 値	

4.6 保護領域

<距離>:	ベクトルの参照。	
	入力: インクレメンタル移動距離 • <距離> [0]:1 番目のジオメトリ軸 • <距離> [1]:2 番目のジオメトリ軸 • <距離> [2]:3 番目のジオメトリ軸	
	出力 (<状態>で 100000 桁を設定している場合のみ)	
	<距離>には、出力値として単位ベクトル v が含まれ、これが WCS 内での詳細な移動方向を定義します。	
	ケース 1: <TestID>、ビット 4 == 1 の場合のベクトル v の形成 入力ベクトル<距離>と<最大距離>は移動平面にかかります。この平面は、違反されたリミット面によって切断されます。2 つの平面の交線によって、ベクトル v の方向が定義されます。方向(符号)は、入力ベクトル<最大距離>と v の間の角度が 90°を超えないよう選択されます。	
	ケース 2: <TestID>、ビット 5 == 1 の場合のベクトル v の形成 ベクトル v は、リミット面上の<距離>に含まれる移動ベクトルの突起方向の単位ベクトルです。リミット面上の移動ベクトルの突起がゼロベクトルの場合、アラームが返されます。	
	パラメータタイプ:	入力/出力
	データタイプ:	VAR REAL [3]
	値の範囲:	-最大 REAL 値 ≤ x[<n>] ≤ +最大 REAL 値

<リミット>:	長さ 5 の配列の参照 • <リミット> [0 - 2]:リミットまでのジオメトリ軸の最小スペース – <リミット> [0]:1 番目のジオメトリ軸 – <リミット> [1]:2 番目のジオメトリ軸 – <リミット> [2]:3 番目のジオメトリ軸 最小スペースは、次のもので遵守されます。 – 作業領域リミット:制限なし – ソフトウェアリミットスイッチ:座標変換が有効でない場合や、5 軸座標変換など、直線機械軸へのジオメトリ軸の明確な割り当てが可能な座標変換が有効な場合。 • <リミット> [3]:たとえば、非線形座標変換のためにジオメトリ軸を割り当てられない直線機械軸の最小スペースが含まれています。この値は、従来のプロテクションゾーンや干渉回避プロテクションゾーンの監視用の制限値にも使用されます。 • <リミット> [4]:たとえば、非線形座標変換のためにジオメトリ軸を割り当てられない回転機械軸の最小スペースが含まれています。 注 この値は、特殊な座標変換用のソフトウェアリミットスイッチの監視にだけ有効です。
パラメータタイプ:	入力
データタイプ:	VAR REAL [5]
値の範囲:	-最大 REAL 値 ≤ x[n] ≤ +最大 REAL 値

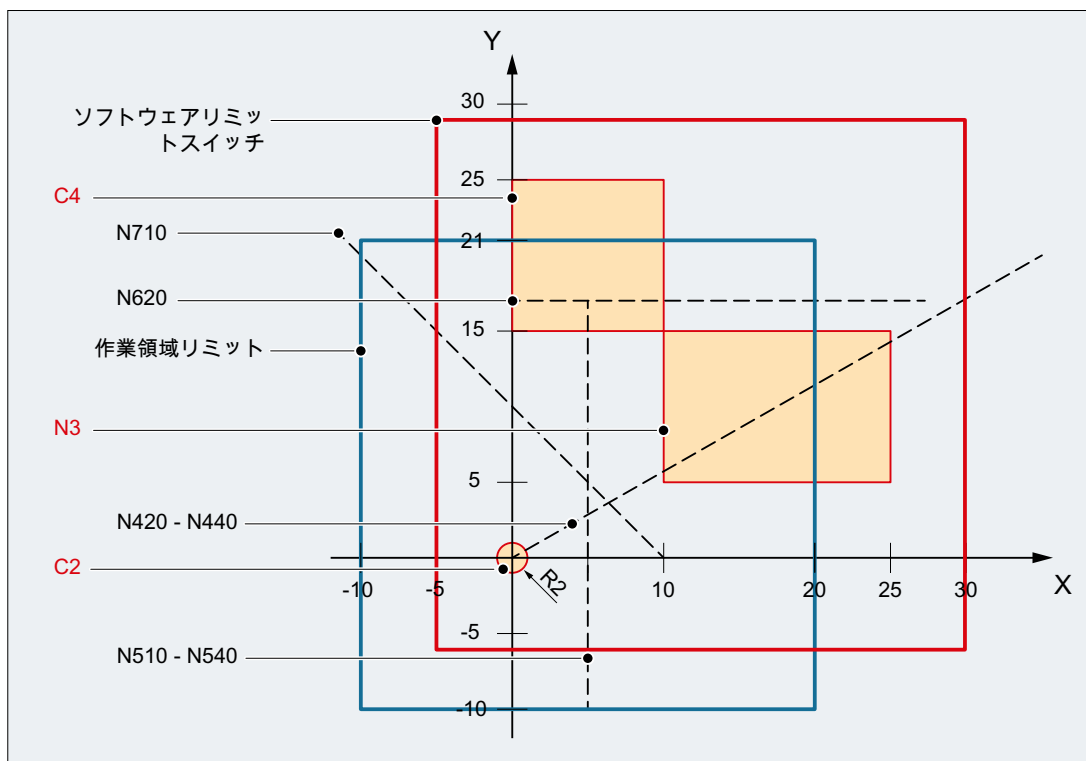
4.6 保護領域

<最大距離>:	軸リミットの指定された最小スペースが、関連する機械軸のどれにも違反していないインクリメンタル移動距離のベクトルを参照。	
	• <距離> [0]:1 番目のジオメトリ軸 • <距離> [1]:2 番目のジオメトリ軸 • <距離> [2]:3 番目のジオメトリ軸	
	移動距離に制限がない場合、この戻りパラメータの内容は<距離>の内容と同じです。	
	<テスト ID>、ビット 4 == 1 の場合: <距離>と<最大距離> <最大距離>と<距離>には、移動平面にかかるベクトルを入力値として含めてください。2 つのベクトルは、互いに線形従属しないものとします。<最大距離>の絶対値は任意です。移動方向の計算については、<距離>の説明を参照してください。	
	パラメータタイプ:	出力
<MeasSys>:	データタイプ:	VAR REAL [3]
	値の範囲:	-最大 REAL 値 ≤ x[<n>] ≤ +最大 REAL 値
	位置と距離の指定のための単位系(インチ/メトリック)(オプション)	
	データタイプ:	BOOL
値:	FALSE (デフォルト)	G グループ 13(G70、G71、G700、G710)の現在有効な G 命令に対応する単位系です。 注 G70 が有効で基本単位系がメトリックの場合(または、G71 が有効で基本単位系がインチの場合)、システム変数\$AA_IW と\$AA_MW は基本単位系で提供され、使用する場合、CALCPOSI 用に変換してください。
	TRUE	基本単位系に準拠する単位系: MD52806 \$MN_ISO_SCALING_SYSTEM

<リミットのテスト>:	監視するリミットのビット指定選択(オプション)	
	データタイプ:	INT
	初期値:	ビット 0、1、2、3、6、7 == 1 (207)
	ビ ツ ト	10 進数 意味
	0	1 ソフトウェアリミットスイッチ
	1	2 作業領域リミット
	2	4 起動した従来のプロテクションゾーン
	3	8 起動待ちの従来のプロテクションゾーン
	4	16 <距離>のソフトウェアリミットスイッチまたは作業領域リミットに違反した場合、 ケース 1 に示すような移動方向が返されます(上記参照)。
	5	32 <距離>のソフトウェアリミットスイッチまたは作業領域リミットに違反した場合、 ケース 2 に示すような移動方向が返されます(上記参照)。
	6	64 起動された干渉回避プロテクションゾーン
	7	128 起動待ちの干渉回避プロテクションゾーン
	8	256 起動された干渉回避プロテクションゾーンと起動待ちの干渉回避プロテクションゾーンのペア
¹⁾ 複数のプロテクションゾーンに同時に違反している場合、指定された移動距離に最も大きな制限を課すプロテクションゾーンが返されます。		

例

リミット位置



この例では、X-Y平面内の有効なソフトウェアリミットスイッチおよび作業領域リミットと、次の3つのプロテクションゾーンが表示されています。

- C2:工具関連のチャンネル別プロテクションゾーン、有効、円形、半径= 2 mm
- C4:ワーク関連のチャンネル別プロテクションゾーン、起動待ち、正方形、辺長= 10 mm
- N3:機械別のプロテクションゾーン、有効、長方形、辺長= 10 mm x 15 mm

NC プログラム

NC プログラムでは最初にプロテクションゾーンと作業領域リミットが定義されています。その後、さまざまなパラメータ設定で CALCPOSI () 機能が呼び出されます。

プログラムコード

```

N10 DEF REAL _START[3]
N20 DEF REAL _DIST[3]
N30 DEF REAL _LIMIT[5]
N40 DEF REAL _MAXDIST[3]
N50 DEF INT _PA
N60 DEF INT _STATUS

```

プログラムコード

```
:工具関連プロテクションゾーン C2
N70 CPROTDEF(2, TRUE, 0)
N80 G17 G1 X-2 Y0
N90 G3 I2 X2
N100 I-2 X-2
N110 EXECUTE(_PA)
; ワーク関連プロテクションゾーン C4
N120 CPROTDEF(4, FALSE, 0)
N130 G17 G1 X0 Y15
N140 X10
N150 Y25
N160 X0
N170 Y15
N180 EXECUTE(_PA)
; 機械別プロテクションゾーン N3
N190 NPROTDEF(3, FALSE, 0)
N200 G17 G1 X10 Y5
N210 X25
N220 Y15
N230 X10
N240 Y5
N250 EXECUTE(_PA)
; プロテクションゾーンを起動または起動待ちにします
N260 CPROT(2, 2, 0, 0, 0)
N270 CPROT(4, 1, 0, 0, 0)
N280 NPROT(3, 2, 0, 0, 0)
; 作業領域リミットを定義します
N290 G25 XX=-10 YY=-10
N300 G26 XX=20 YY=21
N310 _START[0] = 0.
N320 _START[1] = 0.
N330 _START[2] = 0.
N340 _DIST[0] = 35.
N350 _DIST[1] = 20.
N360 _DIST[2] = 0.
N370 _LIMIT[0] = 0.
N380 _LIMIT[1] = 0.
N390 _LIMIT[2] = 0.
N400 _LIMIT[3] = 0.
N410 _LIMIT[4] = 0.
N420 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST)
N430 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,3)
N440 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,1)
N450 _START[0] = 5.
N460 _START[1] = 17.
N470 _START[2] = 0.
```

4.6 保護領域

プログラムコード

```
N480 _DIST[0] = 0.
N490 _DIST[1] = -27.
N500 _DIST[2] = 0.

N510 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,14)
N520 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,, 6)
N530 _LIMIT[1] = 2.
N540 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,, 6)
N550 _START[0] = 27.
N560 _START[1] = 17.1
N570 _START[2] = 0.

N580 _DIST[0] = -27.
N590 _DIST[1] = 0.
N600 _DIST[2] = 0.
N610 _LIMIT[3] = 2.
N620 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,12)
N630 _START[0] = 0.
N640 _START[1] = 0.
N650 _START[2] = 0.
N660 _DIST[0] = 0.
N670 _DIST[1] = 30.
N680 _DIST[2] = 0.
N690 TRANS X10
N700 AROT Z45

N710 _STATUS = CALCPOSI(_START,_DIST, _LIMIT, _MAXDIST)
; N690 および N700 からフレームを再度解除します
N720 TRANS

N730 _START[0] = 0.
N740 _START[1] = 10.
N750 _START[2] = 0.
; ベクトル _DIST および _MAXDIST が移動平面を定義します
N760 _DIST[0] = 30.
N770 _DIST[1] = 30.
N780 _DIST[2] = 0.
N790 _MAXDIST[0] = 1.
N800 _MAXDIST[1] = 0.
N810 _MAXDIST[2] = 1.
N820 _STATUS = CALCPOSI(_START, _DIST, _LIMIT, _MAXDIST,,17)
N830 M30
```

CALCPOSI()の結果

N.. .	<状態>	<最大距離>[0] ≒ X	<最大距離>[1] ≒ Y	備考
42 0	3123	8.040	4.594	N3 に違反しています。
43 0	1122	20.000	11.429	保護領域が監視されず、ワーキングエリアリミットに違反しています。
44 0	1121	30.000	17.143	ソフトウェアリミット監視のみが依然として有効です。
51 0	4213	0.000	0.000	起点が C4 に違反しています。
52 0	0000	0.000	-27.000	起動待ちの C4 が監視されていません。指定された距離をすべて移動できます。
54 0	2222	0.000	-25.000	_LIMIT[1] = 2 のために、移動距離が作業領域リミットによって制限されます。
62 0	4223	-13.000	0.000	C4 までのスペースは、C2 と _LIMIT[3]により、合計 4 mm です。C2 → N3 の 0.1 mm のスペースのために移動距離が制限されることはありません。
71 0	1221	0.000	21.213	平行移動と回転のあるフレームが有効です。_DIST で許容される移動距離は、平行移動および回転した WCS に適用されます。
82 0	102121	18.000	18.000	Y 軸のソフトウェアリミットスイッチに違反しています。詳細な移動方向の計算が<_TESTLIM>=17 で要求されます。この方向は_DIST (0.707, 0.0, 0.707)です。これは、<_STATUS>で 100000 桁が設定されているため、有効です。

他の情報

「原点確立された」軸の状態

CALCPOSI () によって考慮されるすべての機械軸を原点確立してください。

円弧関連距離の指定

すべての円弧関連距離の指定は、常に半径指定として解釈されます。このことは、特に直径指定(DIAMON/DIAM90)が有効になっている径方向軸の場合に考慮してください。

移動距離の縮小

ある軸の指定された移動距離が制限されている場合は、それに比例して他の軸の移動距離も<最大距離>の戻り値で縮小されます。したがって、得られる終点はまだ指定された軌跡上にあります。

回転軸

回転軸は、モジュロ回転軸でない場合にのみ監視されます。

ソフトウェアリミットスイッチ、作業領域リミットおよびプロテクションゾーンのいずれも、関連する複数の軸に対して定義しないこともできます。

ソフトウェアリミットスイッチおよび作業領域リミットの状態

ソフトウェアリミットスイッチと作業領域リミットは、CALCPOSI () の実行中に有効である場合にのみ考慮されます。状態は、たとえば、次のものの影響を受けることがあります。

- マシンデータ:MD21020 \$MC_WORKAREA_WITH_TOOL_RADIUS
- セッティングデータ: \$AC_WORKAREA_CS_...
- NC/PLC インタフェース信号 DB31、... DBX12.2 / 3
- 命令:WALIMON / WALIMOF

ソフトウェアリミットスイッチと座標変換

CALCPOSI () では、さまざまなキネマティックトランスフォーメーション(TRANSMIT など)の場合、移動距離の特定の位置であいまいさが生じるため、ジオメトリ軸(WCS)の位置から機械軸(MCS)の位置を明確に特定できるとはかぎりません。通常の移動動作では、履歴、および WCS の連続移動が MCS の連続移動に対応する、という条件から一義的にならないのが一般的です。したがって、ソフトウェアリミットスイッチを監視するときは、こ

のような場合のあいまいさを解決するために CALCPOSI () 実行時の機械位置が使用されます。

注記

先読み停止

座標変換と組み合わせて CALCPOSI () を使用する場合、機械軸位置の同期のために、CALCPOSI () の前に先読み停止(STOPRE)を先読みと共にプログラムすることは、使用者の責任です。

プロテクションゾーンのスペースと従来のプロテクションゾーン

従来のプロテクションゾーンでは、パラメータ<リミット>[3]で設定した安全距離が、指定された軌跡上の移動動作中、すべてのプロテクションゾーンについて保持されるという保証は**ありません**。保証されるのは、<距離>で返される終点が移動方向に安全距離だけ延長されたときに、プロテクションゾーンに違反しないということだけです。ただし、直線がプロテクションゾーンのすぐ近くを通過することがあります。

プロテクションゾーンのスペースと干渉回避プロテクションゾーン

干渉回避プロテクションゾーンでは、パラメータ<リミット>[3]で設定した安全距離が、指定された軌跡上の移動動作中、すべてのプロテクションゾーンについて保持されるという保証があります。

パラメータ<リミット>[3]で指定された安全距離は、以下が適用される場合にのみ有効になります。

<リミット>[3] > (MD10619 \$MN_COLLISION_TOLERANCE)

ビット 4 がパラメータ<TestLim>で設定されている場合にのみ(実行中の移動方向の計算)、数十万の数値が機能の戻り値(<status>)で設定されているときに、<DIST>で受信した方向ベクトルは有効です。プロテクションゾーンに違反があったため、または座標変換が有効であるために、このような方向を決定できない場合、<DIST>の入力値は変更されないままです。追加エラーメッセージは出力されません。

4.7 特殊動作命令

4.7.1 符号化位置へのアプローチ(CAC、CIC、CDC、CACP、CACN)

「符号化位置へのアプローチ」の軌跡命令を使用すると、位置番号を指定して、マシンデータテーブルに保存されている固定軸に直線軸と回転軸を移動することができます。

4.7 特殊動作命令

構文

CAC (<n>)
 CIC (<n>)
 CACP (<n>)
 CACN (<n>)

意味

CAC (<n>):	位置番号 n からの符号化位置へのアプローチ
CIC (<n>):	実位置番号から前方(+ n)または後方(- n)へ、符号化位置 n の場所へアプローチします
CDC (<n>):	位置番号 n から最短軌跡に沿って指定位置へアプローチします (回転軸の場合のみ)
CACP (<n>):	位置番号 n から正の方向に符号化位置へアプローチします(回転軸の場合のみ)
CACN (<n>):	位置番号 n から負の方向に符号化位置へアプローチします(回転軸の場合のみ)
<n>:	マシンデータ表内の位置番号 値の範囲:0, 1, ... (表の位置の最大番号- 1)

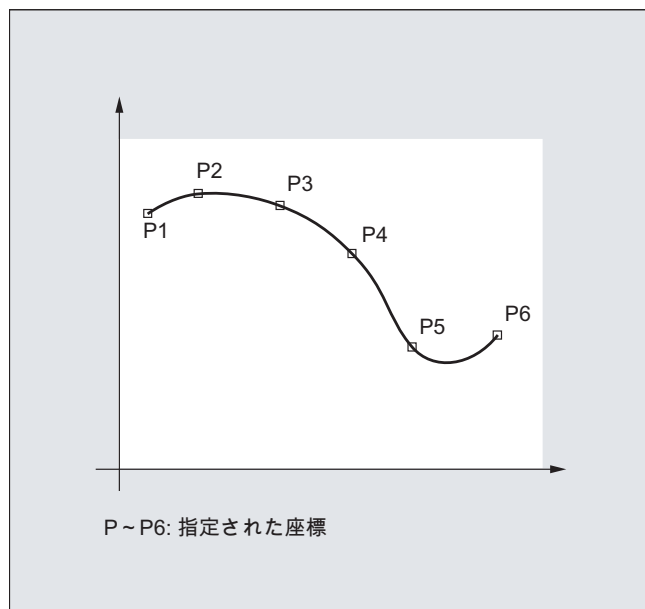
例:位置決め軸の符号化位置へのアプローチ

プログラミングコード	コメント
N10 FA[B]=300	; 位置決め軸 B の送り速度
N20 POS[B]=CAC(10)	; 位置番号 10 の符号化位置へアプローチ
N30 POS[B]=CIC(-4)	; 「現在の位置番号」 - 4 の符号化位置へアプローチ

4.7.2 スプライン補間(ASPLINE、BSPLINE、CSPLINE、BAUTO、BNAT、BTAN、EAUTO、ENAT、ETAN、PW、SD、PL)

不規則な曲線のワーク輪郭は、解析形式で正確には定義できません。このため、表面を数値化するときなどは、これらのタイプの輪郭として、曲線上の一定の数の点を使用した近似値が求められます。数値化されたワークの表面を生成するには、曲線上の点を接続して輪郭を定義します。このために、スプライン補間を使用できます。

スプラインは、2次または3次多項式で形成される曲線を定義します。スプラインの曲線上の点の特性は、**使用するスプラインタイプ**に応じて定義できます。



SINUMERIK ソリューションラインの場合は、次のスプラインタイプを使用できます。

- A スプライン
- B スプライン
- C スプライン

構文

一般:

```
ASPLINE X...Y...Z...A ...B ...C...
BSPLINE X...Y...Z...A ...B ...C...
CSPLINE X...Y...Z...A ...B ...C...
```

B スプラインでは、以下をさらにプログラム指令できます。

PW=<n>

SD=2

PL=<値>

A と C スプラインでは、以下をさらにプログラム指令できます。

BAUTO / BNAT / BTAN

EAUTO / ENAT / ETAN

4.7 特殊動作命令

意味

スプライン補間タイプ:				
ASPLINE:		A スプライン補間を起動する命令		
BSPLINE:		B スプライン補間を起動する命令		
CSPLINE:		C スプライン補間を起動する命令		
		ASPLINE、BSPLINE、および CSPLINE 命令はモーダルに有効で、動作命令のグループに属します。		
曲線上の点とチェック点:				
X...Y...Z...		直交座標の位置		
A...B ...C...				
点の重み(B スプラインのみ):				
PW:		PW 命令を使用すると、曲線上の各点に「点の重み」をプログラム指令できます。		
<n>:		「点の重み」		
		値の範囲:	$0 \leq n \leq 3$	
		ステップ値:	0.0001	
		動作:	$n > 1$	曲線がチェック点に大きく引き付けられます。
			$n < 1$	曲線はチェック点にそれほど引き付けられません。
スプラインの次数(B スプラインのみ):				
SD:		3 次多角形を標準として使用しますが、SD=2 をプログラム指令すると、2 次多角形も使用できます。		
ノード間の距離(B スプラインのみ):				
PL:		ノード間の距離を、内部で適切に計算します。制御装置は PL 命令を使用して、いわゆるパラメータ区間の長さで指定した予約ノードスペースも加工できます。		
<値>:		パラメータ区間の長さ		
		値の範囲:	軌跡寸法と同じ	

スプライン曲線の開始の遷移動作(A または C スプラインのみ):	
BAUTO:	遷移動作は指定されていません。開始を特定するのは、1 番目の点の位置です。
BNAT:	曲率ゼロ
BTAN:	前のブロックの接線方向の遷移(解除位置)
スプライン曲線の終わりの遷移動作(A または C スプラインのみ):	
EAUTO:	遷移動作は指定されていません。終わりを特定するのは、最後の点の位置です。
ENAT:	曲率ゼロ
ETAN:	前のブロックの接線方向の遷移(解除位置)

注記

プログラム指令可能な遷移動作は、B スプラインには影響しません。B スプラインは、その始点と終点で常に、指令多角形に接します。

4.7 特殊動作命令

必要条件

- 工具径補正を使用できます。
- 衝突監視が、投影された平面で実行されます。

例

例 1:B スプライン

プログラムコード 1 (重みはすべて 1 です)

```
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
```

プログラムコード 2 (さまざまな重みがあります)

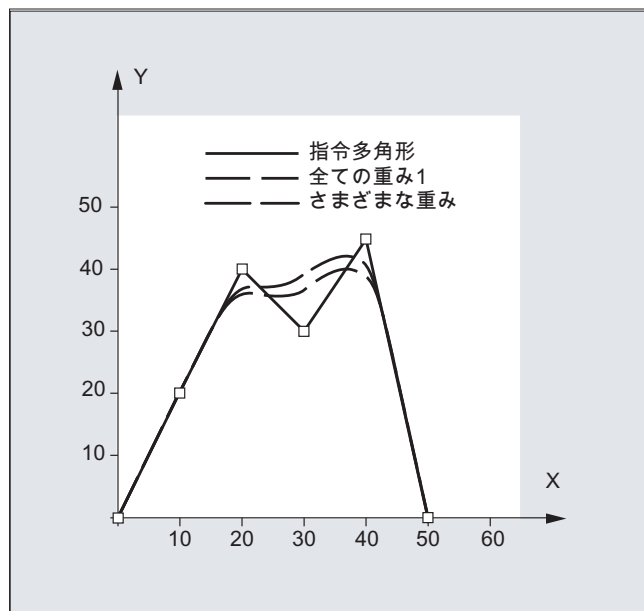
```
N10 G1 X0 Y0 F300 G64
N20 BSPLINE
N30 X10 Y20 PW=2
N40 X20 Y40
N50 X30 Y30 PW=0.5
N60 X40 Y45
N70 X50 Y0
```

プログラムコード 3 (指令多角形)

コメント

```
N10 G1 X0 Y0 F300 G64
N20
N30 X10 Y20
N40 X20 Y40
N50 X30 Y30
N60 X40 Y45
N70 X50 Y0
```

; 該当なし



例 2 :C スプライン、始まりと終わりでの曲率ゼロ

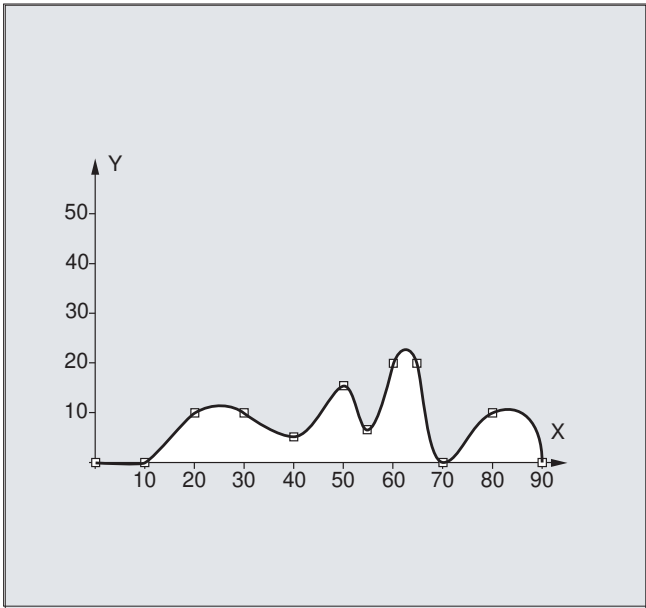
プログラムコード

```

N10 G1 X0 Y0 F300
N15 X10
N20 BNAT ENAT
N30 CSPLINE X20 Y10
N40 X30
N50 X40 Y5
N60 X50 Y15
N70 X55 Y7
N80 X60 Y20
N90 X65 Y20
N100 X70 Y0
N110 X80 Y10
N120 X90 Y0
N130 M30

```

4.7 特殊動作命令



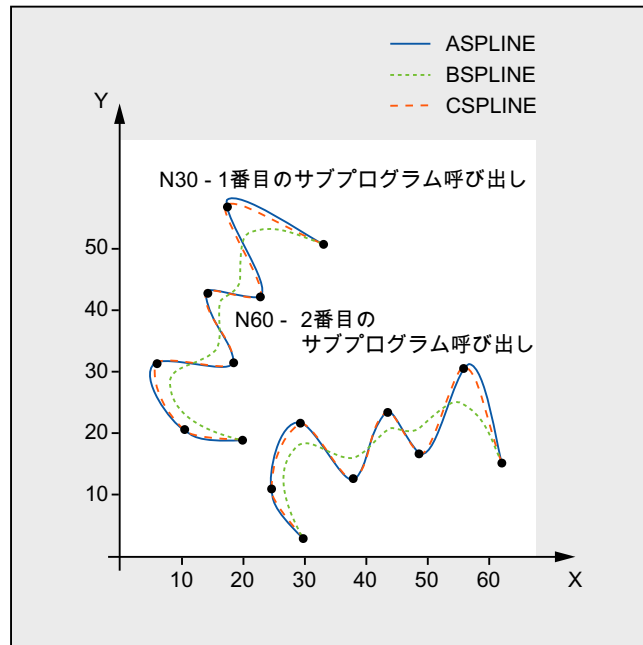
例 3 :スプライン補間(A スプライン)と座標変換(ROT)
メインプログラム

プログラムコード	コメント
N10 G00 X20 Y18 F300 G64	; 起点にアプローチします。
N20 ASPLINE	; 補間タイプの A スプラインを有効にします。
N30 CONTOUR	; 1 番目のサブプログラムを呼び出します。
N40 ROT Z-45	; 座標変換:Z 軸を中心として-45°の WCS を回転します。
N50 G00 X20 Y18	; 輪郭の起点へアプローチします。
N60 CONTOUR	; 2 番目のサブプログラムを呼び出します。
N70 M30	; プログラム終了

サブプログラム「contour」(曲線上の点の座標を含む):

プログラムコード
N10 X20 Y18
N20 X10 Y21
N30 X6 Y31
N40 X18 Y31
N50 X13 Y43
N60 X22 Y42
N70 X16 Y58
N80 X33 Y51
N90 M1

上記の例のプログラム(ASPLINE)で得られたスプライン曲線に加え、以下の図には、B または C スプライン補間(BSPLINE、CSPLINE)を起動すると得られるスプライン曲線も含まれます。



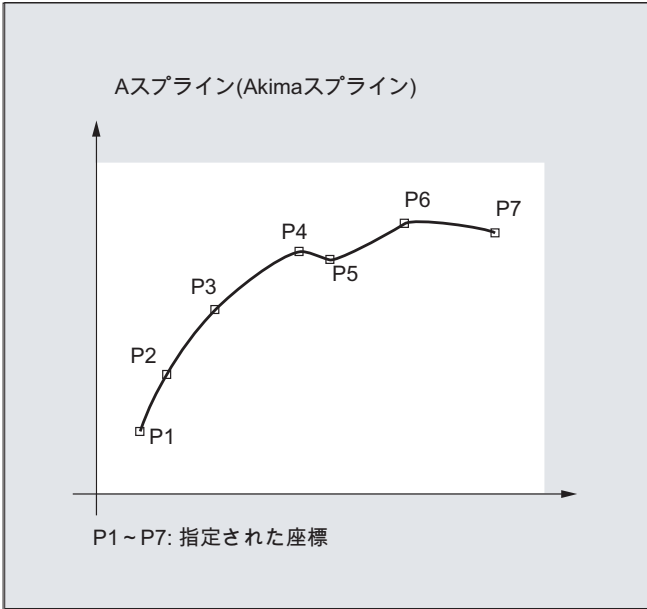
詳細情報

スプライン補間の長所

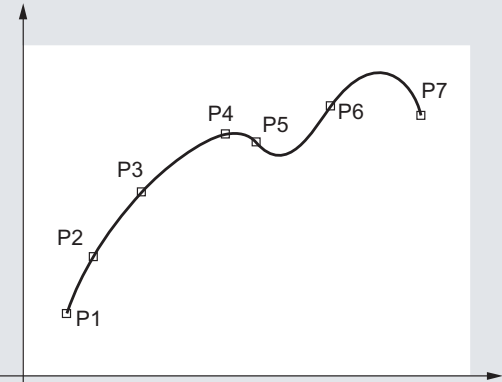
スプライン補間を使用すると、直線ブロック G01 を使用した場合とは対照的に、次の長所が得られます。

- より少ないパートプログラムブロックの数で輪郭が記述できます。
- なめらかな曲線特性により、パートプログラムブロック間遷移で、機械システムへの負担が軽減されます。

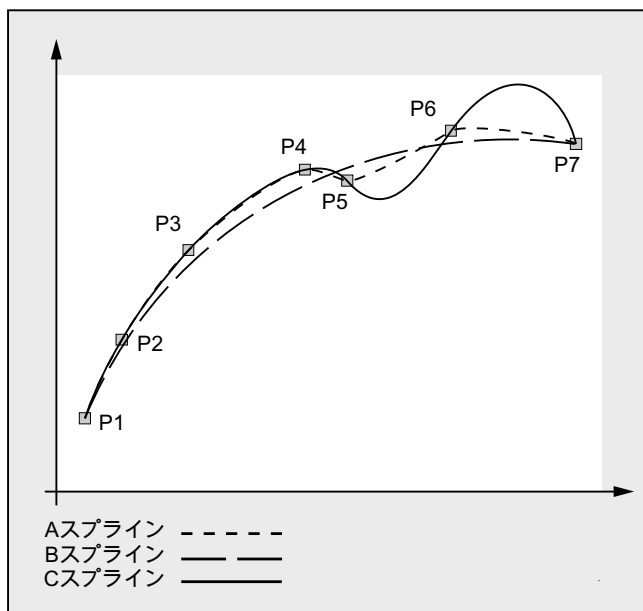
さまざまなスプライントイプの機能と用途

スプライントイプ	機能と使用方法
A スプライン	Aスプライン(Akimaスプライン)  P1～P7: 指定された座標 特性: • 曲線上の指定した中間点を正確に通過します。 • 曲線特性は接線方向ですが、曲率は連続しません。 • 不必要な振動はほとんど発生しません。 • 曲線上の中間点の変化が影響する範囲は局部的です。つまり、曲線上の中間点の変化が影響するのは、隣接する 6 個までの中間点のみです。 用途: A スプラインは特に、大きく変化する曲線(階段状タイプの曲線特性など)の補間に適しています。

スプラインタイプ	機能と使用方法
B スプライン	Bスプライン P1 ~ P7: 指定された座標 特性: • 曲線上の指定した中間点は通過しませんが、その近くを通過します。曲線は中間点に引きつけられません。曲線特性はさらに、係数による中間点の重み付けにより影響されます。 • 曲線特性は、曲率は連続して接線方向です。 • 不必要な振動はまったく発生しません。 • 曲線上の中間点の変化が影響する範囲は局部的です。つまり、曲線上の中間点の変化が影響するのは、隣接する 6 個までの中間点のみです。 用途: B スプラインの主な用途は、CAD システムとのインタフェースです。

スプラインタイプ	機能と使用方法
C スプライン	Cスプライン(3次スプライン)  P1 ~ P7: 指定された座標 特性: • 曲線上の指定した中間点を正確に通過します。 • 曲線特性は、曲率は連続して接線方向です。 • 不必要な振動が、特に曲線が大きく変化する位置でよく発生します。 • 中間点の変化が影響する範囲は全体にわたります。つまり、中間点 が変化した場合は、それが曲線特性全体に影響します。 用途: C スプラインは、解析で定義される曲線(円、放物線、双曲線)上に中 間点がある場合の使用に適しています。

同じ補間点をもつ3つのスプラインタイプの比較演算



スプラインブロックの最小数

G コード ASPLINE、BSPLINE、および CSPLINE は、ブロック終点をスプラインでリンクします。このために、一連のブロック(終点)は同時に計算されます。計算のバッファサイズは、標準では 10 ブロックです。すべてのブロック情報がスプライン終点となるわけではありません。ただし、制御装置には、10 ブロックごとに特定の数のスプライン終点ブロックが必要です。

スプライン タイプ	スプラインブロックの最小数
A スプライン:	各 10 ブロックのうち、 4 ブロック以上をスプラインブロックにしてください。 これには、コメントブロックやパラメータ計算は含まれません。
B スプライン:	各 10 ブロックのうち、 6 ブロック以上をスプラインブロックにしてください。 これには、コメントブロックやパラメータ計算は含まれません。
C スプライン:	必要な最小スプラインブロックの数は、以下の合計です。 MD20160 \$MC_CUBIC_SPLINE_BLOCKS の値+ 1 スプラインセグメントを計算する点の数を MD20160 に入力します。初期設定は 8%です。各 10 ブロックのうち、 9 ブロック以上をスプラインブロックにしてください。

4.7 特殊動作命令

注記

許容値を下回ると同時に、スプラインで使用する軸のいずれかを位置決め軸としてプログラム指令している場合は、アラームが発生します。

短いスプラインブロックの結合

スプライン補間に短いスプラインブロックが発生し、軌跡速度が必要以上に減速するおそれがあります。「短いスプラインブロックの結合」機能を使用すると、これらのブロックが十分な長さになるように結合して、軌跡速度の減速を防止できます。

この機能は、次のチャンネル別マシンデータで有効にします。

MD20488 \$MC_SPLINE_MODE (スプライン補間の設定)

詳細情報:機能マニュアル、基本機能

4.7.3 スプライングループ(SPLINEPATH)

スプライングループで補間する軸を、SPLINEPATH 命令で選択します。スプライン補間のグループ化に使用できるのは、8 つまでの軌跡軸です。

注記

SPLINEPATH を明示的にプログラム指令していない場合は、チャンネルの最初の 3 軸がスプライングループとして移動します。

構文

スプライングループは個別のブロックで定義します。

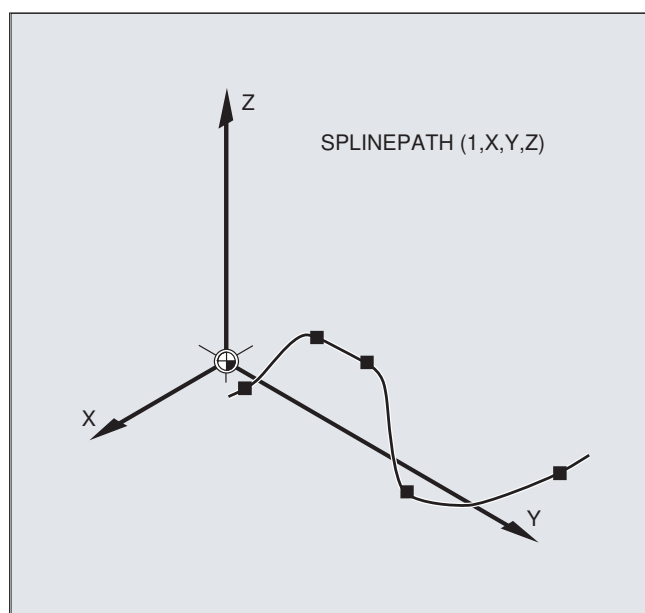
SPLINEPATH (n, X, Y, Z, ...)

意味

SPLINEPATH:	スプライングループを定義する命令
n:	=1 (固定値)
X, Y, Z, ... :	スプライングループで補間する軌跡軸の識別子

例:3 つの軌跡軸によるスプライングループ

プログラムコード	コメント
N10 G1 X10 Y20 Z30 A40 B50 F350	
N11 SPLINEPATH(1,X,Y,Z)	; スプライングループ
N13 CSPLINE BAUTO EAUTO X20 Y30 Z40 A50 B60	; C スプライン:
N14 X30 Y40 Z50 A60 B70	; 中間点
...	
N100 G1 X... Y...	; スプライン補間を選択解除



4.7.4 NCブロック圧縮(COMPON、COMPCURV、COMPCAD、COMPSURF、COMPOF)の有効化/無効化

直線ブロック(さらにパラメータ設定に応じて、円弧、早送りブロック)を圧縮する機能は、G グループ 30 の G 命令を使用して起動/解除します。これらの命令はモーダルです。

構文

```

COMPON / COMPCURV / COMPCAD / COMPSURF
...
COMPOF

```

4.7 特殊動作命令

意味

COMPON:	コンプレッサ機能 COMPON を有効にします
COMPCURV:	コンプレッサ機能 COMPCURV を有効にします
COMPCAD:	コンプレッサ機能 COMPCAD を有効にします
COMPSURF:	コンプレッサ機能 COMPSURF を有効にします
COMPOF:	現在動作中のコンプレッサ機能を解除します。

注記

スムージング機能 G642 と加々速度制限 SOFT を使用して、加工面品質をさらに向上させることができます。これらの命令は、プログラムの先頭に記述してください。

例:COMPCAD

プログラムコード	コメント
N10 G00 X30 Y6 Z40	
N20 G1 F10000 G642	; 起動:スムージング機能 G642
N30 SOFT	; 起動:加々速度制限 SOFT
N40 COMPCAD	; 起動:コンプレッサ機能 COMPCAD
N50 FIFCTRL	
N24050 Z32.499	; 1 番目の移動ブロック
N24051 X41.365 Z32.500	; 2 番目の移動ブロック
...	
N99999 X...Z...	; 最後の移動ブロック
COMPOF	; コンプレッサ機能をオフ
...	

4.7.5 多項式補間(POLY、POLYPATH、PO、PL)

実際には多項式補間(POLY)が使用され、スプライン補間タイプは使用されません。その主な目的は、外部で生成され、しかもスプライン区間を直接記述できるスプライン曲線をプログラム指令するための、インタフェースとなることです。

この補間モードを使用すると、NC が多項式係数の計算処理から解放されます。これは、CAD システムまたはポストプロセッサから直接、係数が供給される場合に適切に使用できます。

構文

3 次多項式:

POLY PO[X]=(xe,a2,a3) PO[Y]=(ye,b2,b3) PO[Z]=(ze,c2,c3) PL=n

5 次多項式と新しい多項式の構文:

POLY X=PO(xe,a2,a3,a4,a5) Y=PO(ye,b2,b3,b4,b5)

Z=PO(ze,c2,c3,c4,c5) PL=n

POLYPATH("AXES","VECT")

注記

NC ブロックでプログラム指令された多項式係数と軸の合計が、ブロック毎に許可される軸の最大数を超えないようにしてください。

意味

POLY :	POLY を含むブロックによる多項式補間を適用します。
POLYPATH :	AXIS と VECT の両方の軸グループに対して多項式補間を選択可能です。
PO[軸識別子/変数] :	終点と多項式係数
X, Y, Z :	軸識別子
xe, ye, ze :	特定の軸の終点の指定。軌跡寸法としての数値範囲
a2, a3, a4, a5 :	係数 a_2 、 a_3 、 a_4 、および a_5 は軌跡寸法としての数値範囲の値で記述されます。それぞれの最後の係数がゼロの場合は、その係数を省略できます。
PL:	多項式が定義されたパラメータ区間の長さ (関数 $f(p)$ の定義範囲) です。 この区間は常に 0 で始まり、p には 0~PL の値を入れることができます。 PL の理論的な数値の範囲: 0.0001~99 999.9999 注: PL の値は、それが配置されているブロックに適用されます。PL をプログラム指令しない場合は、PL=1 が適用されます。

多項式補間の有効化/無効化

多項式補間は、パートプログラムで POLX G 命令を使用して有効にします。

POLY G 命令は、G0、G1、G2、G3、ASPLINE、BSPLINE 、および CSPLINE と共に第 1 グループに属します。

名称と終点のみでプログラム指令された軸(X10 など)は、直線で移動します。NC ブロックの軸をすべて、このようにプログラム指令した場合は、コントローラは G1 の場合と同様に動作します。

多項式補間は、第 1 G グループの別の命令(G0、G1)をプログラム指令すると、再度自動的に無効化されます。

多項式係数

PO 値 (PO [=]) または ...=PO (...) は、軸のすべての多項式係数を指定します。多項式の次数に対応する複数の値を、コンマで区切って指定します。1 ブロック内で、さまざまな軸にさまざまな次数の多項式を指定できます。

POLYPATH サブプログラム

POLYPATH (...) を使用すると、次のように、特定の軸グループに対して、多項式補間を選択して解放できます。

軌跡軸と追加軸のみ:

POLYPATH ("AXES")

旋回軸のみ:

POLYPATH ("VECT")

(方向座標変換での移動時)

解放されない軸は直線で移動します。

多項式補間は両方の軸グループに対して、標準で有効になっています。

多項式補間は、POLYPATH () がパラメータなしでプログラム指令された場合、すべての軸に対して無効です。

例

プログラムコード	コメント
N10 G1 X... Y... Z... F600	
N11 POLY PO[X]=(1,2.5,0.7) PO[Y]=(0.3,1,3.2) PL=1.5	; 多項式補間をオン
N12 PO[X]=(0,2.5,1.7) PO[Y]=(2.3,1.7) PL=3	
...	

プログラムコード	コメント
N20 M8 H126 ...	
N25 X70 PO[Y]=(9.3,1,7.67) PL=5	; 軸の混合データ
N27 PO[X]=(10,2.5) PO[Y]=(2.3)	; PL のプログラム指令なし、PL=1 を適用
N30 G1 X... Y... Z.	; 多項式補間をオフ
...	

例:新しい多項式構文

従来の多項式構文	新しい多項式構文
PO[軸識別子]=(.. , ..)	軸識別子=PO(.. , ..)
PO[PHI]=(.. , ..)	PHI=PO(.. , ..)
PO[PSI]=(.. , ..)	PSI=PO(.. , ..)
PO[THT]=(.. , ..)	THT=PO(.. , ..)
PO[]=(.. , ..)	PO(.. , ..)
PO[変数]=IC(.. , ..)	変数=PO IC(.. , ..)

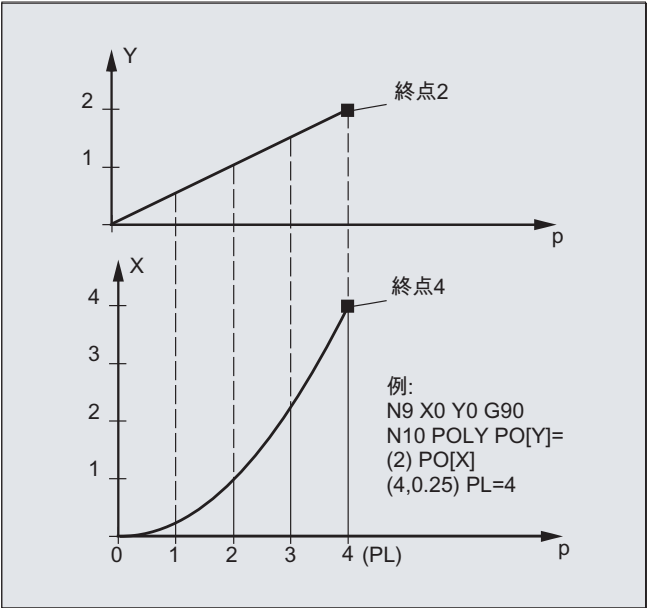
例:X/Y 平面の曲線。

プログラミング

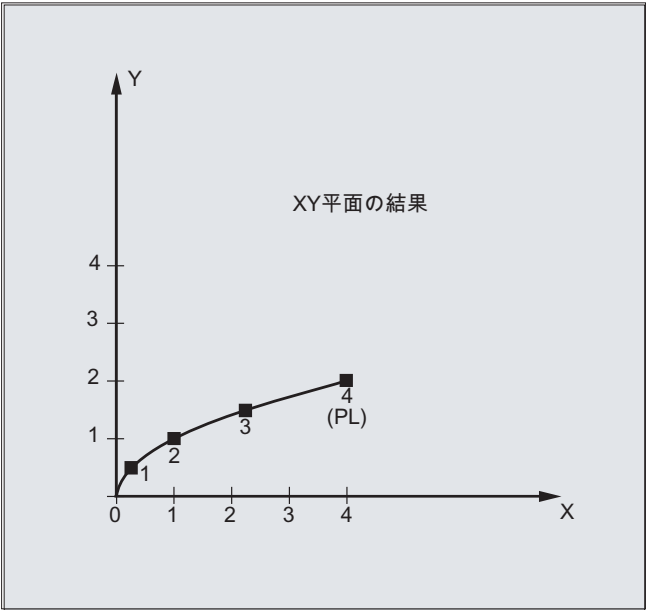
プログラムコード
N9 X0 Y0 G90 F100
N10 POLY PO[Y]=(2) PO[X]=(4,0.25) PL=4

曲線の形状、X(p)と Y(p)

4.7 特殊動作命令



XY 平面の曲線の形状



説明

多項式関数は通常、次の等式で表わします。

$$f(p) = a_0 + a_1p + a_2p^2 + \dots + a_np^n$$

意味: a_i : 定数係数 ($i = 0, 1, \dots, n$)

p : パラメータ

コントローラでは、次のように、最大 5 次までの多項式をプログラム指令できます。

$$f(p) = a_0 + a_1p + a_2p^2 + a_3p^3 + a_4p^4 + a_5p^5$$

これらの係数に具体的な値を割り当てて、線、放物線、べき関数などのさまざまな曲線形状を生成できます。

直線は、 $a_2 = a_3 = a_4 = a_5 = 0$ を使用して、次のように生成します。

$$f(p) = a_0 + a_1p$$

以下も適用されます。

a_0 : 前のブロック終了の軸位置

$p = PL$

$$a_1 = (x_E - a_0 - a_2 * p^2 - a_3 * p^3) / p$$

G 命令 POLY を使用して有効化した多項式補間を使用しない多項式をプログラム指令することができます。この場合は、プログラム指令した多項式が補間されませんが、代わりに、プログラム指令した軸のすべての終点へ直線でアプローチします(G1)。プログラム指令した多項式は、パートプログラムで、指定により多項式補間を有効化した後にのみ移動します(POLY)。

特記事項:分母多項式

PO[] = (...) 命令を使用すると、(軸名称を指定せずに) ジオメトリ軸に普通の分母多項式をプログラム指令できます。つまり、ジオメトリ軸の移動を、2 つの多項式の商として補間します。

このプログラミングオプションを使用すると、円錐曲線(円、楕円、放物線、双曲線)などの形状を正確に表現できます。

4.7 特殊動作命令

例:

プログラムコード	コメント
POLY G90 X10 Y0 F100	; ジオメトリ軸が位置 X10 Y0 へ直線で移動します。
PO[X]=(0,-10) PO[Y]=(10) PO[]=(2,1)	; ジオメトリ軸が 4 分円上を X0 Y10 へ移動します。

分母多項式の定数係数(a₀)は常に 1 とみなされます。プログラム指令した終点は、G90 / G91 には影響されません。

X(p)と Y(p)は、プログラム指令値から、次のように計算されます。

$$X(p) = (10 - 10 * p^2) / (1 + p^2)$$

$$Y(p) = 20 * p / (1 + p^2)$$

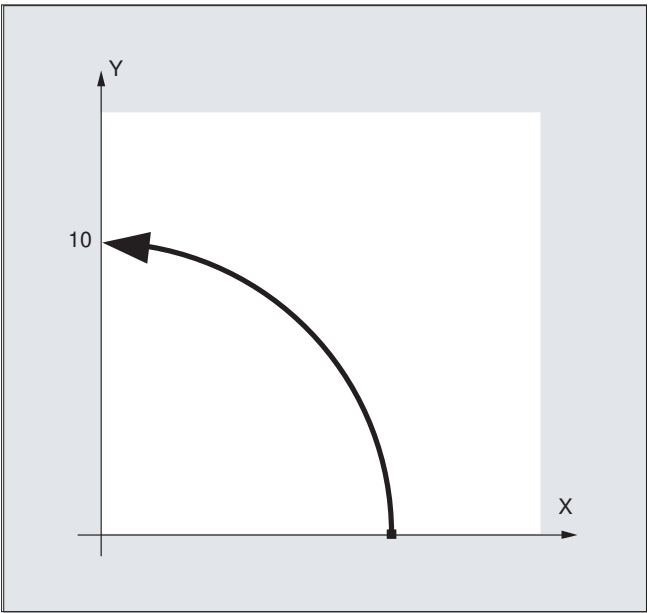
ただし、 $0 \leq p \leq 1$

始点と終点、定数 a₂、PL=1 をプログラム指令した結果、中間結果は次のようになります。

分子(X) = 10 + 0 * p - 10 * p²

分子(Y) = 0 + 20 * p + 0 * p²

分母= 1 + p²



多項式補間が有効で、区間 $[0, PL]$ の範囲内がゼロで分母多項式をプログラム指令した場合は、拒否されてアラームが発生します。分母多項式は、付加軸の移動には無効です。

注記

多項式補間と組み合わせて G41、G42 で工具径補正を有効にして、直線補間モードまたは円弧補間モードと同様に適用できます。

4.7.6 設定可能な軌跡基準(SPATH、UPATH)

多項式補間(POLY、ASPLINE、BSPLINE、CSPLINE、COMP...)の場合、軌跡軸 i の位置は、多項式 $p_i(U)$ で指定されます。曲線パラメータ U は、NC ブロック内で 0 から 1 に移動します。

FGROUP は、軌跡送り速度 F が適用される軸(FGROUP 軸)を選択します。FGROUP 軸の軌跡 S での定速度による補間は、多項式補間時に曲線パラメータ U の変化が通常は一定でないことを意味します。結果として、FGROUP に含まれない軸を選択する場合、その軸と FGROUP 軸の関係について、以下の 2 種類の方法を使用できます。

- 軌跡 S (SPATH)に同期
- 曲線パラメータ U (UPATH)に同期

構文

SPATH
UPATH

意味

SPATH:	FGROUP に含まれない軸は、軌跡 S を基準として移動します。
UPATH:	FGROUP に含まれない軸は、曲線パラメータ U を基準として移動します。

注記

UPATH と SPATH は、軌跡移動による F ワード多項式(FPOLY、FCUB、FLIN)の相互関係も定義します。

4.7 特殊動作命令

境界条件

SPATH と UPATH は、以下の場合は無効です。

- 直線補間(G1)
- 円弧補間(G2、G3)
- ねじブロック(G33、G34、G35、G33x、G63)
- すべての軌跡軸が FGROUP に含まれる場合

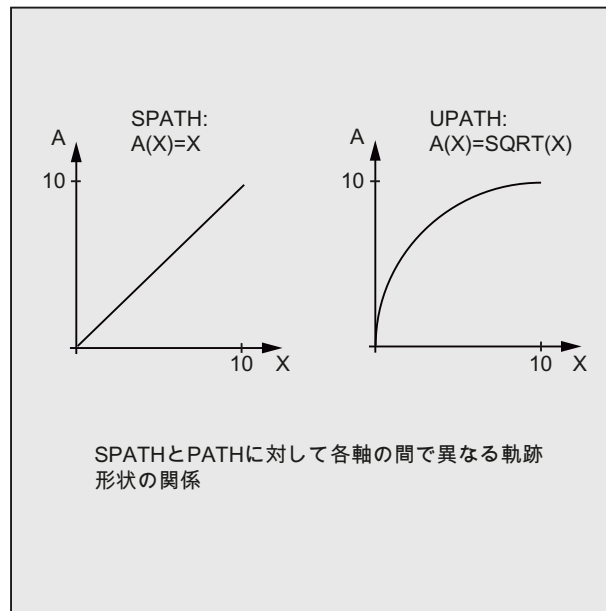
例

次の例は、両タイプのモーションコントロールの相違点を示します。

プログラムコード
N10 FGROUP (X,Y,Z)
N15 G1 X0 A0 F1000 SPATH ; SPATH
N20 POLY PO[X]=(10,10) A10

プログラムコード
N10 FGROUP (X,Y,Z)
N15 G1 X0 A0 F1000 UPATH ; UPATH
N20 POLY PO[X]=(10,10) A10

両方のプログラム区間で、曲線パラメータ U の二乗に応じて N20 の FGROUP 軸の軌跡 S が変わります。したがって、軌跡 X に沿った同期軸 A の位置が、SPATH と UPATH のどちらが有効かによって異なります。



詳細情報

リセットの制御動作、およびマシンデータ/オプションデータ

MD20150 \$MC_GCODE_RESET_VALUES[44]によって定義される G 命令は、リセット後に有効になります(45 番目の G グループ)。

スムージングのタイプの初期状態は、MD20150 \$MC_GCODE_RESET_VALUES[9] (10 番目の G グループ)で定義します。

軸別のマシンデータ MD33100 \$MA_COMPRESS_POS_TOL[<n>]は、拡張された意味があります。これには、コンプレッサ機能とスムージング(G642 による)の許容範囲が含まれています。

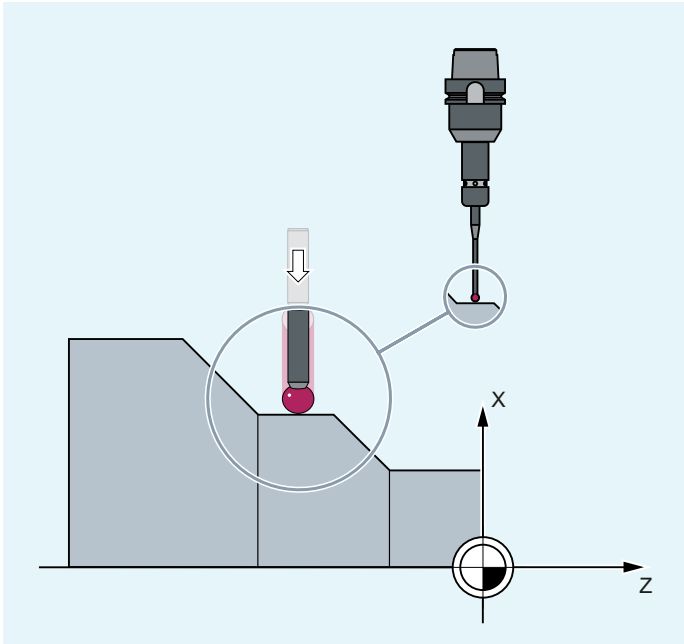
4.7.7 チャネル別計測(MEAS、MEAW)

チャネル別計測の場合、NC チャネルの計測処理は常に、関連するチャネルで実行中のパートプログラムから起動されます。**1 つ**のトリガ事象(プローブの立ち上がりまたは立ち下りエッジ)と、残移動距離削除あり(MEAS)または残移動距離削除なし(MEAW)の**1 つ**の計測モードが、**1 つ**の計測ブロックでプログラム指令されます。これにより、計測ブロックでプログラム指令されている**すべての軸**が計測処理に加わります。

計測ブロックが有効になると同時に、プローブがワークに移動します。プローブの切り替えエッジで、計測ブロックでプログラム指令したすべての軸の位置を計測し、各軸の適切なメモリに書き込みます。

4.7 特殊動作命令

計測の結果は、パートプログラムに読み込むか、シンクロナイズドアクションで機械座標系とワーク座標系の両方に書き込むことができます。



構文

```
MEAS=<TE> G...X...Y...Z...
MEAW=<TE> G...X...Y...Z...
```

意味

MEAS:	残移動距離を削除する計測	
	有効性:	ノンモーダル
MEAW:	残移動距離を削除しない計測	
	有効性:	ノンモーダル
	用途:	プログラム指令された位置に毎回アプローチする計測タスク

<TE>:	計測を開始する事象をトリガします		
	タイプ:	INT	
	値の範囲:	-2, -1, 1, 2	
	値:	(+)1	プローブ 1 の立ち上がりエッジ(計測入力 1)
		-1	プローブ 1 の立ち下がりエッジ(計測入力 1)
		(+)2	プローブ 2 の立ち上がりエッジ(計測入力 2)
		-2	プローブ 2 の立ち下がりエッジ(計測入力 2)
	注:	プローブは最大 2 個です(構成レベルにより異なります)。	
G...:	G0、G1、G2、G3 などの補間タイプ		
X...Y...Z...:	直交座標の終点		

注記

MEAS と MEAW はノンモーダルで、移動命令と一緒にプログラム指令します。送り速度と補間タイプ(G0、G1 など)だけでなく、軸数も、それぞれの計測処理に合わせて補正してください。

例

プログラムコード	コメント
...	
N10 MEAS=1 G1 F1000 X100 Y730 Z40	; 1 番目の計測点でのプローブによる計測ブロックと直線補間です。先読み停止は自動的に起こわれます。
...	

詳細情報

状態確認

プログラムで評価が必要な場合、プローブのスイッチがオンであるかどうかやプローブが切り替えられているかどうかなどの状態を、次のシステム変数で問い合わせることができます。

システム変数	意味	データタイプ	値	
\$A_PROBE[<n>]	プローブ状態	INT	0	プローブのスイッチがオフ。
			1	プローブのスイッチがオン。
\$AC_MEA[<n>]	プローブの切り替え状態 \$AC_MEA[<n>]が、計測の開始時に自動的にリセットされます。	INT	0	プローブは切り替えられていません。
			1	プローブは切り替えられています。

<n> = プローブの番号

計測値の読み取り

チャンネル別計測の場合、ブロックのすべての移動軌跡と位置決め軸の位置が得られます(軸の最大数は制御装置の構成により異なります)。MEAS では、プローブの起動後に、移動が定義された方法で減速されます。

注記

ジオメトリ軸が計測ブロックにプログラム指令されている場合は、現在のすべてのジオメトリ軸の計測値が保存されます。

変換に関与する軸が計測ブロックにプログラム指令されている場合は、この変換に関与するすべての軸の計測値が保存されます。

計測結果の読み取り

プローブで得られた軸の計測値を、次のシステム変数を使用して、パートプログラムおよびシンクロナイズドアクションで読み取ることができます。

システム変数	意味
\$AA_MM[<軸>]	機械座標系のプローブ計測値
\$AA_MW[<軸>]	ワーク座標系のプローブ計測値

4.7.8 軸別計測(MEASA、MEAWA、MEAC) (オプション)

軸別計測の場合、パートプログラムまたはシンクロナイズドアクションで起動することができます。2つの検出器が軸で使用可能な場合、両方の検出器を計測に使用できます。

以下の計測方法を使用できます。

- 残移動距離を削除する計測(MEASA)
- 残移動距離を削除しない計測(MEAWA)
- 残移動距離を削除しない連続計測(MEAC)

プログラム指令軸に MEASA または MEAWA を使用する場合は、各計測で最高 4 つの計測値が取得され、トリガ事象に対応したシステム変数に保存されます。

MEAC を使用した連続計測の場合は、計測結果は FIFO 変数に保存されます。

構文

```
MEASA [<軸>] = (<モード>, <TE1>, ..., <TE4>)
MEAWA [<軸>] = (<モード>, <TE1>, ..., <TE4>)
MEAC [<軸>] = (<モード>, <MeasMem>, <TE1>, ..., <TE4>)
```

注記

MEASA と MEAWA はノンモーダルで、1 ブロックと一緒にプログラム指令できます。一方、MEASA/MEAWA を MEAS/MEAW と一緒に 1 ブロックにプログラム指令した場合、エラーメッセージが表示されます。

意味

MEASA:	残移動距離を削除する軸別計測	
	有効性:	ノンモーダル
MEAWA:	残移動距離を削除しない軸別計測	
	有効性:	ノンモーダル
MEAC:	残移動距離を削除しない軸別連続計測	
	有効性:	ノンモーダル
<軸>:	計測に使用するチャネル軸の名称	

4.7 特殊動作命令

<モード>:	運転モードを示す 2 桁(xx)の数字(計測モードと検出器)		
	1 の位:計測モード トリガ事象を時系列の順に起動するか、プログラム指令された順に起動するかを指定します。		
	x0	計測ジョブをキャンセルします。	
	x1	4 種類までのトリガ事象を同時に有効にできます。	
	x2	4 種類までのトリガ事象を連続して有効にできます。	
	x3	4 種類までのトリガ事象を連続して有効にできますが、起動時にトリガ事象 1 は監視されません(アラーム 21700/21703 はマスクされます)。	
		注: MEAC は、このモードをサポートしていません。	
	10 の位:検出器 計測を行う計測器を指定します。		
	0x (または指定なし)	有効な検出器	
	1x	検出器 1	
	2x	検出器 2	
3x	両方の検出器		
<TE>:	計測を開始する事象をトリガします		
	タイプ:	INT	
	値の範囲:	-2, -1, 1, 2	
	(+)1	プローブ 1 の立ち上がりエッジ	
	-1	プローブ 1 の立ち下がりエッジ	
	(+)2	プローブ 2 の立ち上がりエッジ	
	-2	プローブ 2 の立ち下がりエッジ	
	注: 2 つの検出器により計測処理をおこなう場合、最大で 2 つのトリガ事象をプログラム指令できます(立ち上りエッジまたは立ち下がりエッジ)。両方の検出器の計測値は、2 つのトリガ事象の両方について取得されます。		
	<MeasMem>:	FIFO 番号(リングバッファ)	

例

例 1:モード 1 の、残移動距離を削除する軸別計測(時系列の順に評価)

a) 検出器が 1 基の計測

プログラムコード	コメント
...	
N100 MEASA[X]=(1,1,-1) G01 X100 F100	; 有効な検出器によるモード 1 の計測。プローブ 1 からの立ち上がりエッジ/立ち下がりエッジで、X=100 への移動軌跡の計測信号を待機します。
N110 IF \$AC_MEA[1]==FALSE GOTOF END	; 計測が正常におこなわれたことを確認します。
N120 R10=\$AA_MM1[X]	; 1 番目にプログラム指令のトリガ事象(立ち上がりエッジ)で得られた計測値を保存します。
N130 R11=\$AA_MM2[X]	; 2 番目にプログラム指令のトリガ事象(立ち下がりエッジ)で得られた計測値を保存します。
N140 END;	

b) 検出器が 2 基の計測

プログラムコード	コメント
...	
N200 MEASA[X]=(31,1,-1) G01 X100 F100	; 両方の検出器によるモード 1 の計測。プローブ 1 からの立ち上がりエッジ/立ち下がりエッジで、X=100 への移動軌跡の計測信号を待機します。
N210 IF \$AC_MEA[1]==FALSE GOTOF END	; 計測が正常におこなわれたことを確認します。
N220 R10=\$AA_MM1[X]	; 立ち上がりエッジで検出器 1 の計測値を保存します。
N230 R11=\$AA_MM2[X]	; 立ち上がりエッジで検出器 2 の計測値を保存します。
N240 R12=\$AA_MM3[X]	; 立ち下がりエッジで検出器 1 の計測値を保存します。
N250 R13=\$AA_MM4[X]	; 立ち下がりエッジで検出器 2 の計測値を保存します。
N260 END;	

例 2:モード 2 の、残移動距離を削除する軸別計測(時系列の順に評価)

プログラムコード	コメント
...	
N100 MEASA[X]=(2,1,-1,2,-2) G01 X100 F100	; 有効な検出器によるモード 2 の計測。X=100 への移動軌跡の間、立ち上がりプローブ 1、立ち下がりプローブ 1、立ち上がりプローブ 2、立ち下がりプローブ 2 という順に計測信号を待機します。
N110 IF \$AC_MEA[1]==FALSE GOTOF PROBE2	; プローブ 1 による計測が正常におこなわれたことを確認します。
N120 R10=\$AA_MM1[X]	; 1 番目にプログラム指令のトリガ事象(プローブ 1 の立ち上がりエッジ)で得られた計測値を保存します。

4.7 特殊動作命令

プログラムコード	コメント
N130 R11=\$AA_MM2[X]	; 2 番目にプログラム指令のトリガ事象 (プローブ 1 の立ち上がりエッジ) で得られた計測値を保存します。
N140, PROBE2:	
N150 IF \$AC_MEA[2]==FALSE GOTOF END	; プローブ 2 による計測が正常におこなわれたことを確認します。
N160 R12=\$AA_MM3[X]	; 3 番目にプログラム指令のトリガ事象 (プローブ 2 の立ち上がりエッジ) で得られた計測値を保存します。
N170 R13=\$AA_MM4[X]	; 4 番目にプログラム指令のトリガ事象 (プローブ 2 の立ち上がりエッジ) で得られた計測値を保存します。
N180 END:	

例 3:モード 1 の軸別連続計測(時系列の順に評価)

a) 100 個までの計測値の計測

プログラムコード	コメント
...	
N110 DEF REAL MEASVALUE[100]	
N120 DEF INT loop=0	
N130 MEAC[X]=(1,1,-1) G01 X1000 F100	; 有効な検出器によるモード 1 の計測。\$AC_FIFO1 に計測値を保存します。X=1000 への移動軌跡で、検出器のプローブ 1 の立ち下がりエッジを待機します。
N135 STOPRE	
N140 MEAC[X]=(0)	; 軸位置に到達すると、計測を終了します。
N150 R1=\$AC_FIFO1[4]	; パラメータ R1 に計測値の累計個数を保存します。
N160 FOR loop=0 TO R1-1	
N170 MEASURED VALUE[loop]=\$AC_FIFO1[0]	; \$AC_FIFO1 から計測値を読み出し、保存します。
N180 ENDFOR	

b) 10 個の計測値を得た後の、残移動距離を削除する計測

プログラムコード	コメント
...	
N10 WHEN \$AC_FIFO1[4]>=10 DO MEAC[x]=(0) DELDTG(x)	; 残移動距離削除
N20 MEAC[x]=(1,1,1,-1) G01 X100 F500	
N30 MEAC [X]=(0)	
N40 R1 = \$AC_FIFO1[4]	; 計測値の数
...	

c) 2つのプローブによる立ち下がり/立ち上がり歯面の計測

プログラムコード	コメント
...	
N110 DEF REAL MEASVALUE[16]	
N120 DEF INT loop=0	
N130 MEAC[X]=(1,1,-1,2) G01 X100 F100	; 有効な検出器によるモード1の計測。\$AC_FIFO1により計測値を保存します。X=100への移動軌跡の移動の間に、プローブ1の立ち下がりエッジ、プローブ2の立ち上がりエッジの順に計測信号を待機します。
N140 STOPRE	; 先読み停止
N150 MEAC[X]=(0)	; 軸位置に到達すると、計測を終了します。
N160 R1=\$AC_FIFO1[4]	; パラメータR1に計測値の累計個数を保存します。
N170 FOR loop=0 TO R1-1	
N180 MEASURED VALUE[loop]=\$AC_FIFO1[0]	; \$AC_FIFO1から計測値を読み出し、保存します。
N190 ENDFOR	

詳細情報

計測ジョブ

計測ジョブは、パートプログラムで、またはシンクロナイズドアクションからプログラム指令できます。各軸がいつでも有効にすることができるのは、1つの計測ジョブのみであることに注意してください。

注記

特定の計測処理に合わせて、送り速度を調節してください。

MEASA と MEAWA の場合は、結果の正確性は送り速度に対して保証されます。この送り速度で、同じタイプの1つまでのトリガ事象、および異なるタイプの4つまでのトリガ事象のみが各位置制御周期で発生します。

MEAC による連続計測の場合は、補間クロックサイクルと位置制御サイクルの比率が 1:8 を超えないようにしてください。

トリガ事象

トリガ事象は、プローブ番号と計測信号のトリガ条件(立ち上がりエッジまたは立ち下がりエッジ)から成ります。

それぞれの計測では、使用するプローブのトリガ事象を4つまで処理できます。つまり、それぞれ2つの計測信号エッジによって2つまでのプローブが処理できます。トリガ事象の処理順序と最大数は、選択したモードにより異なります。

4.7 特殊動作命令

2つの検出器により計測処理をおこなう場合、最大で2つのトリガ事象をプログラム指令できます(立ち上がりエッジまたは立ち下がりエッジ)。両方のプローブの計測値が、2つのトリガ事象の両方について取得されます。

PROFIBUS 通信メッセージ 391 (PROFIBUS 通信の初期設定)を使用する場合、各トリガ事象および位置コントローラクロック周期について取得できる計測値は1つです。

PROFIBUS メッセージ 395 を使用する場合、動作モード1のMEACでは、トリガ事象ごとの計測値の数は、各トリガ事象と位置コントローラサイクルについて、合計で立ち上がりの8つの計測値、立ち下がりの8つの計測値まで増やすことができます。

- 1つのプローブ:立ち上がりエッジ用に8つの計測値、立ち下がりエッジ用に8つの計測値
- 2つのプローブ:プローブごとに、立ち上がりエッジ用に4つの計測値、立ち下がりエッジ用に4つの計測値

これは、PROFIBUS メッセージ 395 を使用して、送り速度または速度を上げられることを意味します。

動作モード

運転モードの1番目の桁(10進数の10の位)で、必要な検出器を選択します。取り付けられている検出器は1基のみであるが、2基目がプログラム指令されている場合は、取り付けられている検出器を自動的に選択します。

2番目の桁(10進数の1の位)で、必要な計測モードを選択します。これで、計測処理が、当該の制御装置がサポートするオプションに合わせて次のように調整されます。

- **モード1**
トリガ事象は、発生順に時系列で評価されます。
- **モード2**
トリガ事象は、プログラム指令順に評価されます。
- **モード3**
トリガ事象は、プログラム指令順に評価されますが、起動時にトリガ事象1の監視はおこなわれません。

残移動距離を削除する計測と削除しない計測

MEASA 命令をプログラム指令した場合は、必要な計測値がすべて取得されるまでは残移動距離が削除されません。

MEAWA は、プログラム指令した位置が常にアプローチされる特殊な計測処理で使します。

注記

MEASA は、シンクロナイズドアクションではプログラム指令できません。別の方法として、MEAWA に残移動距離削除を加えて、シンクロナイズドアクションとしてプログラム指令できます。

MEAWA による計測ジョブをシンクロナイズドアクションから起動した場合は、計測値は機械座標系でのみ得ることができます。

ジオメトリ軸/座標変換

ジオメトリ軸に対して軸計測を起動する場合は、その他のジオメトリ軸のすべてに、同じ計測ジョブを明示的にプログラム指令してください。同じことが、座標変換の関連軸にも適用されます。

例:

```
N10 MEASA[Z]=(1,1) MEASA[Y]=(1,1) MEASA[X]=(1,1) G0 Z100
```

または

```
N10 MEASA[Z]=(1,1) POS[Z]=100
```

状態確認

プログラムで評価が必要な場合、プローブのスイッチがオンであるかどうかやプローブが切り替えられているかどうかなどの状態を、次のシステム変数で問い合わせることができます。

システム変数	意味	データタイプ	値	
\$A_PROBE[<n>]	プローブ状態	INT	0	プローブのスイッチがオフ。
			1	プローブのスイッチがオン。

4.7 特殊動作命令

システム変数	意味	データタイプ	値	
\$AC_MEA[<n>]	プローブの切り替え状態 \$AC_MEA[<n>]が、計測の開始時に自動的にリセットされます。	INT	0	プローブは切り替えられていません。
			1	プローブは切り替えられています(計測ブロックでプログラム指令したすべてのトリガ事象がおこなわれています)。

<n> = プローブの番号

注記

計測がシンクロナイズドアクションから起動された場合は、\$AC_MEA は更新されなくなります。この場合は、新しい NC/PLC インタフェース信号 DB31, ... DBX62.3 または同等の変数\$AA_MEA ACT[<軸>]を確認してください。

\$AA_MEA ACT==1:計測が有効です。

\$AA_MEA ACT==0:計測が無効です。

プローブ制限

NC プログラムまたはシンクロナイズドアクションで、PROFIBUS メッセージ 395 の使用時にシステム変数\$A_PROBE_LIMITED を使用してプローブ制限状態を読み取ることができます。

\$A_PROBE_LIMITED[<n>] == 0:プローブの制限の無効化/リセット

\$A_PROBE_LIMITED[<n>] == 1:プローブ制限が有効

<n> = プローブ番号

MEASA/MEAWA の計測結果

プローブで得られた計測値を、次のシステム変数を使用して、パートプログラムおよびシンクロナイズドアクションで読み取ることができます。

システム変数	意味
\$AA_MM1[<軸>]	機械座標系のトリガ事象 1 のプローブ計測値
...	...
\$AA_MM4[<軸>]	機械座標系のトリガ事象 4 のプローブ計測値
\$AA_MW1[<軸>]	ワーク座標系のトリガ事象 1 のプローブ計測値
...	...
\$AA_MW4[<軸>]	ワーク座標系のトリガ事象 4 のプローブ計測値

<軸> = 計測軸

2 基の検出器で計測ジョブを実行する場合は、両方の検出器で 2 つのトリガ事象を、それぞれ得ることができます。システム変数の割り当ては以下のようになります。

\$AA_MM1[<軸>]	または \$AA_MW1[<軸>]	トリガ事象 1 の検出器 1 からの計測値
\$AA_MM2[<軸>]	または \$AA_MW2[<軸>]	トリガ事象 1 の検出器 2 からの計測値
\$AA_MM3[<軸>]	または \$AA_MW3[<軸>]	トリガ事象 2 の検出器 1 からの計測値
\$AA_MM4[<軸>]	または \$AA_MW4[<軸>]	トリガ事象 2 の検出器 2 からの計測値

連続計測(MEAC)

MEAC の計測値は機械座標系で得ることができ、プログラム指令した FIFO[<n>]メモリ(リングバッファ)に保存されます。2 個のプローブを計測用に構成している場合は、2 番目のプローブの計測値は、このために特に設定された(マシンデータで定義した)FIFO[<n>+1]メモリに別に格納されます。

4.7 特殊動作命令

FIFO メモリはリングバッファで、ここには、計測値が循環規則に従って\$AC_FIFO 変数に書き込まれます。

注記

FIFO の内容は、リングバッファから一回だけ読み取ることができます。この計測データを複数回使用する場合は、ユーザーデータに保持してください。

FIFO メモリの計測値の数が、マシンデータで定義した最大値を超えた場合は、計測が自動的に終了します。

計測値を周期的に読み出すと、無限の計測処理を実行できます。この場合は、最低でも新しい計測値の入力と同じ頻度でデータを読み出してください。

詳細情報:機能マニュアル シンクロナイズドアクション

プログラミングエラーに対する保護

以下のプログラミングエラーが検出され、エラーとして表示されます。

- MEAS/MEAW と MEASA/MEAWA のプログラム指令が同一ブロックにあります

例:

```
N01 MEAS=1 MEASA[X]=(1,1) G01 F100 POS[X]=100
```

- MEASA/MEAWA のパラメータの数が<2 または>5 です

例:

```
N01 MEAWA[X]=(1) G01 F100 POS[X]=100
```

- MEASA/MEAWA のトリガ事象が 1/-1/2/-2 ではありません

例:

```
N01 MEASA[B]=(1,1,3) B100
```

- 無効なモードの MEASA/MEAWA です

例:

```
N01 MEAWA[B]=(4,1) B100
```

- MEASA/MEAWA のトリガ事象が 2 回プログラム指令されています

例:

```
N01 MEASA[B]=(1,1,-1,2,-1) B100
```

- MEASA/MEAWA とジオメトリ軸がありません

例:

```
N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) G01 X50 Y50 Z50 F100 ;GEO  
axis X/Y/Z
```

- ジオメトリ軸と計測ジョブが不整合です

例:

```
N01 MEASA[X]=(1,1) MEASA[Y]=(1,1) MEASA[Z]=(1,1,2) G01 X50 Y50  
Z50 F100
```

下記も参照

シンクロナイズドアクション (ページ 1108)

4.7.9 OEM ユーザー用の応用機能(OMA1 ... OMA5、OEMIPO1、OEMIPO2、G810 ... G829)

OEM アドレス

OEM アドレスの意味は、OEM ユーザーが特定します。その機能は、コンパイルサイクルを使用して組み込まれています。5 つの OEM アドレスが予約されています(OMA1 ... OMA5)。アドレス識別子は設定できます。OEM アドレスはどのブロックにもプログラム指令できます。

予約済み G 命令呼び出し

以下の G 命令呼び出しが OEM ユーザーに予約されています。

- OEMIPO1、OEMIPO2 (G グループ 1)
- G810 ... G819 (G グループ 31)
- G820 ... G829 (G グループ 32)

その機能は、コンパイルサイクルを使用して組み込まれています。

4.7 特殊動作命令

機能とサブプログラム

また、OEM ユーザーは、予約機能とサブプログラムをパラメータ転送を使用して設定することもできます。

注記

ワークシミュレーション

SW 4.4 まではコンパイルサイクルはサポートされていません。SW 4.4 では、選択されたコンパイルサイクル(CC)のみがワークシミュレーション用にサポートされています。

したがって、サポートされていないコンパイルサイクルのパートプログラム内の言語命令(OMA1 ... OMA5、OEMIP01/2、G810 ... G829、ユーザー自身の手順と機能)は、その結果アラームメッセージが発生し、個々の処理をおこなわずにシミュレーションがキャンセルされます。

対策:パートプログラム(\$P_SIM query)で、欠落している CC 用の言語要素を個別に処理します。

例:

```
N1 G01 X200 F500
IF (1==$P_SIM)
N5 X300 ;not active for CC simulation
ELSE
N5 X300 OMA1=10
ENDIF
```

4.7.10 コーナ減速による送り速度低下(FENDNORM、G62、G621)

自動コーナ減速を使用すると、コーナに達する前に、送り速度がベル型カーブを描いて減速します。また、セッティングデータで、加工に関連する工具動作の範囲をパラメータ設定することもできます。

- 送り速度減速の開始と終了
- 送り速度を減速するオーバライド
- 当該のコーナの検出

当該のコーナは、コーナ内角の角度が、セッティングデータでパラメータ設定したコーナ角度より小さい角度です。

初期値 FENDNORM は、自動コーナオーバライドの機能を解除します。

構文

FENDNORM

G62 G41

G621

意味

FENDNORM:	自動コーナ減速オフ
G62:	工具径補正が有効なときの 内側コーナ のコーナ減速
G621:	工具径補正が有効なときの すべてのコーナ のコーナ減速

G62 は、工具径補正 G41/G42 と連続軌跡モード G64/G641 が動作している場合の内側コーナーに対してのみ動作します。

次のように計算された減速送り速度で、コーナにアプローチします。

$$F * (\text{送り速度減速オーバーライド}) * \text{送り速度オーバーライド}$$

送り速度低下は、工具がコーナで中心軌跡を基準として方向転換する、ちょうどその位置で最大となります。

G621 は、G62 と同様に、FGROUP で定義した軸のそれぞれのコーナで動作します。

4.7.11 プログラム指令可能な動作終了条件(FINEA、COARSEA、IPOENDA、IPOBRKA、ADISPOSA)

軌跡補間(G601、G602、および G603)のブロック切り替え条件と同様に、パートプログラムのとき、またはコマンド軸/PLC 軸のシンクロナイズドアクションのときの単独軸補間の動作終了条件もプログラム指令できます。

動作終了条件の指令は、単独軸移動のよるパートプログラムブロックとテクノロジサイクルブロックが完了するまでの時間に影響します。これは、FC15/16/18 経由の PLC にも適用されます。

構文

FINEA [<軸>]

COARSEA [<軸>]

IPOENDA [<軸>]

IPOBRKA (<軸>[, <タイミグ>])

ADISPOSA [<軸>] = (<モード>, <範囲のサイズ>)

4.7 特殊動作命令

意味

FINEA:	動作終了条件:「精密イグザクトストップ」	
	有効:	モーダル
COARSEA:	動作終了条件:「汎用イグザクトストップ」	
	有効:	モーダル
IPOENDA:	動作終了条件:「補間停止」	
	有効:	モーダル
IPOBRKA:	ブロック切り替え条件:減速カーブ	
	有効:	モーダル
ADISPOSA:	動作終了条件の許容範囲	
	有効:	モーダル
<軸>:	チャンネル軸名称(X、Y、...)	
<タイミング>:	ブロック切り替えのタイミング、減速カーブを基準にして%で表わします。 • 100% = 減速カーブの開始 • 0% = 減速カーブの終了、IPOENDAと同じ意味です。	
	タイプ:	REAL
<モード>:	許容範囲の基準	
	値の範囲:	0 許容範囲が有効ではありません
		1 設定位置の許容範囲
		2 実位置の許容範囲
	タイプ:	INT
<範囲のサイズ>:	許容範囲のサイズ	
	タイプ:	REAL

例

例 1:動作終了条件:「補間停止」

プログラムコード

```

; 位置決め軸 x を 100 に移動、速度 200 m/min、 加速度 90%、
; 動作終了条件:補間停止
N110 G01 POS[X]=100 FA[X]=200 ACC[X]=90 IPOENDA[X]

```

プログラムコード

```

; シンクロナイズドアクション:
; もし常に、入力 1 が設定された場合は
; 位置決め軸を 50 に移動、速度 200 m/min、加速度 140%、
;      動作終了条件:補間停止
N120 EVERY $A_IN[1] DO POS[X]=50 FA[X]=200 ACC[X]=140
IPOENDA[X]

```

例 2:ブロック切り替え条件:「減速カーブ」

プログラムコード	コメント
	; 初期設定が有効です。
N40 POS[X]=100	; X から位置 100 への位置決め動作です。 ブロック切り替え条件:精密イグザクトストップ
N20 IPOBRKA(X,100)	; ブロック切り替え条件:「減速カーブ」, 100% = 減速カーブの開始
N30 POS[X]=200	; ブロックは X 軸の減速開始と同時に切り替えられます。
N40 POS[X]=250	; X 軸は位置 200 で減速せずに、位置 250 まで移動し続けます。 軸の減速開始と同時にブロックが切り替えられます。
N50 POS[X]=0	; X 軸が減速して位置 0 に戻ります。 位置 0 でブロックが切り替えられ、「精密イグザクトストップ」となります。
N60 X10 F100	; X 軸が軌跡軸として位置 10 に移動します。

詳細情報

動作終了条件用のシステム変数

有効な動作終了条件を、システム変数\$AA_MOTEND を使用して読み取ることができます。

ブロック切り替え条件:「減速カーブ」(IPOBRKA)

ブロック切り替え条件「減速カーブ」を有効にするときに、任意選択のブロック切り替え点のタイミングの値がプログラム指令されている場合、この値は次の位置決め動作に対して有効になり、メインランに同期してセッティングデータに書き込まれます。ブロック切り替え点のタイミングの値が指定されていない場合、セッティングデータの現在値が有効になります。

SD43600 \$SA_IPOBRAKE_BLOCK_EXCHANGE

IPOBRKA は、軸の動作終了条件(FINEA、COARSEA、IPOENDA)がその次に対応する軸へプログラム指令されると、対応する軸に対して無効になります。

4.8 座標変換(フレーム)

追加のブロック切り替え条件:「許容範囲」(ADISPOSA)

ADISPOSA を使用して、ブロック終点周囲の許容範囲(現在位置または指令位置)を追加のブロック切り替え条件として定義できます。ブロック切り替えに対して以下の 2 つの条件を満たしてください。

- ブロック切り替え条件:「減速カーブ」
- ブロック切り替え条件:「許容範囲」

4.8 座標変換(フレーム)

4.8.1 フレーム変数による座標変換

ROT、AROT、SCALE などのフレーム命令 (ページ 342)に加え、ワーク座標系(WCS)もフレーム変数\$P_...FR (データ管理フレーム)および\$P_...FRAME (有効なフレーム)を使用して変換することができます。

次の図は、フレーム変数の構造の概要を示したものです。

- データ管理フレーム
- 動作中のフレーム
- 動作中の全体フレーム:動作中の全フレームの結合
- NCU グローバルフレーム
- チャネル別フレーム

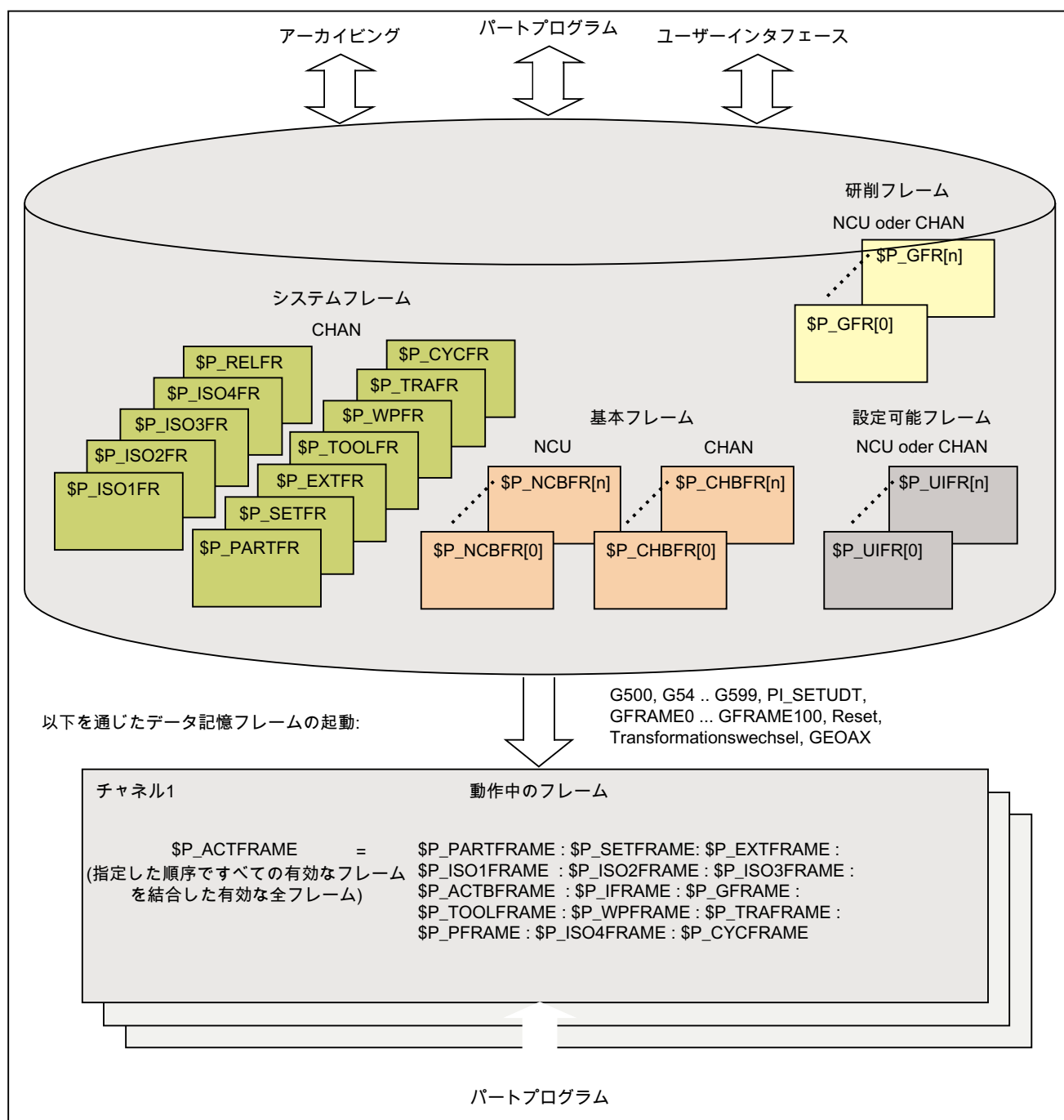


図 4-1 フレーム変数の概要

4.8 座標変換(フレーム)

4.8.1.1 予約フレーム変数(\$P_CHBFRAME、\$P_IFRAME、\$P_PFRAME、\$P_ACTFRAME)

有効:チャンネル別基本フレーム\$P_CHBFRAME[<n>] (\$P_BFRAME)

注記

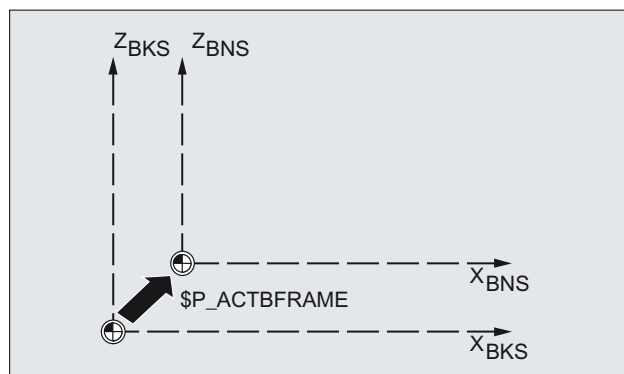
現在の基本フレーム\$P_BFRAME とデータ記憶の基本フレーム\$P_UBFR は、互換性の理由から保持されます。

- $\$P_BFRAME \triangleq \$P_CHBFRAME[0]$
- $\$P_UBFR \triangleq \$P_CHBFR[0]$

フレーム変数\$P_CHBFRAME[<n>]は、基本座標系(BCS)と基本ゼロオフセットシステム(BZS)の間の位置関係を定義します。

現在のチャンネル別基本フレーム\$P_CHBFRAME[<n>]を、NC プログラムですぐに有効にするには、以下の方法を使用できます。

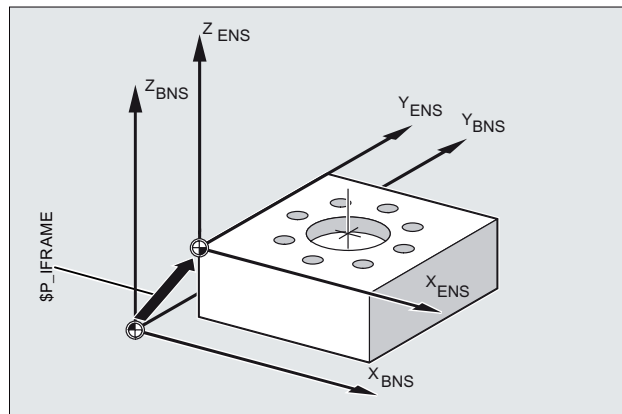
- 命令:
 - G500 (すべての設定可能フレームを無効にし、基本フレームは有効なままにします)
 - G54～G599 (設定可能なゼロオフセット)
- データ記憶のチャンネル別基本フレームを、現在のチャンネル別基本フレームに割り当て
 $\$P_CHBFRAME[<n>] = \$P_CHBFR[<m>]$



有効:チャンネル別設定可能フレーム\$P_IFRAME

フレーム変数\$P_IFRAME は、基本ゼロオフセットシステム(BZS)と設定可能ゼロオフセットシステム(SZS)の間の位置関係を定義します。

- \$P_IFRAME は\$P_UIFR[\$P_IFRNUM]に対応します。
- たとえば、G54 をプログラム指令後に\$P_IFRAME には、G54 で定義した平行移動、回転、スケーリング、およびミラーリングが含まれます。

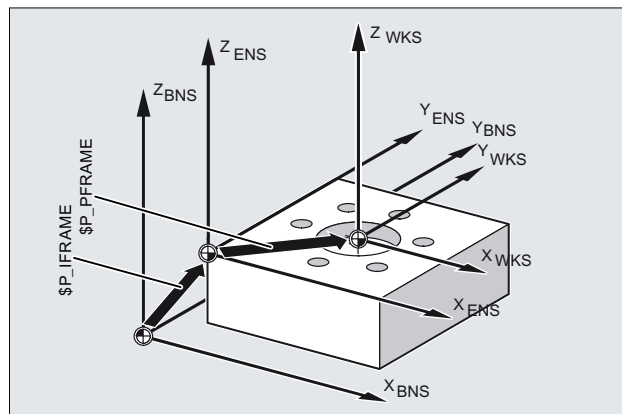


有効:チャネル別プログラマブルフレーム\$P_PFRAME

\$P_PFRAME フレーム変数は、設定可能ゼロオフセットシステム(SZS)とワーク座標系(WCS)の間の位置関係を定義します。

\$P_PFRAME には、以下のいずれかの結果のフレームが含まれます。

- TRANS/ATRANS、ROT/AROT、SCALE/ASCALE、MIRROR/AMIRROR のプログラミングの結果、または
- プログラム指令 **FRAME** への CTRANS、CROT、CMIRROR、CSCALE の割り当ての結果



4.8 座標変換(フレーム)

有効:全体フレーム\$P_ACTFRAME

チャンネル内の有効な全体フレームは、チャンネル内で機能するすべてのフレームを結合したものです。

```
$P_ACTFRAME = $P_PARTFRAME : $P_SETFRAME : $P_EXTFRAME :  
              $P_ISO1FRAME : $P_ISO2FRAME : $P_ISO3FRAME :  
              $P_ACTBFRAME : $P_IFRAME : $P_GFRAME :  
              $P_TOOLFRAME : $P_WPFRAME : $P_TRAFRAME :  
              $P_PFRAME      : $P_ISO4FRAME : $P_CYCFRAME
```

\$P_ACTFRAME は、現在有効なワーク座標系を記述します。

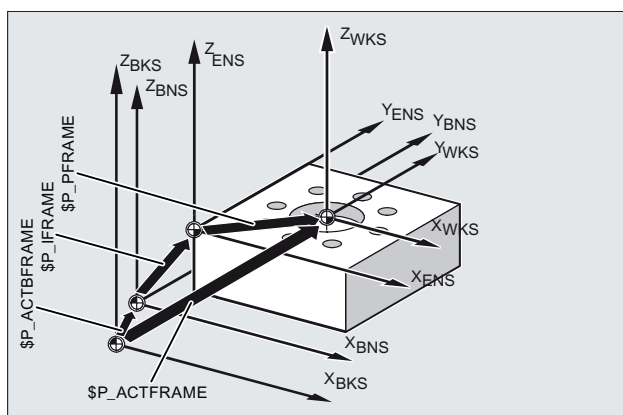
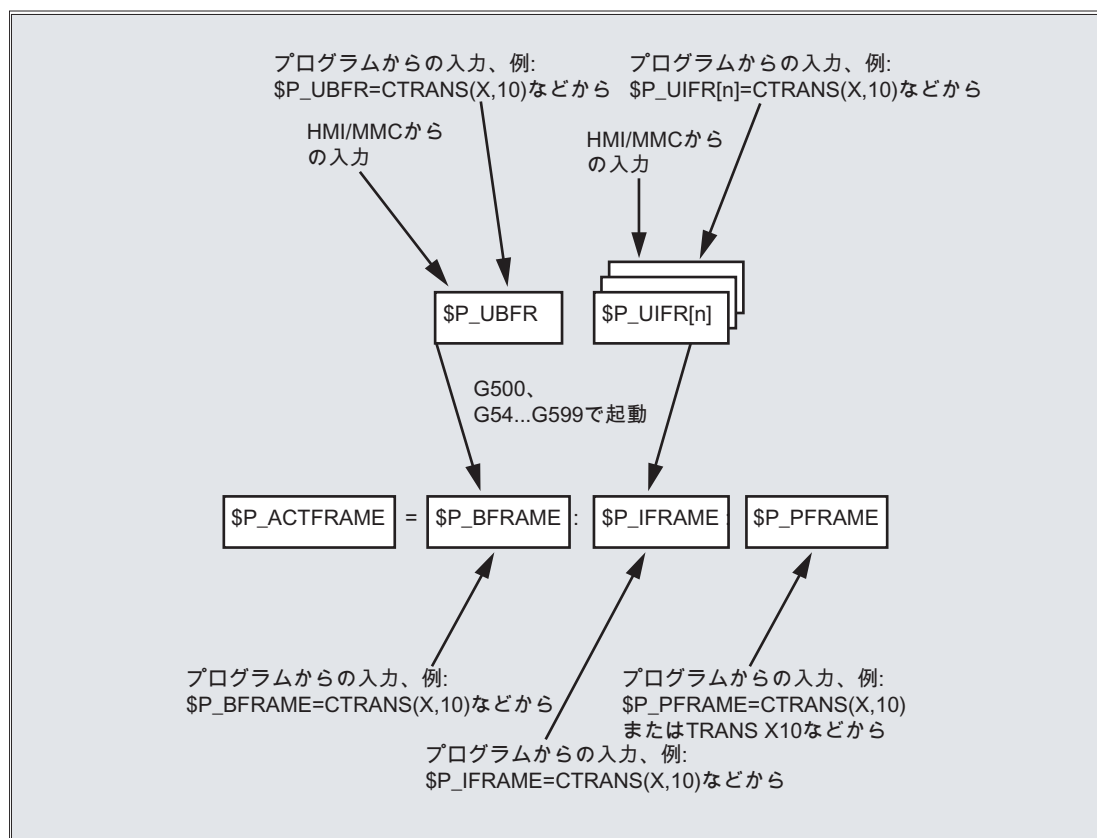


図 4-2 フレーム変数\$P_ACTFRAME

\$P_BFRAME/\$P_CHBFRAME[<n>]、\$P_IFRAME、または\$P_PFRAME のいずれかのフレームが変更された場合、現在の全体フレーム\$P_ACTFRAME が再計算されます。



MD 20110 RESET_MODE_MASK を以下のように設定した場合は、リセット後に基本フレームと設定可能フレームが有効になります。

ビット 0=1、ビット 14=1 --> $\$P_UBFR$ (基本フレーム)が動作します。

ビット 0=1、ビット 5=1 --> $\$P_UIFR$ [$\$P_UIFRNUM$] (設定可能フレーム)が動作します。

データ記憶:チャンネル別基本フレーム $\$P_CHBFR[<n>]$

フレーム変数 $\$P_CHBFR[<n>]$ は、データ記憶の基本フレームを読み取り/書き込みます。データ記憶フレームは、書き込んだときにチャンネル内ですぐには有効になりません。書き込まれたフレームは、以下により有効になります。

- チャンネルリセットと MD20110 $\$MC_RESET_MODE_MASK$ 、ビット 0 == 1 およびビット 14 == 1
- 命令 G500、G54 ... G57、G505 ... G599 (基本フレームの有効化/無効化と、その後の現在の全体フレームの再計算)

4.8 座標変換(フレーム)

データ記憶:チャンネル別設定可能フレーム\$P_UIFR[<n>]

フレーム変数\$P_UIFR[<n>]は、データ記憶の設定可能な基本フレームを読み取り/書き込みます。フレームは、書き込んだときにチャンネル内ですぐには有効になりません。チャンネル内の書き込まれたフレームは、以下により計算されます。

- G500 命令(すべての設定可能フレームまたはゼロオフセットを解除)
- G54 ... G57、G505 ... G599 命令(設定可能フレームまたはゼロオフセットを有効化)

有効な設定可能フレーム	データ記憶フレーム	(対応する命令)
\$P_IFRAME =	\$P_UIFR[0]	G500
	\$P_UIFR[1]	G54
	\$P_UIFR[2]	G55
	\$P_UIFR[3]	G56
	\$P_UIFR[4]	G57
	\$P_UIFR[5]	G505
	\$P_UIFR[6]	G506
	...	...
	\$P_UIFR[99]	G599

4.8.2 フレームへの値の割り付け

4.8.2.1 値の直接割り当て(軸値、角度、スケール)

NC プログラムでフレームまたはフレーム変数に直接、値を割り当てることができます。

構文

構文

```
$P_PFRAME = CTRANS (X, <オフセット値>, Y, <オフセット値>, Z, <オフセット値>, ...)
```

```
$P_PFRAME = ROT (X, <角度>, Y, <角度>, Z, <角度>, ...)
```

```
$P_UIFR[...] = CROT (X, <角度>, Y, <角度>, Z, <角度>, ...)
```

```
$P_PFRAME = CSCALE (X, <スケール>, Y, <スケール>, Z, <スケール>, ...)
```

```
$P_PFRAME = CMIRROR(X, Y, Z)
```

\$P_CHBFRAME[<n>] の構文は \$P_PFRAME と同じです。

意味

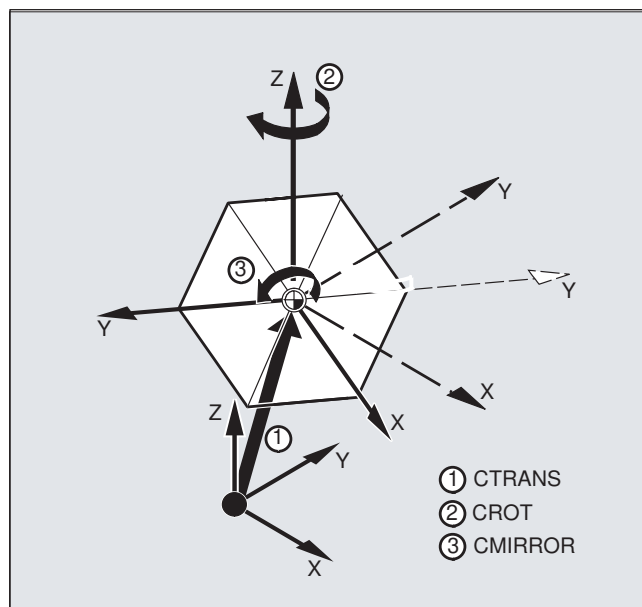
CTRANS:	指定軸の平行移動
CROT:	指定軸を中心とする回転
CSCALE:	指定軸でのスケール変更
CMIRROR:	指定軸の方向逆転
X, Y, Z:	指定ジオメトリ軸の方向のオフセット値
<オフセット値>:	オフセット値
<角度>:	回転角度
<スケール>:	スケール値

例

現在のプログラマブルフレームのフレーム成分への値割り当て

現在のプログラマブルフレームのフレーム成分の移動、回転、ミラーリングのための値割り当て

```
$P_PFRAME = CTRANS(X,10,Y,20,Z,5) :CROT(Z,45) :CMIRROR(Y)
```



4.8 座標変換(フレーム)

フレームの回転成分の書き込み

CROTにより、設定可能なデータ記憶フレーム\$P_UIFRの回転成分の3本の軸すべてに値を割り当てます。

```
$P_UIFR[5] = CROT(X, 0, Y, 0, Z, 0)
```

別の方法として、データ記憶フレームの回転成分の関連する軸に個々の値を直接割り当てます。

```
$P_UIFR[5, Y, RT]=0
```

```
$P_UIFR[5, X, RT]=0
```

```
$P_UIFR[5, Z, RT]=0
```

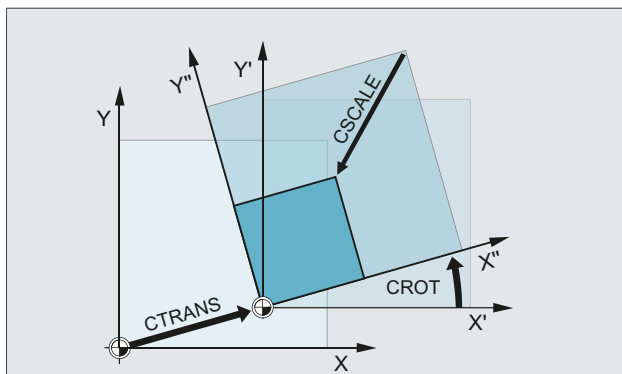
説明

結合演算子: フレームでの複数の操作を相互に結合します。操作は、左から右に連続して処理されます。

例

\$P_PFRAMEでのオフセット、回転、スケーリングの結合操作

```
$P_PFRAME = CTRANS(...) :CROT(...) :CSCALE...
```



4.8.2.2 フレーム成分の読み取りと変更(TR、FI、RT、SC、MI)

この機能を使用すると、特定のオフセット値または回転角度など、フレームの個々のデータにアクセスできます。これらの値を変更したり、別の変数に割り当てたりできます。

構文

<code>R10=\$P_UIFR[\$P_UIFNUM,X,RT]</code>	現在有効な設定可能ゼロオフセット <code>\$P_UIFRNUM</code> の、 <code>X</code> 軸を中心とする回転角度 <code>RT</code> を変数 <code>R10</code> に割り当てます。
<code>R12=\$P_UIFR[25,Z,TR]</code>	設定したフレーム No. 25 のデータセットから、 <code>Z</code> のオフセット値 <code>TR</code> を変数 <code>R12</code> に割り当てます。
<code>R15=\$P_PFRAME[Y,TR]</code>	現在のプログラマブルフレームの <code>Y</code> のオフセット値 <code>TR</code> を変数 <code>R15</code> に割り当てます。
<code>\$P_PFRAME[X,TR] = 25</code>	現在のプログラマブルフレームの <code>X</code> のオフセット値 <code>TR</code> を変更します。 <code>X25</code> は直ちに適用されます。

意味

<code>\$P_UIFRNUM:</code>	この命令は、現在有効な設定可能ゼロオフセットを基準とした位置を自動的に確立します。
<code>P_UIFR[n, ..., ...] :</code>	フレーム番号 <code>n</code> を指定して、設定可能フレーム No. <code>n</code> にアクセスします。
	読み取る、または変更する成分を指定します。
<code>TR:</code>	TR 平行移動
<code>FI:</code>	FI 精密平行移動
<code>RT:</code>	RT 回転
<code>SC:</code>	SC スケール変更
<code>MI:</code>	MI ミラーリング
<code>X, Y, Z :</code>	対応する軸 <code>X</code> 、 <code>Y</code> 、 <code>Z</code> も指定されます (例を参照してください)。

RT 回転の値の範囲

1 番目のジオメトリ軸を中心とした $-180^{\circ} \sim +180^{\circ}$

回転:

2 番目のジオメトリ軸を中心とした $-90^{\circ} \sim +90^{\circ}$

回転:

3 番目のジオメトリ軸を中心とした $-180^{\circ} \sim +180^{\circ}$

回転:

4.8 座標変換(フレーム)

説明

フレームの呼び出し

システム変数\$P_UIFRNUM を指定すると、\$P_UIFR または G54、G55、... で設定した現在のゼロオフセットにアクセスできます

(\$P_UIFRNUM には、現在設定しているフレーム番号が含まれます)。

保存されている他の設定可能な\$P_UIFR フレームはすべて、適切な番号の\$P_UIFR[n]を指定して呼び出します。

予約フレーム変数とユーザー定義フレームの場合は、\$P_IFRAME などの名称を指定してください。

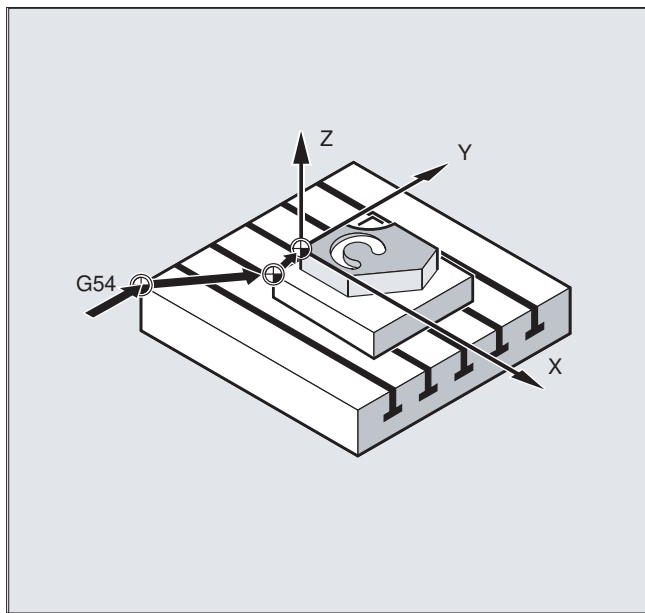
データの呼び出し

アクセスする、または変更する軸名称と値のフレーム成分は、[X, RT]や[Z, MI]のように、角括弧で囲んで記述します。

4.8.2.3 フレームの接続

フレームを別のフレームに割り当てたり、フレームを NC プログラムで相互に結合することができます。

このフレーム結合は、パレット上に配置され、同じ処理で加工する複数のワークの記述に適しています。



フレーム成分に含めることができるのは、パレット処理の記述のための中間値のみです。これらの値の結合により、さまざまなワーク原点を生成します。

例

割り当て

プログラムコード	コメント
DEF FRAME SETTING_1	; ローカルフレーム変数の定義
SETTING_1 = CTRANS(X,10)	; 関数の結果をフレーム変数に割り当て
\$P_PFRAME = SETTING_1	; フレーム変数を現在のフレームに割り当て
DEF FRAME SETTING_4	; ローカルフレーム変数の定義
SETTING_4 = \$P_PFRAME	; 現在のフレームをフレーム変数に保存
...	
\$P_PFRAME = SETTING_4	; 現在のフレームをフレーム変数から取得

結合

演算子: フレームをプログラム指令順に相互に結合します。オフセットや回転などのフレーム成分が、連続して順に実行されます。

プログラムコード	コメント
\$P_IFRAME = \$P_UIFR[15] :	; 2つの設定可能なデータ記憶フレームの
\$P_UIFR[16]	; 結合による結果フレームを
	; 動作中の設定可能な全体フレームに割り当て
	; 用途例:
	; \$P_UIFR[15]: オフセット
	; \$P_UIFR[16]: 回転
\$P_UIFR[3] = \$P_UIFR[4] :	; 2つの設定可能なデータ記憶フレームの
\$P_UIFR[5]	; の結合による結果フレームを
	; 別の設定可能なデータ記憶フレームに割り当て

4.8.2.4 フレーム変数の定義(DEF FRAME)

予約フレーム変数に加えて、ユーザーフレーム変数も定義できます。ユーザー定義のフレーム変数は、FRAME タイプのユーザー変数です。フレームの名称は、ユーザー変数の規則に従って自由に割り当てることができます。

CTTRANS、CROT、CSCALE、および CMIRROR は、ユーザー定義のフレーム変数に値を割り当てます。

構文

```
DEF FRAME <名称>
```

4.8 座標変換(フレーム)

意味

DEF FRAME:	FRAME タイプのユーザー変数を定義します。
<名称>:	フレーム変数の名称

例

「PALETTE」フレーム変数の定義とオフセット値と回転値の割り当て

プログラムコード	コメント
DEF FRAME PALETTE	; PALETTE フレーム変数を定義
PALETTE = CTRANS(...) :CROT(...)	; オフセットと回転のために結合の結果フレームを
	; PALETTE フレーム変数に割り当て

4.8.3 荒削りオフセットと仕上げオフセット(CTRANS、CFINE)

仕上げオフセット

仕上げオフセット CFINE (...) は、次のフレームに適用できます。

- 設定可能フレーム: \$P_UIFR または \$P_IFRAME
- 基本フレーム: \$P_NCBFR[<n>]、\$P_CHBFR[<n>]、\$P_CHBFRAMES[<n>] または \$P_ACTBFRAME
- プログラマブルフレーム: \$P_PFRAME

フレームの仕上げオフセットは、CFINE (...) 命令によりプログラム指令されます。

荒削りオフセット

荒削りオフセット CTRANS (...) は、すべてのフレームに適用できます。

サムオフセット

サムオフセットは、荒削りオフセットと仕上げオフセットを加算したものです。

マシンデータ

仕上げオフセットの有効化

仕上げオフセットは次のマシンデータで有効にします。

MD18600 \$MN_MM_FRAME_FINE_TRANS = 1

構文

仕上げオフセット

- フレーム全体
 - <フレーム> = CFINE(<K_1>,<値>)
 - <フレーム> = CFINE(<K_1>,<値>,<K_2>,<値>)
 - <フレーム> = CFINE(<K_1>,<値>,<K_2>,<値>,<K_3>,<値>)
- フレーム成分
 - <フレーム>[<n>,<K_1>,FI] = <値>

荒削りオフセット

- フレーム全体
 - <フレーム> = CTRANS(<K_1>,<値>)
 - <フレーム> = CTRANS(<K_1>,<値>,<K_2>,<値>)
 - <フレーム> = CTRANS(<K_1>,<値>,<K_2>,<値>,<K_3>,<値>)
- フレーム成分
 - <フレーム>[<n>,<K_1>,TR] = <値>

特にプログラマブルフレーム\$P_PFRAMEの場合は、次のようになります。

- TRANS <K_1> <値>
- TRANS <K_1> <値> <K_2> <値>
- TRANS <K_1> <値> <K_2> <値> <K_3> <値>

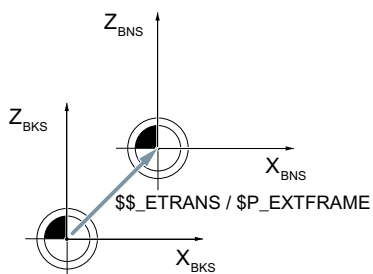
意味

<フレーム>:	フレーム、データ記憶の設定可能フレーム\$P_UIFR[<n>]など
CFINE:	仕上げオフセットまたは追加オフセット
CTrans:	荒削りオフセット、アブソリュートオフセット
TRANS:	プログラマブルフレームのみ:荒削りオフセット、アブソリュートオフセット
<K_n>:	座標軸 X、Y、Z
<値>:	オフセット値

4.8 座標変換(フレーム)

4.8.4 外部ゼロオフセット(\$AA_ETRANS)

外部ゼロオフセットは、基本座標系(BCS)と基本ゼロオフセットシステム(BZS)の間の直線オフセットです。



\$AA_ETRANS による外部ゼロオフセットは、マシンデータのパラメータ設定に応じて、次の2通りの方法で動作します。

1. NC/PLC インタフェース信号による起動後、システム変数\$AA_ETRANS は直接オフセット値として動作します。
2. NC/PLC インタフェース信号による起動後、システム変数\$AA_ETRANS の値は、有効なシステムフレーム\$P:EXTFRAME とデータ記憶フレーム\$P_EXTFR に転送されます。動作中の全体フレーム\$P_ACTFRAME が再計算されます。

マシンデータ

システム変数\$AA_ETRANS との組み合わせでは、次のマシンデータで選択した2つの処理は区別されます。

MD28082 \$MC_MM_SYSTEM_FRAME_MASK、ビット 1 = <値>

<値>	意味
0	機能:\$AA_ETRANS[<軸>]を PLC、HMI、または NC プログラムで直接書き込みます。 次の可能な移動ブロックで\$AA_ETRANS[<軸>]のためにゼロオフセットの後退を可能にします:DB31、...DBX3.0
1	機能:有効なシステムフレーム\$P:EXTFRAME とデータ記憶フレーム\$P_EXTFR を起動します。 \$AA_ETRANS[<軸>]のためにゼロオフセットの後退を可能にします:DB31、...DBX3.0。チャンネルで以下が実行されます。 • チャンネルのすべての移動動作を停止(コマンド軸と PLC 軸以外) • 先読み停止とその後の再解析(STOPRE) • 動作中のフレームの荒削りオフセット\$P_EXTFRAME[<軸>] = \$AA_ETRANS[<軸>] • データ記憶フレームの荒削りオフセット\$P_EXTFR[<軸>] = \$AA_ETRANS[<軸>] • 動作中の全体フレームの再計算\$P_ACTFRAME • プログラム指令された軸でオフセットを後退 • 中断されていた NC プログラムの移動動作を再開

プログラミング

- 構文
\$AA_ETRANS[<軸>] = <値>
- 意味

\$AA_ETRANS:	外部ゼロオフセットを保存するためのシステム変数
<軸>:	チャンネル軸
<値>:	オフセット値

NC/PLC インタフェース信号

DB31, ... DBX3.0 = 0 → 1 ⇒ \$P_EXTFRAME[<軸>] = \$P_EXTFR[<軸>] = \$AA_ETRANS[<軸>]

4.8 座標変換(フレーム)

4.8.5 原点確立状態をリセットして現在値を設定(PRESETON)

PRESETON()は、機械座標系(MCS)の 1 つ以上の軸に対して、新しい現在値を設定します。これは、軸の MCS のゼロオフセットに対応します。これで軸が移動することはありません。

PRESETON によって、同期による先読み停止が起動されます。実際の位置は、停止状態でのみ軸に割り当てられます。

PRESETONS の軸がチャンネルに割り当てられていない場合、次の手順は、軸別の軸切り替え動作の設定によって決まります。

```
MD30552 $MA_AUTO_GET_TYPE
```

原点確立状態

機械座標系で新しい現在値を設定することにより、機械軸の原点確立状態がリセットされます。

```
DB31、... DBX60.4/1.5 = 0 (原点確立済み/原点同期済み検出器 1/2)
```

PRESETON は、レファレンス点が不要な軸にのみ使用することをお勧めします。

元の機械座標系に復帰するには、たとえばパートプログラムからレファレンス点復帰動作により(G74)、機械軸の検出器を再度原点確立してください。

 注意

原点確立状態のリセット

新しい現在値を PRESETON で機械座標系に設定すると、機械軸の原点確立状態が「原点未確立/原点未同期」にリセットされます。

プログラミング

構文

```
PRESETON(<軸_1>, <値_1> [, <軸_2>, <値_2>, ... <軸_8>, <値_8>])
```

意味

PRESETON:	原点確立状態のリセットにより現在値を設定します	
	先読み停止:	あり
	単独ブロック指令:	あり

<軸_x>:	機械軸名称	
	タイプ:	AXIS
	値の範囲:	チャンネルで定義された機械軸の名称
<値_x>:	機械座標系(MCS)での機械軸の新しい現在値 入力には現在有効な単位系(インチ/メトリック)でおこなわれます 有効な直径指定(DIAMON)が考慮されます	
	タイプ:	REAL

詳細情報

NC プログラムでの PRESETON

NC プログラムでの PRESETON の詳細については、『機能マニュアル 基本機能』を参照してください。

シンクロナイズドアクションでの PRESETONS

シンクロナイズドアクションでの PRESETON の詳細については、『機能マニュアル シンクロナイズドアクション』を参照してください。

4.8.6 原点確立状態をリセットせずに現在値を設定(PRESETONS)

PRESETONS()は、機械座標系(MCS)の 1 つ以上の軸に対して、新しい現在値を設定します。これは、軸の MCS のゼロオフセットに対応します。これで軸が移動することはありません。

PRESETONS によって、同期による先読み停止が起動されます。実際の位置は、停止状態でのみ軸に割り当てられます。

PRESETONS の軸がチャンネルに割り当てられていない場合、次の手順は、軸別の軸入れ替え動作の設定によって決まります。

MD30552 \$MA_AUTO_GET_TYPE

原点確立状態

PRESETONS により機械座標系(MCS)で新しい現在値を設定しても、機械軸の原点確立状態は変更されません。

4.8 座標変換(フレーム)

必要条件

• エンコーダタイプ

PRESETONS は、次のエンコードタイプの有効な検出器に対してのみ使用できます。

- MD30240 \$MA_ENC_TYPE[<検出器>] = 0 (シミュレーションされるエンコーダ)
- MD30240 \$MA_ENC_TYPE[<検出器>] = 1 (生信号エンコーダ)

• 原点復帰動作モード

PRESETONS は、有効な検出器の以下の指令モードでのみ使用できます。

- MD34200 \$MA_ENC_REFP_MODE[<検出器>] = 0 (可能なレファレンス点復帰なし)
- MD34200 \$MA_ENC_REFP_MODE[<検出器>] = 1 (インクレメンタル検出器、回転検出器、またはリニア検出器のレファレンス点復帰:エンコーダトラックでゼロパルス)

プログラミング

構文

PRESETONS (<軸_1>, <値_1> [, <軸_2>, <値_2>, ... <軸_8>, <値_8>])

意味

PRESETONS:	原点確立状態をリセットせずに、現在値を設定します	
	先読み停止:	あり
	単独ブロック指令:	あり
<軸_x>:	機械軸名称	
	タイプ:	AXIS
	値の範囲:	チャンネルで定義された機械軸の名称
<値_x>:	機械座標系(MCS)の機械軸の新しい現在の実際値 入力是有効な単位系(インチ/メトリック)でおこなわれます 有効な直径指定(DIAMON)が考慮されます	
	タイプ:	REAL

詳細情報

NC プログラムでの PRESETONS

NC プログラムでの PRESETONS の詳細については、『機能マニュアル 基本機能』を参照してください。

シンクロナイズドアクションでの PRESETONS

シンクロナイズドアクションでの PRESETONS の詳細については、『機能マニュアル シンクロナイズドアクション』を参照してください。

4.8.7 空間の 3 つの計測点からのフレーム計算(MEAFRAME)

MEAFRAME 機能は、計測サイクルのサポートに使用します。これは、最適な 3 点と対応する計測点からフレームを計算します。

ワークが加工位置に位置決めされると、直交機械座標系を基準とした位置は通常、その最適な位置に対してオフセットされて回転されます。正確な加工または計測をおこなうには、高価な部品の調整をおこなうか、パートプログラムで定義した動作を変更する必要があります。

フレームは、空間の 3 点の既知の最適位置のサンプリングで定義できます。サンプリングには、タッチトリガプローブまたは光学センサを使用して、支持板またはプローブボールに正確に取り付けられた特殊な穴に接触します。

構文

MEAFRAME (<最適な点>, <計測点>, <品質>)

意味

MEAFRAME:	ファンクションコール
<最適な点>:	2 次元最適な 3 点の座標を含む実配列
<計測点>:	2 次元計測した 3 点の座標を含む実配列

4.8 座標変換(フレーム)

<品質>:	フレーム演算の品質に関する情報を返す変数		
	タイプ:	VAR REAL	
	値:	-1	最適な点がほぼ直線上にあります。フレームを計算できません。返された FRAME 変数に不明のフレームが含まれます。
		-2	計測点がほぼ直線上にあります。フレームを計算できません。返された FRAME 変数に不明のフレームが含まれます。
		-4	別の理由で、回転マトリクスを計算できませんでした。
		≥ 0.0	変形量の合計(各点間の距離)。これは計測した三角形を最適な三角形と一致する三角形に変換するために必要です。

注記

計測の品質

計測した座標を最適な座標に、回転と平行移動を使用してマッピングするには、計測した点が形成する三角形が、最適な三角形に一致する必要があります。このためには、計測した三角形を最適な三角形に再形成するために必要な偏差平方和を最小にする補正アルゴリズムを使用します。

計測点の実際の変形量を使用して、計測値の品質を判断できるため、MEAFRAME は追加の変数として、この実際の変形量を返します。

注記

MEAFRAME で作成したフレームは、ADDFRAME 機能によってフレーム結合で別のフレームに変換できます(例「ADDFRAME による結合」を参照してください)。

例

例 1:

パートプログラム 1:

プログラムコード

```
...
DEF FRAME CORR_FRAME
```


4.8 座標変換(フレーム)

計測点の設定:

プログラムコード	コメント
<pre> DEF REAL IDEAL_POINT[3,3]= SET(10.0,0.0,0.0,0.0,10.0,0.0,0.0,0.0,10.0) DEF REAL MEAS_POINT[3,3]= SET(10.1,0.2,-0.2,-0.2,10.2,0.1,-0.2,0.2,9.8) DEF REAL FIT_QUALITY=0 DEF REAL ROT_FRAME_LIMIT=5 DEF REAL FIT_QUALITY_LIMIT=3 DEF REAL SHOW_MCS_POS1[3] DEF REAL SHOW_MCS_POS2[3] DEF REAL SHOW_MCS_POS3[3] </pre>	; テスト用 ; 部品位置を最大 5° だけ回転できます。 ; 最適な三角形と計測した三角形の間に最大 3 mm のオフセットが可能です。
プログラムコード	コメント
<pre> N100 G01 G90 F5000 N110 X0 Y0 Z0 N200 CORR_FRAME=MEAFRAME(IDEAL_POINT,MEAS_POINT,FIT_QUALITY) N230 IF FIT_QUALITY < 0 SETAL(65000) GOTO NO_FRAME ENDIF N240 IF FIT_QUALITY > FIT_QUALITY_LIMIT SETAL(65010) GOTO NO_FRAME ENDIF N250 IF CORR_FRAME[X,RT] > ROT_FRAME_LIMIT SETAL(65020) GOTO NO_FRAME ENDIF N260 IF CORR_FRAME[Y,RT] > ROT_FRAME_LIMIT SETAL(65021) GOTO NO_FRAME ENDIF N270 IF CORR_FRAME[Z,RT] > ROT_FRAME_LIMIT SETAL(65022) GOTO NO_FRAME ENDIF N300 \$P_IFRAME=CORR_FRAME </pre>	; 1 番目の RPY 角を制限しています。 ; 2 番目の RPY 角を制限しています。 ; 3 番目の RPY 角を制限しています。 ; サンプルフレームを設定可能フレームで起動します。 ; ジオメトリ軸を最適な点に位置決めしてフレームを確認します。

4.8 座標変換(フレーム)

プログラムコード	コメント
N400 X=IDEAL_POINT[0,0] Y=IDEAL_POINT[0,1] Z=IDEAL_POINT[0,2]	
N410 SHOW_MCS_POS1[0]=\$AA_IM[X]	
N420 SHOW_MCS_POS1[1]=\$AA_IM[Y]	
N430 SHOW_MCS_POS1[2]=\$AA_IM[Z]	
N500 X=IDEAL_POINT[1,0] Y=IDEAL_POINT[1,1] Z=IDEAL_POINT[1,2]	
N510 SHOW_MCS_POS2[0]=\$AA_IM[X]	
N520 SHOW_MCS_POS2[1]=\$AA_IM[Y]	
N530 SHOW_MCS_POS2[2]=\$AA_IM[Z]	
N600 X=IDEAL_POINT[2,0] Y=IDEAL_POINT[2,1] Z=IDEAL_POINT[2,2]	
N610 SHOW_MCS_POS3[0]=\$AA_IM[X]	
N620 SHOW_MCS_POS3[1]=\$AA_IM[Y]	
N630 SHOW_MCS_POS3[2]=\$AA_IM[Z]	
N700 G500	; 設定可能フレームをゼロフレームと同じよう に解除します(入力値もプリセット値もありませ ん)。
No_FRAME	; ゼロフレームでプリセットされるように設定 可能フレームを解除します(入力値はありません)。
M0	
M30	

例 2:フレームの結合

オフセットの MEAFRAME の結合

MEAFRAME 機能はオフセットフレームを返します。このオフセットフレームを、機能(G54 など)の呼び出し時に有効であった設定可能フレーム\$P_UIFR[1]に結合すると、設定可能フレームが得られ、移動や加工でさらに変換することができます。

ADDFRAME による結合

フレーム結合のこのオフセットフレームを別の位置で適用する場合、または他のフレームが設定可能フレームより前に有効になっている場合は、ADDFRAME 機能を使用して、チャネル基本フレームのいずれかまたはシステムフレームに結合できます。

このフレームでは以下を無効にしてください。

- MIRROR によるミラーリング
- SCALE によるスケーリング

指令値と現在位置の入力パラメータはワーク座標です。これらの座標は、必ずメトリック単位またはインチ単位(G71/G70)で、半径を基準として(DIAMOF)、制御装置の基本単位系で指定してください。

ADDFRAME の追加情報については、『機能マニュアル 基本機能』を参照してください。

4.8.8 グローバルフレーム

各制御装置のすべてのチャンネルに使用するグローバルフレームは1セットのみです。グローバルフレームは、すべてのチャンネルから読み取りと書き込みを実行できます。グローバルフレームは、それぞれのチャンネルで有効にします。

グローバルフレームを使用して、オフセットを含む**チャンネル軸**と**機械軸**のスケーリングとミラーリングを実行できます。

幾何学的関係とフレーム結合

グローバルフレームを使用する場合、各軸間の幾何学的関係はありません。したがって、回転を実行することも、ジオメトリ軸識別子をプログラム指令することもできません。

グローバルフレームでは、回転を使用できません。回転のプログラミングは、アラーム18310「Channel %1 Block %2 Frame:rotation not allowed」(フレーム:回転できません)とともに拒否されます。

グローバルフレームとチャンネル別フレームは結合できます。結果として得られるフレームには、すべての軸の回転を含むすべてのフレーム成分が含まれます。回転成分を含むフレームをグローバルフレームに割り当てると、アラーム「Frame:rotation not allowed」(フレーム:回転できません)とともに拒否されます。

グローバルフレーム

グローバル基本フレーム\$P_NCBFR[n]

8個までのグローバル基本フレームを設定できます。

チャンネル別基本フレームも使用できます。

グローバルフレームは、制御装置のすべてのチャンネルから読み取りと書き込みを実行できます。グローバルフレームに書き込むときは、チャンネル協調をおこなってください。これは、待機マーク(WAITMC)などを使用して実行できます。

注記

工作機械メーカー

グローバル基本フレームの数は、マシンデータを介して設定します。

詳細情報:機能マニュアル、基本機能

4.8 座標変換(フレーム)

設定可能フレーム(\$P_UIFR[n])

すべての設定可能フレーム G500、G54～G599 は、グローバルに、またはチャンネル別に設定できます。

注記

工作機械メーカー

すべての設定可能フレームは、マシンデータ

MD18601 \$MN_MM_NUM_GLOBAL_USER_FRAMES を利用して、グローバルフレームとして再設定できます。

チャンネル軸識別子と機械軸識別子は、フレームプログラム命令の軸識別子として使用できます。ジオメトリ軸識別子をプログラミングすると、アラームで拒否されます。

4.8.8.1 チャンネル別フレーム(\$P_CHBFR、\$P_UBFR)

設定可能フレームまたは基本フレームは、オペレータ操作と PLC による OPI およびパートプログラムを介して読み書きできます。

グローバルフレームには仕上げオフセットも使用できます。グローバルフレームは、チャンネル別フレームと同様に、G53、G153、SUPA、および G500 を使用してマスクされます。

工作機械メーカー

チャンネル内の基本フレームの数は、マシンデータ MD28081

\$MC_MM_NUM_BASE_FRAMES を介して設定できます。標準構成は、チャンネル毎に 1 個以上のフレームがあることを前提に構成されています。1 チャンネルあたり 8 個までの基本フレームがサポートされています。8 個の基本フレームに加え、8 個の NCU グローバル基本フレームをチャンネルで使用できます。

チャンネル別フレーム

\$P_CHBFR[n]

システム変数 \$P_CHBFR[n] を使用すると、基本フレームの読み取りと書き込みを実行できます。基本フレームに書き込んだときに、結合された基本フレーム全体は、G500、G54～G599 命令が実行されるまで無効です。この変数は主として HMI または PLC から基本フレームへの書き込み操作を格納するために使用されます。これらのフレーム変数は、データバックアップにより保存されます。

チャンネルの 1 番目の基本フレーム

配列インデックスが 0 の基本フレームは、予約変数 \$P_UBFR への書き込み時に同時に起動されるのではなく、G500、G54...G599 命令の実行時にのみ起動されます。この変数の読み取りと書き込みはプログラムでも実行できます。

\$P_UBFR

\$P_UBFR は \$P_CHBFR[0] と同じです。システム変数に旧バージョンとの互換性があるよう、初期設定では、常に基本フレームがチャンネルに 1 個あります。チャンネル別基本フレームがない場合は、読み取り/書き込み時に次のアラームが発生します。「Frame:statement not permissible」(フレーム: 命令を実行できません)

4.8.8.2 チャンネルの有効フレーム

チャンネルの有効フレームは、パートプログラムから、これらのフレームに関連のあるシステム変数を使用して入力します。これにはシステムフレームも含まれます。現在のシステムフレームの読み取りと書き込みは、これらのシステム変数を使用してパートプログラムで実行できます。

チャンネルの現在有効なフレーム

一覧

現在のシステムフレーム

\$P_PARTFRAME

\$P_SETFRAME

\$P_EXTFRAME

\$P_NCBFRAME[n]

\$P_CHBFRAME[n]

\$P_BFRAME

\$P_ACTBFRAME

\$P_CHBFRMASK と \$P_NCBFRMASK

\$P_IFFRAME

現在のシステムフレーム

\$P_TOOLFRAME

\$P_WPFRAME

\$P_TRAFRAME

\$P_PFRAME

用途:

TCARR と PAROT

現在位置設定と接触計測

外部ゼロオフセット

現在のグローバル基本フレーム

現在のチャンネル基本フレーム

電流 1。チャンネルの基本フレーム

基本フレーム全体

基本フレーム全体

現在の設定可能フレーム

用途:

TOROT と TOFRAME

ワークの基準点

座標変換

現在のプログラマブルフレーム

4.8 座標変換(フレーム)

現在のシステムフレーム

\$P_CYCFRAME

P_ACTFRAME

フレーム結合

用途:

サイクル

現在のフレーム全体

現在のフレームが基本フレーム全体から構成されます

\$P_NCBFRAME [n] 現在のグローバル基本フレーム

システム変数\$P_NCBFRAME[n]を使用して、現在のグローバル基本フレームのフィールド要素を読み書きできます。結果として得られる基本フレーム全体は、チャンネルでの書き込み処理により計算されます。

変更されたフレームは、そのフレームがプログラムされたチャンネル内でのみ起動されます。制御装置のすべてのチャンネルについてフレームを変更する場合は、\$P_NCBFR[n]と\$P_NCBFRAME[n]を同時に書き込んでください。そのあと、その他のチャンネルでG54などでフレームを起動します。基本フレームに書き込むたびに、基本フレーム全体が再計算されます。

\$P_CHBFRAME[n] 現在のチャンネル基本フレーム

システム変数\$P_CHBFRAME[n]を使用して、現在のチャンネル基本フレームのフィールド要素を読み書きできます。結果として得られる基本フレーム全体は、チャンネルでの書き込み処理により計算されます。基本フレームに書き込むたびに、基本フレーム全体が再計算されます。

\$P_BFRAME 現在の、チャンネルの1番目の基本フレーム

予約フレーム変数\$P_BFRAMEを使用して、チャンネル内で有効な配列インデックスが0の現在の基本フレームを、パートプログラム内で読み書きできます。書き込んだ基本フレームは、直ちに計算されます。

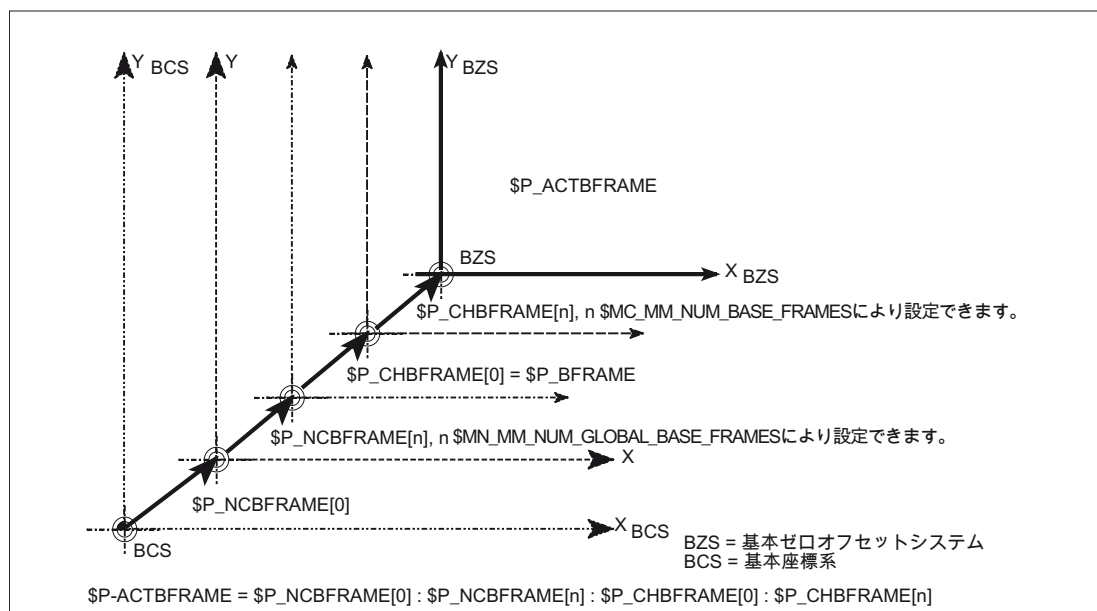
\$P_BFRAMEは\$P_CHBFRAME[0]と同じです。システム変数には常に、有効な初期値を保持します。チャンネル別基本フレームがない場合は、読み取り/書き込み時に次のアラームが発生します。「Frame:statement not permissible」(フレーム: 命令を実行できません)

\$P_ACTBFRAME 基本フレーム全体

\$P_ACTFRAME 変数は、結合された基本フレーム全体を特定します。この変数は読み取り専用です。

\$P_ACTFRAME は、以下に対応します。

\$P_NCBFRAME[0] : ... : \$P_NCBFRAME[n] : \$P_CHBFRAME[0] : ... : \$P_CHBFRAME[n]



$\$P_CHBFRMASK$ と $\$P_NCBFRMASK$ 基本フレーム全体

システム変数 $\$P_CHBFRMASK$ と $\$P_NCBFRMASK$ を使用して、どの基本フレームを基本フレーム「全体」の計算に含めるかを選択できます。これらの変数はプログラム内でのみプログラム指令でき、OPI を介して読み取ることができます。変数の値はビットマスクとして解釈され、 $\$P_ACTFRAME$ のどの基本フレームフィールドの要素が計算に含まれるかを指定します。

$\$P_CHBFRMASK$ を使用して、どのチャンネル別基本フレームを計算に含めるかを指定できます。また、 $\$P_NCBFRMASK$ を使用して、どのグローバル基本フレームを計算に含めるかを指定できます。

変数をプログラム指令すると、基本フレーム全体とフレーム全体が再計算されます。リセット後および初期設定では、 $\$P_CHBFRMASK$ と $\$P_NCBFRMASK$ の値は次のようになります。

$\$P_CHBFRMASK = \$MC_CHBFRAME_RESET_MASK$

$\$P_NCBFRMASK = \$MC_CHBFRAME_RESET_MASK$

例:

$\$P_NCBFRMASK = 'H81' ; \$P_NCBFRAME[0] : \$P_NCBFRAME[7]$

$\$P_CHBFRMASK = 'H11' ; \$P_CHBFRAME[0] : \$P_CHBFRAME[4]$

4.8 座標変換(フレーム)

\$P_IFRAME 現在の設定可能フレーム

予約フレーム変数\$P_IFRAME を使用して、チャンネル内で有効な現在の設定可能フレームを、パートプログラム内で読み書きできます。書き込んだ設定可能フレームは、すぐに計算されます。

グローバル設定可能フレームの場合、変更されたフレームは、そのフレームがプログラムされたチャンネル内でのみ機能します。制御装置のすべてのチャンネルについてフレームを変更する場合は、P_UIFR[n]と\$P_IFRAME を同時に書き込んでください。それから、その他のチャンネルで G54 などに対応するフレームを起動してください。

\$P_PFRAME 現在のプログラマブルフレーム

\$P_PFRAME は、TRANS/ATRANS、G58/G59、ROT/AROT、SCALE/ASCALE、MIRROR/AMIRROR のプログラム指令の結果、またはプログラマブルフレームへの CTRANS、CROT、CMIRROR、CSCALE の割り当ての結果のプログラマブルフレームです。

設定可能ゼロオフセットシステム(SZS)とワーク座標系(WCS)の間の位置関係を確立する現在のプログラマブルフレーム変数。

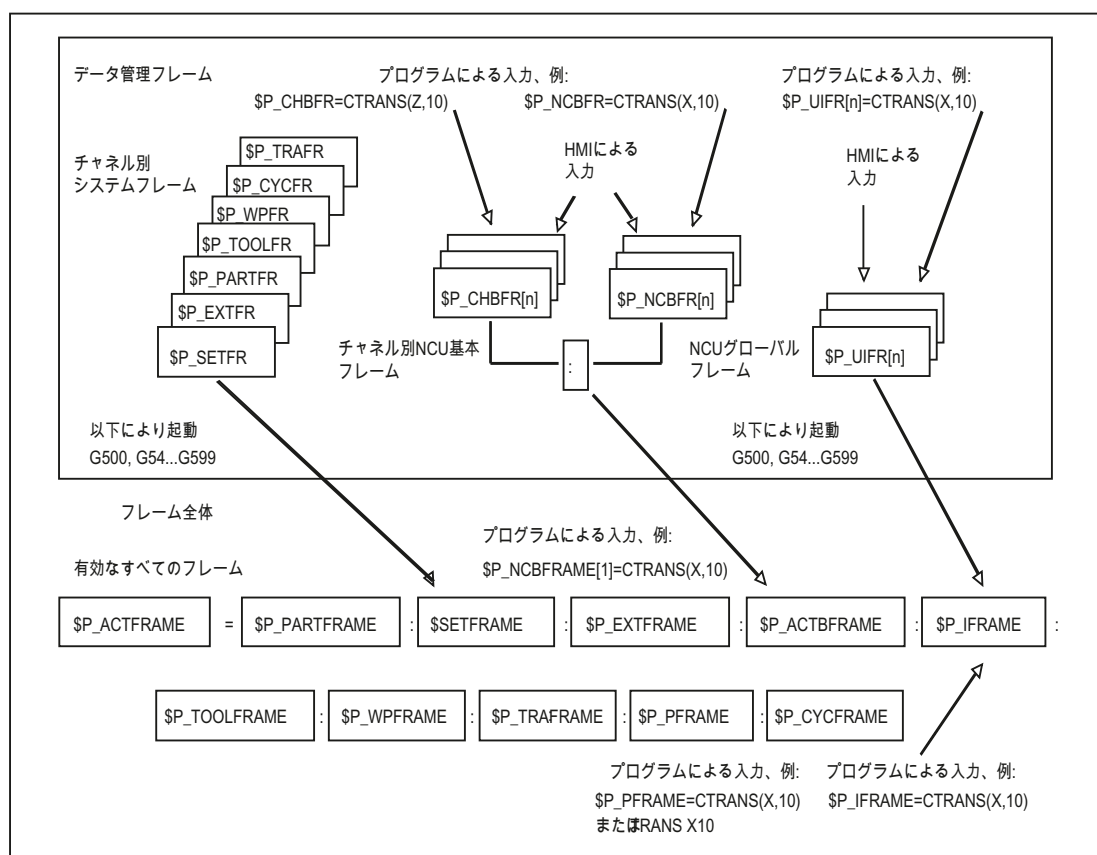
P_ACTFRAME 現在のフレーム全体

結果として得られる現在のフレーム全体\$P_ACTFRAME は、基本フレームのすべて、現在の設定可能フレーム、およびプログラマブルフレームを結合したものです。現在のフレームは、フレーム成分の変更のたびに必ず更新されます。

\$P_ACTFRAME は、以下に対応します。

\$P_PARTFRAME : \$P_SETFRAME : \$P_EXTFRAME : \$P_ACTBFRAME : \$P_IFRAME :
\$P_TOOLFRAME : \$P_WPFRAME : \$P_TRAFRAME : \$P_PFRAME : \$P_CYCFRAME

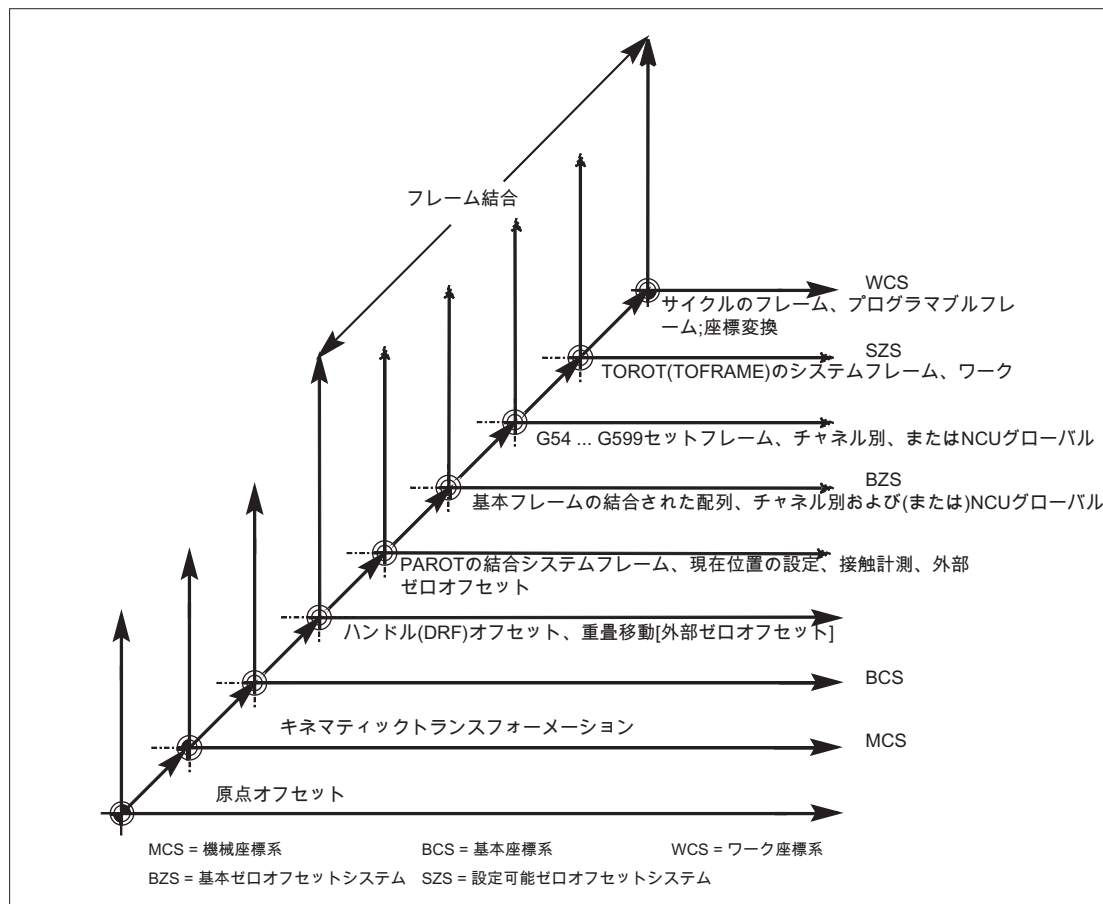
4.8 座標変換(フレーム)



4.9 座標変換

フレーム結合

現在のフレームは、上記で指定した現在のフレーム全体に従って、基本フレーム全体、設定可能フレーム、システムフレーム、およびプログラマブルフレームから構成されています。



4.9 座標変換

4.9.1 座標変換タイプの一般的なプログラミング

座標変換は、コントローラを別の機械のキネマティクスに設定します。座標変換は、NCプログラムのファンクションコールで有効になります。これらのパラメータを使用すると、空間の工具の向きと回転軸の旋回移動の両方を定義できます。

方向座標変換(3軸、4軸、および5軸座標変換)では、プログラム指令した位置データが常に工具の先端と対応し、空間でその工具は加工する平面に直角に追従します。直交座標系

は、基本座標系から機械座標系に変換され、ジオメトリ軸に対応します。これらは、動作点を記述します。仮想回転軸は、空間の工具の向きを記述します。

キネマティック座標変換(TRANSMIT、TRACYL、TRAANG)の場合は、位置を直交座標系でプログラム指令できます。コントロールシステムは、プログラム指令した直交座標系の移動動作を、実際の機械軸の移動動作に変換します。

プログラミング

方向座標変換の適用

TRAORI (<座標変換番号>, <配向ベクトル>, <回転軸オフセット>)

キネマティック座標変換の適用

TRANSMIT (<座標変換番号>)

TRACYL (<作業直径>, <座標変換番号>)

TRAANG (<傾斜軸の角度>, <座標変換番号>)

有効な座標変換の無効化

TRAFOOF

方向座標変換

3 軸、4 軸、5 軸座標変換(TRAORI)

機械の作業領域の空間に構成された表面を最適に加工するために、工作機械には、3 つの直線軸 X、Y、および Z に加え、追加の軸が必要です。空間の向きを記述するこれらの付加軸は、後続の章で旋回軸と呼ばれます。それらは、異なったキネマティックスで 4 つの機械タイプの回転軸として使用できます。

1. 2 軸旋回ヘッド
固定式工具テーブルによる直線軸に並行な 1 つの回転軸つきカルダン式工具ヘッドなど
2. 2 軸回転テーブル
2 つの軸を中心にして回転する工具テーブル付き固定式スイベルヘッドなど
3. 単独軸スイベルヘッドと単独軸回転テーブル
1 つの軸を中心にして回転する工具テーブルによる回転した工具付き回転可能スイベルヘッドなど
4. 2 軸スイベルヘッドと単独軸回転テーブル
1 つの軸を中心にして回転可能な工具テーブルと工具自体を中心にして回転する工具のある回転可能なスイベルヘッド付き

3 軸と 4 軸座標変換は 5 軸座標変換の特殊なタイプで、5 軸座標変換と同じ方法でプログラム指令します。

4.9 座標変換

汎用 3/4/5/6 軸座標変換」の対象機能は、直交回転軸の座標変換とカルダン式ミリングヘッドの座標変換の両方に対応しており、他のすべての方向座標変換と同様に、これらの 4 種類の機械に対しても TRAORI で有効にできます。汎用 5/6 軸座標変換では、工具オリエンテーションにさらに 3 次元の自由度があり、そのため工具は空間で工具方向に対応する工具軸を中心として回転できます。

詳細:『機能マニュアル』、「座標変換」

キネマティックスに影響しない工具オリエンテーションの初期設定

ORIRESET

TRAORI による方向座標変換が有効の場合は、ORIRESET を使用して、3 つまでの旋回軸の初期設定を任意選択パラメータ A、B、C で指定できます。プログラム指令パラメータは、座標変換で定義した旋回軸の順序に従って回転軸に割り当てられます。ORIRESET(A, B, C) をプログラム指令すると、各旋回軸がその現在の位置から指定した初期設定位置へと、直線的に同期しながら移動します。

キネマティックトランスフォーメーション

TRANSMIT と TRACYL

旋盤でのミリングの場合、以下の加工タイプは取り決めた座標変換に対して有効にできます。

- TRANSMIT による回転式チャックでの正面加工
- TRACYL による、円筒上の任意の連続する溝加工

TRAANG

傾斜切り込みの切り込み軸のオプション設定が必要な場合(研削加工の場合など)は、TRAANG を使用して宣言した座標変換に対して設定可能な角度をプログラム指令できます。

直交 PTP 移動

キネマティックトランスフォーメーションには「直交 PTP 移動」が含まれます。最大 8 つの異なる関節継手位置(STAT=...)をプログラム指令できます。ただし、位置が直交座標系でプログラム指令され、機械の移動は機械座標系で実行されます。

詳細:『機能マニュアル』、「座標変換」

座標変換重畳

2 つの座標変換は、連続して切り替えることができます。ここで結合された 2 番目の座標変換では、軸の移動部分は 1 番目の座標変換から取得されます。

1 番目の変換は、次のいずれかです。

- 方向座標変換 TRAORI
- 極座標変換 TRANSMIT
- 円筒座標変換 TRACYL
- 傾斜軸座標変換 TRAANG

2 番目の座標変換は傾斜軸 TRAANG にしてください。

4.9.1.1 座標変換の旋回移動

移動と旋回移動

プログラム指令の向きの移動動作は、主に機械の種類で特定されます。TRAORI による 3 軸、4 軸、5 軸タイプの座標変換の場合は、回転軸、または直線軸まわりの回転で工具の旋回移動を記述します。

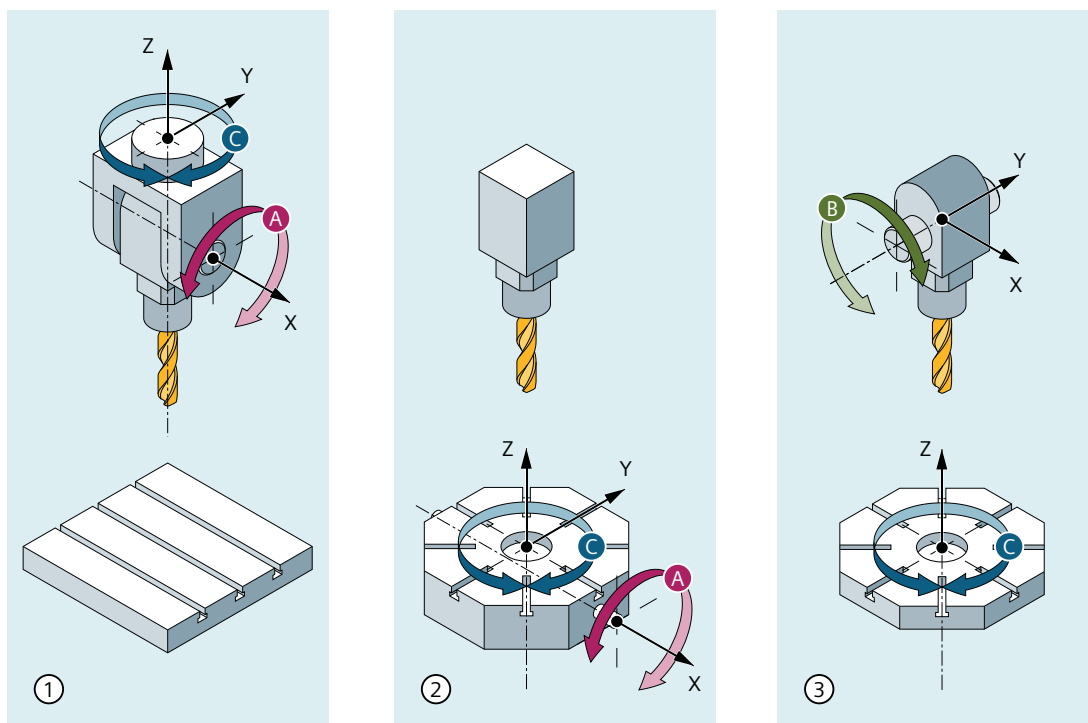
方向座標変換に関わる回転軸の位置の変更により、その他の機械軸で補正移動が発生します。工具先端の位置はそのまま変更されません。

工具の旋回移動は、用途に合わせて仮想軸の回転軸識別子 A...、B...、C...を使用してプログラム指令できます。この場合、オイラー角または RPY 角、方向ベクトルまたは面法線ベクトル、円錐の回転軸、または円錐面上の中間旋回用に正規化ベクトルを入力します。

TRANSMIT、TRANSMIT、TRACYL、および TRAANG によるキネマティックトランスフォーメーションの場合は、コントローラが、プログラム指令した直交座標系の移動を、実際の機械軸の移動に割り当てます。

3 軸、4 軸、5 軸座標変換 TRAORI の機械のキネマティックス

工具または工具テーブルは、最大 2 つの回転軸を使用して回転できます。スイベルヘッドと回転テーブル(いずれも単独軸のもの)の組み合わせも可能です。



- ① 軸の順序が CA の機械タイプ 1
 - 工具ホルダは X 軸と Z 軸を中心に回転可能
 - ワークテーブル固定
- ② 軸の順序が AC の機械タイプ 2
 - 工具ホルダ固定
 - ワークテーブルは X 軸と Z 軸を中心に回転可能
- ③ 軸の順序が BC の機械タイプ 3
 - 工具ホルダは Y 軸を中心に回転可能
 - ワークテーブルは Z 軸を中心に回転可能

機械のタイプ	方向のプログラミング
3 軸座標変換の機械のタイプ 1 とタイプ 2	工具オリエンテーションのプログラミングは、回転軸に垂直な平面上のみです。 2 つの平行移動の軸(直線軸)、および 1 つの回転の軸(回転軸)があります。
4 軸座標変換の機械のタイプ 1 と 2	工具オリエンテーションのプログラミングは、回転軸に垂直な平面上のみです。 3 つの平行移動の軸(直線軸)、および 1 つの回転の軸(回転軸)があります。
5 軸座標変換機械タイプ 3 単独軸スィベルヘッド、および単独軸回転テーブル	方向座標変換のプログラミング 3 つの直線軸、および 2 つの直交する回転軸によるキネマティックスです。 回転軸は、3 つの直線軸の 2 つに平行です。2 つの直交する直線軸により、1 番目の回転軸が移動します。これにより、3 つ目の直線軸が工具と一緒に回転します。2 番目の回転軸は、ワークを回転します。

汎用 5/6 軸座標変換

機械のタイプ	方向座標変換のプログラミング
汎用 5/6 軸座標変換機械タイプ 4 工具自身を中心に回転する工具付き 2 軸スィベルヘッド、および単独軸回転テーブル	方向座標変換のプログラミング 3 つの直線軸、および 3 つの直交する回転軸によるキネマティックス。 回転軸は、3 つの直線軸の 2 つに平行です。2 つの直交する直線軸により、1 番目の回転軸が移動します。これにより、3 つ目の直線軸が工具と一緒に回転します。2 番目の回転軸は、ワークを回転します。基本の工具オリエンテーションは、THETA 回転角度による、工具自身を中心とした追加回転でもプログラム指令できます。

「汎用 3 軸、4 軸、および 5/6 軸座標変換」を呼び出すと、基本の工具オリエンテーションをおこなうこともできます。回転軸の方向に関する制限事項は、ここでは適用されません。回転軸が互いに正確に垂直ではないか、既存の回転軸が直線軸に正確に平行はでない場合は、「汎用 5/6 軸座標変換」により、工具オリエンテーションに関しては良好な結果が得られます。

4.9 座標変換

キネマティックトランスフォーメーション TRANSMIT、TRACYL、および TRAANG

旋削機械のフライス加工、または研削中に傾斜切り込みに設定できる軸の場合は、初期設定では宣言している座標変換に従って、スレーブ軸の配置が適用されます。

TRANSMIT	極座標変換の適用
回転式チャックでの正面加工	回転軸 回転軸に垂直な切り込み軸 回転軸に平行な長手軸

TRACYL	円筒補間の適用
円筒上の任意の軌跡の溝加工	回転軸 回転軸に垂直な切り込み軸 回転軸に平行な長手軸

TRAANG	傾斜軸座標変換の適用
傾斜切り込み軸による加工	回転軸 パラメータ設定可能な角度の切り込み軸 回転軸に平行な長手軸

直交 PTP 移動

機械は機械座標で移動し、以下を使用してプログラム指令します。

TRAORI	座標変換の適用
PTP ポイントツーポイント移動	直交座標系(MCS)位置へのアプローチ
CP	BCS の直交軸の軌跡移動
STAT	関節継手の位置は座標変換により異なります
TU	軸が最短軌跡で移動するときの角度

汎用 5 軸/6 軸座標変換による PTP 移動

機械が、機械座標と工具オリエンテーションを使用して移動します。この移動は、キネマティックスまたは方向ベクトルに関係なく、回転軸の位置、およびオイラー角ベクトルと RPY 角ベクトルのは両方を使用してプログラム指令できます。

このような場合は、回転軸補間、大円弧補間のベクトル補間、または円錐面の配向ベクトルの補間が可能です。

例:ユニバーサルミリングヘッドでの 3 軸～5 軸座標変換

工作機械には、軸が 5 軸以上あります。

- 直線移動のための 3 つの平行移動軸。この軸で、作業領域の任意の位置に動作点を移動します。
- 設定可能な角度(通常は 45°)に配置された 2 つの回転する旋回軸。この軸で、45°刻みで半球に制限されている空間内の位置で工具を旋回します。

4.9 座標変換

4.9.1.2 方向座標変換 TRAORI の一覧

TRAORI と組み合わせて使用できるプログラミングタイプ

機械のタイプ	動作中の座標変換 TRAORI によるプログラミング
機械タイプ 1、2、または 3 の 2 軸スイベルヘッドまたは 2 軸回転テーブル、または単独軸スイベルヘッドと単独軸回転テーブルの組み合わせです。	旋回軸の軸の順序と工具の旋回方向は、機械のキネマティックスに応じて、マシンデータを使用して 機械別基準で設定するか、 または、機械のキネマティックスとは無関係に、ワーク別基準で変更可能な向きに設定することができます。 基準系の旋回軸の回転方向は、次のようにプログラム指令します。 - ORIMKS 基準系=機械座標系 - ORIWKS 基準系=ワーク座標系 初期設定は ORIWKS です。 旋回軸は、次のようにプログラミングします。 直接の機械軸位置 A、B、C A2、B2、C2 仮想軸のプログラム指令角度 - オイラー角(標準)による ORIEULER - RPY 角による ORIRPY - 1 番目の定義の仮想旋回軸による ORIVIRT1 - 2 番目の定義の仮想旋回軸による ORIVIRT2 これは次の補間タイプ間で違いがあります: 直線補間 - 旋回軸または機械軸の ORIAXES 大半径円弧補間(配向ベクトルの補間) - 旋回軸からの ORIVECT ベクトル成分の A3、B3、C3(方向/面法線)の指定による旋回軸のプログラミング 得られた工具オリエンテーションのプログラミング ブロック始点の面法線ベクトルの A4、B4、C4 ブロック終点で面に垂直なベクトルの A5、B5、C5 LEAD 工具オリエンテーションのリード角 TILT 工具オリエンテーションの傾斜角

機械のタイプ	動作中の座標変換 TRAORI によるプログラミング
	円錐面の配向ベクトルの補間 次の補間を使用した、空間の任意の場所で円錐面への向きの変更: - 平面での ORIPLANE(大半径円弧補間) - 円錐面での右回り方向への ORICONCW - 円錐面での左回り方向への ORICONCCW 、A6、B6、C6 方向ベクトル(円錐の回転軸) -円錐面での OICONIO 補間で、 A7、B7、C7 中間ベクトル(最初または最終の向き)による、または - 円錐面での ORICONTO、接線方向の遷移 以下による、軌跡に対する向きの変更 - ORICURVE 以下による 2 つの接点の移動の指定 PO[XH]=(xe, x2, x3, x4, x5)旋回多項式(5 次以下) PO[YH]=(ye, y2, y3, y4, y5)旋回多項式(5 次以下) PO[ZH]=(ze, z2, z3, z4, z5)旋回多項式(5 次以下) - ORIPATHS 後退移動のときの工具の方向、および軌跡長に対応する工具の A8、B8、C8 の再旋回状態の 旋回処理のスムージング。

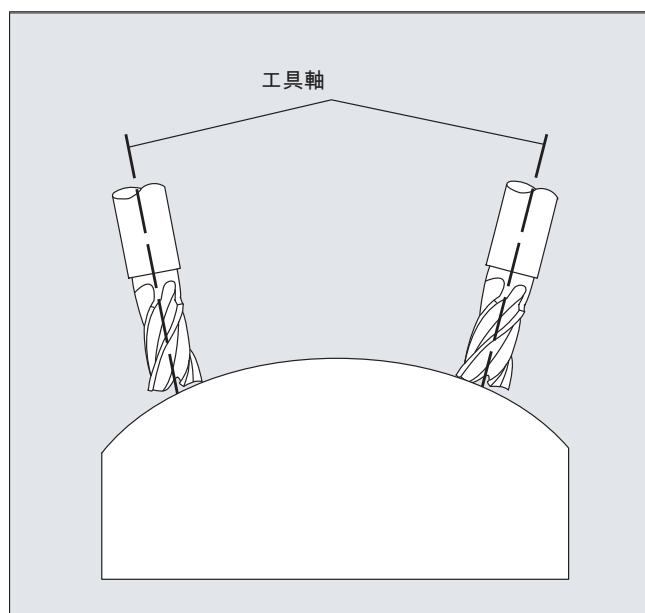
4.9 座標変換

機械のタイプ	動作中の座標変換 TRAORI によるプログラミング
機械タイプ 1 と 3 付加工具自身がその工具を中心として回転する、その他の機械タイプには、3 番目の回転軸が必要です。 方向座標変換は汎用 6 軸座標変換などです。配向ベクトルの回転。	工具オリエンテーションの回転のプログラミング LEAD 角、面法線ベクトルに対する角度 PO[PHI] 5 次までの多項式のプログラミング TILT 軌跡タンジェント(Z 方向)を中心とした回転角度 PO[PSI] 5 次までの多項式のプログラミング THETA 回転角度(Z の工具方向を中心とした回転) THETA=ブロック終点の到達値 THETA=AC(...)指令へのアブソリュートノンモーダル切り替え THETA=IC(...)結合指令へのノンモーダル切り替え THETA=Θ。プログラム指令角度 G90/G91 の補間 PO[THT]=(..) 5 次までの多項式のプログラミング 回転ベクトルのプログラミング - ORIOTA アブソリュート回転ベクトル - ORIOTR 相対回転ベクトル - ORIOTT 接線方向の回転ベクトル
軌跡に対する向きの変更のための軌跡に対する旋回、または軌跡の接線方向の回転ベクトルの回転	以下による、軌跡に対する向きの変更 - ORIPATH 軌跡に対する工具オリエンテーション - ORIPATHS 旋回処理が不連続変化した場合の回転ベクトルのプログラミング - ORIOTC 接線方向の回転ベクトル、軌跡タンジェントに対する回転

4.9.2 3 軸、4 軸、5 軸座標変換(TRAORI)

4.9.2.1 ユニバーサル工具ヘッドの一般的な関係

3 次元曲線の加工で最適な切削条件を得るために、工具の設定角度を変えることができます。



これを可能にする機械構成が軸データに格納されています。

5 軸座標変換

カルダン式工具ヘッド

ここで、3 つの直線軸(X、Y、Z)および 2 つの旋回軸(C、A)は、工具の設定角度と動作点を定義します。2 つの旋回軸の 1 つは傾斜軸として作成し、ここでの A' の例では、多くの場合、45°の角度で配置します。

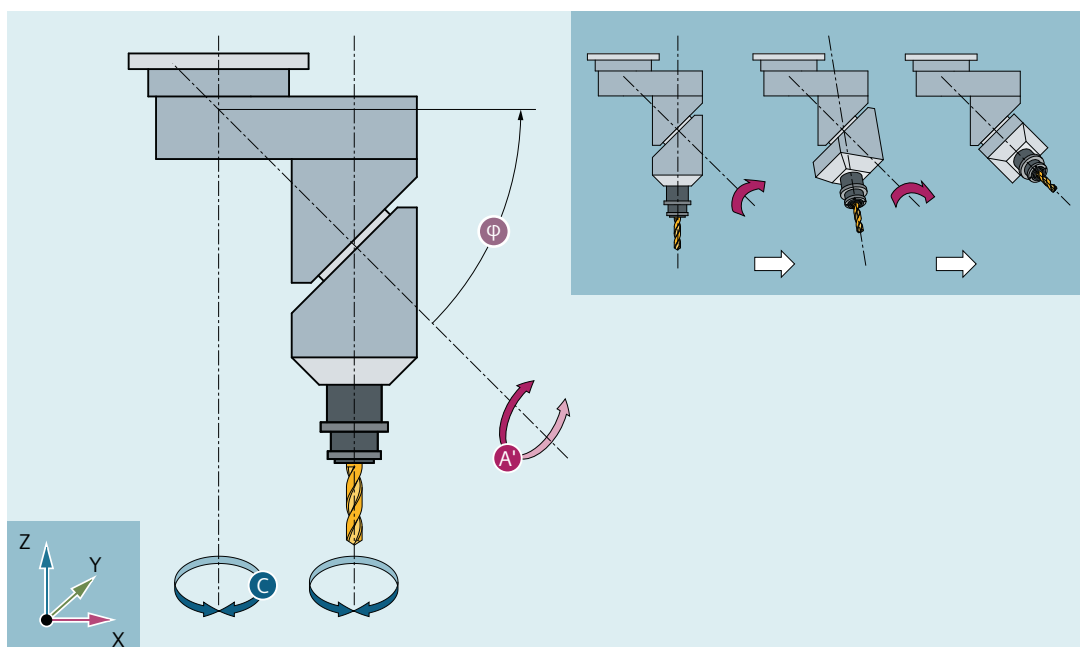
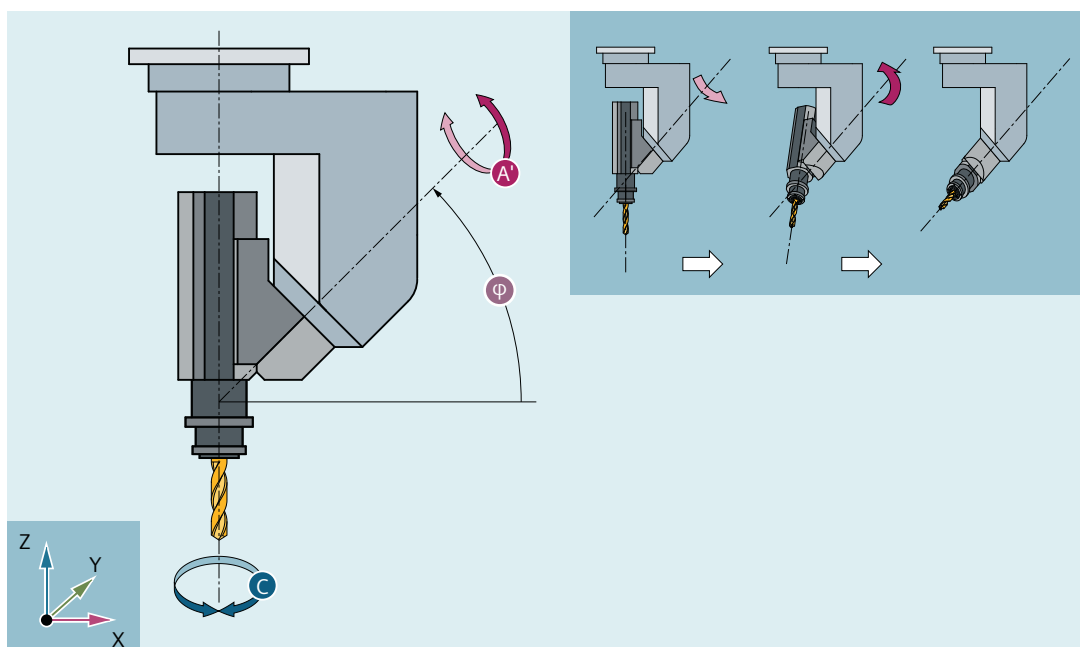


図 4-3 カルダン式工具ヘッドタイプ 1

ここで示す例では、その配置は、カルダン式工具ヘッド付き CA タイプの機械のキネマティックスで説明されています。

旋回軸の軸の順序と工具の旋回方向は、機械のキネマティックスに合わせてマシンデータを使用して設定できます(→ 機械メーカー)。



この例では、A'は X 軸に対して角度 φ の位置にあります。

次の関係が一般的に有効です。

A'は X 軸に対して角度 ϕ の位置にあります X 軸

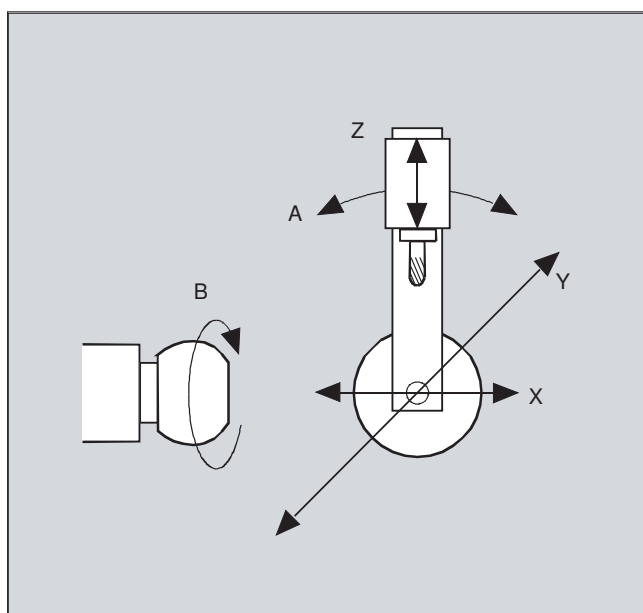
B'は Y 軸に対して角度 ϕ の位置にあります Y 軸

C'は Z 軸に対して角度 ϕ の位置にあります Z 軸

角度 ϕ は、マシンデータを使用して、 $0^{\circ} \sim +89^{\circ}$ の範囲で設定できます。

直線軸旋回機構の場合

これは、移動するワークと移動する工具の配置です。キネマティックスは、3 つの直線軸 (X、Y、Z)、および 2 つの直交する回転軸から成ります。たとえば、1 番目の回転軸が 2 つの直線軸の合成スライドで移動し、工具は 3 番目の直線軸に平行に停止しているとします。2 番目の回転軸は、ワークを回転します。3 番目の直線軸(旋回軸)は合成スライドの平面にあります。



回転軸の軸の順序と工具の旋回方向は、機械のキネマティックスに合わせてマシンデータを使用して設定できます。

以下の関係が可能です。

軸:	軸の順序:
1 番目の回転軸	A A B B C C
2 番目の回転軸	B C A C A B
直線軸旋回機構	Z Y Z X Y X

4.9 座標変換

工具の旋回方向について設定可能な軸の順序についての**追加情報**は、『機能マニュアル』、「座標変換」を参照してください。

4.9.2.2 3 軸、4 軸、5 軸座標変換(TRAORI)

2 つまたは 3 つの平行移動軸、および 2 つの回転軸を構成できます。座標変換では、回転軸が旋回平面と直交していることを前提としています。

工具の旋回は回転軸に垂直な平面上でのみ可能です。座標変換は、可動式工具と可動式ワークの機械タイプをサポートしています。

3 軸と 4 軸座標変換は、5 軸座標変換と同じ方法で設定され、プログラム指令します。

構文

```
TRAORI (<n>)  
TRAORI (<n>,<X>,<Y>,<Z>,<A>,<B>)  
TRAFOOF
```

意味

TRAORI:	1 番目に指定した方向座標変換を起動します	
TRAORI (<n>):	n で指定した方向座標変換を起動します	
<n>:	座標変換の番号	
	値:	1 または 2
	例: TRAORI(1)は方向座標変換 1 を有効にします	
<X>,<Y>,<Z>:	工具が指す配向ベクトル成分	
<A>,:	回転軸のプログラマブルオフセット	
TRAFOOF:	座標変換の解除	

工具オリエンテーション

工具に対して選択された旋回方向に応じて、工具長補正が工具オリエンテーションの方向に働くように、有効な作業平面(G17、G18、G19)を NC プログラムで設定してください。

注記

座標変換が有効なときは、位置決めデータ(X、Y、Z)は常に工具の先端に対応しています。座標変換に関わる回転軸の位置を変更すると、それに合わせて、残りの機械軸が補正移動をおこないます。このため、工具先端の位置は変わりません。

方向座標変換は常に、工具先端から工具アダプタまでを示します。

旋回軸のオフセット

方向座標変換を有効にすると、旋回軸に対して追加オフセットを直接プログラム指令できます。

正しい順序でプログラム指令している場合は、パラメータを省略できます。

例:

TRAORI (, , , , A, B) (オフセットを 1 つだけ入力する場合)

直接プログラミングに代わる方法として、旋回軸の追加オフセットを、現在動作中のワークオフセットから自動的に転送することもできます。転送はマシンデータに設定します。

例

TRAORI (1, 0, 0, 1)	; 工具の基本の向きは Z 方向です
TRAORI (1, 0, 1, 0)	; 工具の基本の向きは Z 方向です
TRAORI (1, 0, 1, 1)	; 工具の基本の向きは Y/Z 方向です (-45°の位置になります)

4.9.2.3 向きのプログラミングのタイプと初期設定(ORIRESET)

TRAORI による工具オリエンテーションのプログラミング

プログラム指令可能な TRAORI 方向座標変換と組み合わせるには、直線軸 X、Y、Z の他に回転軸識別子 A...、B...、C... も使用して、軸位置、または角度またはベクトル成分で仮想軸をプログラム指令できます。旋回軸と機械軸では、さまざまなタイプの補間を実行できます。現在有効な PO[角度]旋回多項式、および PO[軸]軸多項式に関係なく、異なったタイプの多くの多項式をプログラム指令できます。これには G1、G2、G3、CIP または POLY が含まれます。

4.9 座標変換

工具オリエンテーションの変更は、配向ベクトルでプログラム指令できる場合もあります。このような場合では、各ブロックの最終的な向きを、ベクトルの直接プログラミングまたは回転軸位置のプログラミングによって設定できます。

3 軸～5 軸座標変換の向きのプログラミングタイプ

以下に示す向きのプログラムのバージョンは、相互に排他的です。

A、B、C	回転軸位置の直接入力。
A2、B2、C2	オイラー角または RPY 角を使用した仮想軸の角度のプログラム
A3、B3、C3	ベクトル成分項目
LEAD、TILT	軌跡と面に対するリード角と傾斜角の指定
A4、B4、C4 A5、B5、C5	ブロックの始点とブロックの終点にある、面法線ベクトル
A6、B6、C6 A7、B7、C7	テーパ面変換における配向ベクトルの補間
A8、B8、C8	工具の方向転換、後退運動の方向と軌跡長さ

工具オリエンテーションの初期設定のアプローチ(ORIRESET)

ORIRESET(...) を介して、関連する機械のキネマティックスの旋回軸を、現在の位置から、プログラムされた初期状態位置へ、線形的かつ同期的に移動します。軸の基本位置がプログラムされていない場合、関連するマシンデータ \$MC_TRAFO5_ROT_AX_OFFSET_1/2 の位置が使用されます。

回転軸の有効なフレームは無視されます。

機械のキネマティックス CA(チャネル軸名称 C、A)の例

命令	説明
ORIRESET(90, 45)	軸 C: 90° 軸 A: 45°
ORIRESET(, 30)	軸 C: \$MC_TRAFO5_ROT_AX_OFFSET_1/2[0] 軸 A: 30°
ORIRESET()	軸 C: \$MC_TRAFO5_ROT_AX_OFFSET_1/2[0] 軸 A: \$MC_TRAFO5_ROT_AX_OFFSET_1/2[1]

機械のキネマティックス CAC(チャンネル軸名称 C、A、B)の例

命令	説明
ORIRESET(90, 45, 90)	軸 C: 90° 軸 A: 45° 軸 B: 90°
ORIRESET()	軸 C: \$MC_TRAFO5_ROT_AX_OFFSET_1/2[0] 軸 A: \$MC_TRAFO5_ROT_AX_OFFSET_1/2[1] 軸 B: \$MC_TRAFO5_ROT_AX_OFFSET_1/2[2]

注記

ORIRESET...)による工具オリエンテーションの初期状態への移動は、有効なオリエンテーション座標変換 TRAORI...)とともにのみ実行できます。

LEAD、TILT、および THETA 回転のプログラミング

リード角 LEAD と傾斜角 TILT。

3 軸～5 軸座標変換に関しては、工具オリエンテーションの回転を LEAD 角と TILT 角でプログラム指令します。

回転角度 THETA

3 番目の回転軸による座標変換の場合、工具自身の回転を、ベクトル成分のオリエンテーション、および角度 LEAD と TILT のプログラムのために、THETA 回転角度でプログラムできます。

4.9.2.4 工具オリエンテーションのプログラミング(A...、B...、C...、LEAD、TILT):

工具オリエンテーションのプログラム指令のときは、次のオプションを使用できます。

1. 回転軸の移動の直接プログラミング。基本座標系または機械座標系では、向きの変更が常に発生します。同期軸として旋回軸を移動します。
2. A2, B2, C2 を使用した角度定義に従ったオイラー角または RPY 角のプログラミング。
3. A3, B3, C3 を使用した方向ベクトルのプログラミング。方向ベクトルは、工具先端から工具アダプタの方向を指します。
4. A4, B4, C4 によるブロック始点、および A5, B5, C5 によるブロック終点の面法線ベクトルのプログラミング(正面削り)。
5. リード角 LEAD と傾斜角 TILT を使用したプログラミング。

4.9 座標変換

6. A6, B6, C6 を使用した正規化ベクトルとしての円錐の回転軸、または A7, B7, C7, を使用した円錐面の中間の向きプログラミング。「円錐面に沿った向きのプログラミング (ORIPLANE、ORICONxx)」を参照してください。
7. A8, B8, C8, を使用した後退移動のときの再旋回、工具の方向と軌跡長のプログラミングについては、「旋回処理のスージング (ORIPATHS A8=, B8=, C8=)」を参照してください。

注記

すべての場合において、方向座標変換が動作中の場合にのみ、向きをプログラム指令できます。

長所:これらのプログラムは、あらゆる機械のキネマティックスに適用できます。

G 命令による工具オリエンテーションの定義

注記

工作機械メーカー

マシンデータを使用すると、オイラー角と RPY 角を切り替えることができます。従って、マシンデータを適切に設定した場合は、有効なグループ 50 の G 命令に対応した形でも、対応していない形でも切り替えが可能です。以下の設定オプションを選択できます。

1. 回転軸と回転角度を定義する両方マシンデータを G 命令でゼロに設定している場合は、A2, B2, C2 を使用してプログラム指令された角度が **マシンデータにより異なります**。また、向きのプログラミングの角度の定義が、オイラー角または RPY 角として解釈されます。
2. 回転軸を定義するマシンデータを G 命令で 1 に設定した場合は、動作中のグループ 50 の G 命令によって、切り替えが **異なります**。
A2, B2, C2 を使用してプログラムされた角度は、動作中の G 命令 ORIEULER、ORIRPY、ORIVIRT1、ORIVIRT2、ORIXPOS、および ORIPY2 に従って解釈されます。回転軸でプログラムした値も、動作中のグループ 50 の G 命令に従って旋回角度として解釈されます。
3. 旋回角度を定義するマシンデータを G 命令で 1 に、回転軸を定義するマシンデータを G 命令で 0 に設定した場合は、動作中のグループ 50 の G 命令によって、切り替えは **変わりません**。
A2, B2, C2 を使用してプログラムした角度は、動作中の G 命令 ORIEULER、ORIRPY、ORIVIRT1、ORIVIRT2、ORIXPOS、および ORIPY2 のいずれかに従って解釈されます。回転軸でプログラムした値は、有効なグループ 50 の G 命令にかかわらず、常に回転軸の位置として解釈されます。

構文

回転軸の位置

G1 X<値> Y<値> Z<値> A<値> B<値> C<値>

オイラー角

G1 X<値> Y<値> Z<値> A2<値> B2<値> C2<値>

方向ベクトル

G1 X<値> Y<値> Z<値> A3<値> B3<値> C3<値>

ブロック始点での面法線ベクトル

G1 X<値> Y<値> Z<値> A4<値> B4<値> C4<値>

ブロック終点での面法線ベクトル

G1 X<値> Y<値> Z<値> A5<値> B5<値> C5<値>

リード角

LEAD=<値>

傾斜角

TILT=<値>

意味

G1:	直線補間
X, Y, Z:	直線軸の位置
A、B、C:	回転軸の位置
A2=、B2=、C2=:	角度のプログラム(オイラー角または RPY 角)
A3=、B3=、C3=:	WCS による X、Y、Z 座標の方向ベクトル。
A4=、B4=、C4=:	WCS による X、Y、Z 座標のブロック始点の面法線ベクトル。
A5=、B5=、C5=:	WCS による X、Y、Z 座標のブロック終点の面法線ベクトル。
LEAD= :	リード角 ¹⁾
TILT= :	傾斜角 ¹⁾
1) 角度の指定の解釈は、MD21094 \$MC_ORIPATH_MODE での設定によって異なります。	

詳細情報

5 軸プログラムは通常、CAD/CAM システムで生成され、制御装置では入力されません。そのため、次の説明は主に、ポストプロセッサのプログラマを対象としています。

4.9 座標変換

向きのプログラムには、以下のコマンドを使用できます。

命令	意味
ORIEULER:	回転処理 ZX'Z"のオイラー角
ORIRPY:	回転処理 XY'Z"の RPY 角
ORIRPY2:	回転処理 ZY'X"の RPY 角
ORIVIRT1:	以下を通じて自由に定義可能な、回転処理を伴う仮想旋回軸: MD21120 \$MC_ORIAX_TURN_TAB_1
ORIVIRT2:	以下を通じて自由に定義可能な、回転処理を伴う仮想旋回軸: MD21130 \$MC_ORIAX_TURN_TAB_2
ORIAXPOS:	回転軸位置を伴う仮想旋回軸

注記

工作機械メーカーはマシンデータを使用して、さまざまなタイプを定義できます。工作機械メーカーの取扱説明書を参照してください。

オイラー角のプログラム ORIEULER、回転処理 Z X' Z"

A2、B2、C2 で向きプログラム ORIEULER を行うときのプログラム指令値は、オイラー角 (° 単位) として解釈されます。

新しい向きのベクトルは、元の向きのベクトルにおける以下 3 つの回転の結果として生じます。

1. 座標軸 Z を中心とする回転軸 A2
2. 新しい座標軸 X' を中心とする回転軸 B2
3. 座標軸 Z" を中心とする回転軸 C2

この場合、C2 の値(新しい Z 軸を中心とする回転)は無意味であり、プログラム指令する必要はありません。

RPY 角のプログラム ORIRPY、回転処理 X Y' Z"

A2、B2、C2 で向きのプログラム RPY を行うときのプログラム指令値は、RPY 角 (° 単位) として解釈されます。

注記

ORIEULER を使用したプログラムとは対照的に、ORIRPY では 3 つの値がすべて配向ベクトルに作用します。

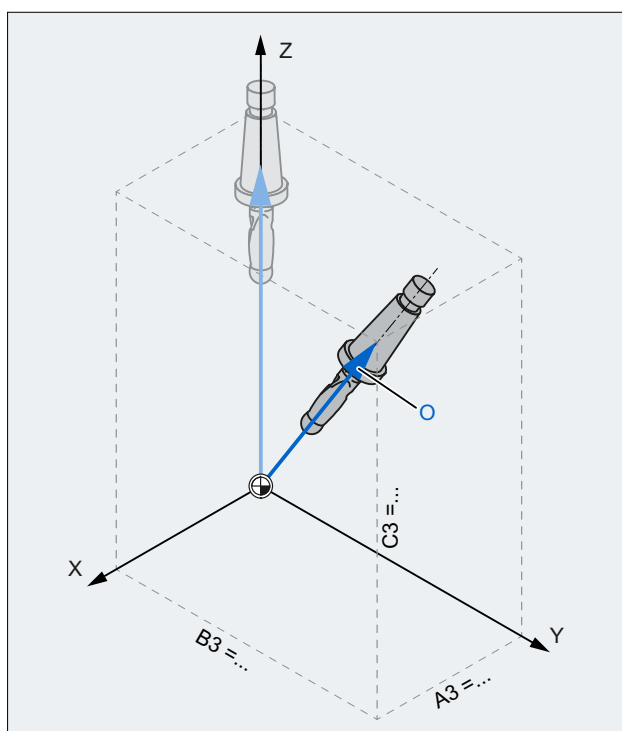
新しい向きのベクトルは、元の向きのベクトルにおける以下 3 つの回転の結果として生じます。

1. 座標軸 X を中心とする回転軸 A2
2. 新しい座標軸 Y' を中心とする回転軸 B2
3. 座標軸 Z' を中心とする回転軸 C2

方向ベクトルのプログラミング

方向ベクトル成分は、A3, B3, C3 でプログラム指令します。ベクトルは工具アダプタの方向を指し、ベクトル長には意味はありません。

プログラム指令していないベクトル成分は、ゼロとに設定されます。



X、Y、Z WCS の座標軸

A3、方向ベクトルのコンポーネント

B3、C3

⊕ 配向ベクトル

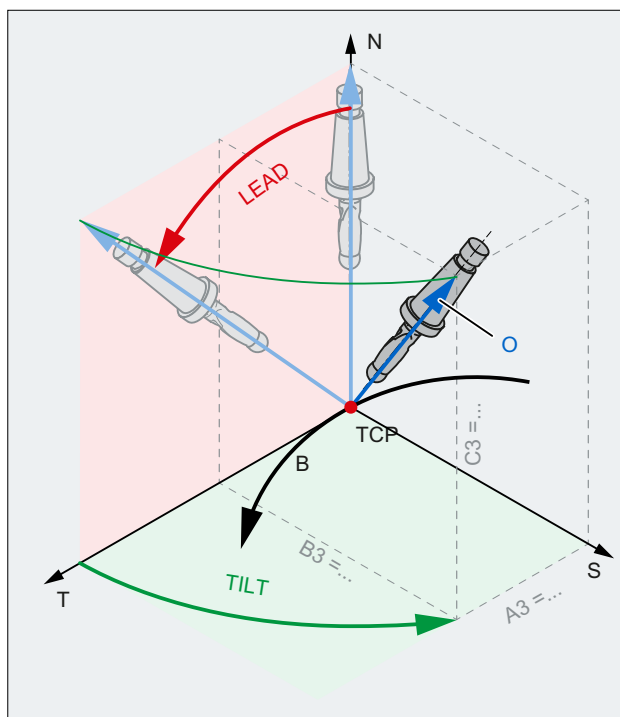
図 4-4 方向ベクトルのプログラミング

4.9 座標変換

LEAD と TILT による工具オリエンテーションのプログラム

得られる工具オリエンテーションは以下により特定されます。

- 軌跡タンジェント
- 面法線ベクトル
A4, B4, C4 のブロック始点、および A5, B5, C5 のブロック終点
- リード角 LEAD
軌跡タンジェントと面法線ベクトルで定義した平面での角度
- ブロック終点での傾き角 TILT
面法線ベクトルに対する軌跡タンジェントに垂直な平面上の角度



- T 軌跡タンジェント
S 軌跡タンジェントと直交
N 面法線
B パス
TCP 工具中心点
O 配向ベクトル

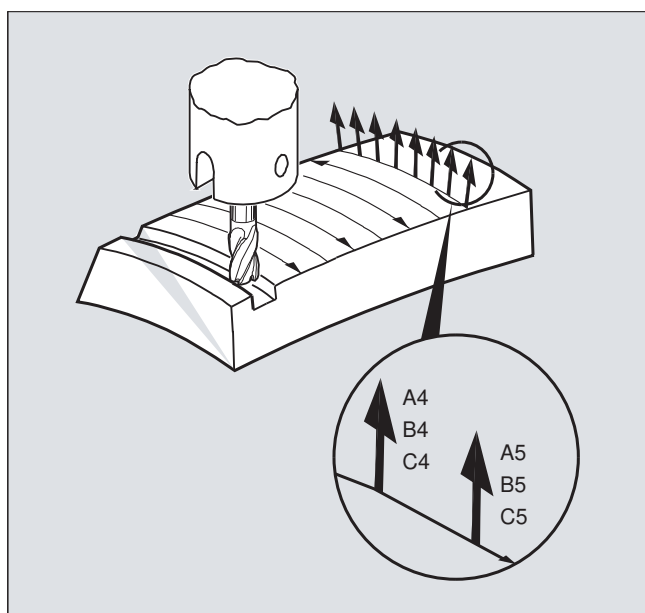
図 4-5 LEAD TILT のプログラム

注記**3次元工具オフセットでの内側コーナの動作**

ブロックが内側コーナで短縮されると、プログラムされた工具オリエンテーションもブロック終点で調整されることになります。

4.9.2.5 正面削り (A4、B4、C4、A5、B5、C5)

正面削りを使用すると、あらゆる種類の曲面を加工できます。



このタイプの3次元加工の場合は、ワーク表面の3次元軌跡を行単位で記述してください。

通常はCAMで実行される計算では、工具の形状と寸法が考慮されます。その後、計算を完了したNCブロックが、ポストプロセッサ経由で制御装置に読み取られます。

軌跡曲率のプログラミング**面の記述**

軌跡曲率は、以下の成分を使用した面法線ベクトルで記述します。

A4, B4, C4 ブロック始点の開始ベクトル

A5, B5, C5 ブロック終点の終了ベクトル

4.9 座標変換

ブロックに開始ベクトルのみが含まれる場合、面法線ベクトルはブロック全体を通じて一定です。ブロックに終了ベクトルのみが含まれる場合は、前のブロックの終了値から、大円弧補間で、プログラム指令の終了値へと補間が実行されます。

開始ベクトルと終了ベクトルをプログラム指令した場合は、大円弧補間で、2方向の間で補間が実行されます。これにより、連続的に滑らかな軌跡を作成できます。

有効な G17～G19 平面にかかわらず、初期設定では、面法線ベクトルは Z 方向を指します。ベクトルの長さには意味はありません。

プログラム指令していないベクトル成分は、ゼロに設定されます。

ORIWKS が有効な場合(「旋回軸の基準(ORIWKS、ORIMKS): (ページ 774)」を参照してください)、面法線ベクトルは動作中のフレームを基準とし、フレーム回転でも回転されます。

機械製造メーカー

面法線ベクトルは、マシンデータで設定した制限値内で軌跡タンジェントに直交している必要があります。直交していない場合は、アラームが発生します。

4.9.2.6 旋回軸の基準(ORIWKS、ORIMKS):

ワーク座標系で向きをプログラム指令するには、以下のいずれかを使用できます。

- オイラー角または RPY 角
- 配向ベクトル

上記を使用した場合は、ORIMKS/ORIWKS を使用して、回転移動の進路を設定できます。

注記

工作機械メーカー

旋回補間のタイプは、次のマシンデータで設定します。

MD21104 \$MC_ORI_IPO_WITH_G_CODE

= FALSE:G 命令 ORIWKS と ORIMKS で基準が設けられます。

= TRUE:51 番目のグループの G 命令(ORIAXES、ORIVECT、ORIPLANE など)で基準が設定されます。

構文

ORIMKS=...

ORIWKS=...

意味

ORIMKS:	機械座標系の回転
ORIWKS:	ワーク座標系の回転

注記

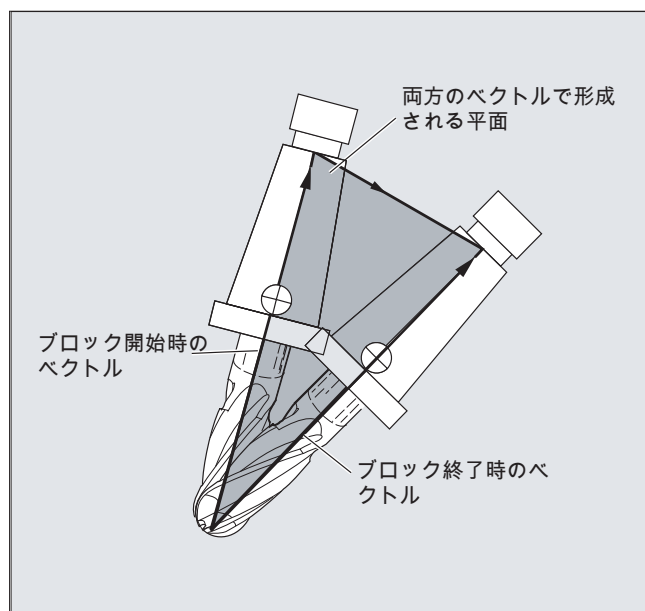
基本設定は ORIWKS です。どの機械で実行するかが明確でない 5 軸プログラムの場合は、常に ORIWKS を選択してください。機械が実際にどのように移動を実行するかは、機械のキネマティックスにより異なります。

ORIMKS を使用すると、(機器類との衝突を避けるためなどに)実際の機械の移動をプログラム指令できます。

詳細情報

ORIMKS を使用した場合、工具がおこなう移動は機械のキネマティックスにより**異なります**。空間の固定点で工具先端の向きが変わった場合は、各回転軸位置の間で直線補間がおこなわれます。

ORIWKS を使用した場合、工具がおこなう移動は機械のキネマティックスには**影響されません**。固定された工具先端の向きが変わると、開始ベクトルと終了ベクトルで設定された平面で工具が移動します。



4.9 座標変換

特異点

注記

ORIWKS

5 軸機械が特異点設定領域で旋回移動をおこなうには、機械軸が大きく移動する必要があります。(たとえば、C を回転軸、A を旋回軸として回転するスィベルヘッドでは、 $A=0$ のすべての位置は特異点です。)

工作機械メーカー

機械軸への過負荷を避けるために、特異点位置近傍の工具軌跡速度は速度制御により大幅に減速されます。

使用するマシンデータ

\$MC_TRAFO5_NON_POLE_LIMIT

\$MC_TRAFO5_POLE_LIMIT

上記のマシンデータで座標変換のパラメータを設定すると、極近傍の旋回移動で極を通過するため、高速加工をおこなうことができます。

特異点は MD \$MC_TRAFO5_POLE_LIMIT でのみ処理できます。

詳細情報: 『機能マニュアル』、「座標変換」

4.9.2.7 旋回軸のプログラミング(ORIAXES、ORIVECT、ORIEULER、ORIRPY、ORIRPY2、ORIVIRT1、ORIVIRT2)

空間の工具の向きを記述する「旋回軸」機能は、回転軸のオフセットをプログラム指令して実行します。工具自身の回転もおこなうと、さらに 3 次元の自由度が提供されます。この場合は、3 番目の回転軸により空間で工具の向きが決まります。これには 6 軸座標変換が必要です。工具自身の回転は、回転ベクトルの補間タイプに従って、THETA の回転角度を使用して定義します(「工具オリエンテーションの回転(ORIROTA、ORIROTR、ORIROTT、ORIROTC、THETA) (ページ 786)」を参照してください)。

軸識別子 A2、B2、および C2 を使用して旋回軸をプログラム指令します。

構文

```
N...ORIAXES/ORIVECT ; 直線補間または大円弧補間
N...G1 X Y Z A B C
```

```
N...ORIPLANE ; 平面の旋回補間
```

N...ORIEULER/ORIRPY/ORIRPY2 : 旋回角度 オイラー角/RPY 角
 N...G1 X Y Z A2= B2= C2= ; 仮想軸の角度のプログラミング

N...ORIVIRT1/ORIVIRT2 ; 仮想旋回軸の定義 1/2
 N...G1 X Y Z A3= B3= C3= ; 方向ベクトルのプログラミング

注記

旋回軸の他の回転軸オフセットをプログラム指令して、空間の円錐面に沿って向きを変更できます。「円錐面に沿った向きのプログラミング(ORIPLANE、ORICONCW、ORICONCCW、ORICONTO、ORICONIO) (ページ 779)」を参照してください。

意味

ORIAxes:	機械軸または旋回軸の直線補間
ORIVect:	大円弧補間(ORIPLANE と同じ)
ORIMKS: ORIWKs:	機械座標系の回転 ワーク座標系の回転 詳細については、「旋回軸の基準(ORIWKs、ORIMKS): (ページ 774)」を参照してください。
A= B= C=:	機械軸位置のプログラミング
ORIEULER:	オイラー角による向きのプログラミング
ORIRPY:	RPY 角による向きのプログラミング 回転の順序は XYZ で、 • A2 は X を中心とする回転の角度 • B2 は Y を中心とする回転の角度 • C2 は Z を中心とする回転の角度
ORIRPY2:	RPY 角による向きのプログラミング 回転の順序は ZYX で、 • A2 は Z を中心とする回転の角度 • B2 は Y を中心とする回転の角度 • C2 は X を中心とする回転の角度
A2= B2= C2=:	仮想軸の角度のプログラミング

4.9 座標変換

ORIVIRT1/ORIVIRT2:	仮想回転軸を使用した向きのプログラミング 定義 1: MD21120 \$MC_ORIAX_TURN_TAB_1 による定義 定義 2: MD21130 \$MC_ORIAX_TURN_TAB_2 による定義
A3= B3= C3=:	方向軸の方向ベクトルのプログラミング

詳細情報

工作機械メーカー

MD21102 \$MC_ORI_DEF_WITH_G_CODE は、プログラム指令角度 A2、B2、C2 の定義方法を指定します。

定義は、MD21100 \$MC_ORIENTATION_IS_EULER (標準)、または G グループ 50 (ORIEULER、ORIRPY、ORIVIRT1、ORIVIRT2)に従います。

MD21104 \$MC_ORI_IPO_WITH_G_CODE は、有効な補間モードタイプを定義します。
ORIWS/ORIMKS と ORIAxes/ORIVECT のどちらにするかを定義します。

JOG モード

この動作モードの回転角度の補間は常に直線補間です。移動キーによる連続移動とステップ移動時は、1 つの回転軸のみ移動できます。複数の回転軸を、手動パルス発生器を使用して同時に移動できます。

回転軸を手動で移動する場合は、チャンネル別の送り速度オーバーライドスイッチ、または早送りオーバーライドスイッチが早送りオーバーライドで機能します。

次のマシンデータを使用すると、個別の速度を設定できます。

MD21160 \$MC_JOG_VELO_RAPID_GEO

MD21165 \$MC_JOG_VELO_GEO

MD21150 \$MC_JOG_VELO_RAPID_ORI

MD21155 \$MC_JOG_VELO_ORI

注記

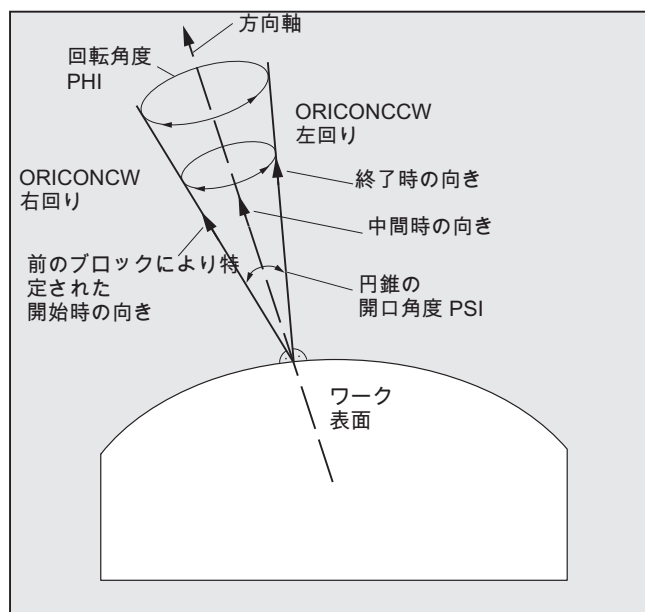
「マテハン装置用座標変換」を備えた SINUMERIK 840D sl

「手動の直交移動」機能を使用して、JOG モードでの複数のジオメトリ軸の平行移動を、基準系 MCS、WCS、および TCS で、それぞれに個別に設定できます。

詳細情報: 『機能マニュアル』、「座標変換」

4.9.2.8 円錐面に沿った向きのプログラミング(ORIPLANE、ORICONCW、ORICONCCW、ORICONTO、ORICONIO)

拡張旋回をおこなうと、空間の円錐面に沿って向きの変更を実行できます。ORICONxx モーダル命令を使用して、配向ベクトルを円錐面で補間します。旋回の終了は、平面上の補間のための ORIPLANE でプログラム指令できます。旋回の開始は通常、前のブロックで定義します。



4.9 座標変換

プログラミング

旋回の終了は、A2, B2, C2 によるオイラー角または RPY 角での角度のプログラミングの指定、または A, B, C による回転軸位置のプログラミングにより定義します。円錐面に沿った回転軸には、さらに以下の詳細なプログラミングが必要です。

- A6、B6、C6 による円錐の回転軸のベクトル
- 識別子 NUT による開口角度 PSI
- A7, B7, C7 による円錐面の中間旋回

注記

円錐の回転軸のための方向ベクトル A6, B6, C6 のプログラミング

旋回の終了は必ずプログラミングする必要はありません。旋回の終了を指定していない場合は、360°の円錐面全体が補間されます。

NUT=角度による、円錐の開口角度のプログラミング

旋回の終了を指定してください。

この方法では、360°の円錐面全体は補間できません。

円錐面の中間旋回 A7, B7, C7 のプログラミング

旋回の終了を指定してください。向きの変更と回転方向を、旋回の開始、旋回の終了、および旋回の中間という 3 つのベクトルで一義的に定義します。この 3 つはそれぞれ、異なるベクトルです。プログラム指令した旋回の中間が、旋回の開始または終了に平行である場合は、直線の旋回大円弧補間が、開始ベクトルと終了ベクトルで定義された平面で実行されます。

円錐面での拡張旋回補間

N...ORICONCW または ORICONCCW

N...A6= B6= C6= A3= B3= C3=

または

N...ORICONTO

N...G1 X Y Z A6= B6= C6=

または

N...ORICONIO

N...G1 X Y Z A7= B7= C7=

N...PO[PHI]=(a2, a3, a4, a5)

N...PO[PSI]=(b2, b3, b4, b5)

円錐面の補間

円錐の右回り/左回り方向の方向ベクトルと旋回の終了、または

接線方向の遷移と
旋回の終了の指定、

または

旋回の終了と

円錐面の中間旋回の指定で

回転角度の多項式と

開口角度の多項式を使用

パラメータ

ORIPLANE:	平面での補間(大円弧補間)
ORICONCW:	円錐面での右回り方向の補間
ORICONCCW:	円錐面での左回り方向への補間
ORICONTO:	接線方向の遷移による円錐面の補間
A6= B6= C6=:	円錐の回転軸のプログラミング(正規化ベクトル)
NUT=角度:	円錐の開口角度(° 単位)
NUT=+179:	180°以下の移動角度
NUT=-181:	180°以上の移動角度
ORICONIO:	円錐面での補間
A7= B7= C7=:	中間旋回(正規化ベクトルとしてのプログラミング)
PHI:	円錐の方向軸を中心とした向きの回転角度
PSI:	円錐の開口角度
使用可能な多項式 PO[PHI]=(a2, a3, a4, a5) PO[PSI]=(b2, b3, b4, b5)	それぞれの角度とは別に、 5 次までの多項式もプログラム指令できます

例:さまざまな向きの変更

プログラムコード	コメント
...	
N10 G1 X0 Y0 F5000	
N20 TRAORI(1)	; 方向座標変換をオンします
N30 ORIVECT	; 工具オリエンテーションをベクトルとして補間します
...	; 平面の工具オリエンテーション
N40 ORIPLANE	; 大円弧補間を選択します
N50 A3=0 B3=0 C3=1	
N60 A3=0 B3=1 C3=1	; Y/Z 平面での向きは 45°回転し、ブロック終点で目的の向き (0, 1/√2, 1/√2) に到達します。
...	
N70 ORICONCW	; 円錐面の向きのプログラミングです。
N80 A6=0 B6=0 C6=1 A3=0 B3=0 C3=1	配向ベクトルは、方向 (0, 0, 1) の円錐面で目標の向き (1/√2, 0, 1/√2) まで右回りに補間されます。回転角度は 270°です。
N90 A6=0 B6=0 C6=1	; 工具オリエンテーションは、同じ円錐面で完全に 1 回転します。

詳細情報

空間の任意の位置での円錐面に沿った向きの変更を記述する場合は、工具オリエンテーションの回転の中心となるベクトルが必要です。旋回の開始と終了も指定してください。前のブロックから旋回の開始を得て、旋回の終了はプログラム指令、または他の条件を使用して定義してください。

ORIVECT に対応する ORIPLANE 平面のプログラミング

角度多項式による大半径円弧補間のプログラミングは、輪郭の直線補間と多項式補間に対応します。工具オリエンテーションは、旋回の開始と終了で定義した平面で補間されます。さらに、多項式をプログラム指令する場合は、配向ベクトルを平面から傾けることもできます。

平面の円弧のプログラミング G2/G3、CIP、および CT

拡張旋回は、平面上の円弧の補間に対応します。G2/G3 などの中心点データまたは半径データのある円弧、中間点 CIP による円弧、および接線方向の円弧 CT の、プログラミングオプションについて詳しくは、章「円弧補間 (ページ 212)」を参照してください。

旋回のプログラミング**円錐面上の配向ベクトルの補間 ORICONxx**

円錐面上の向きを補間するために、G グループ 51 から、以下の 4 つの補間タイプを選択できます。

1. 旋回の終了と円錐の方向、または開口角度の指定による円錐面の右回り方向の補間 ORICONCW。方向ベクトルは、識別子 A6、B6、C6、および識別子 NUT=0~180°の範囲の円錐の開口角度でプログラム指令します。
2. 旋回の終了と円錐の方向、または開口角度の指定による円錐面の左回り方向の補間 ORICONCCW。方向ベクトルは、識別子 A6、B6、C6、および識別子 NUT=0~180°の範囲の円錐の開口角度でプログラム指令します。
3. 旋回の終了、および中間旋回の指定による円錐面の補間 ORICONIO。中間旋回は識別子 A7、B7、C7 でプログラム指令します。
4. 接線方向の遷移、および旋回の終了の指定による円錐面の補間 ORICONTO。方向ベクトルは、識別子 A6、B6、C6 でプログラム指令します。

4.9.2.9 2 つの接点の向きの指定(ORICURVE、PO[XH]=、PO[YH]=、PO[ZH]=)**空間の、もう 1 つの曲線を使用した向き変更 ORICURVE のプログラミング**

空間で、曲線に沿って工具先端を移動する以外に、向きの変更をプログラム指令するもう 1 つの方法は、ORICURVE を使って、工具の 2 番目の接点の移動をプログラム指令します。

この方法を使うと、工具ベクトル自身をプログラム指令したときと同様に、工具オリエンテーションの変更を一義的に定義できます。

工作機械メーカー

軸識別子に関する工作機械メーカーの注釈を参照してください。マシンデータで軸識別子を設定すると、工具の旋回軌跡をもう 1 つプログラム指令できます。

プログラミング

このタイプの補間を使用すると、空間の 2 つの曲線を表わす点(G1 によって)または多項式(POLY を使用)をプログラム指令できます。円弧とインボリュート曲線はプログラム指令できません。BSPLINE スプライン補間と「短いスプラインブロックの結合」機能も有効です。

「短いスプラインブロックの結合」機能についての追加情報は、『機能マニュアル』、「基本機能」を参照してください。

他のタイプのスプライン(ASPLINE と CSPLINE)および COMP...を使用したコンプレッサの有効化は、許容されていません。

座標の旋回多項式をプログラミングするときに、工具の 2 つの接点の移動を 5 次まで事前設定できます。

空間の追加曲線による拡張旋回補間と座標の多項式

N...ORICURVE ;工具の 2 番目の接点の移動の指定と当該座標の追加多項式
 N...PO[XH]=(xe, x2, x3, x4, x5)
 N...PO[YH]=(ye, y2, y3, y4, y5)
 N...PO[ZH]=(ze, z2, z3, z4, z5)

パラメータ

ORICURVE	工具の 2 つの接点間の移動を指定した旋回補間。
XH YH ZH	空間の曲線としての工具追加輪郭の 2 番目の接点の座標識別子
PO[XH]=(xe, x2, x3, x4, x5) PO[YH]=(ye, y2, y3, y4, y5) PO[ZH]=(ze, z2, z3, z4, z5)	使用可能な多項式 当該のそれぞれの終点を使用する以外に、空間の曲線を多項式でプログラム指令することもできます。
xe, ye, ze	空間の曲線の終点
xi, yi, zi	5 次までの多項式の係数

注記**2 番目の旋回軌跡をプログラムするための識別子 XH YH ZH**

識別子は、他の識別子や直線軸と競合しないように選択してください。

XYZ の各軸

、および以下のような回転軸

A2 B2 C2 オイラー角または RPY 角

A3 B3 C3 方向ベクトル

A4 B4 C4 または A5 B5 C5 面法線ベクトル

A6 B6 C6 回転ベクトルまたは A7 B7 C7 中間点座標

またはその他の補間パラメータ

4.9.3 旋回多項式(PO[角度]、PO[座標])

G グループ 1 による、現在有効な多項式補間にかかわらず、2 つのタイプの 5 次までの旋回多項式をプログラムして、3 軸から 5 軸の座標変換をおこなうことができます。

1. 角度の多項式:

旋回の開始と終了で定義した平面に対するリード角 LEAD と傾斜角 TILT です。

2. 座標の多項式: 工具の基準点の工具オリエンテーションのための空間の 2 番目の曲線の XH、YH、ZH です。

6 軸座標変換を使用すると、回転ベクトル THT の回転を 5 次までの多項式でプログラム指令して、工具オリエンテーションだけでなく、工具自身の回転も実行できます。

構文

タイプ 1 **角度**の旋回多項式

N... PO[PHI]=(a2, a3, a4, a5)

3 軸～5 軸座標変換

N... PO[PSI]=(b2, b3, b4, b5)

タイプ 2 座標の旋回多項式

N... PO[XH]=(xe, x2, x3, x4, x5) 工具オリエンテーションの 2 番目の旋回軌跡の座標識別子
 N... PO[YH]=(ye, y2, y3, y4, y5)
 N... PO[ZH]=(ze, z2, z3, z4, z5)

上記のいずれの場合も、6 軸座標変換を実行すると、以下のような多項式をプログラム指令して、**回転**実行することもできます。

N... PO[THT]=(c2, c3, c4, c5) 軌跡に対する回転の補間
 または
 N... PO[THT]=(d2, d3, d4, d5) 配向ベクトルの向きの変更に対するアブソリュート補間、相対補間、および接線方向の補間

of the orientation vector.これが可能なのは、THETA の回転角度によるプログラム指令と補間が可能なオフセットを含む回転ベクトルを座標変換がサポートしている場合のみです。

意味

PO[PHI]	旋回の開始と終了の間にある平面の角度
PO[PSI]	旋回の開始と終了の間の平面からの、向きの傾斜角
PO[THT]	THETA でプログラムしたグループ 54 の G 命令のうちいずれかの回転ベクトルの回転により生じた回転角度
PHI	リード角 LEAD
psi	傾斜角 TILT
THETA	Z の工具方向を中心とする回転
PO[XH]	工具の基準点の X 座標
PO[YH]	工具の基準点の Y 座標
PO[ZH]	工具の基準点の Z 座標

詳細情報

次の場合は、旋回多項式をプログラム指令できません。

- ASPLINE、BSPLINE、CSPLINE スプライン補間が有効である。
タイプ 1(角度の旋回多項式)は、スプライン補間を除く、すべてのタイプの補間に使用できます。つまり、早送り G00 または送り速度 G01 による直線補間、POLY を使用した多項式補間、および円弧/インボリュート補間 G02、G03、CIP、CT、INVCW、および INCCCW に使用できます。
ただし、タイプ 2(座標の旋回多項式)は、早送り G00 または送り速度 G01 による直線補間、または POLY を使用した多項式補間が有効な場合にかぎり、使用できます。
- ORIAXES の軸補間を使用して向きを補間する。この場合は、旋回軸 A と B に対して多項式の PO[A]と PO[B]を使用して直接プログラム指令できます。

ORIVECT、ORIPLANE、および ORICONxx を使用したタイプ 1 の旋回多項式

タイプ 1 の旋回多項式は、ORIVECT、ORIPLANE、および ORICONxx を使用して大半円弧補間、および円錐面の補間に使用できます。

ORICURVE を使用したタイプ 2 の旋回多項式

空間の追加曲線による補間の ORICURVE が有効な場合は、配向ベクトルの直交成分が補間され、使用できるのはタイプ 2 の旋回多項式のみとなります。

4.9.4 工具オリエンテーションの回転(ORIOTA、ORIOTR、ORIOTT、ORIOTC、THETA)

可動式工具を備えた機械タイプの工具の向きも変更する場合は、各ブロックに旋回の終了をプログラム指令してください。機械のキネマティクスに応じて、旋回軸の旋回方向または配向ベクトル THETA の回転方向をプログラム指令できます。以下の回転ベクトルに対して、さまざまな補間タイプをプログラム指令できます。

- ORIOTA:アブソリュート回転方向への回転角度です。
- ORIOTR:旋回の開始と終了の間に形成される平面に対する回転角度です。
- ORIOTT:配向ベクトルの変更に対する回転角度です。
- ORIOTC:軌跡タンジェントに対する接線方向の回転角度です。

構文

補間タイプ **ORIROTA** が有効な場合にのみ、回転角度または回転ベクトルを、4 つのモードすべてで次のようにプログラム指令できます。

1. 直接、回転軸位置 A, B, C として
2. A2, B2, C2 によるオイラー角(° 単位)
3. A2, B2, C2 による RPY 角(° 単位)
4. A3, B3, C3 による方向ベクトル(THETA=<値>を使用した回転角度)

ORIROTR または **ORIROTT** が有効な場合は、回転角度を THETA でのみ、直接プログラム指令できます。

向きを変更せずに、別のブロックで回転をプログラム指令することもできます。この場合、**ORIROTR** と **ORIROTT** は無効です。この場合は、回転角度は常に、アブソリュート方向(**ORIROTA**)を基準として解釈されます。

```

N...ORIROTA                回転ベクトルの補間を定義します
N...ORIROTR
N...ORIROTT
N...ORIROTC
N...A3= B3= C3= THETA=<値>  配向ベクトルの回転を定義します
N...PO[THT]=(d2, d3, d4, d5)  回転角度を 5 次多項式で補間します

```

意味

ORIROTA:	アブソリュート回転方向への回転角度
ORIROTR:	旋回の開始と終了間の平面に対する回転角度
ORIROTT:	向きの変更に対する接線方向の回転ベクトルの回転角度です。
ORIROTC:	軌跡タンジェントに対する接線方向の回転ベクトルの回転角度です。
THETA:	配向ベクトルの回転
THETA=<値>:	ブロック終点へ到達するまでの回転角度(° 単位)
THETA=Θ _e :	回転ベクトルの終了角度が Θ _e の回転角度
THETA=AC (...):	アブソリュート指令へのノンモーダル切り替え
THETA=AC (...):	インクレメンタル指令へのノンモーダル切り替え

4.9 座標変換

Θe:	G90 によるアブソリュート回転ベクトル、かつ G91 (インクレメンタル指令)による相対回転ベクトルの両方の終了角度が有効です
PO[THT]=(...):	回転角度の多項式

例:向きの回転

プログラムコード	コメント
N10 TRAORI	; 方向座標変換を起動します。
N20 G1 X0 Y0 Z0 F5000	; z 軸の回転
N30 A3=0 B3=0 C3=1 THETA=0	; z 方向に回転角度 0
N40 A3=1 B3=0 C3=0 THETA=90	; x 方向に約 90°回転
N50 A3=0 B3=1 C3=0 PO[THT]=(180,90)	; 向き
N60 A3=0 B3=1 C3=0 THETA=IC(-90)	; y 方向に約 180°回転
N70 ORIROTT	; そのまま一定で、90°回転
N80 A3=1 B3=0 C3=0 THETA=30	; 向きの変更に対する回転角度
	; x/y 平面に対して 30°の角度の回転ベクトル

ブロック N40 の補間時には、0°(初期値)から 90°(最終値)までの回転角度を直線的に補間します。ブロック N50 では、回転角度が放物線 $\theta(u) = +90u^2$ に従って、90°から 180°まで変化します。N60 では、向きを変更せずに、回転を実行することもできます。

N80 を使用すると、工具オリエンテーションは Y 方向から X 方向に向かって回転します。向きの変更は X/Y 平面でおこなわれ、回転ベクトルはこの平面に対して 30°の角度を表わします。

詳細情報

ORIOTA

回転角度 THETA は、空間のアブソリュート方向を基準として補間されます。回転の基本方向はマシンデータで定義します。

ORIOTR

回転角度 THETA は、旋回の開始と終了で定義した平面に対して補間されます。

ORIOTT

回転角度 THETA は、向きの変更に対して補間されます。THETA=0 の場合は、向きの変更に対して接線方向に回転ベクトルが補間されます。また、向きの「傾斜角 PSI」に対して 1 つ以上の多項式をプログラム指令している場合を除き、ORIOTR と同じです。この場合の結果は、向きは変更されますが、平面では実行されません。その後、さらに回転角度

THETA を使用して、回転ベクトルを補間すると、常に向きの変更を示す特定の値を作成します。

ORIROTC

回転ベクトルは、オフセットを使用して軌跡タンジェントに対して補間されます。このオフセットは THETA 角度でプログラム指令可能です。5 次以下の多項式 $PO[THT] = (c2, c3, c4, c5)$ をオフセット角に対してプログラム指令することもできます。

4.9.5 軌跡に対する向き

4.9.5.1 軌跡に対する向きのタイプ

この拡張機能を使用すると、相対的な向きがブロック終点だけでなく、軌道全体にわたって成立します。前のブロックで成立した向きを、大円弧補間によって、プログラム指令した旋回の終了へ渡します。軌跡に対して求める向きをプログラミングする方法は、基本的に次の 2 つです。

1. 工具回転と同様に、工具オリエンテーションは、ORIPATH、ORPATHTS を使用して軌跡に対して補間されます。
2. 配向ベクトルは、通常の方法でプログラム指令し、補間します。配向ベクトルの回転を、ORIROTC を使用して軌跡タンジェントに対して開始します。

構文

旋回補間のタイプと工具の回転は、以下を使用してプログラム指令します。

N...ORIPATH	軌跡に対する向き
N...ORIPATHS	旋回処理のスムージングによる、軌跡に対する向き
N...ORIROTC	軌跡に対する回転ベクトルの補間

軌道のコーナで発生する向きの不連続変化は、ORIPATHS でスムージングできます。戻し移動の方向と軌跡長は、A8=X、B8=Y、および C8=Z 成分を使用してベクトルでプログラム指令します。

4.9 座標変換

ORIPATH/ORIPATHS を使用すると、軌道全体に対して、次の 3 種類の角度を使用して軌跡タンジェントに対してさまざまな基準をプログラム指令できます。

- LEAD = 軌跡と面に対するリード角を指定します
- TILT = 軌跡と面に対する傾斜角を指定します
- THETA = 軌道全体に対する

回転角度ですまた THETA 回転角度の他に、PO[THT] = (...) で 5 次以下の多項式をプログラム指令できます。

注記

工作機械メーカー

工作機械メーカーの取扱説明書を参照してください。軌跡に対する向きに、その他の設定をおこなうには、マシンデータとセッティングデータを使用して設定できます。

詳細情報:『機能マニュアル』、「座標変換」

意味

角度 LEAD 角と TILT 角の補間には、マシンデータを使用して、以下のさまざまな設定が可能です。

- LEAD と TILT でプログラム指令した工具オリエンテーションの基準は、ブロック全体にわたって保持されます。
- リード角 LEAD:接線と法線ベクトルに垂直な方向を中心とする回転 TILT:法線ベクトルを中心とする向きの回転
- リード角 LEAD:接線と法線ベクトルに垂直な方向を中心とする回転 傾斜角 TILT:軌跡タンジェント方向の向きの回転。
- 回転角度 THETA:6 軸座標変換で旋回軸として動作する、3 番目の付加回転軸による工具自身の回転

注記

軌跡に対する向きの OSC、OSS、OSSE、OSD、および OST との組み合わせの禁止

軌跡に対する旋回補間、つまり ORIPATH または ORIPATHS、および ORIOTC は、グループ 34G 命令による旋回処理のスムージングと組み合わせてプログラムすることはできません。このため、OSOF を有効にしてください。

4.9.5.2 軌跡に対する工具オリエンテーションの回転(ORIPATH、ORIPATHS、回転角度)

6 軸座標変換を使用すると、3 番目の回転軸で工具自身を回転させ、空間内の求める向きに工具を配置できます。ORIPATH または ORIPATHS を使用して、軌跡に対して工具オリエンテーションを回転させると、THETA を使用した回転角度で、さらに回転をプログラム指令できます。または、工具方向と垂直な平面にあるベクトルを使って、LEAD 角と TILT 角をプログラム指令できます。

機械製造メーカ

工作機械メーカの説明書を参照してください。LEAD 角と TILT 角の補間は、マシンデータを使用して、別々の設定にすることができます。

構文

工具オリエンテーションと工具の回転

軌跡に対する工具オリエンテーションのタイプは、ORIPATH または ORIPATHS で有効にします。

N...ORIPATH	軌跡に対する旋回のタイプを有効にします
N...ORIPATHS	旋回処理のスムージングにより、軌跡に対する旋回のタイプを有効にします
下記の 3 つの回転角度を有効にします:	
N...LEAD=	面法線ベクトルに対する、プログラム指令の向きの角度
N...TILT=	面法線ベクトルに対する軌跡タンジェントに直交する平面上の、プログラム指令の向きの角度
N...THETA=	3 番目の回転軸の工具方向の向きの変更に対する回転角度

ブロック終点での角度の値は、LEAD=値、TILT=値、または THETA=値でプログラム指令します。定数の角度にの他に、3 つの角度のすべてに対して 5 次までの多項式をプログラム指令できます。

N...PO[PHI]=(a2, a3, a4, a5)	リード角 LEAD の多項式
N...PO[PSI]=(b2, b3, b4, b5)	傾斜角 TILT の多項式
N...PO[THT]=(d2, d3, d4, d5)	回転角度 THETA の多項式

4.9 座標変換

より高次の多項式係数がゼロのときは、省略してプログラミングできます。
 例:PO[PHI]=a2 の結果は、LEAD 角の放物線となります。

意味

軌跡に対する工具オリエンテーション

ORIPATH :	軌跡に対する工具オリエンテーション
ORIPATH S:	軌跡に対する工具オリエンテーション、旋回処理の不連続変化をスムーズ グします
LEAD:	軌跡タンジェントと面法線ベクトルで定義された平面の、面法線ベクトルに対 する角度
TILT:	Z 方向の向きの回転、または軌跡タンジェントを中心とした回転
THETA:	Z へ向かう工具方向を中心とする回転
PO[PHI] :	LEAD 角の旋回多項式
PO[PSI] :	TILT 角の旋回多項式
PO[THT] :	THETA の回転角度の旋回多項式

注記

回転角度 THETA

旋回軸が動作する 3 番目の回転軸で、工具自身を回転させるには、6 軸座標変換が必要です。

4.9.5.3 軌跡に対する工具回転の補間(ORIOTC、THETA)

回転ベクトルによる補間

ORIOTC でプログラム指令した、軌跡タンジェントに対する工具回転の回転ベクトルは、
 回転角度 THETA でプログラム指令できるオフセットを使用して補間することもできます。
 したがって、PO[THT]を使用して、5 次以下の多項式をオフセット角に対してプログラム指
 令できます。

構文

N...ORIOTC 軌跡タンジェントに対する工具の回転を開始します

N...A3= B3= C3= THETA=値 配向ベクトルの回転を定義します

N...A3= B3= C3= PO[THT]=(c2, c3, c4, c5) 5 次以下の多項式でオフセット角度を補間します

向きを変更せずに、個別のブロックで回転をプログラム指令することもできます。

意味

6 軸座標変換の軌跡に対する工具回転の補間

ORIOTC:	軌跡タンジェントに対する接線方向の回転ベクトルを開始します
THETA=値:	ブロック終点までの到達回転角度(° 単位)
THETA=Θ _e :	回転ベクトルの終了角度が Θ _e の回転角度
THETA=AC (...):	アブソリュート指令へのノンモーダル切り替え
THETA=IC (...):	インクレメンタル指令へのノンモーダル切り替え
PO[THT]=(c2, c3, c4, c5):	5 次多項式でオフセット角度を補間します

注記

回転ベクトルの補間 ORIOTC

軌跡タンジェントに対する工具回転を、工具オリエンテーションとは逆方向に開始できるのは、6 軸座標変換の場合のみです。

ORIOTC が動作中の場合

回転ベクトル ORIOTA はプログラム指令できません。プログラム指令すると、アラーム 14128 「Absolute programming of tool rotation with active ORIOTC」 (ORIOTC が動作中に工具回転のアブソリュートプログラム指令がされました)が発生します。

3 軸～5 軸座標変換の工具の旋回方向

工具の旋回方向は、3 軸～5 軸座標変換の場合と同様に、オイラー角、RPY 角、または方向ベクトルでプログラム指令できます。空間の工具の向きの変更は、大円弧補間 ORIVECT、

4.9 座標変換

旋回軸の直線補間 **ORIAxes**、円錐面上のすべての補間 **ORICONxx**、そして、さらに工具の 2 つの接点による空間の曲線の補間 **ORICURVE** をプログラミングして実行することもできます。

G.....:	回転軸移動の詳細
X, Y, Z:	直線軸の詳細
ORIAxes:	機械軸または旋回軸の直線補間
ORIVECT:	大円弧補間(ORIPLANE と同じ)
ORIMKS:	機械座標系での回転
ORIWKS:	ワーク座標系での回転 説明については、「工具オリエンテーションの回転」の章を参照してください。
A= B= C=:	機械軸位置のプログラミング
ORIEULER:	オイラー角による向きのプログラミング
ORIRPY:	RPY 角による向きのプログラミング
A2= B2= C2=:	仮想軸の角度のプログラミング
ORIVIRT1:	仮想旋回軸を使用した向きのプログラミング
ORIVIRT2:	(定義 1)、マシンデータ \$MC_ORIAX_TURN_TAB_1 による定義 (定義 2)、マシンデータ \$MC_ORIAX_TURN_TAB_2 による定義
A3= B3= C3=:	方向軸の方向ベクトルのプログラミング
ORIPLANE:	平面での補間(大円弧補間)
ORICONCW:	円錐面での右回り方向の補間
ORICONCCW:	円錐面での左回り方向への補間
ORICONTO:	接線方向の遷移による円錐面の補間
A6= B6= C6=:	円錐の回転軸のプログラミング(正規化ベクトル)
NUT=角度	円錐の開口角度(° 単位)
NUT=+179	180°以下の移動角度
NUT=-181	180°以上の移動角度
ORICONIO:	円錐面での補間

A7= B7= C7=:	中間旋回(正規化ベクトルとしてのプログラミング)
ORICURVE XH YH ZH (例: 多項式 PO[XH]=(xe, x2, x3, x4, x5)などによ る)	工具の2つの接点間の移動を指定した旋回補間。終点だけでなく、追加の曲線多項式もプログラム指令できます。

注記

ORIXES が有効で、工具オリエンテーションを旋回軸で補間した場合は、回転角度はブロック終点で、軌跡に対して開始されるだけです。

4.9.5.4 旋回処理のスミージング(ORIPATHS A8=, B8=, C8=)

向きの変更を輪郭上で一定の加速度で実行すると、軌跡移動が、特に輪郭のコーナ部で不必要に中断される場合があります。結果として生じる旋回処理の不連続変化は、別の中間ブロックを挿入してスミージングできます。再旋回時に ORIPATHS が有効な場合は、向きの変更が一定の加速でおこなわれます。この段階で、工具を後退できます。

工作機械メーカー

この機能を起動するための、事前設定されたマシンデータとセッティングデータについては、工作機械メーカーの注意事項を参照してください。

次のマシンデータを使用すると、後退ベクトルの解釈方法を設定できます。

1. TCS では、工具方向で Z 座標を定義します。
2. WCS では、有効平面で Z 座標を定義します。

「軌跡に対する向き」機能についての**追加情報**は、『機能マニュアル』、「座標変換」を参照してください。

構文

輪郭のコーナでは、軌跡に対する工具オリエンテーション全体を一定に保つために、さらに詳細なプログラミングが必要です。この移動の方向と軌跡長は、A8=X、B8=Y、C8=Z 成分を使用してベクトルでプログラム指令します。

```
N...ORIPATHS A8=X B8=Y C8=Z
```

4.9 座標変換

意味

ORIPATHS:	軌跡に対する工具オリエンテーション; 旋回処理の不連続変化がスムージングされます
A8= B8= C8=:	方向と軌跡長のベクトル成分
X, Y, Z:	工具方向への後退移動

注記

方向ベクトル A8、B8、C8 のプログラミング

このベクトルの長さが正確にゼロの場合は、後退移動は実行されません。

ORIPATHS

軌跡に対する工具オリエンテーションは、ORIPATHS で有効にします。有効にしない場合は、直線大円弧補間で、向きが旋回の開始から旋回の終了に渡されます。

4.9.6 旋回の圧縮(COMPON、COMPCURV、COMPCAD、COMPOF、COMPSURF)

方向座標変換(TRAORI)が有効で、工具オリエンテーションがプログラム指令されている(タイプを問わず)NC プログラムは、指定制限内であれば、圧縮できます。

プログラミング

工具オリエンテーション

方向座標変換(TRAORI)が有効な場合は、5 軸の機械では、工具の向きを次のようにプログラム指令できます(キネマティックスは影響しません)。

- 方向ベクトルのプログラミング:
A3=<...> B3=<...> C3=<...>
- オイラー角または RPY 角のプログラミング:
A2=<...> B2=<...> C2=<...>

工具の回転

6 軸の機械では、工具オリエンテーションに加えて工具の回転もプログラム指令できます。回転角度は、以下のようにプログラム指令します。

THETA=<...>

「工具オリエンテーションの回転(ORIOTA、ORIOTR、ORIOTT、ORIOTC、THETA) (ページ 786)」を参照してください。

注記

追加の回転をプログラムした NC ブロックは、回転角度が一次で変化する場合にのみ圧縮できます。つまり、回転角度に対して、PO[THT] = (...)を使用して多項式をプログラムすることはできません。

圧縮可能な NC ブロックの一般的構造

したがって、圧縮可能な NC ブロックの一般的な構文は、以下のようになります。

N...X=<...> Y=<...> Z=<...> A3=<...> B3=<...> C3=<...> THETA=<...> F=<...>

または

N...X=<...> Y=<...> Z=<...> A2=<...> B2=<...> C2=<...> THETA=<...> F=<...>

注記

位置データは直接(X90 など)、またはパラメータ設定で間接的に($X=R1*(R2+R3)$)入力できます。

回転軸位置を使用した工具オリエンテーションのプログラミング

工具オリエンテーションは、回転軸位置を使用して、次のような構文で指定することもできます。

N...X=<...> Y=<...> Z=<...> A=<...> B=<...> C=<...> THETA=<...> F=<...>

この場合は、大半径円弧補間が実行されるかどうかに応じて、2つの異なる方法で圧縮が実行されます。大半径円弧補間が行われない場合は、圧縮された向きの変更が回転軸の軸多項式により通常の方法で表されます。

輪郭精度

選択した圧縮モード(MD20482 \$MC_COMPRESSOR_MODE)に応じて、設定した軸別の許容範囲(MD33100 \$MA_COMPRESS_POS_TOL)、または以下のチャネル別の許容範囲(セッティングデータで設定)が、ジオメトリ軸と旋回軸の圧縮で有効になります。

SD42475 \$SC_COMPRESS_CONTUR_TOL (最大輪郭誤差)

SD42476 \$SC_COMPRESS_ORI_TOL (工具オリエンテーションの最大角度誤差)

SD42477 \$SC_COMPRESS_ORI_ROT_TOL (工具の回転角度の最大角度誤差) (6軸の機械でのみ使用可能です)

4.9 座標変換

起動と解除

コンプレッサ機能は、モーダル G 命令 COMPON、COMPCURV、COMPCAD、または COMPSURF を使用して起動できます。

コンプレッサ機能は COMPOF で終了します。

「NC ブロック圧縮(COMPON、COMPCURV、COMPCAD、COMPSURF、COMPOF)の有効化/無効化 (ページ 691)」を参照してください。

注記

旋回動作が圧縮されるのは、大半径円弧補間が有効なときのみです(つまり、旋回開始と旋回終了で特定される平面で工具オリエンテーションが変更されます)。

大半径円弧補間は、次の条件下で実行されます。

- MD21104 \$MC_ORI_IPO_WITH_G_CODE = 0、
ORIWKS が有効で、
向きを (A3, B3, C3 または A2, B2, C2 を使用した) ベクトルとしてプログラム指令している。
- MD21104 \$MC_ORI_IPO_WITH_G_CODE = 1 であり、
ORIVECT または ORIPLANE が有効である。
工具オリエンテーションは、方向ベクトルとして、または回転軸位置を使用してプログラム指令できます。G 命令 ORICONxx と ORICURVE のいずれかが有効な場合、または旋回角度の多項式(PO[PHI]と PO[PSI])をプログラムしている場合は、大半径円弧補間は実行されません。

例

以下のプログラムの例では、多角形定義により近似される円弧を圧縮します。工具オリエンテーションは同時に、円錐面を移動します。プログラムされた向きの変更が次々と、ただし、不連続的に実行された場合は、コンプレッサ機能により、旋回動作が滑らかになります。

プログラミング	コメント
DEF INT NUMBER=60 DEF REAL RADIUS=20 DEF INT COUNTER DEF REAL ANGLE N10 G1 X0 Y0 F5000 G64 //sort match rate lower first \$SC_COMPRESS_CONTUR_TOL=0.05 \$SC_COMPRESS_ORI_TOL=5 TRAORI COMPCURV N100 X0 Y0 A3=0 B3=-1 C3=1	 ; 輪郭の最大誤差= 0.05 mm ; 旋回の最大誤差= 5° ; 移動は、多角形から作成された円弧を表わします。旋回は、z 軸を中心として、開口角度 45°の円錐上を移動します。

プログラミング	コメント
N110 FOR COUNTER=0 TO NUMBER	
N120 ANGLE=360*COUNTER/NUMBER	
N130 X=RADIUS*cos(角度) Y=RADIUS*sin(角度)	
A3=sin(角度) B3=-cos(角度) C3=1	
N140 ENDFOR	

4.9.7 向きの特性の有効化と無効化(ORISON、ORISOF)

「向きの特性のスミージング」は、G グループ 61 の命令を使用して、パートプログラムで有効化または無効化できます。これらの命令はモーダルです。

必要条件

- 5/6 軸座標変換を使用したシステム
- コンプレッサ機能 COMPCAD が有効

構文

```
ORISON
...
ORISOF
```

意味

ORISON:	向きの特性のスミージングを起動
ORISOF:	向きの特性のスミージングを解除

例

プログラムコード	コメント
...	
TRAORI()	; 方向座標変換の適用。
COMPCAD	; COMPCAD コンプレッサ機能の起動
ORISON	; オリエンテーションのスミージングの起動。
\$SC_ORISON_TOL=1.0	; 工具オリエンテーションの最大角度誤差= 1.0°
G91	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	

4.9 座標変換

プログラムコード	コメント
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
X10 A3=1 B3=0 C3=1	
X10 A3=-1 B3=0 C3=1	
...	
ORISO	; 旋回スムージングの解除。
...	

向きが XZ 平面上で-45°から+45°へと、90°旋回します。旋回処理のスムージングにより、向きが、最大角度値の-45°または+45°のどちらにも達しなくなります。

4.9.8 キネマティックトランスフォーメーション

4.9.8.1 面端座標変換(TRANSMIT)の起動

前面座標変換(TRANSMIT)は、TRANSMIT 命令を使用してパートプログラムまたはシンクロナイズドアクションで有効になります。

構文

TRANSMIT

TRANSMIT (<n>)

意味

TRANSMIT:	1 番目の TRANSMIT データセットで TRANSMIT を有効にします。
TRANSMIT (n):	n 番目の TRANSMIT データセットで TRANSMIT を有効にします。

注記

チャンネルで TRANSMIT 座標変換が有効になり、次のことが適用されます:

- 座標変換の解除: TRAFOOF
- 別の座標変換の適用: 例: TRACYL、TRAANG、TRAORI

4.9.8.2 円筒補間(TRACYL)の起動

円筒座標変換(TRACYL)は、TRACYL 命令を使用してパートプログラムまたはシンクロナイズドアクションで有効になります。

構文

TRACYL (<d>)

TRACYL (<d>,<n>)

TRACYL (<d>,<n>,<k>)

意味

TRACYL (<d>):	1 番目の TRACYL データセットと作業直径<d>で TRACYL を有効にします。	
TRACYL (<d>,<n>):	<n>番目の TRACYL データセットと作業直径<d>で TRACYL を有効にします。	
<d>:	基準または作業直径 この値は 1 より大きくします。	
<n>:	TRACYL データセット番号(任意選択)	
	値の範囲:	1, 2
<k>:	パラメータ<k>は座標変換タイプ 514 にのみ関連します。	
	k = 0:	溝壁補正なし
	k = 1:	溝壁補正あり
	パラメータが指定されていない場合、次のパラメータ設定された基本位置が適用されます: \$MC_TRACYL_DEFAULT_MODE_<n> <n> = TRACYL データセット番号	

注記

チャンネル内で有効な TRACYL 座標変換は、以下によってオフに切り替えることができます。

- 座標変換の解除:TRAFOOF
- 別の座標変換の有効化:TRAANG、TRANSMIT、TRAORI など

4.9 座標変換

例

プログラムコード	コメント
...	
N40 TRACYL(40.)	; 1 番目の TRACYL データセットと作業直径 40 mm で TRACYL を有効にします。
...	

詳細情報

プログラム構成

TRACYL 座標変換 513 (溝壁オフセットありの TRACYL)を使用して溝を加工するパートプログラムは通常、以下の手順から成ります。

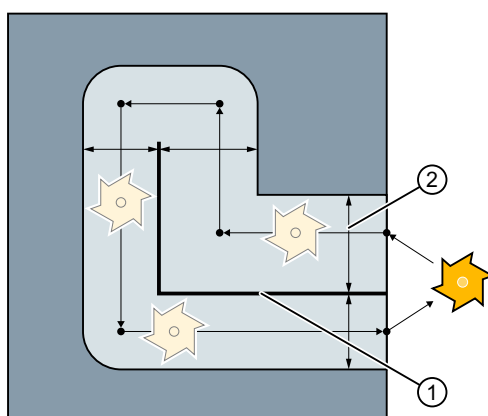
1. 工具の選択
2. TRACYL の選択
3. 適切な座標オフセット(フレーム)の選択
4. 位置決め
5. OFFN のプログラム指令
6. TRC の選択
7. 移動ブロック(TRC を位置決め後に溝壁へアプローチする)
8. 溝の中心線の輪郭
9. TRC の選択解除
10. 後退ブロック(TRC を後退後に溝壁から離れる)
11. 位置決め
12. TRAFOOF
13. オリジナルの座標シフト(フレーム)の再選択

輪郭オフセット(OFFN)

TRACYL 座標変換 513 を使用して溝を加工するには、パートプログラムで、溝の中心線と溝幅の半分を OFFN アドレスを使用してプログラム指令します。

溝壁の損傷を避けるために、OFFN は工具径補正が有効なときにのみ機能します。

パートプログラム内で OFFN を変更できます。これにより、溝の中心線を中心からオフセットすることができます。



- ① OFFN
- ② プログラム指令軌跡

注記

溝壁の反対側の損傷を避けるために、OFFN には工具の半径以上の値を指定してください。

注記

OFFN は、TRACYL と併用した場合と、TRACYL と併用しない場合では、動作が異なります。TRACYL と併用していない場合でも、TRC が有効なときは OFFN が含まれるため、OFFN は、TRAFOOF の後でゼロにリセットしてください。

通知

OFFN の動作は、座標変換タイプによって異なります。

TRACYL 座標変換 513 (溝壁オフセットありの TRACYL) の場合、溝幅の半分を OFFN にプログラム指令します。

TRACYL 座標変換 512 (溝壁オフセットありの TRACYL) の場合、OFFN 値は TRC の許容範囲として機能します。

工具径補正(TRC)

TRACYL 座標変換 513 の場合、TRC は溝壁ではなく、プログラム指令された溝の中心を基準にして考慮されます。工具を溝壁の左に移動させるには、命令 G41 の代わりに G42 をプログラム指令するか、OFFN の値を負の符号付きで指定してください。

工具直径

TRACYL と直径が溝幅より小さい工具を使用した場合、直径が溝幅と同じ工具を使用した場合と同じ溝壁の形状は生成されません。精度を上げるには、工具の直径が溝幅より少しだけ小さくなるよう選択することをお勧めします。

4.9 座標変換

軸の使用

注記

以下の軸は、位置決め軸と揺動軸のいずれにも使用できません。

- 円筒面の外周面方向のジオメトリ軸(Y 軸)
- 溝壁補正用の追加の直線軸(Z 軸)

4.9.8.3 角度がプログラム指令可能な傾斜角座標変換(TRAANG)の有効化(TRAANG)

角度がプログラム指令可能な傾斜角座標変換(TRAANG)は、TRAANG 命令を使用して、パートプログラムまたはシンクロナイズドアクションで有効にします。

キネマティック結合の制限

マシンデータによるパラメータ設定の場合、対応する座標軸に対して傾斜軸の角度 α はマシンデータを使用して直交座標系で定義されています。角度 α は、TRAANG(< α >)を使用して呼び出される場合は変更できます。変更された角度はキネマティックモデルに対する変更になり、他のアプリケーションはこの変更により影響を受けることがあります。そのため、角度の指定は、角度がキネマティック結合から生じる角度と同一である場合だけ許可されます。

注記

角度の定義

傾斜軸が座標軸を中心にして回転する場合のみ、角度は定義されます。そのため、ジオメトリ軸は 1 つだけで、座標軸の 1 つと並行でなく、主面にあります。

構文

```
TRAANG  
TRAANG (  
TRAANG (, <n>  
TRAANG (< $\alpha$ >  
TRAANG (< $\alpha$ >, <n>)
```


意味

TRAANG: TRAANG () :	1 番目の TRAANG データセットと最後の有効な角度<α>で TRAANG を有効にします。	
TRAANG (, <n>) :	<n>番目の TRAANG データセットと最後の有効な角度<α>で TRAANG を有効にします。	
TRAANG (<α>) :	1 番目の TRAANG データセットと角度<α>で TRAANG を有効にします。	
TRAANG (<α> , <n>) :	<n>番目の TRAANG データセットと角度<α>で TRAANG を有効にします。	
<α> :	傾斜軸の角度(任意選択)	
	値の範囲:	-90° < α < + 90°
	角度を指定しない場合は、次のマシンデータでパラメータ設定した初期状態が有効になります。 MD2xxxx \$MC_TRAANG_ANGLE_<n>	
<n> :	TRAANG データセット番号(任意選択)	
	値の範囲:	1, 2

注記

チャンネルで有効な傾斜角座標変換 TRAANG は、以下を行って無効にします。

- 座標変換の解除:TRAFOOF
- 別の座標変換の適用:例: TRACYL、TRANSMIT、TRAORI

例

プログラムコード	コメント
N20 TRAANG (45)	; 1 番目の TRAANG データセットと角度 45°で TRAANG を有効にします。

4.9.8.4 研削盤での傾斜プランジ切削(G5, G7)

G 命令の G7 と G5 は、「傾斜軸」 (TRAANG)による研削盤での傾斜プランジ切削のプログラム指令を簡素化するのに使用されます。その結果、プランジ切削時に傾斜軸だけが移動します。

プランジ切削動作の必要な終了位置だけを X と Z にプログラム指令します。G7 では、X 軸の現在位置から開始して、NC が傾斜軸のプログラム指令された終了位置と角度 α を計算し、アプローチします。

4.9 座標変換

開始位置は、2 つの直線の交点から計算されます。

- X 軸の現在値から少し離れた Z 軸に平行な直線
- プログラム指令された終了位置を通る傾斜軸に平行な直線

この後の G5 で、傾斜軸はプログラム指令された終了位置に移動します。

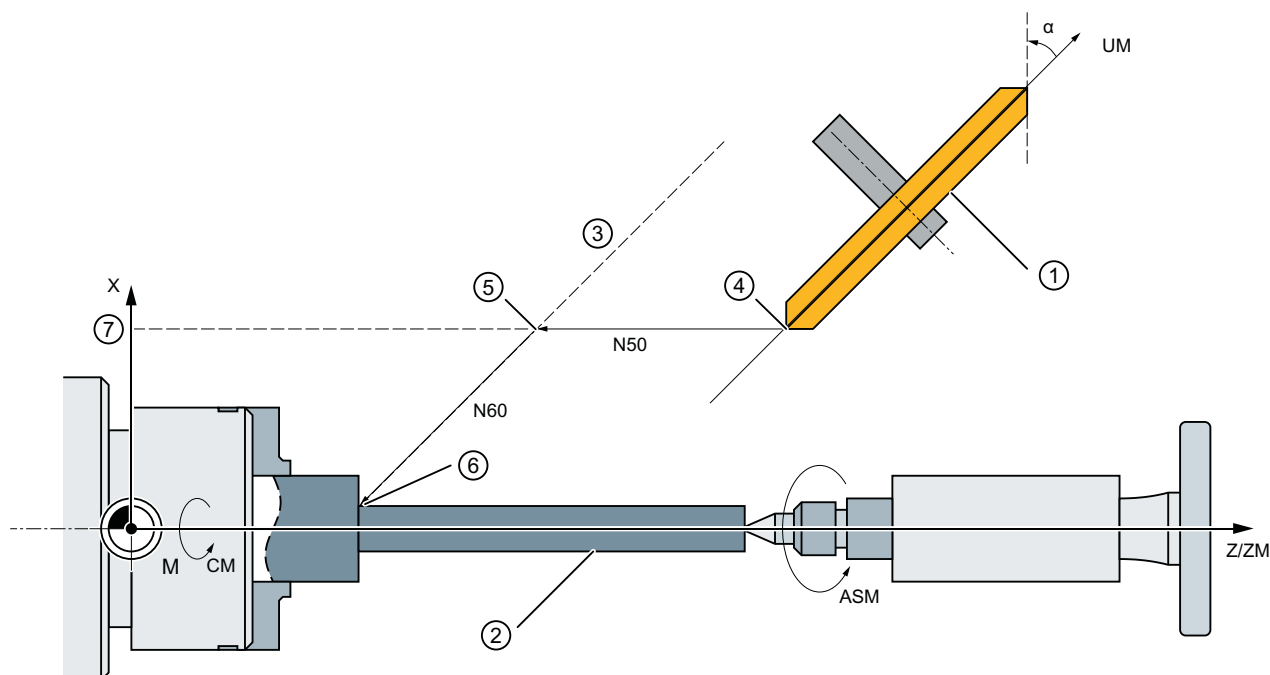
構文

```
G7 <終了位置_x> <終了位置_z>  
G5 <終了位置_x>
```

意味

G7:	傾斜プランジ切削の開始地点を計算し、アプローチします。
G5:	傾斜軸をプログラム指令された終了位置に移動します。
<終了位置_x>:	X 軸の終了位置
<終了位置_z>:	Z 軸の終了位置

例



- ① 研削といし
- ② ワーク
- ③ プログラム指令された終了位置を通過して傾斜軸に平行
- ④ 開始位置
- ⑤ プランジ切削:開始位置
- ⑥ プランジ切削:終了位置
- ⑦ X 軸の現在値から少し離れて Z 軸に平行

X ジオメトリ軸

Z ジオメトリ軸

ZM 機械軸

UM 機械軸

図 4-6 傾斜軸のプログラミング

プログラムコード	コメント
N...G18	; XZ 平面を選択します
N40 TRAANG (45.0)	; TRAANG 座標変換、角度 = 45°を有効にします。
N50 G7 X40 Z70 F4000	; 開始位置を計算してアプローチします。
N60 G5 X40 F100	; 傾斜軸を終了位置に移動します。
N70 ...	

4.9 座標変換

4.9.9 座標変換重畳(TRACTION)の起動

設定した座標変換重畳は、TRAFOON (ページ 836)または TRACON 命令を使用したパートプログラムまたはシンクロナイズドアクションで有効になります。

TRACON によるキネマティック結合でパラメータ設定された座標変換を有効にするためには、システム変数\$NT_TRACON_CHAIN でパラメータ設定する必要があります。

構文

```
TRACON(<Trafo_No>,<Par_1>,...,<Par_n>,<Par_n+1>)
...
TRAFOOF
```

意味

TRACON:	座標変換重畳を有効にします 別の変換が以前に動作中であった場合、その変換は、TRACON()によって自動的に解除されます。		
<座標変換番号>:	座標変換重畳の番号		
	タイプ:	INT	
	値の範囲:	0 ... 2	
	値:	0, 1	1 番目/座標変換重畳のみ
		2	2 番目の座標変換重畳
		指定なし	0 または 1 と同じ意味
		注: 0、1、2 以外の値を指定すると、エラーメッセージが生成されます。	

<code><Par_1>, ...,</code> <code><Par_n>,</code> <code><Par_n+1>:</code>	座標変換重畳のパラメータ	
	<code><Par_1>, ...,</code> <code><Par_n></code>	チェーンの 1 番目の座標変換のパラメータ 実際のは座標変換タイプに依存します。 • TRAORI:2 パラメータ • TRACYL:1~2 パラメータ
	<code><Par_n+1></code>	チェーンの 2 番目の座標変換のパラメータ(TRAANG) 傾斜軸の Δ 角度です。
パラメータが設定されていない場合は、初期設定または最後に使用されたパラメータが有効となります。以前のパラメータで初期設定が有効な場合は、指定したパラメータが確実に目的の順序で使用されるよう、コンマを使用してください。特に、<座標変換番号>の指定が不要な場合でも、1 つ以上のパラメータの前にはコンマが必要です - たとえば、TRACON(, 3.7)。		
TRAFOOF:	最後に有効にした座標変換(重畳)を解除します。	

例

プログラムコード	コメント
...	
N230 TRACON (1, 45.)	; 1 番目の座標変換重畳を有効にします。 ; 以前に有効であった座標変換は自動的に選択解除されます ; 傾斜軸の角度は 45°です。
...	
N330 TRACON (2, 40.)	; 2 番目の座標変換重畳を有効にします。 ; 傾斜軸の角度は 40°です。
...	
N380 TRAFOOF	; 2 番目の座標変換重畳を解除します。
...	

4.9 座標変換

4.9.10 直交 PTP 移動

4.9.10.1 直交 PTP 移動(PTP、PTPG0、PTPWOC、CP)の有効化/無効化

直交ポイントツーポイント、つまり PTP 移動は G グループ 49 命令を使用して NC プログラムで有効化/無効化されます。

これらの命令はモーダルです。デフォルト設定は、直交軌跡移動(CP)での移動です。

CP とは対照的に、有効な PTP 移動の場合、直交ターゲット点のみが変換され、機械軸は同期制御で移動されます。

直交ターゲット点を機械軸値に一義的に変換するには、位置データと角度データに加えて、軸位置を識別するための情報も必要です。このデータは、設定可能なアドレス STAT (ページ 811)および TU (ページ 816)から取り出されます。

必要条件

座標変換 TRAORI_DYN、TRAORI_STAT、TRANSMIT_K、RCTRA または ROBX は有効であること(通常 TRAORI および TRANSMIT)。

RCTRA または ROBX はキネマティック結合に使用できません。

構文

PTP / PTPG0 / PTPWOC
...
CP

意味

PTP:	ポイントツーポイント移動 PTP の起動 G0 および G1 ブロック内のプログラムされた直交位置には、同期軸移動でアプローチします。
PTPG0:	ポイントツーポイント移動 PTPG0 の起動 同期軸移動でアプローチされるプログラムされた直交位置は、G0 ブロック内にのみ存在します。G1 ブロックでは、CP 軌跡移動に切り替えられます。

PTPWOC:	ポイントツーポイント移動 PTPWOC を有効化 (方向座標変換が有効な場合のみ可能) ただし、PTP と同様に、回転軸および旋回軸によって生じる補正移動はありません。
CP:	ポイントツーポイント移動の無効化と軌跡移動 CP の有効化 直交軌跡移動は CP で実行されます。

注記**PTPWOC**

PTPWOC を座標変換 RCTRA または ROBX と併用することは無意味です。RCTRA または ROBX はキネマティック結合に使用できません。

例

参照:

- 例 1:ROBX 座標変換による 6 軸ロボットの PTP 移動 (ページ 819)
- 例 2:総合 5 軸座標変換の場合の PTP 移動 (ページ 820)
- 例 3:PTPGO および TRANSMIT (ページ 821)

4.9.10.2 ジョイント(STAT)の位置を指定します

直交座標付きの位置データと工具オリエンテーションの指定だけでは、機械の位置を一義的に識別するには不十分です。同じ工具オリエンテーションに対して、複数のジョイント位置が考えられるからです。関連するキネマティックに応じて、8 つもの異なったジョイント位置が考えられます。これらの異なったジョイント位置は、座標変換に固有です。

不確定性を回避するために、ジョイント位置は STAT アドレスで指定されます。

注記

制御装置は PTP 移動に対してのみ、プログラムされた STAT 値を考慮します。プログラムされた STAT 値は、CP 移動の場合は無視されます。有効な座標変換での移動中に、位置の変更は通常、おこなえないからです有効な CP で移動する場合、ターゲット点の位置は起点から取られます。

構文

STAT=<値>

意味

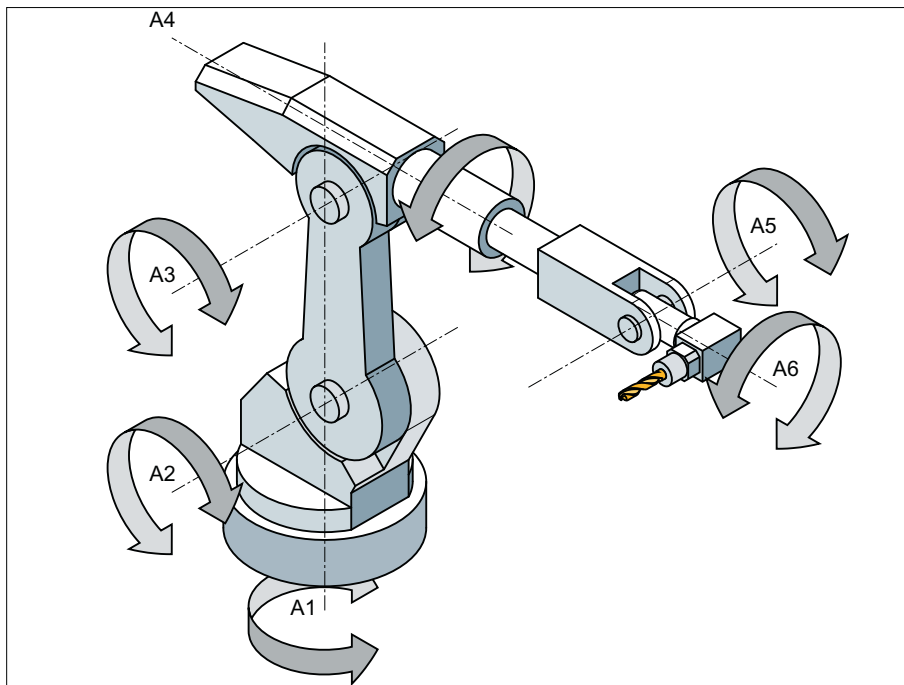
STAT:	ジョイント位置を指定するための設定可能なアドレス
<値>:	2 進数または 10 進数値 考えられる位置ごとに 1 ビットが含まれます。ビットの有効桁は特定の座標変換によって定義されます。

STAT の使用は、フライス主轴を備えた 6 軸の多関節ロボットの例で説明されます。ROBX ロボット座標変換を使用してキネティックトランスフォーメーションが実行されます(前提条件:コンパイルサイクル「RMCC/RCBX 座標変換拡張ロボティクス」がロードされ、有効であること)。

注記

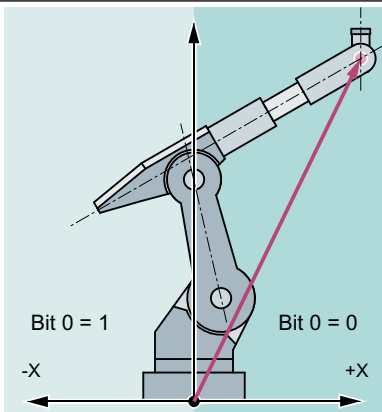
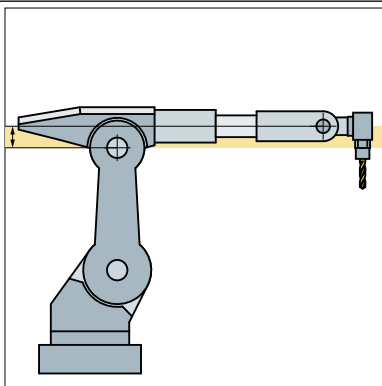
例: 従来のマシンデータのパラメータ設定による ROBX のみ

例は、キネマティック結合ではなく、マシンデータのパラメータ設定によってだけ可能です。



軸 A1、A2、および A3 は、多関節ロボットのメイン軸です。ヘッド軸またはハンド/リスト軸としても指定される軸 A4、A5、および A6 が、メイン軸とともに作業領域に配置されます。ハンド/リスト軸の移動オプションの追加により、特定の加工作業での必要に応じてフライス主轴を空間で旋回できます。同じ工具オリエンテーションを実現するために、さまざまな関節継手位置を使用できます。

加工に必要な関節継手位置は、調節可能な STAT アドレスのビット 0～2 のプログラミングで選択します。

ビット 0	ハンド/リスト軸の交点の位置(A4, A5, A6)		 例:ハンド/リスト軸の交点が基本範囲内にあります。
	= 0	基本範囲(右ショルダー) ハンド/リスト軸の交点の X 値が A1 座標系を基準にして正の場合、ロボットは基本範囲内にあります。	
	= 1	オーバーヘッド範囲(左ショルダー) ハンド/リスト軸の交点の X 値が A1 座標系を基準にして負の場合、ロボットはオーバーヘッド範囲内にあります。	
ビット 1	軸 3 の位置		
	ビット 1 の値が変化する角度は、特定のロボットタイプによって異なります。軸 3 および 4 が交差するロボットには以下が適用されます。		
	= 0	A3 < 0° (下部エルボー)	
	= 1	A3 ≥ 0° (上部エルボー)	
	注記: 軸 3 と 4 の間にオフセットのあるロボットの 場合、ビット 1 の値が変化する角度はこのオフセットの大きさによって異なります。		 A3 と A4 の間のオフセット
ビット 2	軸 5 の位置		
	= 0	A5 ≥ 0° (ハンドフリップなし)	
	= 1	A5 < 0° (ハンドフリップ)	

4.9 座標変換

プログラム例:

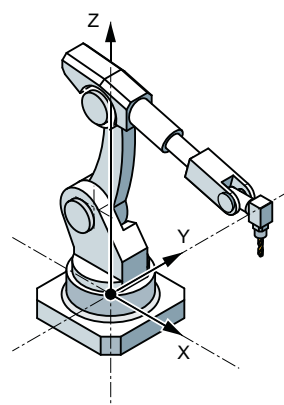
プログラムコード	コメント
...	
N14 T="T8MILLD20" D1	; \$TC_DP3[1,1]=132.95
N16 ORIMKS	
N17 G1 PTP X1665.67 Y0 Z1377.405 A=0 B=0 C=0 STAT=... F2000	; STAT 値は関節継手位置を定義します (以下を参照)
...	

STAT=1 ('B001')

→ 左ショルダー

→ 下部継手

→ ハンドフリップなし

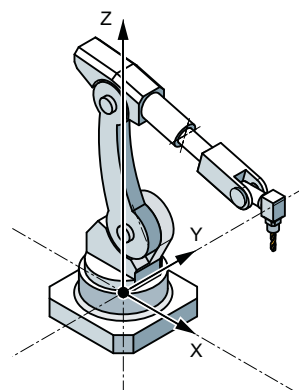


STAT=2 ('B010')

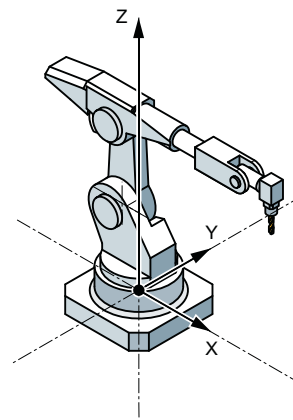
→ 右ショルダー

→ 上部継手

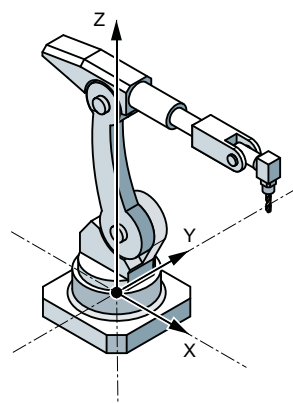
→ ハンドフリップなし



STAT=5 ('B101') → 左ショルダー
 → 下部継手
 → ハンドフリップ



STAT=6 ('B110') → 右ショルダー
 → 上部継手
 → ハンドフリップ



TRANSMIT_K (従来は TRANSMIT)

TRANSMIT_K の場合、アドレス STAT を使用して極に関する不確定性を解決します。

回転軸が 180°回転する必要があったり、CP の輪郭が極を通過する場合、以下が適用されます。

ビット 0	\$NT_POLE_SIDE_FIX[n] = 1 または 2 に対してのみ関連 マシンデータのパラメータ設定の場合は、\$MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 1 または 2:	
	= 0	回転軸が+180°移動、つまり右回りに回転します。
	= 1	回転軸が-180°移動、つまり左回りに回転します。

4.9 座標変換

ビット 1	\$NT_POLE_SIDE_FIX[n] = 0 に対してのみ関連 マシンデータのパラメータ設定の場合、\$MC_TRANSMIT_POLE_SIDE_FIX_1/2 = 0:		
	= 0	軸が極を通過します。回転軸は回転しません。	
	= 1	軸が極の周囲を回転します。STAT のビット 0 が関係します。	

4.9.10.3 軸角度の符号を指定します(TU)

回転軸が特別な移動方法(中間点など)の必要なしに+180°以上または-180°未満の軸角度にもアプローチできるようにするために、軸角度の符号を設定可能アドレス TU で指定してください。

注記

制御装置は PTP 移動に対してプログラムされた TU 値のみを考慮します。CP 移動は無視されます。

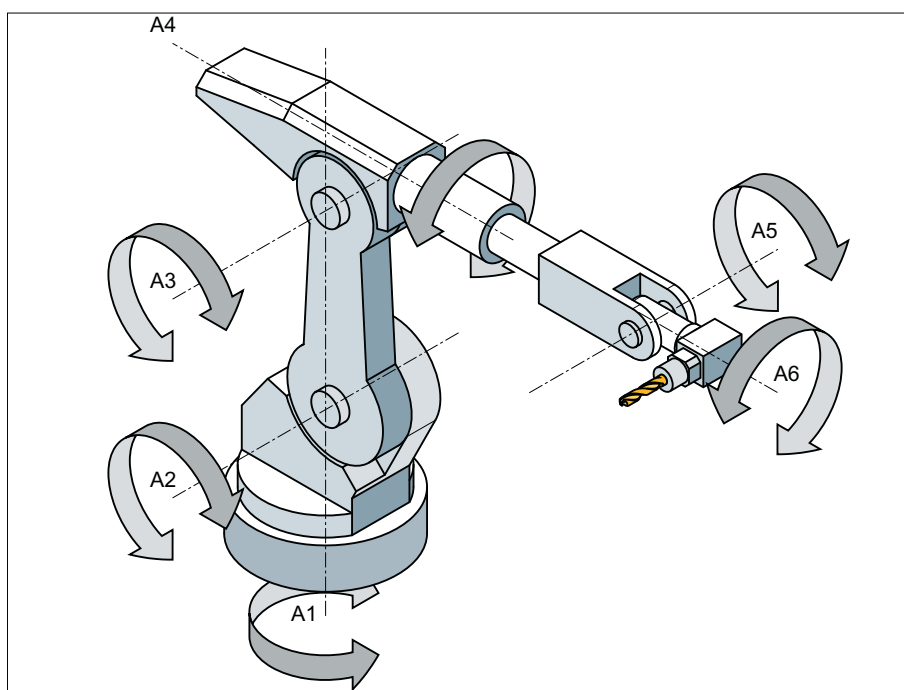
構文

TU=<値>

意味

TU:	軸角度の符号を指定するための設定可能アドレス			
<値>:	2 進数または 10 進数値 座標変換に関係する軸ごとに、軸角度(θ)の符号、すなわち移動方向を示すビットが存在します。			
	ビット	= 0	軸角度の符号: +	軸角度の範囲: $0^\circ \leq \theta < 360^\circ$
		= 1	軸角度の符号: -	軸角度の範囲: $-360^\circ < \theta < 0^\circ$

例:6 軸の多関節ロボット



ビット	意味	値	軸角度の符号	軸角度
ビット 0 ¹⁾	A1 の軸角度の符号	= 0	+/-	$\geq 0^\circ$
		= 1	-	$< 0^\circ$
ビット 1 ¹⁾	A2 の軸角度の符号	= 0	+/-	$\geq 0^\circ$
		= 1	-	$< 0^\circ$
ビット 2 ¹⁾	A3 の軸角度の符号	= 0	+/-	$\geq 0^\circ$
		= 1	-	$< 0^\circ$
ビット 3 ¹⁾	A4 の軸角度の符号	= 0	+/-	$\geq 0^\circ$
		= 1	-	$< 0^\circ$
ビット 4 ¹⁾	A5 の軸角度の符号	= 0	+/-	$\geq 0^\circ$
		= 1	-	$< 0^\circ$

4.9 座標変換

ビット	意味	値	軸角度の符号	軸角度
ビット 5 ¹⁾	A6 の軸角度の符号	= 0	+/-	$\geq 0^\circ$
		= 1	-	$< 0^\circ$

¹⁾ ロボット軸のチャネル軸番号から得られた現在の TU ビット番号例では、ロボット軸(A1～A6)がチャネルの最初の 6 本の軸であるため、TU ビット 0～5 が使用されます。ロボット軸の別のチャネル軸の割り当てでは、ロボット軸の TU ビット番号が、それに対応して変わります(例:ロボット軸は 3～8 番目のチャネル軸です。つまり、TU ビット 2～7 がロボット軸に使用されます)。

従って、TU=19 (TU='B010011 に対応)は以下を意味します。

ビット	値		軸角度
0	= 1	⇒	$\theta_{A1} < 0^\circ$
1	= 1	⇒	$\theta_{A2} < 0^\circ$
2	= 0	⇒	$\theta_{A3} \geq 0^\circ$
3	= 0	⇒	$\theta_{A4} \geq 0^\circ$
4	= 1	⇒	$\theta_{A5} < 0^\circ$
5	= 0	⇒	$\theta_{A6} \geq 0^\circ$

注記

移動範囲が $\pm 360^\circ$ を超える軸の場合は、軸が常に最短の軌跡に沿って移動します。これは、軸位置を TU 情報で固有の指定をすることができないためです。

位置に対して TU がプログラム指令されていない場合は、MD30455

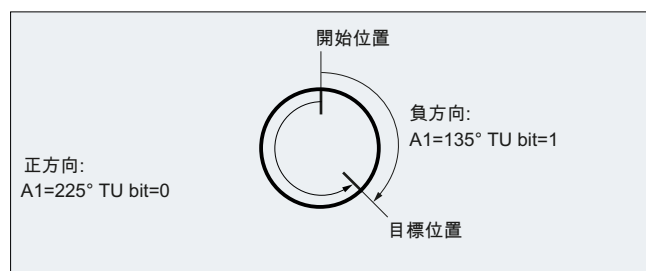
\$MA_MISC_FUNCTION_MASK に応じて、より短い、または長い軌跡を移動します。

TRANSMIT_K (従来は TRANSMIT)

TRANSMIT が有効の PTP 移動の場合、TU のアドレスは意味がありません!

例

次の図に示す回転軸の位置は、負または正方向へアプローチ可能です。回転位置は、アドレス A1 でプログラム指令されます。移動方向は、TU が指定されている場合にのみ絶対的にクリアです。



4.9.10.4 例 1:ROBX 座標変換による 6 軸ロボットの PTP 移動

以下の使用例では、直交 PTP 移動および関連する NC 命令を例として示しています。

注記

従来のマシンデータのパラメータ設定でのみ

この例はキネマティック結合では動作しません。

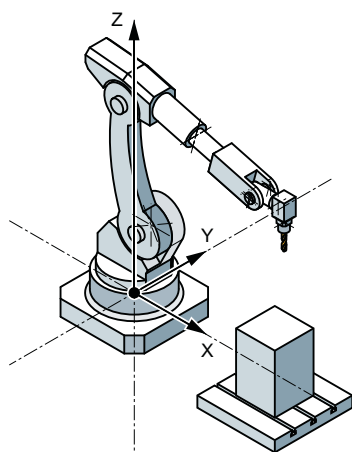


図 4-7 フライス主軸を備えた 6 軸の多関節ロボット

```
N1 G90
N2 T="T8MILLD20" D1 M6
N3 TRAORI
```

4.9 座標変換

```

; $P_UIFR[1]=CTrans(X,1500,Y,0,Z,400):CROT(X,0,Y,0,Z,-90)
N4 G54
N5 M3 S20000
N6 ORIWKS
N7 ORIVIRT1
N8 CYCLE832(0.01,_FINISH,1)
;HOME
N9 TRAFOOF
N10 G0 RA1=0.0000 RA2=-90.0000 RA3=90.0000 A=0.0000 B=90.0000 C=0.0000
N11 TRAORI
N12 G54
N13 G0 PTP X1369.2426 Y956.7528 Z502.5517 A=135.5761 B=-33.2223
C=161.1435 STAT='B010' TU='B001011'
N14 G0 X1355.1242 Y1014.9394 Z424.9695 A=135.8491 B=-33.1439
C=160.9941 STAT='B010' TU='B001011'
N15 G1 CP X1354.8361 Y1016.1269 Z423.3862 A=136.0635 B=-33.0819 C=160.8770
F1000
N16 G1 X1336.4283 Y1016.1269 Z426.6311 A=136.0484 B=-32.2151 C=160.9643
F2000
N17 G1 X1317.9831 Y1016.1269 Z429.6730 A=136.0175 B=-31.3394 C=161.0655
;HOME
N18 TRAFOOF
N19 G0 RA1=0.0000 RA2=-90.0000 RA3=90.0000 A=0.0000 B=90.0000 C=0.0000
N20 M30

```

4.9.10.5 例 2:総合 5 軸座標変換の場合の PTP 移動

前提条件:直角の CA キネマティックスがベースとして使用されていること。

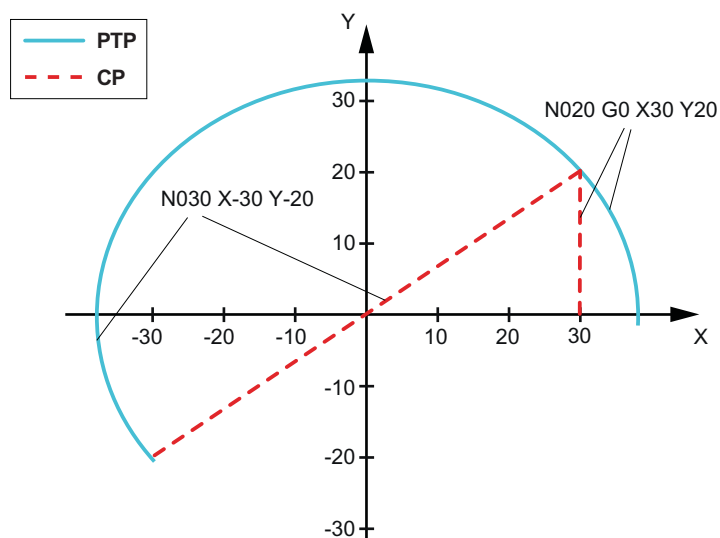
プログラムコード	コメント
TRAORI	; 座標変換 CA キネマティックスをオンにします
PTP	; PTP 移動を起動します
N10 A3=0 B3=0 C3=1	; 回転軸位置 C=0 A=0
N20 A3=1 B3=0 C3=1	; 回転軸位置 C=90 A=45
N30 A3=1 B3=0 C3=0	; 回転軸位置 C=90 A=90
N40 A3=1 B3=0 C3=1 STAT=1	; 回転軸位置 C=270 A=-45

回転軸位置の明確なアプローチ位置は、次のように選択します。

ブロック N40 で、回転軸は **STAT=1** のプログラミングの結果 – 開始点(C=90, A=90)から終点(C=270, A=-45)までの距離より長い距離を移動します。一方、**STAT=0** では、回転軸は終点(C=90, A=45)までの最短距離に沿って移動します。

4.9.10.6 例 3:PTPGO および TRANSMIT

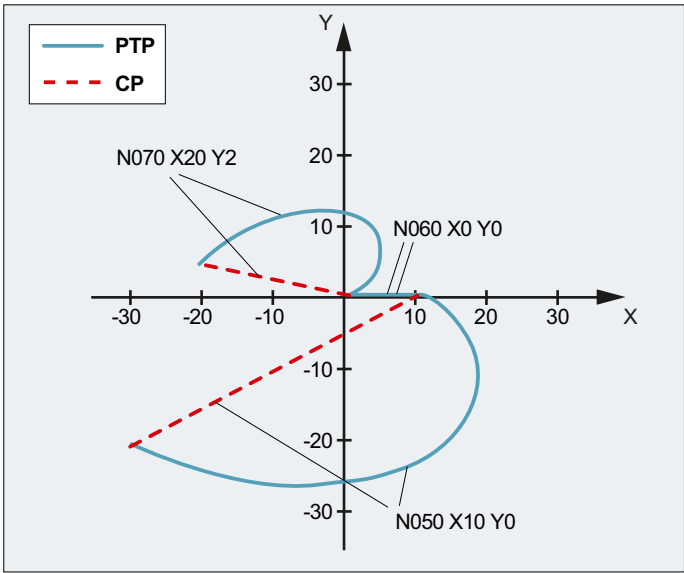
PTPGO および TRANSMIT での極を中心とした移動



プログラムコード	コメント
N001 G0 X30 Z0 F10000 T1 D1 G90	; 初期設定、アブソリュート指令
N002 SPOS=0	
N003 TRANSMIT	;TRANSMIT 座標変換
N010 PTPGO	; G0 ブロックごとに、自動的に PTP - その後、再び CP。
N020 G0 X30 Y20	
N030 X-30 Y-20	
N120 G1 X30 Y20	
N110 X30 Y0	
M30	

4.9 座標変換

PTPG0 および TRANSMIT での極からの移動



プログラミング	コメント
N001 G0 X90 Z0 F10000 T1 D1 G90	; 初期設定
N002 SPOS=0	
N003 TRANSMIT	; TRANSMIT 座標変換
N010 PTPG0	; G0 ブロックごとに、自動的に PTP - その後、再び CP。
N020 G0 X90 Y60	
N030 X-90 Y-60	
N040 X-30 Y-20	
N050 X10 Y0	
N060 X0 Y0	
N070 X-20 Y2	
N170 G1 X0 Y0	
N160 X10 Y0	
N150 X-30 Y-20	
M30	

4.9.11 座標変換の選択時の制約事項

機能

座標変換は、パートプログラムまたは MDI で選択できます。注:

- 中間移動ブロックは挿入されません(面取り/半径)。
- スプラインブロック処理を完了してください。完了しない場合は、メッセージが表示されます。
- 精密工具補正の選択を解除してください(FTOCOF)。解除しない場合は、メッセージが表示されます。
- 工具径補正を解除してください(G40)。解除しない場合は、メッセージが表示されます。
- 動作中の工具長オフセットは、制御装置によって座標変換に含まれます。
- 制御装置は座標変換前に、現在動作中のフレームを解除します。
- 制御装置は、座標変換に対応する軸に対して有効な動作範囲制限を解除します(WALIMOF に対応します)。
- プロテクションゾーンの監視は解除されます。
- 連続軌跡の制御と丸み付けは中断されます。
- マシンデータで指定したすべての軸を、ブロックに対して同期させてください。
- 交換した軸は、元に戻ります。戻らない場合は、メッセージが表示されます。
- 関連軸に対して、メッセージが出力されます。

工具交換

工具は、工具径補正機能が選択解除されているときにのみ交換できます。

工具長補正の変更と工具径補正の選択/解除は、同一ブロックにプログラム指令しないでください。

フレーム変更

基本座標系のみを基準とする命令はすべて使用できます(FRAME、工具径補正)。ただし、G91 (インクレメンタル指令)によるフレーム変更は、座標変換が動作していない場合とは異なり、個別に処理されません。移動するインクレメンタル値は、どのフレームが前のブロックで有効であったかにかかわらず、新しいフレームのワーク座標系で使用されます。

4.10 キネマティック結合

例外

座標変換で影響を受ける軸は、下記の目的では使用できません。

- 事前設定軸(アラーム)
- チェック点へアプローチ(アラーム)
- レファレンス点復帰動作(アラーム)

4.9.12 座標変換の解除(TRAFOOF)

予約 TRAFOOF 手順は、動作中の座標変換とフレームをすべて解除します。

注記

座標変換を選択解除するには、選択と同じ二次条件 (ページ 823)が適用されます。
この後に必要なフレームは、プログラミングを更新して有効にしてください。

構文

...
TRAFOOF

意味

TRAFOOF:	動作中の座標変換/フレームをすべて解除
----------	---------------------

4.10 キネマティック結合

4.10.1 要素の削除(DELOBJ)

DELOBJ () 機能は、割り当てられたシステム変数をデフォルト値にリセットすることで要素を「解除」します。

- キネマティック結合の要素
- プロテクションゾーン、プロテクションゾーン要素、干渉ペア
- 座標変換データ

構文

```
[<RetVal>=] DELOBJ(<CompType>[, , ,<NoAlarm>]) )  
[<RetVal>=] DELOBJ(<CompType>,<Index1>[, ,<NoAlarm>])  
[<RetVal>=] DELOBJ(<CompType>[,<Index1>][ ,<Index2>][ ,<NoAlarm>])
```

4.10 キネマティック結合

意味

DELOBJ:	キネマティック結合、プロテクションゾーン、プロテクションゾーン要素、干渉ペア、座標変換データからの要素の削除
---------	--

<CompType>:	削除する要素タイプ	
	データタイプ:	STRING
	値:"KIN_CHAIN_ELEM" 意味:すべてのキネマティック要素のシステム変数: \$NK_...	
	値:"KIN_CHAIN_SWITCH" 意味:システム変数\$NK_SWITCH[<i>]	
	値:"KIN_CHAIN_ALL" 意味:すべてのキネマティック要素とスイッチ。 「KIN_CHAIN_ELEM」および「KIN_CHAIN_SWITCH」による DELOBJ の連続呼び出しと同じです。	
	値:"PROT_AREA" 意味:プロテクションゾーンのシステム変数: • \$NP_PROT_NAME • \$NP_CHAIN_NAME • \$NP_CHAIN_ELEM • \$NP_1ST_PROT	
	値:"PROT_AREA_ELEM" 意味:機械プロテクションゾーンまたは自動工具プロテクションゾーンのプロテクションゾーン要素のシステム変数 • \$NP_NAME • \$NP_NEXT • \$NP_NEXTP • \$NP_COLOR • \$NP_D_LEVEL • \$NP_USAGE • \$NP_TYPE • \$NP_FILENAME • \$NP_PARA • \$NP_OFF • \$NP_DIR • \$NP_ANG	
	値:"PROT_AREA_COLL_PAIRS" 意味:干渉ペアのシステム変数: • \$NP_COLL_PAIR • \$NP_SAFETY_DIST	

4.10 キネマティック結合

	値:"PROT_AREA_ALL" 意味:すべてのプロテクションゾーン、プロテクションゾーン要素、および干渉ペア(システム変数 \$NP_...) 「PROT_AREA」「PROT_AREA_ELEM」、および「PROT_AREA_COLL_PAIRS」による DELOBJ の連続呼び出しと同じです。 値:"TRAFO_DATA" 意味:すべての座標変換のシステム変数\$NT_...	
<Index1>:	削除される 1 番目の要素のインデックス(オプション)	
	データタイプ:	INT
	初期値:	-1
	値の範囲:	$-1 \leq x \leq (\text{設定された要素の最大数} - 1)$
	値	意味
	0, 1, 2,	削除される要素のインデックス
	-1	指定されたタイプのすべての要素が削除されます。<インデックス 2>は評価されません。
<Index2>:	削除する最後の要素の開始インデックス(任意選択) <Index2>がプログラム指令されていない場合、<Index1>で指定された要素のシステム変数だけが削除されます。	
	データタイプ:	INT
	初期値:	<Index1>で指定された要素のシステム変数だけが削除されます。
	値の範囲:	$\text{<Index1>} < x \leq (\text{定された要素の最大数} - 1)$
<NoAlarm>:	アラームのマスク(任意選択)	
	データタイプ:	BOOL
	初期値:	FALSE
	値	意味
	FALSE	エラーが発生した場合(<RetVal><0)、プログラムの処理は停止してアラームが表示されます。.
	TRUE	エラーが発生した場合、プログラムの処理は停止せず、アラームは表示されません。 用途:戻り値に対応したユーザー固有の応答

<RetVal>:	機能の戻り値	
	データタイプ:	INT
	値の範囲:	0, -1, -2, ... -7
	値	意味
	0	エラーは発生していません
	-1	パラメータなしで機能呼び出します。最低でもパラメータ<要素タイプ>を指定してください。
	-2	<要素タイプ>は、未知の要素を示します。
	-3	<インデックス 1>は-1 より小さくなります。
	-4	<インデックス 1>は、設定された要素の数より大きくなります。
	-5	<インデックス 1>は、要素グループを削除する場合、-1 以外の値となります。
	-6	<インデックス 2>は、<インデックス 1>より小さくなります。
	-7	<インデックス 2>は、設定された要素の数より大きくなります。
	-8	不十分な書き込み許可

4.10.2 名称を使用したインデックスの特定(NAMETOINT)

ユーザー固有の文字列または名称が、タイプ **STRING** のシステム変数配列に入力されます。システム変数の識別子と名称に基づいて、**NAMETOINT()** 機能が名称に所属するインデックス値を決定します。インデックス値はその名前でシステム変数配列に保存されます。

構文

```
<RetVal> = NAMETOINT(<SysVar>,<Name>[,<NoAlarm>])
```

4.10 キネマティック結合

意味

NAMETOINT:	システム変数インデックスの特定	
<SysVar>:	タイプ STRING のシステム変数配列の名称	
	データタイプ:	STRING
	値の範囲:	タイプ STRING のすべての NC システム変数配列の名称
<Name>:	システム変数のインデックスの特定の対象となる文字列または名称	
	データタイプ:	STRING
<NoAlarm>:	アラームのマスク(任意選択)	
	データタイプ:	BOOL
	初期値:	FALSE
	値	意味
	TRUE	エラーが発生した場合、プログラムの処理は停止せず、アラームは表示されません。 用途:戻り値に対応したユーザー固有の応答
	FALSE	エラーが発生した場合(<RetVal><0)、プログラムの処理は停止してアラームが表示されます。
<RetVal>:	システム変数インデックスまたはエラーメッセージ	
	データタイプ:	INT
	値の範囲:	$-1 \leq x \leq$ (設定された要素の最大数 -1)
	値	意味
	≥ 0	検索された名称が、指定されたシステム変数インデックスで見つかりました。
	-1	検索された名称が見つからなかったか、エラーが発生しました。

例

プログラムコード	コメント
DEF INT INDEX	

4.11 キネマティック結合付きの衝突防止

プログラムコード	コメント
\$NP_PROT_NAME[27]="Cover"	
...	
INDEX = NAMETOINT("\$NP_PROT_NAME","Cover")	; INDEX == 27

4.11 キネマティック結合付きの衝突防止

注記

プロテクションゾーン

以下の章で指定されたプロテクションゾーンは、「ジオメトリック機械モデリング」機能を指しています。

この機能についての詳細は、『機能マニュアル』、「監視と補正」を参照してください。

4.11.1 干渉ペアをチェックします(COLLPAIR)

4.11.2 干渉回避のマシンモデルの再計算要求(ROTA)

キネマティックチェーン\$NK_...、幾何学的マシンモデリング、または干渉回避\$NP_...がパートプログラムに記述されている場合、干渉回避の NC 内部機械モデルで変更が有効になるように、続いて ROTA 処理を呼び出してください。

構文

ROTA[(<Par>)]

意味

ROTA:	干渉回避のマシンモデルの再計算要求
	- 先読み停止をおこないます。 - ブロック内では単独にします。

4.11 キネマティック結合付きの衝突防止

<Par>:	パラメータ(オプション)		
	データタイプ:	STRING	
	値:	---	パラメータなし。 マシンモデルが再計算されました。プロテクションゾーンの状態は維持されます。
		「R」	マシンモデルが再計算されました。プロテクションゾーンは、\$NP_INIT_STAT に従って、それぞれの初期状態に設定されます。

必要条件

シミュレーション

PROTA 処理は、シミュレーション(simNC)と組み合わせてパートプログラムで使用しないでください。

例:シミュレーションが有効な間、PROTA 呼び出しを回避します。

プログラムコード	コメント
...	
IF \$P_SIM == FALSE	; IF シミュレーションが有効ではありません
PROTA	; THEN 干渉モデルを再計算します
ENDIF	; ENDIF
...	

下記も参照

プロテクションゾーン状態の設定(PROTS) (ページ 832)

4.11.3 プロテクションゾーン状態の設定(PROTS)

PROTS () 処理は、プロテクションゾーンの状態を、指定値に設定します。

構文

PROTS (<State>[, <Name_1>, ..., <Name_n>])

意味

PROTS:	プロテクションゾーンの状態の設定		
	• ブロック内では単独にします。		
<State>:	指定されたプロテクションゾーンを設定する状態		
	データタイプ:	CHAR	
	値:	「A」 または 「a」	状態:有効
		「I」 または 「i」	状態:無効
		「P」 または 「p」	状態:起動待ちまたは PLC 制御 ¹⁾
		「R」 または 「r」	状態:初期化ステータスの NC 内部値 ²⁾
	<Name_1> <Name_n>:	指定された状態に設定される複数のプロテクションゾーンの名称(オプション)	
名称を指定しない場合、指定された状態がすべての定義済みプロテクションゾーンに対して設定されます。			
データタイプ:		STRING	
値の範囲:		パラメータ設定されたプロテクションゾーン名称	
注:		パラメータとして指定できるプロテクションゾーンの最大数は、プログラム行ごとの最大許容文字数によって異なります。	

¹⁾ 起動と解除は以下によっておこなわれます。DB10.DBX234.0 - DBX241.7

²⁾ ステータスは、初期化ステータスの NC 内部値に設定されます。つまり、システム変数 \$NP_INIT_STAT の、前回の PROTA() (ページ 831)呼び出し時における値です。

4.11.4 :2 つのプロテクションゾーンの間隔の決定(PROTD)

PROTD() 機能は、2 つのプロテクションゾーンの間隔を計算します。

4.11 キネマティック結合付きの衝突防止

機能の特性

- 間隔の計算は、プロテクションゾーン状態(起動済み、解除、起動待ち)とは無関係に実行されます。
- 2つのプロテクションゾーンの間隔の計算には、\$NP_USAGE = "C"または"A"とマークされたプロテクションゾーン項目のみが使用されます。プロテクションゾーンで\$NP_USAGE = "V"とマークされたプロテクションゾーン項目は考慮されません。
- すべてのプロテクションゾーン項目が\$NP_USAGE = "V"とマークされたプロテクションゾーンは、間隔の計算に使用できません。
- 間隔の計算は、前のブロックの終了時に有効だった位置で実行されます。
- メインラン計算に含まれる重畳(DRF オフセットや外部ゼロオフセットなど)は、機能解析時間での有効な値で間隔の計算に含まれます。

注記

同期制御

PROTD () 機能を使用する場合、必要に応じて STOPRE 先読み停止でメインランと先読みを同期化するのはユーザーの責任で行います。

干渉

指定された 2 つのプロテクションゾーンの間干渉がある場合、機能は 0.0 の値を返します。両方のプロテクションゾーンが互いに接触または交差している場合に、干渉が発生します。

干渉チェックの安全距離(MD10622 \$MN_COLLISION_SAFETY_DIST)は、間隔の計算で考慮されません。

構文

```
[<RetVal> =] PROTD ([<Name_1>], [<Name_2>], VAR <Vector> [, <System>])
```

意味

PROTD:	:2 つの指定されたプロテクションゾーンの間隔を計算します。		
	• ブロック内では単独にします。		
<RetVal>:	機能の戻り値:2 つのプロテクションゾーンのアブソリュート間隔値または干渉時の 0.0(上記の干渉の節を参照してください)		
	データタイプ:	REAL	
	値の範囲:	0.0 ≤ x ≤ +max.REAL 値	

4.11 キネマティック結合付きの衝突防止

<Name_1>, <Name_2>:	間隔を計算する 2 つのプロテクションゾーンの名称(任意選択)		
	データタイプ:	STRING	
	値の範囲:	パラメータ設定されたプロテクションゾーン名称	
	初期値:	"" (空の文字列) プロテクションゾーンを指定しない場合、この機能は干渉モデル内のすべての起動済みおよび起動待ちプロテクションゾーンから、現在の最小の間隔を計算します。	
<Vector>:	戻り値:プロテクションゾーン<Name_2>からプロテクションゾーン<Name_1>までの 3 次元の間隔ベクトル: • <Vector>[0]:ワールド座標系内の X 座標 • <Vector>[1]:ワールド座標系内の Y 座標 • <Vector>[2]:ワールド座標系内の Z 座標 干渉時: <Vector> == ゼロベクトル		
	データタイプ:	VAR REAL [3]	
	値の範囲:	<Vector> [n]:0.0 ≤ x ≤ ±最大 REAL 値	
<System>:	間隔および間隔ベクトルの単位系(インチ/メトリック)(任意選択)		
	データタイプ:	BOOL	
	値:	FALSE (初期設定)	G グループ 13(G70、G71、G700、G710)の現在有効な G 命令に対応する単位系です。
		TRUE	設定されている基本単位系に対応する単位系 MD52806 \$MN_ISO_SCALING_SYSTEM

4.12 キネマティック結合による座標変換

4.12.1 座標変換の有効化(TRAFOON)

キネティックチェーンで定義される座標変換は、予約された TRAFOON 手順によって有効化されます。呼び出しはブロック内では単独である必要があります。

注記
または、キネティックチェーンで定義される座標変換は、TRAORI や TRANSMIT などの通常の NC 命令によって有効にすることもできます。このために、ゼロに等しくない適切な値を \$NT_TRAFO_INDEX のシステム変数に入力する必要があります。
\$NT_TRAFO_INDEX の詳細については、「システム変数パラメータマニュアル」を参照してください。

構文

TRAFOON (<Trafoname>,<Diameter>,<k>)

意味

TRAFOON:	キネティックチェーンで定義される座標変換を有効化するための手順		
<Trafoname>:	座標変換データセットの名称		
	データタイプ:	STRING	
	値の範囲:	\$NK_NAME によって定義される座標変換データセットのすべての名称	
	注記:		
	座標変換データセットの名称は一義的でなければなりません。 \$NT_NAME で一度だけ発生するようにします。		
<Diameter>:	基準または作業直径(TRAACYL のみ)		
	データタイプ:	REAL	
	値は 1 より大きくしてください。		

4.12 キネマティック結合による座標変換

<k>:	溝壁補正の使用を定義します(TRACYL のみ)。		
	データタイプ:	BOOL	
	値:	FALSE	溝壁補正を使用しない場合
		TRUE	溝壁補正を使用する場合
	TRACYL 座標変換タイプ 514 に対応します(溝壁補正はプログラム指令可能です)。<k>を指定していない場合、\$NT_CNTRL[<n>]の 10 ビットのパラメータ設定が適用されます。		

例

プログラムコード	コメント
TRAFOON["Trans_1"]	Trans_1 の名称による座標変換を有効にします。

4.12.2 機械計測後の方向座標変換の変更(CORRTrafo)

キネティックチェーンによって定義された方向座標変換がある機械では、予約した CORRTrafo 機能を使用して、機械計測後に機械のキネマティックモデルの旋回軸の補正ベクトルや方向ベクトルを変更することができます。

注記

CORRTrafo 機能で書き込まれた補正值は、座標変換で直ちに有効になりません。補正值は、座標変換の解除後、NEWCONF 後、および座標変換の選択後まで有効になりません。

構文

```
<Corr_Status> = CORRTrafo(<Corr_Vect>, <Corr_Index>, <Corr_Mode>,
[ <No_Alarm>])
```

4.12 キネマティック結合による座標変換

意味

CORRTRAFO:	ファンクションコール
------------	------------

4.12 キネマティック結合による座標変換

<Corr_Status>: 値:	機能の戻り値	
	データタイプ:	INT
	値:	0 機能が異常なく実行されました。
		1 動作中の座標変換がありません。
		2 現在動作中の座標変換は、方向座標変換ではありません。
		3 動作中の方向座標変換は、キネティックチェーンで定義されていませんでした。
		10 <Corr_Index>コールパラメータは負です。
		11 <Corr_Mode>コールパラメータは負です。
		12 サブチェーンのセクションへの無効な参照です(<Corr_Index>の一の位)。値はサブチェーンの旋回軸の数より大きくしないでください。
		13 サブチェーンの旋回軸への無効な参照です(<Corr_Index>の一の位)。値はサブチェーンの旋回軸の数未満としてください。
		14 サブチェーンへの無効な参照です(<Corr_Index>の十の位)。値 0 と 1 のみが許容されます(パートまたは工具チェーンを参照)。この異常番号は、<Corr_Index>が参照するサブチェーンが存在しない場合に発生します。
		15 <Corr_Index>パラメータ(\$NT_CORR_ELEM_P または \$NT_CORR_ELEM_T)で参照されたセクションに補正エレメントがありません。
		20 無効な補正モードです(<Corr_Mode>の 1 の位)。値 0 と 1 のみが許容されます。
		21 無効な補正モードです(<Corr_Mode>の 10/100 の位)。軸方向の書き込み時にゼロに等しくしてはいけないのは一の位のみです。
		30 <Corr_Mode>の 100 の位が無効です。値 0 と 1 のみが許容されます。
		31 <Corr_Mode>の 1000 の位が無効です。値 0 と 1 のみが許容されます。
		40 軸方向として扱われる方向ベクトルがゼロベクトルです。このエラーは、<Corr_Mode>の 1000 の位が 0

4.12 キネマティック結合による座標変換

		に等しい場合に発生する場合があります。このパラメータの 1000 の位が 1 に等しい場合(最大補正の監視が無効化される)、ゼロベクトルも書き込むことができます。
	41	オフセットベクトルの補正では、少なくとも 1 つの座標における現在値との差分が、設定データ SD41610 \$SN_CORR_TRAFO_LIN_MAX によって指定された最大値より大きくなります。<Corr_Vect>パラメータはエラーベクトルによって上書きされます。これは、アラームによって処理が中止された場合にも適用されます(<No_Alarm>パラメータを参照)。 補正值が許容制限を超えた場合、エラーベクトルには、所定の補正值と制限値の間に、正しい符号の差分があります。 制限を超えていないコンポーネントの内容はゼロです。
	42	方向ベクトルの補正では、現在の座標と比べた角変位は、設定データ SD41611 \$SN_CORR_TRAFO_DIR_MAX によって指定された最大値より大きくなります。
	43	書き込み権限がないため、システム変数を書き込もうとして拒否されました。
<Corr_Vect>:	補正ベクトル 補正ベクトルの内容は、次のパラメータ<Corr_Index>および<Corr_Mode>によって定義されます。 <Corr_Status> = 41 の場合、ベクトルの内容は上書きされます(前述の通り)。	
データタイプ:	REAL	

4.12 キネマティック結合による座標変換

<Corr_Index>:	補正エレメントを変更するセクション/ 方向ベクトルを変更する旋回軸のインデックス		
	データ イプ:	INT	
	<Corr_Index>パラメータは 10 進数コードです(1 から 10 の位):		
	1 の 位:	セクションのインデックスまたはサブチェーンの旋回軸を含みます。	
	10 の 位:	サブチェーンを指します。	
		0x	ワークチェーン
		1x	工具チェーン

4.12 キネマティック結合による座標変換

<Corr_Mode>:	補正モード	
	データタイプ:	INT
	<Corr_Index>パラメータは 10 進数コードです(1 から 1000 の位):	
	1 の位:	補正するエレメントを指定します。
		xxx0 直線オフセットベクトルの補正
		xxx1 旋回軸の方向ベクトルの補正
	10 の位:	<Corr_Index>が指す補正エレメントの変更方法を指定します。
		xx0x 補正ベクトルは、補正エレメントにすぐ書き込まれます。 このタイプは、補正エレメントをすぐ書き込むために使用できます。関連するシステムデータ (\$NK_OFF_DIR[<n>, ...])のインデックス<n>が既知である必要はありません。
		xx1x 0 と同様ですが、転送された補正值が世界座標で解釈される点は異なります。 0 と 1 のタイプの違いは、初期状態(すべての旋回軸の一が 0 に等しい)のキネティックチェーンに他の回転が含まれる場合に必ず発生します。
		xx2x 1 と同様ですが、補正值がセクション全体を指すという点が異なります。すなわち、補正エレメントに値が入力されると、セクション全体が補正值によって定義された長さに達するようになります。
		注記: 値 1 および 2 は、旋回軸の方向ベクトルを書き込む場合には使用できません。
	100 の位:	<Corr_Vect>パラメータの内容の解釈方法を指定します。
		x0xx 転送された補正ベクトル<Corr_Vect>には、補正エレメントの新しい長さ全体、または<Corr_Index>が<Corr_Mode>の 10 の位と組み合わせて参照するセクションが含まれます(絶対補正)。
		x1xx 転送された補正ベクトル<Corr_Vect>には、補正エレメントの現在の長さ、または<Corr_Index>が<Corr_Mode>の 10 の位と組み合わせて参照す

4.12 キネマティック結合による座標変換

			るセクションと比べた違いのみが含まれます(増分補正)。
		注記: 旋回軸の方向ベクトルの補正では、100 の位の内容は 0 でなければなりません。	
	1000 の位:	補正が次の最大値によって制限されるべきかどうかを指定します。 - SD41610 \$SN_CORR_TRAFO_LIN_MAX または - SD41611 \$SN_CORR_TRAFO_DIR_MAX	
		0xxx	最大補正の監視が有効です。
		1xxx	最大補正の監視は有効ではありません。
<No_Alarm>:	エラー発生時の動作(戻り値 > 0) (オプション)		
	データタイプ:	BOOL	
	値:	FALSE (初期設定)	エラーが発生した場合、プログラムの処理は停止し、アラーム 14103 が表示されます。
		TRUE	エラーが発生した場合、プログラムの処理は停止せず、アラームは表示されません。 用途:戻り値に対応したユーザー固有の応答

注記

機能呼び出し時にエラーが発生した場合、アラームが出力されるか、異常番号が返されますので(<No_Alarm>パラメータを参照してください)、エラー状態に適した方法で対応できます。問題の原因はアラームパラメータで詳細に記述されます。アラームの代わりに返された異常番号は、アラームパラメータと同じです。

CORRTRAFO の詳細情報

方向座標変換を伴う機械のキネマティック構造は、世界座標系の原点から始まる 1 つまたは 2 つのキネティックチェーン(サブチェーン)によって説明されます。2 つのチェーンの内の 1 つ、**工具チェーン**は、工具のレファレンス点で終了し、もう 1 つのチェーンである**ワークチェーン**は、基本座標系の原点で終了します。

4.12 キネマティック結合による座標変換

CORRTrafo 機能は、機械上のレバーアームの長さや軸方向を、方向座標変換によって特別な補正エレメントに書き込みます。キネティックチェーンは、\$NK_TYPE で定義されるタイプ OFFSET のエレメントなどを使用して説明されます。

CORRTrafo はセクションで動作する

2 つのサブチェーンは、それぞれ最大 4 つのセクションに分けることができます。

- セクション 1 はチェーンの起点で始まり、最初の回転軸で終了します。
- セクション 2 は回転軸 1 と回転軸 2 の間のセクションです。
- セクション 3 は回転軸 2 と回転軸 3 の間のセクションです。
- セクション 4 は回転軸 3 と工具チェーンまたはワークチェーンの終点の間のセクションです。

各セクションは、タイプ OFFSET または ROT_CONST の一定のチェーンエレメントを含む場合があります。

以下の図は、2 つの回転軸による方向座標変換を示します。

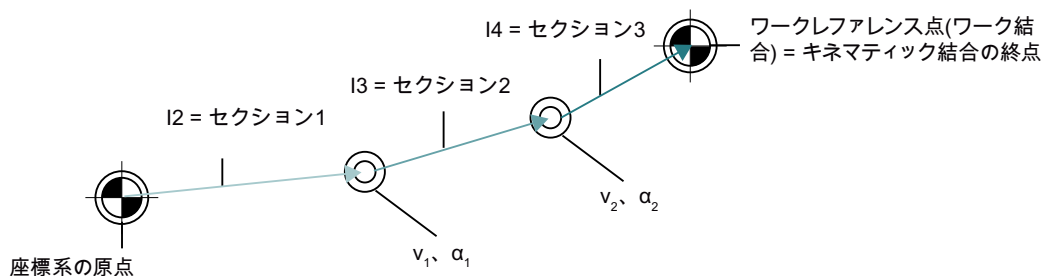


図 4-8 CORRTrafo の例

セクションは明確に定義されます。キネマティックサブチェーンを起点から終点まで動かした場合、最初のセクションはインデックス 0 となり、次はインデックス 1、などとなります。このとき、最後のセクションのインデックスは、常に回転軸の数と等しくなります。

補正エレメント

\$NT_CORR_ELEM_T[<n>, 0 ... 3]または\$NT_CORR_ELEM_P[<n>, 0 ... 3]システム変数によって、各セクションの一定のキネティックチェーンエレメント(\$NK_TYPE[<n>] = "OFFSET"タイプのチェーンエレメント)を参照できます。機械計測中に特定された補正值は、CORRTrafo 機能を使用してこれらのエレメントに書き込まれます。

座標変換インデックス = 1 の例:

- `$NT_CORR_ELEM_T[1,0] = "C_AXIS_OFFSET"`; セクション 1 の C 軸(旋回軸 1)のオフセットが補正エレメントとして定義されます。
- `$NT_CORR_ELEM_T[1,1] = "B_AXIS_OFFSET"`; セクション 2 の B 軸(旋回軸 2)のオフセットが補正エレメントとして定義されます。
- `$NT_CORR_ELEM_T[1,2] = "BASE_TOOL_OFFSET"`; セクション 3 の工具レファレンス点からの B 軸のオフセットが、補正エレメントとして定義されます。

`$NT_CORR_ELEM_T/P[<n>, 0 ... 3]`の参照順は、上述のセクションに対応する必要があります。つまり、最初の旋回軸の前にある`$NT_CORR_ELEM_T/P[<n>, 0]`に存在できるチェーンエレメントは 1 つのみである、などです。

CORRTrafo 機能は、機械を計測することで特定した値を、この方法で定義した補正エレメントに書き込みます。補正值の変更は、<Corr_Mode>パラメータ経由で CORRTrafo で定義されます。

チェーンを閉じる

`$NT_CNTRL[<n>]`システム変数でビット 7 またはビット 8 が設定された場合、追加の一定チェーンエレメント(チェーンの終点から機械の原点への接続を確立する)が、ワークチェーンの終点(ビット 7)または工具チェーンの起点(ビット 8)で、内部に自動的に挿入されます(「クローズチェーン」)。

これらの自動的に挿入されたエレメントは、外部に書き込むことはできず、読み込むことしかできません(`$AC_TRAFO_CORR_ELEM_P/T` システム変数を参照)。

工具チェーンを閉じる点

`$NT_CLOSE_CHAIN_T` システム変数が空でない場合、工具チェーンはチェーンの終点ではなく、指定されたチェーンエレメントの終点で閉じています。この点の外側にあるその他のチェーンエレメントは、座標変換が有効になったときに、結果として対応するワークオフセットとなります。

旋回軸のインデックス

旋回軸の間の一定オフセットに加えて、旋回軸の方向ベクトルも CORRTrafo 機能を使用して書き込むことができます。旋回軸のインデックスとは、キネマティックサブチェーンが起点から終点まで動くときに生じる、カウントがゼロで始まるインデックスです。そのため、旋回軸のインデックスは常に前のセクションのインデックスと同じになります。

旋回軸のインデックスは、`$AC_TRAFO_ORIAX_LOC` システム変数によって特定することもできます。

4.13 工具オフセット

チェーンエレメントの最大許容変化率

チェーンエレメントの最大許容変化率は、旋回軸のオフセットベクトル用の SD41610 \$SN_CORR_TRAFO_LIN_MAX と方向ベクトル用の SD41611 \$SN_CORR_TRAFO_DIR_MAX の 2 つの設定データによって、制限することができます。SD41610 \$SN_CORR_TRAFO_LIN_MAX は最大量を指定し、それによって個別のベクトルコンポーネントそれぞれを、その指令値に関して変更できます。SD41611 \$SN_CORR_TRAFO_DIR_MAX は最大角を指定し、それによって軸ベクトルの方向を、その指令値に関して変更できます。指令値は常に、CORRTrafo を呼び出したときに有効になる座標変換で有効な、対応する値です。これは、キネマティックデータの変更された内容が、座標変換の有効化後に CORRTrafo 機能の動作方法に影響を与えないことを意味します。

4.13 工具オフセット

4.13.1 オフセットメモリ

オフセットメモリの構成

すべてのデータフィールドは T および D 番号によって呼び出すことができます。また、すべてのデータフィールドに、工具の形状の指定だけでなく、工具タイプなどのその他のエントリも含まれています。

ユーザー刃先データ

ユーザー刃先データは、マシンデータで設定できます。工作機械メーカーの取扱説明書を参照してください。

工具パラメータ

注記

オフセットメモリの個々の値

オフセットメモリ P1～P25 の個々の値は、プログラムでシステム変数を使用して読み取りと書き込みができます。その他のパラメータはすべて予約されています。

工具パラメータ \$TC_DP6～\$TC_DP8、\$TC_DP10、および \$TC_DP11 には、工具タイプに応じてもう 1 つの意味があります。\$TC_DP15～\$TC_DP17、\$TC_DP19、および \$TC_DP20 も同様です。

4.13 工具オフセット

工具パラメータ番号(DP)	システム変数の意味	備考
\$TC_DP1	工具タイプ	概要は、リストを参照してください
\$TC_DP2	刃先位置	旋盤工具のみ
ジオメトリ	長補正	
\$TC_DP3	長さ 1	割当先
\$TC_DP4	長さ 2	タイプとレベル
\$TC_DP5	長さ 3	
ジオメトリ	半径	
\$TC_DP6 ¹⁾	半径 1 / 長さ 1	フライス/旋削/研削工具
\$TC_DP6 ²⁾	直径 d	メタルソー
\$TC_DP7 ¹⁾	長さ 2 / コーナ半径、先細フライス工具	フライス工具
\$TC_DP7 ²⁾	切り込み幅 b のコーナ半径	メタルソー
\$TC_DP8 ¹⁾	フライス工具のラウンディング半径 1	フライス工具
\$TC_DP8 ²⁾	突出長 k	メタルソー
\$TC_DP9 ^{1) 3)}	ラウンディング半径 2	予備
\$TC_DP10 ¹⁾	角度 1 工具の先端	先細フライス工具
\$TC_DP11 ¹⁾	角度 2 工具縦軸	先細フライス工具
磨耗	長さと径補正	
\$TC_DP12	長さ 1	
\$TC_DP13	長さ 2	
\$TC_DP14	長さ 3	
\$TC_DP15 ¹⁾	半径 1 / 長さ 1	フライス/旋削/研削工具
\$TC_DP15 ²⁾	直径 d	メタルソー
\$TC_DP16 ¹⁾	長さ 2 / コーナ半径、先細フライス工具、切り込み幅 b のコーナ半径	フライス工具
\$TC_DP16 ³⁾		メタルソー
\$TC_DP17 ¹⁾	フライス工具のラウンディング半径 1	フライス加工/3 次元正面削り
\$TC_DP17 ²⁾	突出長 k	メタルソー
\$TC_DP18 ^{1) 3)}	ラウンディング半径 2	予備
\$TC_DP19 ¹⁾	角度 1 工具の先端	先細フライス工具
\$TC_DP20 ¹⁾	角度 2 工具縦軸	先細フライス工具
工具ベース寸法/アダプタ	長さオフセット	

4.13 工具オフセット

工具パラメータ番号(DP) システム変数の意味		備考
\$TC_DP21	長さ 1	
\$TC_DP22	長さ 2	
\$TC_DP23	長さ 3	
テクノロジー		
\$TC_DP24	逃げ角	旋盤工具のみ
\$TC_DP25		予備

- 1) 3次元正面削り用フライス工具にも適用
- 2) メタルソー工具タイプの場合
- 3) 予約済み

備考

形状変数では、複数の入力成分を使用できます(長さ 1 または半径など)。これらは合計されて 1 つの値(全長 1、全半径)となり、計算に使用されます。

不要なオフセット値には、値ゼロを割り当ててください。

輪郭工具の工具パラメータ\$TC-DP1 to \$TC-DP23

注記

表に含まれない\$TC_DP7 などの工具パラメータは使用されません。つまり、その内容は意味がありません。

工具パラメータ番号(DP)	意味	切削 Dn		備考
\$TC_DP1	工具タイプ			400～599
\$TC_DP2	刃先位置			
ジオメトリ	長補正			
\$TC_DP3	長さ 1			
\$TC_DP4	長さ 2			
\$TC_DP5	長さ 3			
ジオメトリ	半径			
\$TC_DP6	半径			
ジオメトリ	制限角度			

工具パラメータ番号 (DP)	意味	切削 Dn		備考
\$TC_DP10	最小角度			
\$TC_DP11	最大角度			
磨耗	長さ と 径補正			
\$TC_DP12	磨耗長さ 1			
\$TC_DP13	磨耗長さ 2			
\$TC_DP14	磨耗長さ 3			
\$TC_DP15	磨耗半径			
磨耗	制限角度			
\$TC_DP19	最小磨耗角度			
\$TC_DP20	最大磨耗角度			
工具ベース寸法/アダプタ	長さオフセット			
\$TC_DP21	長さ 1			
\$TC_DP22	長さ 2			
\$TC_DP23	長さ 3			

基本値と磨耗値

得られる値はそれぞれ、基本値と磨耗値の合計です(半径の場合は\$TC_DP6 + \$TC_DP15、など)。基本計測(\$TC_DP21 - \$TC_DP23)の値も 1 番目の刃先の工具長に加算されます。他のすべてのパラメータも、標準工具の有効工具長に影響する場合があります、また、この(アダプタ、旋回工具ホルダ、セッティングデータ)工具長にも影響します。

制限角度 1 と 2

制限角度 1 と 2 はそれぞれ、刃先中心点から刃先基準点へのベクトルを基準に、左回りにカウントされます。

4.13.2 追加オフセット

4.13.2.1 追加オフセットの選択(DL)

追加オフセットは、加工のなかでプログラム指令できる処理オフセットと見なすことができます。このオフセットは、刃先の形状データを表わしています。したがって、工具切削データの 1 つの成分です。

4.13 工具オフセット

追加オフセットのデータは、DL 番号(DL:ロケーションに依存;使用ロケーションでのオフセット)でアドレス指定し、操作画面を使用して入力します。

用途

使用するロケーションが原因で発生する寸法誤差は、追加オフセットで補正できます。

構文

DL=<番号>

意味

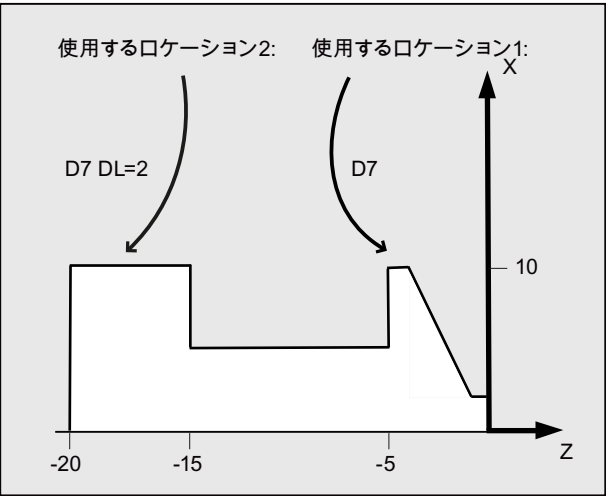
DL:	追加オフセットを起動する命令
<番号>:	有効にする追加工具オフセットデータを、<番号>パラメータで指定します。

注記

マシンデータを使用して、追加オフセットの数を定義し、さらにそれらのオフセットを起動します(→工作機械メーカーのデータに従っているか、十分に注意してください)。

例

次のように、2 個の軸受座に同じ刃先を使用します。



プログラムコード	コメント
N110 T7 D7	; リボルバがロケーション 7 に位置決めされます。D7 および DL=1 が有効になり、次のブロックで移動します。
N120 G0 X10 Z1	
N130 G1 Z-6	
N140 G0 DL=2 Z-14	; D7 の他に DL=2 も有効になり、次のブロックで移動します。
N150 G1 Z-21	
N160 G0 X200 Z200	; 工具交換点へのアプローチ
...	

4.13.2.2 摩耗値とセットアップ値の指定(\$TC_SCPxy[t,d]、\$TC_ECPxy[t,d])

4.13.2.3 追加オフセットの削除(DELDL)

DELDL 命令は、工具の刃先の追加オフセットを削除します(メモリ空間が解放されます)。定義した摩耗値とセットアップ値が両方とも削除されます。

構文

```
DELDL [<t>,<d>]
DELDL [<t>]
DELDL
<状態>=DELDL [<t>,<d>]
```

意味

DELDL:	追加オフセットを削除する命令
<t>:	工具の T 番号
<d>:	工具刃先の D 番号
DELDL [<t>,<d>]:	工具<t>の刃先<d>の追加オフセットをすべて削除します。
DELDL [<t>]:	工具<t>のすべての刃先の追加オフセットをすべて削除します。
DELDL:	工具オフセットユニットのすべての工具のすべての刃先の追加オフセットをすべて削除します(このチャンネルでこの命令をプログラム指令した場合)。

4.13 工具オフセット

<状態>:	削除のステータスです	
	値:	意味
	0	削除が正常に完了しました。
	-	オフセットが削除されていません(パラメータ設定で指定した工具刃先が 1 個のみの場合)か、または完全には削除されていません(パラメータ設定で指定した工具刃先が複数の場合)。

注記

動作中の工具の摩耗値とセットアップ値は削除できません(基本的に、D または工具データの削除動作と同じです)。

4.13.3 工具補正の特殊処理

工具長さと摩耗の符号の評価は、セッティングデータ SD42900～SD42960 で制御できます。

これは、ジオメトリ軸のミラーリング時または加工平面変更時の摩耗成分の動作、および工具方向の熱変位補正にも同様に適用されます。

磨耗値:

以下で摩耗値を参照する場合は、実際の摩耗値の合計(\$TC_DP12～\$TC_DP20)、摩耗値によるサムオフセット(\$SCPX3～\$SCPX11)、およびセットアップ値(\$ECPX3～\$ECPX11)として理解してください。

サムオフセットの詳細については、『機能マニュアル』、「工具管理」を参照してください。

セッティングデータ

SD42900 \$SC_MIRROR_TOOL_LENGTH	工具長成分のミラーリング、および工具ベース寸法の成分です。
SD42910 \$SC_MIRROR_TOOL_WEAR	工具長成分の摩耗値のミラーリングです。
SD42920 \$SC_WEAR_SIGN_CUTPOS	摩耗成分の符号を、刃先位置に応じて評価します。
SD42930 \$SC_WEAR_SIGN	摩耗寸法の符号を反転します。

SD42935 \$SC_WEAR_TRANSFORM	摩耗値を座標変換します。
SD42940 \$SC_TOOL_LENGTH_CONST	工具長成分をジオメトリ軸へ割り当てます。
SD42950 \$SC_TOOL_LENGTH_TYPE	工具タイプに依存しない工具長成分の割り当てをおこないます。
SD42960 \$SC_TOOL_TEMP_COMP	工具方向の熱変位補正值です。工具オリエンテーションをプログラム指令した場合も機能します。

詳細情報

変更されたセッティングデータの適用

上記のセッティングデータを変更すると、次に工具刃先が選択されるまで、工具成分が再評価されません。工具がすでに有効で、この工具のデータを再評価する場合は、工具を再度選択してください。

これは、得られる工具長が、軸のミラーリングの状態が変化のために変更された場合も同様に適用されます。ミラーリング命令の後には、変更された工具長成分を起動するために、工具を再度選択してください。

旋回工具ホルダと新しいセッティングデータ

セッティングデータ SD42900～SD42940 は、動作中の旋回工具ホルダの成分には影響を与えません。ただし、旋回工具ホルダの計算では常に、工具に対して、長さの合計結果は(工具長+摩耗+工具ベース寸法)になります。セッティングデータによる変更はすべて、長さの合計結果の計算に含まれます。つまり、旋回工具ホルダのベクトルは加工平面には影響されません。

注記

旋回工具ホルダの使用時には、多くの場合、これはミラーリングされない基本単位系のすべての工具の定義に役立ちます。これには、ミラーリングされる加工のみに使用される工具も含まれます。ミラーリングされた軸を使用して加工しているとき、工具ホルダは工具の実際の位置が正確に記述されるように回転します。すべての工具長成分は、セッティングデータを介した個々の成分評価の制御の必要なく、個々の軸のミラーリング状態に応じて、正しい方向で自動的に動作します。

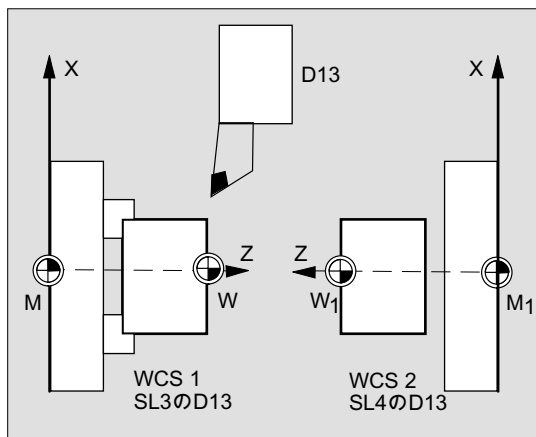
その他のオプション用途

旋回工具ホルダ機能の使用は、旋削工具の物理的なオプションが機械に存在しない場合にも便利です。これは、向きの異なる工具が取り外しできないように設置されている場合も同様です。工具のサイズ決めは、基本の向きで一様におこなうことができます。この場合、加工に関連する寸法は実際の工具ホルダに応じて計算されます。

4.13 工具オフセット

4.13.3.1 工具長のミラーリング

セッティングデータ SD42900 \$SC_MIRROR_TOOL_LENGTH と SD42910 \$SC_MIRROR_TOOL_WEAR をゼロに設定していない場合は、工具長成分と基本寸法の成分を、摩耗値とその関連軸を使用してミラーリングできます。

**SD42900 \$SC_MIRROR_TOOL_LENGTH**

ゼロ以外のセッティングデータ:

工具長成分(\$TC_DP3、\$TC_DP4、および\$TC_DP5)および基本寸法成分(\$TC_DP21、\$TC_DP22、および\$TC_DP23)を、その関連軸に対してミラーリングします。また、符号の反転によるミラーリングもおこないます。

摩耗値はミラーリングされません。摩耗値もミラーリングする場合は、セッティングデータ SD42910 \$SC_MIRROR_TOOL_WEAR を設定してください。

SD42910 \$SC_MIRROR_TOOL_WEAR

ゼロ以外のセッティングデータ:

工具長成分(その関連軸はミラーリングされています)の摩耗値も、符号の反転によりミラーリングされます。

4.13.3.2 摩耗の符号評価

セッティングデータ SD42920 \$SC_WEAR_SIGN_CUTPOS と SD42930 \$SC_WEAR_SIGN の設定がゼロ以外の場合は、摩耗成分の符号評価を反転できます。

SD42920 \$SC_WEAR_SIGN_CUTPOS

ゼロ以外のセッティングデータ:

当該の刃先位置を持つ工具(旋削工具と研削工具、工具タイプは 400)の場合は、加工平面の摩耗成分の符号評価は刃先位置により異なります。このセッティングデータは、当該の刃先位置のない工具タイプに対しては無意味です。

次の表では、SD42920(ゼロ以外)で符号を反転した寸法を、X で示しています。

刃先位置	長さ 1	長さ 2
1		
2		X
3	X	X
4	X	
5		
6		
7		X
8	X	
9		

注記

SD42920 と SD42910 による符号評価は、相互には影響しません。たとえば、両方のセッティングデータを使用して寸法の符号を変更した場合は、符号はそのまま変更されません。

SD42930 \$SC_WEAR_SIGN

ゼロ以外のセッティングデータ:

すべての摩耗寸法の符号を反転します。これは、工具の長さの他に、工具半径、ラウンディング半径などのその他の変数の両方に影響します。

正の摩耗寸法を入力した場合は、工具が「より短く」、「より細く」なります。「工具補正の特殊処理」、「変更されたセッティングデータの適用」の各章を参照してください。

4.13.3.3 動作中の加工運転の座標系(TOWSTD、TOWMCS、TOWWCS、TOWBCS、TOWTCS、TOWKCS)

機械のキネマティックス、または旋回工具ホルダの適用に応じて、これらの座標系のいずれかで計測した摩耗値を変換するか、または適切な座標系に座標変換します。

4.13 工具オフセット

動作中の加工運転の座標系

次の座標系でおこなわれる工具長補正を使用すると、対応するグループ 56 の G 命令で、動作中の工具に工具長摩耗成分を取り入れることができます。

- 機械座標系(MCS)
- 基本座標系(BCS)
- ワーク座標系(WCS)
- 工具座標系(TCS)
- キネマティックトランスフォーメーションの工具座標系(KCS)

構文

TOWSTD
TOWMCS
TOWWCS
TOWBCS
TOWTCS
TOWKCS

意味

TOWSTD:	工具長摩耗値のオフセットの初期設定値
TOWMCS:	MCS の工具長のオフセット
TOWWCS:	WCS の工具長のオフセット
TOWBCS:	BCS の工具長のオフセット
TOWTCS:	工具ホルダの基準点での工具長のオフセット(旋回工具ホルダ)
TOWKCS:	工具ヘッドの工具長のオフセット(キネマティックトランスフォーメーション)

詳細情報

機能の違い

次の表に、重要な機能の違いを示しています。

G 命令	摩耗値	有効な旋回工具ホルダ
TOWSTD	初期値、工具長	摩耗値が回転します。

TOWMCS	MCS の摩耗値です。旋回工具ホルダが動作中でない場合、TOWMCS は、TOWSTD と同じです。	得られる工具長のベクトルのみが回転し、摩耗は考慮されません。
TOWWCS	WCS のなかで摩耗値を MCS に変換します。	工具ベクトルを、摩耗を考慮せずに、TOWMCS として計算します。
TOWBCS	BCS のなかで摩耗値を MCS に変換します。	工具ベクトルを、摩耗を考慮せずに、TOWMCS として計算します。
TOWTCS	ワーク座標系のなかで摩耗値を MCS に変換します。	工具ベクトルを、摩耗を考慮せずに、TOWMCS として計算します。

TOWWCS、TOWBCS、TOWTCS:摩耗値が工具ベクトルに加算されます。

直線的座標変換

MCS のなかで定義した工具長が意味を持つのは、BCS からの直線座標変換によって MCS を生成した場合のみです。

非直線座標変換

たとえば、TRANSMIT による非直線座標変換が動作中の場合に、要求する座標系として MCS を指定すると、自動的に BCS を使用します。

キネマティックトランスフォーメーションが無効で、旋回工具ホルダが無効の場合

キネマティックトランスフォーメーションが無効で、旋回工具ホルダも無効の場合は、(WCS を除く)他の 4 つの座標系がすべて統合されます。この場合のみ、WCS が他の座標系と異なります。評価の必要があるのは工具長のみであるため、各座標系間の平行移動は無関係です。

計算に含める摩耗値

セッティングデータ **SD42935 \$SC_WEAR_TRANSFORM** は、次の 3 つの摩耗成分のどれが、回転の対象となるかを定義します。

- 磨耗
- サムオフセット(仕上げ)
- 荒削りサムオフセット

4.13 工具オフセット

この回転では、アダプタの座標変換、または以下の G 命令のいずれかが有効な場合には旋回工具ホルダを使用します。

- TOWSTD
基本設定です。工具長の補正の場合。
- TOWMCS
機械座標系(MCS)内の摩耗値です。
- TOWWCS
ワーク座標系(WCS)内の摩耗値です。
- TOWBCS
基本座標系(BCS)の摩耗値です。
- TOWTCS
工具ホルダ固定部での工具座標系内の摩耗値(工具ホルダレファレンス点 T)です。
- TOWKCS
キネマティックトランスフォーメーションの工具ヘッドの座標系の摩耗値です。

注記

個々の摩耗成分の評価(ジオメトリ軸への割り当て、符号評価)は、次の要素に影響を受けます。

- 有効な平面
 - アダプタ座標変換
 - セッティングデータ:
 - SD42910 \$SC_MIRROR_TOOL_WEAR
 - SD42920 \$SC_WEAR_SIGN_CUTPOS
 - SD42930 \$SC_WEAR_SIGN
 - SD42940 \$SC_TOOL_LENGTH_CONST
 - SD42950 \$SC_TOOL_LENGTH_TYPE
-

4.13.3.4 工具長と平面変更

セッティングデータ SD42940 \$SC_TOOL_LENGTH_CONST の設定がゼロ以外の場合は、平面の変更時に、長さ、摩耗、および基本寸法などの工具長成分を、旋削工具と研削工具のジオメトリ軸に割り当てることができます。

SD42940 \$SC_TOOL_LENGTH_CONST

ゼロ以外のセッティングデータ:

工具長成分(長さ、摩耗、および工具ベース寸法)のジオメトリ軸への割り当ては、加工平面の変更時(G17～G19)には変わりません。

次の表は、旋削工具と研削工具(工具タイプ 400～599)のジオメトリ軸への工具長成分の割り当てを示します。

内容	長さ 1	長さ 2	長さ 3
17	Y	X	Z
*)	X	Z	Y
19	Z	Y	X
-17	X	Y	Z
-18	Z	X	Y
-19	Y	Z	X

*) 各値が 0 以外で、6 個のリスト値のいずれにも等しくない場合は、値は 18 として使用します。

次の表は、他のすべての工具(工具タイプ< 400 または> 599)に対する工具長成分のジオメトリ軸への割り当てを示しています。

動作平面	長さ 1	長さ 2	長さ 3
*)	Z	Y	X
18	Y	X	Z
19	X	Z	Y
-17	Z	X	Y
-18	Y	Z	X
-19	X	Y	Z

*) 各値が 0 以外で、6 個のリスト値のいずれにも等しくない場合は、値は 17 として使用します。

注記

この表では、3 つまでのジオメトリ軸を X、Y、Z に指定していることが前提です。補正と軸の割り当てを特定するのは、軸識別子ではなく、軸の順序です。

4.13 工具オフセット

4.13.4 オンライン工具補正

4.13.4.1 多項式機能(FCTDEF)の定義

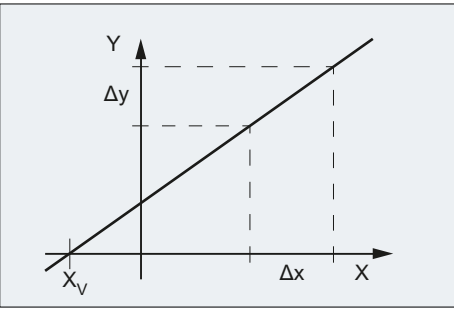
特定の目立て方法(目立てローラなど)には、目立てローラを送り込むと研削砥石の半径が連続して(直線的に)縮小するという特徴があります。この方法では、目立てローラの切り込みとそれぞれの長さの磨耗値の書き込みの間に一次関数が必要となります。直線機能は、予約手順 FCTDEF(...)を使用して三次多項式機能まで定義されます。

直線方程式

$y = f(x) = a_0 + a_1 \cdot x_1$

a_1 :直線の傾き、 $a_1 = \Delta x / \Delta y$

a_0 : $a_0 = -a_1 \cdot X_v$ の場合の X 軸に沿った直線の移動



構文

FCTDEF (<Func>,<LLimit>,<ULimit>,<a0>,<a1>,<a2>,<a3>)

意味

FCTDEF (...):	PUTFTOCF(...)の多項式機能の定義: $y = f(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3$	
<Func>:	ファンクション番号	
	データタイプ:	INT
	値の範囲:	1, 2, 3
<LLimit>:	下限値	
	データタイプ:	REAL
<ULimit>:	上限値	
	データタイプ:	REAL

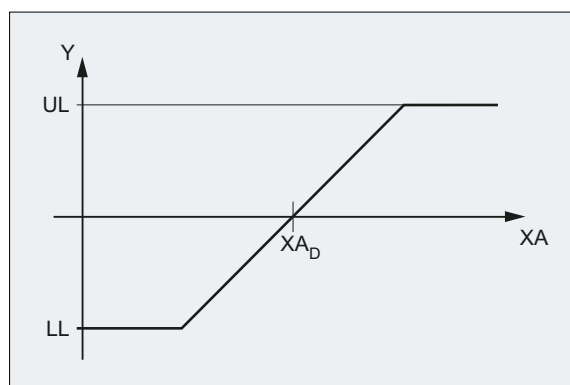
<a0>,<a1>,<a2>,<a3>:	多項式関数の係数です。	
	データタイプ:	REAL

例

定義

- ファンクション番号:1
- 下限値および上限値: -100、100
- 特性の変化率: $a_1 = 1$
- 動作点は特性の中心に位置する必要があります。機能が NC プログラムで定義される場合の WCS の軸 XA のセットポイント位置に基づいて、特性を負の Y 方向に移動してください。 $a_0 = -a_1 * XA_D = -1 * \AA_IW
- $a_2 = a_3 = 0$

Characteristic



UL 上限値

LL 下限値

XA_D 機能が NC プログラムで定義されている場合の軸 XA のセットポイント

プログラミング

プログラムコード	コメント
FCTDEF(1,-100,100,-\$AA_IW[XA],1)	; 機能の定義

4.13 工具オフセット

4.13.4.2 オンライン工具補正の連続書き込み(PUTFTOCF)

予約手順 PUTFTOCF(...)を使用して、オンライン工具補正が FCTDEF(...) (ページ 860) で以前に定義した多項式機能に基づいて実行されます。

注記

オンライン工具オフセットをシンクロナイズドアクションを使用して実現することもできます。

詳細情報については、『ファンクションマニュアル』の「シンクロナイズドアクション」を参照。

構文

PUTFTOCF (<Func>, <RefVal>, <ToolPar>, <Chan>, <Sp>)

意味

PUTFTOCF (...):	FCTDEF(...)で定義した多項式機能を使用して、オンライン工具補正をブロックごとに連続書き込み	
<Func>:	FCTDEF(...)による機能定義で定義された機能番号	
	データタイプ:	INT
	値の範囲:	1, 2, 3
<RefVal>:	オフセットの算出基準となる指令値です(軸のセットポイントなど)。	
	データタイプ:	VAR REAL
<ToolPar>:	オフセット値がインクルードされる摩耗パラメータ(長さ 1、2、または 3)の番号です。	
	データタイプ:	INT
<Chan>:	オンライン工具補正が有効となるチャンネルの番号です。 注記: オフセットが有効なチャンネルで無効な場合にのみ必要です。	
	データタイプ:	INT

<Sp>:	オンライン工具補正が有効となる主軸の番号です。 注記: オフセットが現在使用されている有効な工具ではなく、無効なといしに適用される場合にのみ必要です。	
	データタイプ:	INT

4.13.4.3 オンライン工具補正の書き込み、個別(PUTFTOC)

機能

予約手順 PUTFTOC(...)を使用して、オンライン工具補正が固定のオフセット値に基づいて実行されます。

構文

PUTFTOC (<CorrVal>,<ToolPar>,<Chan>,<Sp>)

意味

PUTFTOC (...):	オンライン工具補正の書き込み	
<CorrVal>:	摩耗パラメータに追加されるオフセット値。	
	データタイプ:	REAL
<ToolPar>:	オフセット値がインクルードされる摩耗パラメータ(長さ 1、2、または 3)の番号です。	
	データタイプ:	INT
<Chan>:	オンライン工具補正が有効となるチャンネルの番号です。 注記: オフセットが有効なチャンネルで無効な場合にのみ必要です。	
	データタイプ:	INT

4.13 工具オフセット

<Sp>:	オンライン工具補正が有効となる主軸の番号です。	
	注記: オフセットが現在使用されている有効な工具ではなく、無効なといしに適用される場合にのみ必要です。	
	データタイプ:	INT

4.13.4.4 オンライン工具オフセットの起動/解除(FTOCON/FTOCOF)

オンライン工具補正は、G 命令 FTOCON および FTOCOF を使用して有効または無効にできます。

構文

```
FTOCON
...
FTOCOF
```

意味

FTOCON:	オンライン工具補正を起動します 命令は、オンライン工具補正が有効になるチャンネルでプログラミングしてください。
FTOCOF:	オンライン工具長補正の解除 命令は、オンライン工具補正が無効になるチャンネルでプログラミングしてください。 注記: FTOCOF では、軸は工具オフセットを超えて移動しません。ただし、PUTFTOC/PUTFTOCF で計算された値は切削固有オフセットデータに残ります。 オンライン工具補正を最終的に無効にするには、工具(T...)を FTOCOF 後にもう一度選択/選択解除してください。

4.13.5 3次元工具径補正

4.13.5.1 3次元外周加工用の3次元工具径補正(CUT3DC、CUT3DCD、ISD)

3次元外周加工用の3次元工具径補正(3D TRC)は、限界面を考慮しない場合、モーダルで有効なG命令、CUT3DCまたはCUT3DCDを使用して選択されます。

実際の有効化は、G41またはG42を使用して実施されます。工具径補正はG40によって無効化されます。

構文

```
G41/G42 ORIC/ORID ISD=...CUT3DC/CUT3DCD CDOF2 X...Y...Z...
...
G40 X...Y...Z...
```

意味

CUT3DC:	外周加工用の3D TRC(5軸座標変換が有効な場合のみ)	
CUT3DCD:	標準工具とは違う外周加工用の工具を基準にした3D TRC(5軸座標変換が有効な場合のみ) 半径差は工具パラメータ\$TC_DP15によって指定されます。	
G41/G42 X...Y...Z...:	工具径補正の有効化	
	G41:	輪郭の左側の工具径補正
	G42:	輪郭の右側の工具径補正
	注記: 有効化は線形ブロック(G0/G1)で実施する必要があります。	
CDOF2:	3次元外周加工用の衝突検出の無効化	
ORIC/ORID:	外側コーナーでの向きの変更の動作が、G命令ORICおよびORIDによって指定されます。	
	ORIC:	外側コーナーでの向きの変更を、挿入する円弧ブロックに重畳します。
	ORID:	外側コーナーでの向きの変更が、挿入する円弧ブロックの前に実行されます。

4.13 工具オフセット

ISD=<値>:	ISD アドレスを使用すると、外周加工と 有効な 3 次元 工具径補正について、工具の切り込み深さを変更できます。	
	<値>:	切り込み深さの長さ
G40 X...Y...Z...:	工具径補正の無効化 注記: 無効化は線形ブロック(G0/G1)でジオメトリ軸の移動によって実施する必要があります。	

注記

3D TRC を選択するための G 命令はアプローチブロックで、つまり通常は G41 または G42 を含むブロックで使用されます。

G41 または G42 は、補正に関連するジオメトリ軸の移動動作なしで、ブロックでプログラム指令することもできます。この場合、アプローチブロックはそのようなブロックに続く最初の移動ブロックとなります。

有効な工具径補正による 3D TRC タイプの変更は、アラームなしで無視されます。

例

プログラムコード	コメント
\$TC_DP1[1,1] = 120	; 工具 D1 の定義:
\$TC_DP3[1,1] = 20	; タイプ (エンドミル)
\$TC_DP6[1,1]=8	; 長さ補正ベクトル
	; 半径
N10 X0 Y0 Z0 T1 D1 F12000	; 工具の選択。
N20 TRAORI(1)	; 座標変換の起動。
N30 G42 ORIC ISD=10 CUT3DC G64 X30	; 3 次元外周加工の起動、
	; 外側コーナーでの連続的な向きの変更、
	; 切り込み深さ:10 mm
N40 ORIWKS A30 B15	; 軸位置の指定によるコーナーでの向きの変更。
N50 Y20 A3=1 C3=1	; 向きが変更される移動ブロック、
	; 方向ベクトルによる向きの指定。
N60 X50 Y30	; 一定の向きの移動ブロック、
N70 Y50 A3=0.5 B3=1 C3=5	; 向きが変更される移動ブロック。
N80 M63	; 移動情報のないブロック。
N90 X0 ISD=20	; 切り込み深さが変更される移動ブロック。
N100 G40 Y0	; 工具径補正を解除。
N110 M30	

詳細情報

経路と向き

ここでは、軌跡(主輪郭)、および対応する向きを定義して、使用する外周加工のタイプを決めます。この加工タイプでは、軌跡上の工具の形状は関係ありません。工具の接触点の半径のみによって決まります。

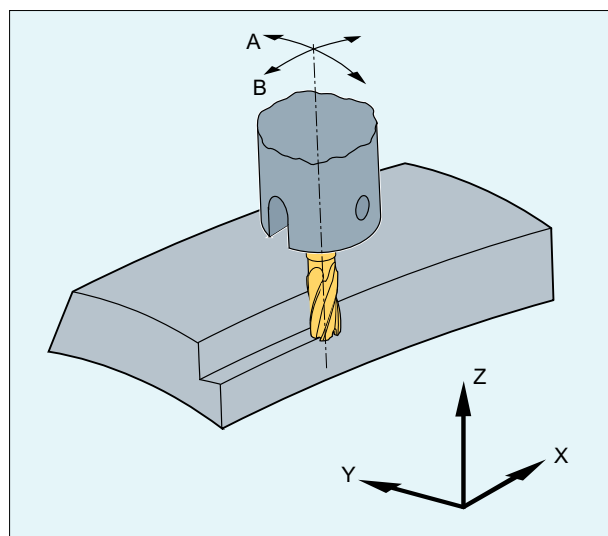


図 4-9 外周削り

アプローチ動作

アプローチ動作は常に 3 次元タイプの工具径補正の基本となります。

外側コーナーでの動作

グループ 18 の G 命令(コーナーでの動作、工具オフセット)は、3D TRC による外周加工に関して、2½D TRC の条件と同じ方法で使用されます。

- G450:挿入円(工具がワークのコーナーを、円弧軌跡を描いて回ります)
2½D TRC のソリューションとは逆に、外側コーナーに挿入される輪郭要素は常に半径 0 の円弧であり、そこで工具径補正は他のプログラムで指令された経路と同じように動作します。円弧の代わりに円錐を挿入することはできません。この場合、DISC アドレスは重要でないため、使用されません。
- G451:等間隔の軌跡の交点(工具がワークのコーナーから後退します)
交点は、関連する 2 ブロックのオフセット曲線を延長し、かつ工具オリエンテーションに垂直な平面上のコーナーで 2 ブロックの交点を定義して特定します。

交点の処理(G451)は、工具の向きの変更を含む少なくとも 1 つのブロックが、関連する移動ブロックの間に挿入された場合は使用されません。この場合、円弧は常にコーナーに挿入されます。

4.13 工具オフセット

外側コーナーでの向きの変更の動作

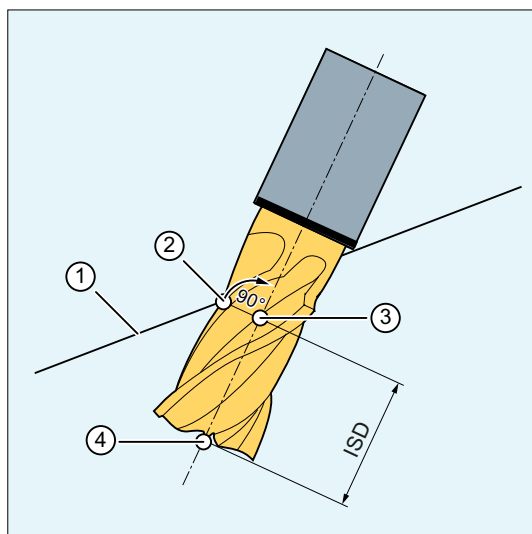
ORIC と ORID G 命令を使用して、コーナーを形成する 2 つのブロックとの間にプログラム指令した向きの変更を、挿入円弧ブロックの前に処理する(ORID)か、円弧ブロックと同時に処理する(ORIC)かを特定します。

切り込み深さ

フライス工具の切り込み深さとは、工具の先端からフライス工具のレファレンス点までの距離です。

フライス工具のレファレンス点は、プログラムされた軌跡上のフライス工具加工点を、工具の長手軸に垂直に投影した点です。

工具の円筒面の加工点位置は、切り込み深さとともに設定されます。



- ① プログラム指令軌跡
- ② フライス工具の加工点
- ③ フライス工具基準点
- ④ フライス工具の先端

ISD 切り込み深さ(InSersion Depth)

図 4-10 切り込み深さ

標準工具とは違う工具を基準にした工具径補正

標準工具とは違う工具を基準にした外周加工用の 3D TRC は、CUT3DCD 命令によって選択されます。この工具径補正は、プログラムされた輪郭が標準工具の中心点軌跡を基準にしていて、標準工具とは違う工具以外の工具が加工に使用されている場合に適用します。3次元工具径補正を計算する場合は、有効な工具の半径の摩耗値(\$TC_DP15)とプログラム

されている工具オフセット OFFN および TOFFR/TOFFLR のみが考慮されます。動作中の工具の基本半径(\$TC_DP6)は考慮されません。

CUT3DC を使用した、外周削りの傾斜側壁のポケット加工

この 3 次元工具径補正では、加工する表面の法線方向の切り込みにより、フライス半径の差を補正します。フライス工具面が位置する平面は、挿入深さ ISD がそのまま同じ場合は、そのまま変更されません。たとえば、標準工具より半径の小さいフライス工具は、限界面でもあるポケットの底には届きません。自動工具切り込みの場合は、この限界面を制御装置が認識している必要があります。「3 次元外周加工、限界面の考慮(CUT3DCC、CUT3DCCD) (ページ 875)」の章を参照してください。

Advanced Surface / Top Surface

注記

工具径補正 CUT3DCD を「Advanced Surface」または「Top Surface」オプション(ライセンスが必要)と組み合わせて適用する場合、「Advanced Surface」/「Top Surface」の推奨設定を遵守してください。

設定データの確認用の特別なテストプログラムが SIOS ポータルに設けられています。

→ Advanced Surface/Top Surface (<https://support.industry.siemens.com/cs/ww/en/view/109738423>)用のテストプログラム

4.13.5.2 3 次元正面削りのための 3 次元工具径補正の選択(CUT3DF、CUT3DFS、CUT3DFF、CUT3DFD)

3 次元正面削りのための 3 次元工具径補正(3D TRC)は、モーダルで有効な G 命令 CUT3DF、CUT3DFS、CUT3DFF または CUT3DFD によって選択されます。

実際の有効化は、G41 または G42 を使用して実施されます。

3 次元正面削りでは、工具径補正を計算するため、加工面の面法線を定義する必要があります。これは G41 または G42 によって、A4、B4、C4 および A5、B5、C5 アドレスを介して行う必要があります。

工具径補正は G40 によって無効化されます。

構文

```
G41/G42 ORIC/ORID CUT3DF/CUT3DFS/CUT3DFF/CUT3DFD
X...Y...Z...A4=...B4=...C4=...A5=...B5=...C5=...
...
G40 X...Y...Z...
```

4.13 工具オフセット

意味

CUT3DFS:	向きが一定の正面削りの 3D TRC。工具オリエンテーションは、G17～G19 で定義され、フレームには影響されません。	
CUT3DFF:	向きが一定の正面削りの 3D TRC。工具オリエンテーションは、G17～G19 で定義した方向で、場合によっては、フレームによって回転します。	
CUT3DF:	向きが変わる正面削りの 3 次元工具補正(動作中の 5 軸座標変換の場合のみ)。	
CUT3DFD:	標準工具とは違う向きが変わる正面削り用の工具を基準にした 3D TRC (5 軸座標変換が有効な場合のみ) 半径差は工具パラメータ \$TC_DP15 によって指定されます。 注記: CUT3DFD は「3 次元正面削りの面法線のスムーズ化」と組み合わせた場合のみ可能です。これは「Top Surface」機能(要ライセンス)を CYCLE832(...)から呼び出すことによって有効になります。	
G41/G42 X...Y...Z...:	工具オフセットの有効化 3 次元正面削りでは G41 と G42 による動作は同じです。 注記: 有効化は線形ブロック(G0/G1)で実施する必要があります。	
A4/5=...B4/5=...C4/5=...:	加工面の面法線の定義	
	A4=...B4=...C4=...: ..:	ブロック始点での定義
	A5=...B5=...C5=...: ..:	ブロック終点での定義

4.13 工具オフセット

ORIC/ORID:	外側コーナーでの向きの変更の動作が、G 命令 ORIC および ORID によって指定されます。	
	ORIC:	外側コーナーでの向きの変更を、挿入する円弧ブロックに重畳します。
	ORID:	円弧ブロックの前に向きの変更を実行します
G40 X...Y...Z...:	工具径補正の無効化 注記: 無効化は線形ブロック(G0/G1)でジオメトリ軸の移動によって実施する必要があります。	

注記

G41 または G42 は、補正に関連するジオメトリ軸の移動動作なしで、ブロックでプログラム指令することもできます。この場合、アプローチブロックはそのようなブロックに続く最初の移動ブロックとなります。

有効な工具径補正による 3D TRC タイプの変更は、アラームなしで無視されます。

例

例 1: CUT3DF による 3 次元正面削り

プログラムコード	コメント
N10	; 工具 D1 の定義:
N20 \$TC_DP1[1,1]=121	; 工具タイプ (トラス形フライス工具)
N30 \$TC_DP3[1,1]= 20	; 長さ補正
N40 \$TC_DP6[1,1]=5	; 半径
N50 \$TC_DP7[1,1]=3	; 丸み付け半径
N60	
N70	
N80 X0 Y0 Z0 A0 B0 C0 G17 T1 D1 F12000	; 工具の選択。
N90 TRAORI (1)	; 方向座標変換を選択。
N100 B4=-1 C4=1	; 平面の定義。
N110 G41 ORID CUT3DF G64 X10 Y0 Z0	; 工具オフセットの有効化。
N120 X30	
N130 Y20 A4=1 C4=1	; 外側コーナー、新しい平面定義。
N140 B3=1 C3=5	; ORID による向きの変更。
N150 B3=1 C3=1	; ORID による向きの変更。
N160 X-10 A5=1 C5=2 ORIC	
N170 A3=-2 C3=1	; ORIC による向きの変更。
N180 A3=-1 C3=1	; ORIC による向きの変更。

4.13 工具オフセット

プログラムコード	コメント
N190 Y-10 A4=-1 C4=3	; 新しい平面定義。
N200 X-20 Y-20 Z10	; 前のブロックがある内側コーナー。
N210 X-30 Y10 A4=1 C4=1	; 内側コーナー、新しい平面定義。
N220 A3=1 B3=0.5 C3=1.7	; ORIC による向きの変更。
N230 X-20 Y30 A4=1 B4=-2 C4=3 ORID	
N240 A3 = 0.5 B3=-0.5 C3=1	; 向きの変更。
N250 X0 Y30 C4=1	; 経路の動き、新しい平面、
	; 相当するプログラミングによる向き。
N260 BSPLINE X20 Z15	; スプライン開始、向きの相当するプログラミング
N270 X30 Y25 Z18	; スプライン中も有効であり続ける。
N280 X40 Y20 Z13	
N290 X45 Y0 PW=2 Z8	
N300 Y-20	
N310 G2 ORIMKS A30 B45 I-20 X25 Y-40 Z0	; ヘリカル、軸プログラミングによる向き。
N320 G1 X0 A3=-0.123 B3=0.456 C3=2.789 B4=-1 C4=5 B5=-1 C5=2	; 経路動作、向き、一定でない平面。
N330 X-20 G40	; 工具径補正を解除。
N340 M30	

例 2:CUT3DFD によって CAD システムから生成された NC プログラム(セクション)

プログラムコード	コメント
N01 G710	
N03 T="12"	
N06 S5305 M03	
N07 G642	
; 工具長さを考慮しながら、MCS の起点にアプローチします。	
G00 G90 X-250.62787 Y-38.37944 A=DC(253.12719) B-12.49543	
G00 G90 Z251.80052	
; MCS での位置決めの終了。	
;	
TRAORI(1)	; 方向座標変換を選択。
G500	
D1	
CYCLE832(0.01, _TOP_SURFACE_SMOOTH_ON + _ORI_FINISH, 1)	; CYCLE832 を呼び出し: ; 輪郭許容範囲 = 0.01 mm、 ; 処理タイプ: スムージングを行う最上面、 ; 旋回の許容範囲の入力による仕上げ ; 向きの許容範囲 = 1 度
CUT3DFD	
N08 G90 G94	
N09 G00 X-269.21195 Y128.32027 Z1.18577 A3=-.216361688 B3=.934284397 C3=-.283373051	; ブロック N09～N10 は、一定の向きのワークへのアプローチ動作 の高速部位です。

4.13 工具オフセット

プログラムコード	コメント
N10 G00 X-251.90301 Y53.57752 Z23.85561	
N11 G01 X-247.57578 Y34.89183 Z29.52308 F50000.00000	; ブロック N11～N21 では、アプローチ動作の低速部位が生成されます。工具はすでにワークの近くにあり、さらに、CYCLE832 からの金型加工設定 (COMPSURF など) が有効になっています (有効な G01 による)。金型加工動作のための、このいわゆる過渡段階の経路は、輪郭許容範囲の約 1000 倍としてください (この例では 10 mm)。
N12 X-247.69126 Y33.82182 Z24.78219 F1061.00000	
N13 X-247.76560 Y33.13299 Z21.73022	
N14 X-247.82755 Y32.55897 Z19.18691	
N15 X-247.87918 Y32.08062 Z17.06748	
N16 X-247.92220 Y31.68200 Z15.30129	
N17 X-247.95805 Y31.34981 Z13.82947	
A3=-.216361686 B3=.934284391 C3=-.283373071	
N18 X-247.98792 Y31.07299 Z12.60295	
A3=-.216360662 B3=.934280801 C3=-.283385691	
N19 X-248.01282 Y30.84230 Z11.58085	
A3=-.216336015 B3=.934194446 C3=-.283689030	
N20 X-248.03357 Y30.65006 Z10.72910	
A3=-.216233089 B3=.933833626 C3=-.284952647	
N21 X-248.05086 Y30.48986 Z10.01931	
A5=-.060687572 B5=.974940255 C5=-.214029243	
A3=-.215712821 B3=.932005189 C3=-.291263295	
N22 G41 X-248.06237 Y30.32400 Z9.36695	; N22 から、面法線は初めてブロック全体で完全に定義されます (つまり、前のブロック N21 の終点に面法線が存在し、そのため、ブロック N22 の始点での面法線と、ブロック N22 の終点での面法線も存在します)。したがって、G41/G42 による工具オフセットの切り替えのための前提条件が満たされました。
A5=-.060431854 B5=.973045457 C5=-.222554556	
A3=-.214974689 B3=.929398552 C3=-.300007025	
F1061.03295	
N23 X-248.07130 Y30.15119 Z8.71082	
A5=-.060165696 B5=.971048883 C5=-.231179920	
A3=-.214177198 B3=.926684940 C3=-.308841625	
N24 X-248.07829 Y29.97126 Z8.05094	
A5=-.059884286 B5=.968941717 C5=-.239928784	
A3=-.213318480 B3=.923853466 C3=-.317789237	
N25 X-248.08317 Y29.78487 Z7.38844	
A5=-.059584206 B5=.966718449 C5=-.248807482	
A3=-.212397895 B3=.920898045 C3=-.326854594	
N26 X-248.08578 Y29.59254 Z6.72679	
A5=-.059263963 B5=.964380907 C5=-.257793037	
A3=-.211418355 B3=.917822366 C3=-.336012474	
...	

注記

CUT3DFD による 3 次元正面削りでは、G41/G42 による工具オフセットを有効化するため、面法線の定義が必要です。面法線を定義せずに G41/G42 をプログラム指令すると、アラームが出力されます。

下記も参照

CYCLE832 - 高速設定 (ページ 1313)

詳細情報

3 次元正面削り

このタイプの 3 次元加工の場合は、ワーク表面の 3 次元軌跡を行単位で記述してください。通常は CAM で実行される計算では、工具の形状と寸法が考慮されます。ポストプロセッサが、NC ブロックの他に、工具の向き(動作中の 5 軸座標変換用)、および必要な 3 次元工具補正の G 命令を、パートプログラムに書き込みます。これは、機械のオペレータが、NC 軌跡の計算に使用する工具とは違った、少し小さい工具を使用することができることを意味します。

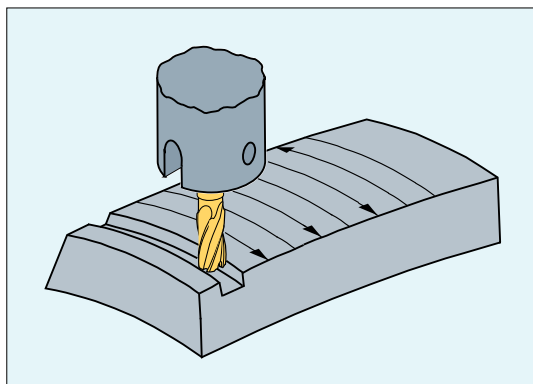


図 4-11 正面削り

アプローチ動作

アプローチ動作は常に 3 次元タイプの工具径補正の基本となります。

外側コーナーでの動作

外側コーナーは正面削りのための半径ゼロの円弧として扱われます。それにより、円弧面は最初のブロックの終端タンジェントから 2 番目のブロックの開始タンジェントまで伸びます。このようにして、ブロック移行中に向きを変えることができます。そのため、円弧は常に外側コーナーに輪郭要素として挿入されます。正面削りでは交点手順はありません。

外側コーナーでの向きの変更の動作

ORIC と ORID G 命令を使用して、コーナーを形成する 2 つのブロックとの間にプログラム指令した向きの変更を、挿入円弧ブロックの前に処理する(ORID)か、円弧ブロックと同時に処理する(ORIC)かを特定します。

標準工具とは違う工具を基準にした工具径補正

標準工具とは違う工具を基準にした 3 次元工具径補正は、CUT3DFD 命令を使用して選択します。この工具径補正は、プログラムされた輪郭が標準工具の中心点軌跡を基準にしている、標準工具とは違う工具以外の工具が加工に使用されている場合に適用します。3 次元工具径補正を計算する場合は、有効な工具の半径の摩耗値(\$TC_DP15)とプログラムされている工具オフセット OFFN および TOFFR/TOFFLR のみが考慮されます。動作中の工具の基本半径(\$TC_DP6)は **考慮されません**。

CUT3DFD による 3 次元正面削りは、「3 次元正面削りの面法線のスムーズ化」と組み合わせた場合のみ可能です。これは「Top Surface」機能(要ライセンス)を CYCLE832(...)から呼び出すことによって有効になります。有効化は、G41/G42 によって工具オフセットを有効にする **前に** 実施する必要があります。ただし、工具の接触点の直前ではなく、その経路長さ 1 つ前に実施します。これは輪郭許容範囲の約 1000 倍に対応します(例えば 1000 x 0.01 mm = 10 mm)。無効化は逆に順番で行ってください。まず G40 によって工具オフセットをオフにし、その後輪郭許容範囲の約 1000 倍に対応する経路長さの後で、CUT2D (または類似のもの)によって無効化します。

「3 次元正面削りの面法線のスムーズ化」を使用できるようにするには、「多項式による面法線の補間」機能も有効にする必要があります。

MD28291 \$MC_MM_SMOOTH_SURFACE_NORMALS = TRUE

注記

CUT3DFD による「Top Surface」と組み合わせた 3 次元正面削りでは、「Top Surface」に関連する推奨設定を遵守してください。

設定データの確認用の特別なテストプログラムが SIOS ポータルに設けられています。

→ Top Surface 用のテストプログラム (<https://support.industry.siemens.com/cs/ww/en/view/109738423>)

下記も参照

CYCLE832 - 高速設定 (ページ 1313)

4.13.5.3 3 次元外周加工、限界面の考慮(CUT3DCC、CUT3DCCD)

工具オリエンテーションが連続して、または絶えず変化する 3 次元外周削りでは、多くの場合、定義した標準工具に対して工具中心点軌跡をプログラム指令します。実際には、適切な標準工具を使用できない場合が多いため、標準工具との差が小さい工具($\leq 5\%$)を使用できます。

4.13 工具オフセット

CUT3DCCD は、プログラム指令した標準工具の記述とは違う実際の工具の限界面を考慮します。NC プログラムは、標準工具の中心点軌跡を記述します。

円筒工具を使用した場合の CUT3DCC では、プログラム指令した標準工具が到達する限界面が考慮されます。NC プログラムは、加工面の輪郭を記述します。

限界面の面法線ベクトルは、3 次元正面削りの A4、B4、C4 および A5、B5、C5 によって指定されます。

構文

```
G41/G42 CUT3DCCD/CUT3DCC CDOF2 X...Y...Z...A4=...B4=...C4=...A5=...B5=...C5=...
...
G40 X...Y...Z...
```

意味

CUT3DCCD:	工具中心点軌跡で標準工具とは違う工具を使用した、限界面を考慮した外周加工のための 3D TRC:限界面への切り込み	
CUT3DCC:	3 次元径補正を使用した、限界面を考慮した外周加工のための 3D TRC:加工面の輪郭	
G41/G42 X...Y...Z...:	工具径補正の有効化	
	G41:	輪郭の左側の工具径補正
	G42:	輪郭の右側の工具径補正
	注記: 有効化は線形ブロック(G0/G1)で実施する必要があります。	
CDOF2:	3 次元外周加工用の衝突検出の無効化	
A4/5=...B4/5=...C4/5 =...:	限界面の面法線の定義	
	A4=...B4=...C4=... :	ブロック始点での定義
	A5=...B5=...C5=... :	ブロック終点での定義
G40 X...Y...Z...:	工具径補正の無効化 注記: 無効化は線形ブロック(G0/G1)でジオメトリ軸の移動によって実施する必要があります。	

注記

3D TRC を選択するための G 命令はアプローチブロックで、つまり通常は G41 または G42 を含むブロックで使用されます。

G41 または G42 は、補正に関連するジオメトリ軸の移動動作なしで、ブロックでプログラム指令することもできます。この場合、アプローチブロックはそのようなブロックに続く最初の移動ブロックとなります。

有効な工具径補正による 3D TRC タイプの変更は、アラームなしで無視されます。

例

プログラムコード	コメント
N10 \$TC_DP1[1,1]=120	; 円筒フライス工具
N20 \$TC_DP6[1,1]=10	
N30 \$TC_DP15[1,1]=-3	
...	
; 円筒フライス工具と CUT3DCCD による処理	
N110 TRAORI	; 座標変換の起動。
N120 A4=0 B4=0 C4=1	; ブロック始点での限界面の面法線の定義。
N130 X0 Y0 Z0 A0 C0 T1 D1 F20000	
N140 X10 Y0 Z0 G41 CUT3DCCD CDOF2 G64	; 限界面を考慮して、衝突検出をオフにした 3 次元外周加工の有効化。
N150 X20	
N160 X30 A45	; 鈍角 ==> 切り込みなし
N170 X40 A-45	; 鋭角 ==> 切り込みあり
N180 X55	
N190 Y10 Z10	; 工具方向の移動。
N200 Y20	
N210 C45	; 向きの純粋な変更。
N220 Y30 C90	
N230 A5=-1 B5=0 C5=2 Y40	; 表面の変更。
N240 Y50 G40	; 工具径補正を解除。
...	

詳細情報**工具タイプ**

工具タイプ(工具パラメータ\$TC_DP1)を使用します。シャンクが円筒形のタイプのフライス工具(円筒またはエンドミル、トーラス形フライス工具、限られた状況においては、ボールエンドミル)のみを使用できます。これは番号 111 および 155～157 を除く工具タイプ 1～399 に対応します。

4.13 工具オフセット

コーナ R ありの標準工具

標準工具によるコーナ R は、工具パラメータ\$TC_DP7 で定義します。工具パラメータ\$TC_DP16 は、標準工具との比較による実際の工具のコーナ R の誤差を表わします。

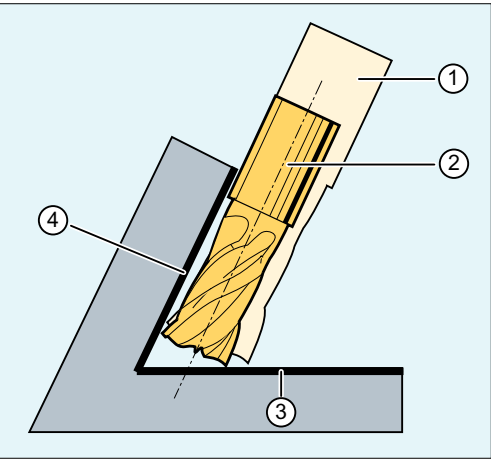
例:標準工具に比べて半径の小さいトーラス形フライス工具

工具タイプ	軸半径(R)	コーナ半径(r)
コーナ R のある標準工具	$R = \$TC_DP6$	$r = \$TC_DP7$
コーナの丸み付けのある実際の工具 工具タイプ 121 と 131 のトーラス形フライス工具 (コーナの丸み付けのあるエンドミル)	$R' = \$TC_DP6 + \$TC_DP15 + OFFN$	$r' = \$TC_DP7 + \TC_DP16

この例では、\$TC_DP15 + OFFN と\$TC_DP16 はいずれも負です。

CUT3DCCD による 3D TRC:限界面までの切り込みによる工具中心点軌跡

該当する標準工具より半径の小さい工具を使用した場合は、長手方向に切り込むフライス工具を使用して、ポケットの底(ベース)に達するまで加工を続行します。工具は、加工面と限界面が成すコーナから、できるだけ切り屑を取り除きます。これは、外周削りと正面削りを組み合わせた加工タイプでおこなわれます。半径が小さい工具と同じように、半径の大きな工具の場合は、逆方向に切り込みます。



- ① 標準工具
- ② 半径が小さい工具での限界面までの切り込み
- ③ 限界面
- ④ 加工面

G グループ 22 の他のすべての工具補正とは逆に、CUT3DCCD に指定した工具パラメータ \$TC_DP6 は、工具半径には関係がなく、補正結果には影響しません。補正オフセットは、工具半径の摩耗値(工具パラメータ\$TC_DP15)と、限界面に垂直なオフセットを計算するためにプログラム指令された工具オフセット OFFN の合計から得られます。

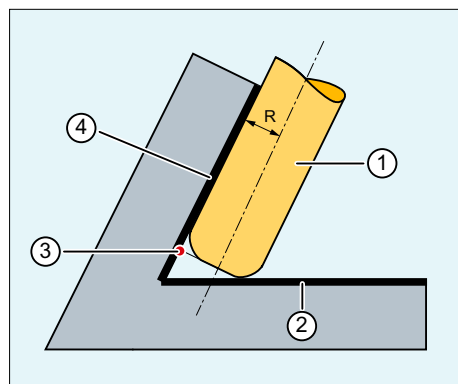
生成したパートプログラムは、加工する表面が軌跡の右側か、左側かを指定しません。したがって、半径が正の値で、オリジナル工具の摩耗値が負の値であることを前提とします。負の摩耗値は常に、より小さい直径の工具を表わします。

円筒工具の使用

円筒工具の使用時には、加工面と限界面が鋭角(90°未満)を成す場合にのみ、切り込みが必要です。トーラス形フライス工具(コーナR付きのエンドミル)を使用する場合は、鋭角と鈍角の両方の場合に、長手方向の工具の切り込みが必要です。

CUT3DCC による 3D TRC:加工面の輪郭

トーラス形フライス工具による CUT3DCC が有効な場合は、プログラム指令軌跡は、同じ直径を持つ仮想の円筒フライス工具を基準とします。得られる軌跡基準点は、次のトーラス形フライス工具の図に示されています。



- ① トーラス形フライス工具
- ② 限界面
- ③ 経路基準点
- ④ 加工面
- R シャフト半径(工具半径)

加工面と限界面が成す角度は、同じブロック内で鋭角から鈍角に、またはその逆に変化する場合があります。

実際には、標準工具より大きな工具も、小さな工具も使用できます。ただし、結果としては、コーナ半径は負ではなく、工具半径の符号はそのまま保持してください。

4.13 工具オフセット

CUT3DCC の場合は、NC パートプログラムは加工面の輪郭を基準とします。従来の工具径補正と同じように、以下で構成される工具半径の合計を使用します。

- 工具半径(工具パラメータ\$TC_DP6)
 - 摩耗値(工具パラメータ\$TC_DP15)
 - 限界面に垂直なオフセットを計算するためにプログラム指令された工具オフセット OFFN
- 限界面の位置は、次の差分から定義されます。

標準工具の寸法 - 工具半径(工具パラメータ\$TC_DP6)

Advanced Surface / Top Surface

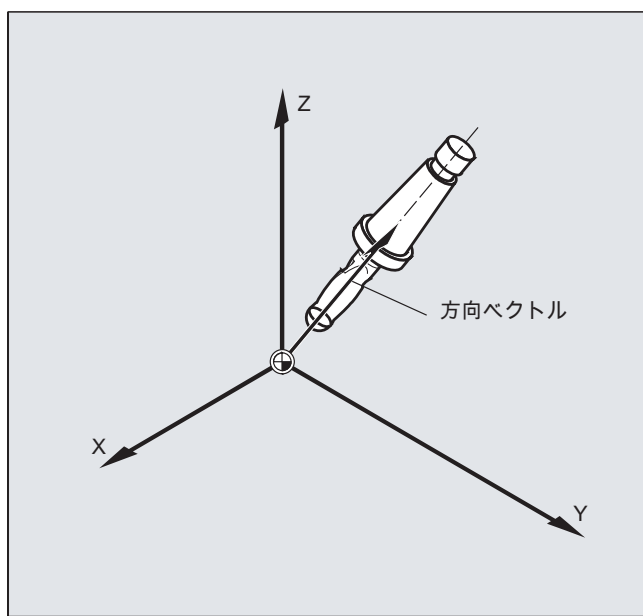
注記

工具径補正 CUT3DCC / CUT3DCCD を「Advanced Surface」または「Top Surface」機能(ライセンスが必要)と組み合わせて適用する場合、「Advanced Surface」/「Top Surface」の推奨設定を遵守してください。設定データの確認用の特別なテストプログラムが SIOS ポータルに 設けられています。

→ Advanced Surface/Top Surface 用のテストプログラム (<https://support.industry.siemens.com/cs/ww/en/view/109738423>)

4.13.6 工具オリエンテーション(ORIC、ORID、OSOF、OSC、OSS、OSSE、ORIS、OSD、OST)

工具オリエンテーションという用語は、空間の工具の幾何学的配置を表わします。5 軸工作機械の工具オリエンテーションは、プログラム命令を使用して設定できます。



OSD と OST で起動する向きの回転移動は、工具オリエンテーションの補間タイプに対応して、それぞれ実行されます。

ベクトル補間が有効な場合は、スムージングした旋回処理も、ベクトル補間を使用して補間されます。また、回転軸補間が有効な場合は、旋回は、回転軸移動を使用して直接スムージングされます。

プログラミング

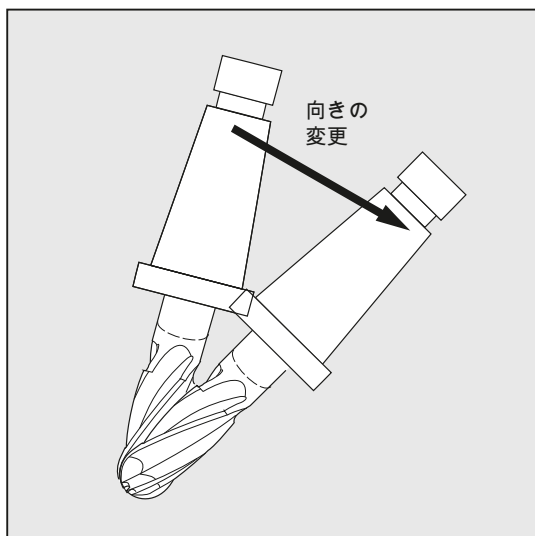
向きの変更のプログラミング:

工具オリエンテーションの変更は、次のようにプログラム指令できます。

- 回転軸 A、B、C の直接プログラミング(回転軸補間)
- オイラー角または RPY 角
- 方向ベクトル(A3 または B3 または C3 の指定によるベクトル補間)
- LEAD/TILT (正面削り)

基準となる座標系は、機械座標系(ORIMKS)または現在のワーク座標系(ORIWKs)です。

4.13 工具オフセット



工具オリエンテーションのプログラミング:

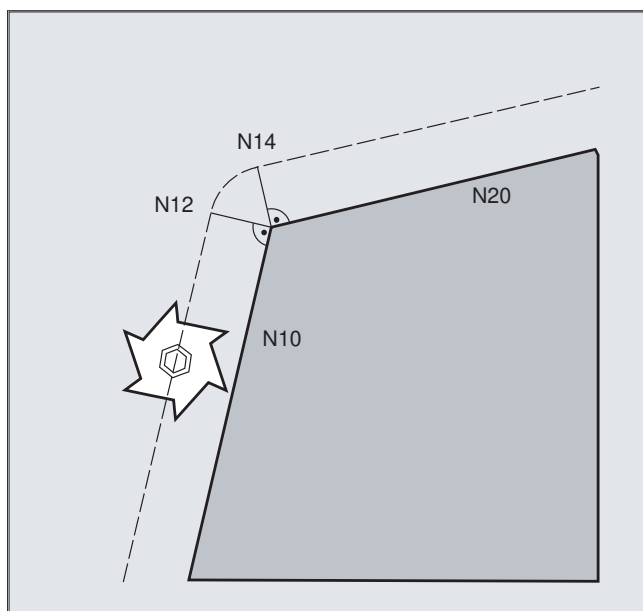
ORIC:	向きの移動と軌跡移動の並行動作
ORID:	向きの移動と軌跡移動の連続動作
OSOF:	スムージングなしの旋回
OSC:	連続旋回のスムージング
OSS:	ブロックの開始時だけの旋回のスムージング
OSSE:	ブロックの開始時と終了時の旋回のスムージング
ORIS:	°/mm の単位で起動される旋回のスムージングによる向きの変更の速度(OSS と OSSE で有効)
OSD:	次のセッティングデータでスムージング距離が指定された旋回のスムージング: SD42674 \$SC_ORI_SMOOTH_DIST
OST:	次のセッティングデータでベクトル補間の許容角度範囲(° 単位)が指定された旋回のスムージング: SD42676 \$SC_ORI_SMOOTH_TOL 回転軸補間の場合は、指定した許容範囲が、旋回軸の最大変動範囲であることが前提となります。

注記

工具オリエンテーションのスミージングのすべての命令(OSOF、OSC、OSS、OSSE、OSD、およびOST)は、G グループ 34 にまとめられています。これらの命令はモーダルです。つまり、同時に有効となるのは、その中の 1 つだけです。

例**例 1:ORIC**

オリエンテーションの変更がある 2 つ以上のブロックが、プログラムされている移動ブロック N10 と N20 の間にプログラムされている場合(例: A2=...B2=...C2=...), かつ ORIC が有効な場合、アブソリュート角度の変更に従って、挿入円弧ブロックがこれらの中間ブロックの間に配分されます。

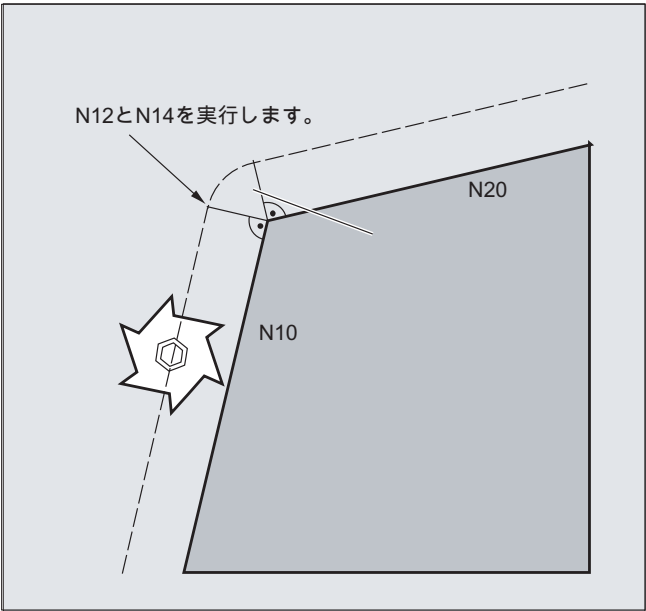


プログラムコード	コメント
ORIC	
N8 A2=... B2=... C2=...	
N10 X... Y... Z...	
N12 C2=... B2=...	; 外側コーナの挿入円弧ブロックは、向きの変更に応じて、N12 と N14 の間に配分されます。円弧移動と向きの変更は、並行して実行されます。
N14 C2=... B2=...	
N20 X =...Y=... Z=... G1 F200	

4.13 工具オフセット

例 2 :ORID

ORID が有効な場合は、2 個の移動ブロック間のすべてのブロックが、1 番目の移動ブロックの終わりに実行されます。一定の向きをもった円弧ブロックは、2 番目の移動ブロックの直前で実行されます。



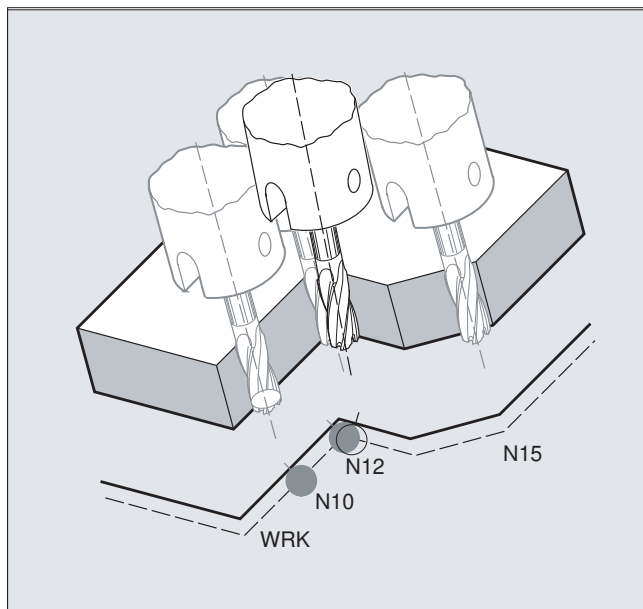
プログラムコード	コメント
ORID	
N8 A2=... B2=... C2=...	
N10 X... Y... Z...	
N12 A2=... B2=... C2=...	; N12 と N14 ブロックは、N10 の終わりで実行されます。その後、円弧ブロックが実際の向きで実行されます。
N14 M20	; 補助機能、など
N20 X... Y... Z...	

注記

外側の輪郭で向きの変更をおこなう方法は、外側コーナの最初の移動ブロックで有効なプログラム命令で特定されます。

向きが変更されない場合:ブロック境界で向きが変更されない場合は、工具の断面が円弧で、両方の輪郭に接しています。

例 3 :内側コーナでの向きの変更



プログラムコード

```

ORIC
N10 X ...Y... Z... G1 F500
N12 X ...Y... Z... A2=... B2=... C2=...
N15 X ...Y... Z... A2=... B2=... C2=...

```

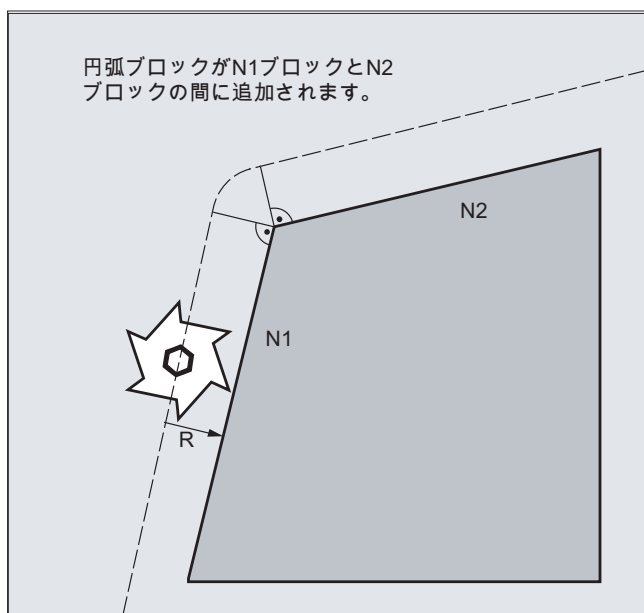
詳細情報

外側コーナでの動作

カッターの半径による円弧ブロックは常に、外側コーナで挿入されます。

ORIC と ORID プログラム命令を使用して、ブロック N1 と N2 の間にプログラム指令した向きの変更を、挿入円弧ブロックの前に処理するか、円弧ブロックと同時に処理するかを特定します。

4.13 工具オフセット



向きを外側コーナで変更する必要がある場合は、補間と同時に実行するか、または補間とは別に、軌跡移動時に実行することができます。

ORID をプログラム指令すると、挿入ブロックは、最初に軌跡移動なしブロックで実行します。コーナを生成する円弧ブロックは、2 個の移動ブロックのうちの 2 番目の直前に挿入されます。

複数の旋回ブロックを外部コーナで挿入し、ORIC を選択している場合は、向きの変更の絶対値に従って、挿入した個々のブロック間に円弧移動が配分されます。

OSD または OST による旋回のスムージング

G642 によるスムージング時には、輪郭軸と旋回軸の最大変動範囲が大きく異ならないようにしてください。2 つの許容範囲のうちの小さい方により、スムージング移動のタイプと許容角度範囲のいずれか、または両方が特定され、旋回処理のスムージングが比較的強めにおこなわれるため、許容輪郭誤差がより小さくなります。

OSD と OST を有効にすると、指定したスムージング距離と許容角度範囲を使用して、旋回処理によるわずかな誤差を、大きな輪郭誤差を生じることなく、「十分に」スムージングできます。

注記

OSD と OST のいずれか、または両方で旋回を丸み付けするときは、G642 による輪郭の丸み付け処理(および旋回処理)とは異なり、別のブロックを形成することなく、丸み付け移動を直接、プログラム指令したオリジナルのブロックに追加します。

OSD と OST のいずれか、または両方を使用しても、工具オリエンテーション補間のタイプを変更する(ベクトル→回転軸、回転軸→ベクトル)場合、ブロック遷移部は丸み付けできません。これらのブロック遷移部は、必要に応じて、標準丸み付け機能 G641、G642 と G643 で丸み付けできます。

4.13.7 D 番号の任意割り当て、刃先番号

4.13.7.1 D 番号、刃先番号の任意割り当て(CE アドレス)

D 番号

D 番号は、オフセット番号として使用できます。また、CE アドレスで刃先番号をアドレス指定することもできます。システム変数\$TC_DPCE を使用して、刃先番号を書き込むことができます。

初期設定:補正番号 == 工具刃先番号

マシンデータを使用して、最大 D 番号(刃先番号)、および工具ごとの刃先の最大数を定義します(→ 工作機械メーカー)。後述の命令は、指定した最大刃先番号(MD18105)が、工具ごとの刃先(MD18106)の最大数より大きい場合のみ使用できます。工作機械メーカーの指示に従ってください。

詳細情報

機能マニュアル、工具

4.13.7.2 D 番号の任意割り当て: D 番号のチェック(CHKDNO)

CKKDNO 命令を使用すると、既存の D 番号が一義的に割り当てられているかどうかを確認できます。TO ユニット内で定義されたすべての工具の D 番号は、2 回以上出現してはなりません。予備工具は考慮されません。

4.13 工具オフセット

構文

```
state=CHKDNO (Tno1,Tno2,Dno)
```

意味

state:	= TRUE:	D 番号は確認範囲に、一義的に割り当てられています。
	= FALSE:	D 番号が競合しているか、またはパラメータが無効です。Tno1、Tno2、および Dno は、競合をおこすパラメータを返します。これらのデータをパートプログラムで使用できます。
CHKDNO (Tno1,Tno2):	指定した範囲のすべての D 番号を確認します。	
CHKDNO (Tno1):	Tno1 のすべての D 番号を、他のすべての工具に対して確認します。	
CHKDNO:	すべての工具のすべての D 番号を、他のすべての工具に対して確認します。	

4.13.7.3 D 番号の任意割り当て: D 番号の名称変更(GETDNO、SETDNO)

D 番号は一義的に割り当ててください。1 つの工具の 2 個の異なる刃先が同じ D 番号にならないようにしてください。

GETDNO

この命令は、工具番号 **t** の工具の特定の刃先(**ce**)の D 番号を返します。入力したパラメータに対して D 番号が存在しない場合は、**d=0** が設定されます。D 番号が無効な場合は、32000 を超える値が返されます。

SETDNO

この命令は、D 番号の値 **d** を工具 **t** の刃先(**ce**)に割り当てます。この命令の結果は **state** (TRUE または FALSE)で返されます。指定したパラメータのデータブロックが存在しない場合は、値 FALSE を返します。構文に誤りがあると、アラームが発生します。D 番号は、明示的には 0 に設定できません。

構文

```
d = GETDNO (t,ce)
state= SETDNO (t,ce,d)
```

意味

D:	工具刃先の D 番号
T:	工具の T 番号
ce:	工具の刃先番号(CE 番号)
state:	命令が実行できたかどうか(TRUE または FALSE)を表わします。

D 番号の名称変更の例

プログラミング	コメント
\$TC_DP2[1,2]=120	
\$TC_DP3[1,2] = 5.5	
\$TC_DPCE[1,2] = 3	; 刃先番号 CE
...	
N10 def int DNoOld, DNoNew = 17	
N20 DNoOld = GETDNO(1,3)	
N30 SETDNO(1,3,DNoNew)	

これで、新しい D 番号 17 が刃先 CE=3 に割り当てられます。この刃先のデータが、NC アドレス D を使用したプログラミングとシステム変数の両方で、D 番号 17 によってアドレス指定されます。

4.13.7.4 D 番号の任意割り当て: 指定した D 番号に対して T 番号の特定(GETACTTD)

定義済み機能 GETACTTD により、絶対 D 番号に関連付けられた T 番号を特定します。一義的かどうかはチェックされません。工具オフセットユニット内に同じ D 番号が複数ある場合は、検索で見つかった最初の工具の T 番号を返します。

構文

<状態>=GETACTTD(<TNo>,<DNo>)

意味

GETACTTD()	ファンクションコール	
:		
<DNo>:	検索する T 番号の D 番号です。	
	データタイプ:	INT

4.13 工具オフセット

<TNo>:	見つかった T 番号です。			
	データタイプ:	VAR INT		
<status>:	結果			
	データタイプ:	INT		
	値:	0	T 番号が見つかりました。<Tno>には T 番号の値が入っています。	
		-1	指定した D 番号の T 番号は存在しません; <Tno>=0 です。	
		-2	D 番号はアブソリュートではありません。<TNo>は、値<Dno>の D 番号を含む、最初に見つかった工具の値を受け取ります。	
		-5	上記以外のなんらかの理由により、この機能を実行できません。	

4.13.7.5 D 番号の任意割り当て: D 番号の無効化(DZERO)

4.13.8 工具ホルダキネマティクス

必要条件

工具ホルダは以下の場合にのみ、空間で考えられるすべての方向に工具を旋回できます。

- 2つの回転軸 v_1 および v_2 が存在する。
- 回転軸が互いに直交している。
- 工具縦軸が 2 番目の回転軸 v_2 に直交している。

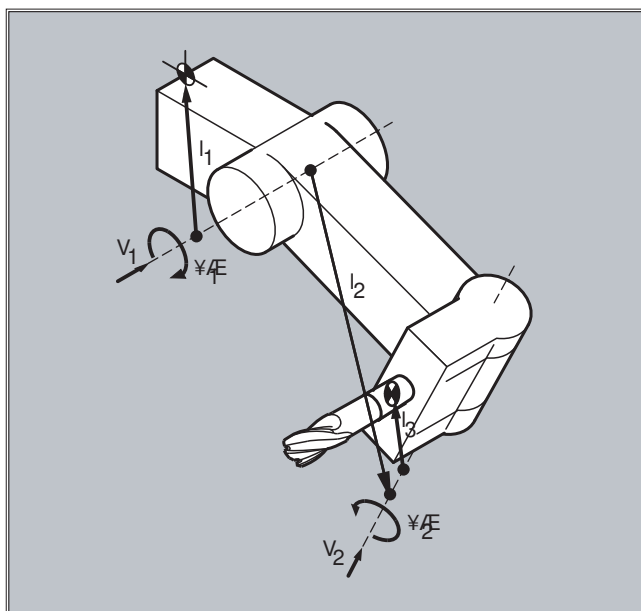
さらに、考えられるすべての旋回が設定可能な機械に対しては、以下の必要条件が適用されます。

- 工具長手軸が 1 番目の回転軸 v_1 に直交している。

機能

最大 2 つの回転軸 v_1 または v_2 による工具ホルダのキネマティックスは、17 個のシステム変数 $\$TC_CARR1[m] \sim \$TC_CARR17[m]$ を使用して定義します。工具ホルダの記述には、以下が含まれます。

- 工具ホルダの 1 番目の回転軸 I_1 からのベクトル距離、1 番目の回転軸から 2 番目の回転軸 I_2 までのベクトル距離、2 番目の回転軸から工具の基準点までのベクトル距離 I_3 。
- 両方の回転軸 v_1 、 v_2 の方向ベクトル。
- 2 つの軸を中心とした回転角度 α_1 、 α_2 。回転角度は、回転軸ベクトルの視野方向の、右回転方向が正の角度です。



分解キネマティックス(工具と部品の両方を回転できる)を備えた機械では、システム変数が $\$TC_CARR18[m] \sim \$TC_CARR23[m]$ で拡張されています。

パラメータ

旋回工具ホルダのシステム変数の機能			
名称	x 成分	y 成分	z 成分
l_1 オフセットベクトル	$\$TC_CARR1[m]$	$\$TC_CARR2[m]$	$\$TC_CARR3[m]$
l_2 オフセットベクトル	$\$TC_CARR4[m]$	$\$TC_CARR5[m]$	$\$TC_CARR6[m]$

4.13 工具オフセット

旋回工具ホルダのシステム変数の機能			
v_1 回転軸	\$TC_CARR7[m]	\$TC_CARR8[m]	\$TC_CARR9[m]
v_2 回転軸	\$TC_CARR10[m]	\$TC_CARR11[m]	\$TC_CARR12[m]
α_1 回転角度 α_2 回転角度	\$TC_CARR13[m] \$TC_CARR14[m]		
l_3 オフセットベクトル	\$TC_CARR15[m]	\$TC_CARR16[m]	\$TC_CARR17[m]

旋回工具ホルダのシステム変数の拡張			
名称	x 成分	y 成分	z 成分
l_4 オフセットベクトル	\$TC_CARR18[m]	\$TC_CARR19[m]	\$TC_CARR20[m]
軸識別子回転軸 v_1 回転軸 v_2	回転軸 v_1 と v_2 の軸識別子(ゼロで初期化) \$TC_CARR21[m] \$TC_CARR22[m]		
キネマティック スタイブ Tool Part Mixed モード	\$TC_CARR23[m] キネマティックスタイブ T -> 工具のみが回転可能 (初期設定)	キネマティックスタイブ P -> 部品のみが回転可能	キネマティックスタイブ M 部品と工具が回転可能
回転軸 v_1 回転軸 v_2 のオフセット	回転軸 v_1 と v_2 の角度(° 単位)(初期設定 \$TC_CARR24[m] \$TC_CARR25[m]を前提とする)		
回転軸 v_1 回転軸 v_2 の角度オフセット	回転軸 v_1 と v_2 のカップリングギヤシステムのオフセット(° 単位) \$TC_CARR26[m] \$TC_CARR27[m]		
角度増分 v_1 回転軸 v_2 回転軸	回転軸 v_1 と v_2 のカップリングギヤシステムのオフセット(° 単位) \$TC_CARR28[m] \$TC_CARR29[m]		
最小位置回転軸 v_1 回転軸 v_2	回転軸 v_1 と v_2 のソフトウェア リミットの最小位置\$TC_CARR30[m] \$TC_CARR31[m]		

旋回工具ホルダのシステム変数の拡張			
最大位置回転軸 v_1 回転軸 v_2	回転軸 v_1 と v_2 のソフトウェアリミットの最大位置\$TC_CARR32[m] \$TC_CARR33[m]		
工具ホルダ名称	工具ホルダには、番号の代わりに名称を付けられます。\$TC_CARR34[m]		
ユーザー: 軸名称 1 軸名称 2 識別子 位置	ユーザーの計測サイクルの用途で使用\$TC_CARR35[m] \$TC_CARR36[m] \$TC_CARR37[m] \$TC_CARR38[m] \$TC_CARR39[m] \$TC_CARR40[m]		
仕上げ オフセット	基本パラメータの値に 加算可能なパラメータ。		
l_1 オフセットベクトル	\$TC_CARR41[m]	\$TC_CARR42[m]	\$TC_CARR43[m]
l_2 オフセットベクトル	\$TC_CARR44[m]	\$TC_CARR45[m]	\$TC_CARR46[m]
l_3 オフセットベクトル	\$TC_CARR55[m]	\$TC_CARR56[m]	\$TC_CARR57[m]
l_4 オフセットベクトル	\$TC_CARR58[m]	\$TC_CARR59[m]	\$TC_CARR60[m]
v_1 回転軸	\$TC_CARR64[m]		
v_2 回転軸	\$TC_CARR65[m]		

注記

パラメータの意味

「m」は、プログラム指令する工具ホルダの番号を指定します。

\$TC_CARR47～\$TC_CARR54 および\$TC_CARR61～\$TC_CARR63 は定義されません。また、読み取りや書き込みをおこなった場合、アラームが発生します。

軸の距離ベクトルの始点/終点は自由に選択できます。2つのそれぞれの軸を中心とした回転角度 α_1 、 α_2 は、工具ホルダの初期状態を 0° として定義されます。この方法を使用すると、工具ホルダのキネマティックスを、任意の数だけプログラム指令できます。

回転軸が 1 つのみ、またはまったくない工具ホルダは、回転軸のいずれかまたは両方の方向ベクトルをゼロに設定すると記述できます。

回転軸を装備しない工具ホルダでは、距離ベクトルが、加工平面(G17～ G19)の変化に影響されない成分を持つ追加の工具オフセットとして機能します。

4.13 工具オフセット

拡張パラメータ

回転軸のパラメータ

システム変数は、\$TC_CARR24[m]～\$TC_CARR33[m]項目により拡張されており、次のように記述されます。

回転軸 v_1 、 v_2 のオフセット	旋回工具ホルダの初期設定のために、回転軸 v_1 または v_2 の位置を変更。
回転軸 v_1 、 v_2 の角度オフセット/ 角度増分	回転軸 v_1 と v_2 のカップリングギヤシステムのオフセットまたは角度増分。プログラム指令した、または計算した角度は、 $\text{phi} = s + n * d$ (n は整数) で得られる値の整数値に切り上げられます。
回転軸 v_1 、 v_2 の最小位置と最大位置	回転軸 v_1 と v_2 の回転軸の制限角度(ソフトウェアリミット)の最小位置と最大位置。

ユーザー用パラメータ

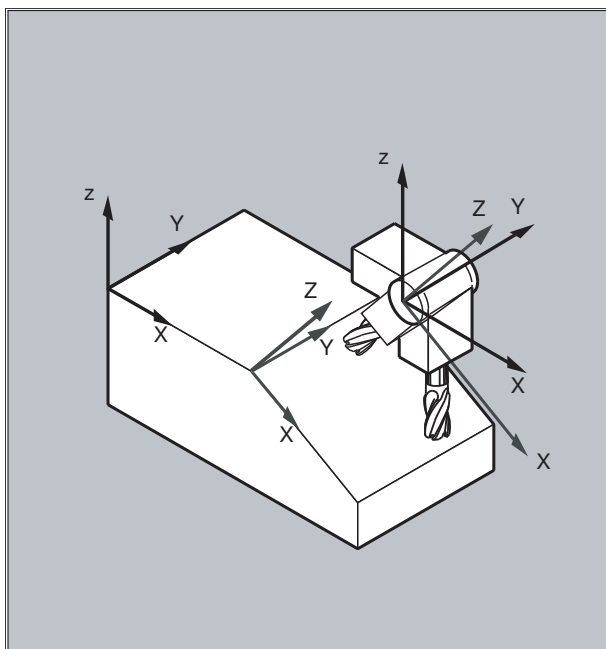
\$TC_CARR34～\$TC_CARR40 には、ユーザーが自由に使用できるパラメータが含まれます。また、これらのパラメータは、SW 6.4 以前では、標準では、NCK 内で使用できない、または意味をもちません。

仕上げオフセットパラメータ

\$TC_CARR41～\$TC_CARR65 には、基本パラメータの値に加算可能な仕上げオフセットパラメータが含まれます。基本パラメータに割り当てる仕上げオフセット値は、40 の値をパラメータ番号に加算すると得られます。

例

次の例で使用する工具ホルダは、Y 軸を中心とした回転で全体を記述できます。



プログラムコード	コメント
N10 \$TC_CARR8[1]=1	; 工具ホルダ 1 の 1 番目の回転軸の Y 成分の定義です。
N20 \$TC_DP1[1,1] = 120	; エンドミルの定義です。
N30 \$TC_DP3[1,1]= 20	; 20 mm 長さのエンドミルの定義です。
N40 \$TC_DP6[1,1]=5	; 5 mm 半径のエンドミルの定義です。
N50 ROT Y37	; Y 軸を中心とした 37°の回転によるフレーム定義です。
N60 X0 Y0 Z0 F10000	; 開始位置へアプローチします。
N70 G42 CUT2DF TCOFR TCARR=1 T1 D1 X10	回転フレームで径補正と工具長補正を設定して、工具ホルダ 1、工具 1 を選択します。
N80 X40	; 37°の回転で加工を実行します。
N90 Y40	
N100 X0	
N110 Y0	
N120 M30	

4.13 工具オフセット

詳細情報

分解キネマティックス

分解キネマティックス(工具とワークの両方が回転する)を備えた機械では、システム変数が \$TC_CARR18 [m] ～ \$TC_CARR23 [m] 項目で拡張されており、次のように記述されます。

以下を備えた旋回工具テーブル:

- 3 番目の回転軸で回転できる工具テーブルの基準点 I_4 に対する 2 番目の回転軸 V_2 のベクトル距離。

以下を備えた回転軸:

- 回転軸 V_1 と V_2 の参照用の 2 個のチャネル識別子(その位置は、必要に応じて、旋回工具ホルダの向きの特定に使用します)。

以下のいずれかのキネマティックスタイプ値:

- キネマティックスタイプ T:工具のみが回転可能です。
- キネマティックスタイプ P:部品のみが回転可能です。
- キネマティックスタイプ M:工具と部品が回転可能です。

工具ホルダデータの消去

すべての工具ホルダデータセットのデータは、\$TC_CARR1 [0]=0 を使用して削除できます。

キネマティックスタイプ \$TC_CARR23 [T]=T には、使用可能な 3 種類の文字(T、P、M)の大文字または小文字のいずれかを割り当ててください。このため、これは削除しないでください。

工具ホルダデータの変更

記述した値はいずれも、パートプログラムで新しい値を割り当てて変更できます。T、P、および M 以外の文字を割り当てると、旋回工具ホルダを起動しようとしたときにアラームが発生します。

工具ホルダデータの読み取り

記述した値はいずれも、パートプログラムでそれを変数に割り当てると、読み取ることができます。

仕上げオフセット

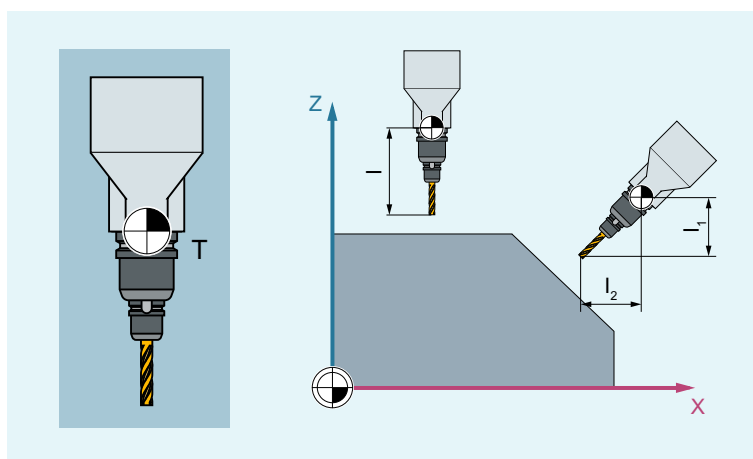
不適切な仕上げオフセット値が検出されるのは、旋回工具ホルダが有効なときに、旋回工具ホルダがこの値を含み、同時にセッティングデータが

SD42974 \$SC_TOCARR_FINE_CORRECTION = TRUE の場合のみです。

最大許容仕上げオフセットは、マシンデータの許容値に制限されます。

4.13.9 旋回工具ホルダの工具長補正(TCARR、TCOABS、TCOFR、TCOFRX、TCOFRY、TCOFRZ)

空間の工具ホルダの向きと、したがって工具の向きが変更すると、その工具長成分も変化します。



T 工具ホルダレファレンス点

l_1 、 工具長成分

l_2

手動設定または、空間の向きが固定された工具ホルダの交換などによりリセットされた後には、工具長成分も再度決定してください。G グループ 42「工具ホルダ」命令はこのために使用されます。

構文

TCARR=<m>
TCOABS
TCOFR
TCOFRZ/TCOFRY/TCOFRX

意味

項目	タイプ	意味	
TCARR=<m>	アドレス	工具ホルダ要求	
		<m>	工具キャリアの番号
TCOABS	G 命令	現在の工具ホルダの向きから工具長成分を特定します。	

4.13 工具オフセット

項目	タイプ	意味
TCOFR	G 命令	動作中のフレームの向きから工具長成分を特定します。
TCOFRZ / TCOFRY / TCOFRX	G 命令	TCOFRX/TCOFRY/TCOFRZ 命令は対応する方向(X/Y/Z)に向けられた工具を想定し、方向付け可能な工具ホルダの設定角度を計算し、有効なフレームの工具が同じ方向を向くようにします。
		TCOFRZ Z 方向に向いた工具は調整され、有効なフレームでも Z 方向を向いています。
		TCOFRY Y 方向に向いた工具は調整され、有効なフレームでも Y 方向を向いています。
		TCOFRX X 方向に向いた工具は調整され、有効なフレームでも X 方向を向いています。

詳細情報

工具ホルダの選択(TCARR)

工具ホルダデータブロックはジオメトリを説明する工具ホルダに接続されます。工具ホルダの選択時に工具が有効な場合、工具は新たに選択された工具ホルダに割り当てられ、工具ホルダのジオメトリデータが直ちに有効になります。動作中の工具を変更したり、再プログラムする必要はありません。新しい工具が有効になると、有効な工具ホルダに取り付けられたかのように処理されます。

TCARR=0 の場合、動作中の工具ホルダは選択解除されます。

注記

工具ホルダの現在のジオメトリデータは、対応するシステム変数を使用して NC プログラムで定義されます(「工具ホルダデータブロックへのアクセス」を参照)。

注記

コントローラの側から見て、工具ホルダ番号<m>と工具番号 <n>を自由に組み合わせることができます。ただし、実際の用途では、加工または構造上の理由で一部の組み合わせを除外することができます。コントローラは、組み合わせに矛盾があるかどうかをチェックしません。

工具ホルダデータブロックへのアクセス

工具ホルダの現在のジオメトリデータは、以下のようにして NC プログラムからアクセスできます。

- 書き込み:

`$TC_CARR<n> [<m>] = <値>`

- 読み取り:

`<値> = $TC_CARR<n> [<m>]`

(`<値>`はデータタイプ REAL の変数でなければなりません)

参照された工具ホルダが定義されていない場合は、アラームが表示されます。

工具ホルダデータブロックをゼロに設定

工具ホルダデータブロックのすべての値は、1 つの命令を使用してパートプログラム内から検出できます。

`$TC_CARR1[0] = 0`

個々の工具ホルダデータブロックは予約処理 DELTC を使用し、選択して削除できます。

工具ホルダ選択の影響

工具ホルダと工具が両方とも有効になったときに、工具ホルダは有効になります。工具ホルダのみ選択しても、機能しません。

工具ホルダ選択の影響は、TCARR とともにプログラム指令された G グループ 42 (工具ホルダ)からの命令によります。

このグループからの G 命令の変更は、工具ホルダの動作時に工具長成分を再計算させます。

工具ホルダの向きから工具長補正の計算(TCOABS)

TCOABS は、システム変数 \$TC_CARR13 および \$TC_CARR14 に保存されている工具ホルダの現在の旋回角度から工具長補正を計算します。検討されている向きは有効なフレームの向きとは無関係です。

システム変数による工具ホルダのキネマティクスの定義の場合、NC プログラミングマニュアルの第 1 章「工具ホルダのキネマティクス (ページ 890)」を参照してください。

有効なフレームの向きから工具長成分の決定(TCOFR)

TCOFR の場合、工具ホルダの旋回角度は有効なフレームの向きから決定されます。ただし、工具ホルダデータに保存された値は変更されません。これらは、回転角度を 1 つのフレームから計算する際に生じる曖昧さを解決するためにも使用されます(以下のパラグラフを参照)。

4.13 工具オフセット

あいまいさ

2つの軸の場合、フレームによって定義される特定の工具の向きは一般に、**2つの異なる**回転角度ペアによって設定できます。これらの2つの可能な位置の中で、コントローラはプログラム指令された回転角度にできるだけ近い回転角度の設定を選びます。

注記

あいまいさが生じる可能性がある場合、通常、フレームから想定される角度を工具ホルダデータに保存する必要があります。

有効なフレームの工具オリエンテーション(TCOFRZ/TCOFRY/TCOFRX)

TCOFRX/TCOFRY/TCOFRZ 命令は、有効な向きにかかわらず、工具ホルダの設定角度を計算するために X/Y/Z 方向で向きを想定します。加工平面(G17/G18/G19)の選択は有効なフレームの向きに影響を与えません。ただし、工具長補正の計算に影響を与えます。

フレーム変更

工具の選択後にフレームを変更できます。フレームを変更しても、工具長補正成分には影響ありません。

工具ホルダデータに保管されている回転角度は、フレームで定義されている回転角度によって影響されません。TCOFR から TCOABS への変更時、工具ホルダデータのオリジナルの(プログラムされた)回転角度が再び有効になります。

工具長補正の再計算

フレームの変更後に工具長補正を再計算するために、工具を再度選択してください。工具長補正を計算するときは、工具ホルダの回転角度を中間のステップで計算します。2つの回転軸を持つ工具ホルダの場合は通常、2つの回転角度ペアがあり、これらを使用すると、動作中のフレームに合わせて工具オリエンテーションを調整できます。したがって、工具ホルダデータに格納された回転角度の値は、少なくとも機械的に設定した回転角度にほぼ対応している必要があります。

注記

実際にあいまいが生じる場合では、フレームから期待される近似角度を工具ホルダデータに保存する必要があります。

注記

工具オリエンテーションは、動作中のフレームに合わせて手動で調整してください。

注記

コントロールシステムは、機械の調整機能に対してフレームの向きを使用して計算される回転角度をチェックできません。工具ホルダの回転軸がフレームの向きによって計算される工具オリエンテーションに到達できないように機械が設計されている場合、アラームが出力されます。

注記

精密工具補正と、可動式工具ホルダでの工具長オフセットの各機能を組み合わせることはできません。これらの両機能を同時に呼び出した場合は、エラーメッセージを出力します。

注記

TOFRAME (ページ 371)の場合、選択した工具ホルダのオリエンテーション方向に基づいてフレームを定義できます。

注記

方向座標変換が有効(3 軸、4 軸、または 5 軸座標変換)な場合は、原点位置からずれている旋回工具ホルダをアラームを発生させずに選択できます。

注記

回転しない動作中のフレームの場合、TCOFRX/TCOFRY/TCOFRZ を使用したオリエンテーションは、工具オリエンテーションがすでに X/Y/Z 方向に向いているので解決するのに手間はかかりません。

Z 方向での向き(TOFRAME)

G グループ 53 (工具関連のフレームの回転)の TOFRAME 命令の場合、その Z 方向が選択した工具ホルダのオリエンテーション方向に平行に調整されているフレームを定義できます。

工具ホルダが無効か、または向きの変化しない工具ホルダが有効である場合、新しいフレームの Z 方向は以下ようになります。

- G17 では前の Z 方向と同じ。
- G18 では前の Y 方向と同じ。
- G19 では前の X 方向と同じ

4.13 工具オフセット

標準サイクルと計測サイクルの転送パラメータ

標準サイクルと計測サイクルの転送パラメータの場合は、定義済みの以下の数値の範囲が適用されます。

角度値では、以下の値範囲が定義されます。

回転対称軸 ...	数値の範囲
1 番目のジオメトリ軸	$-180^{\circ} \leq \text{角度} \leq +180^{\circ}$
2 番目のジオメトリ軸	$-90^{\circ} \leq \text{角度} \leq +90^{\circ}$
3 番目のジオメトリ軸	$-180^{\circ} \leq \text{角度} \leq +180^{\circ}$

注記

角度の値を標準サイクルまたは計測サイクルに渡すときは、以下のこと注意しておこなってください。

NC の演算精度よりも小さい値はゼロに丸められます!

回転位置の NC の計算最小単位は、以下のマシンデータで定義してください。

MD10210 \$MN_INT_INCR_PER_DEG

4.13.10 機械計測に従って、旋回可能な工具ホルダを修正(CORRTC)

工具ホルダの計測したキネマティックチェーンエレメントを、CORRTC 機能を使用して特別な補正エレメントに書き込むことができます。

注記

CORRTC 機能で書き込まれた補正值は、工具ホルダで直ちに有効になりません。補正值は、工具ホルダ、NEWCONF を解除し工具ホルダを選択してはじめて有効になります。

構文

```
<_Corr_Status> = CORRTC(<_Corr_Vect>, <_Corr_Index>,
<_Corr_Mode>, [ <_No_Alarm>])
```

意味

CORRTC:	ファンクションコール			
<_Corr_Status>:	機能の戻り値			
	データタイプ:	INT		
	値:	0	機能が異常なく実行されました。	
		1	工具ホルダが有効になっていません。	
		2	動作中の工具ホルダが、キネマティックチェーンで定義されていませんでした。	
		10	<_Corr_Index>コールパラメータは負です。	
		11	<_Corr_Mode>コールパラメータは負です。	
		12	サブチェーンのセクションへの無効な参照です(_CORR_INDEX)。	
		13	_CORR_INDEX パラメータ(\$TC_CARR_CORR_ELEM)で参照されたセクションで、補正エレメントが定義されていませんでした。	
		20	<_CORR_MODE>の 100 の位が無効です。値 0 と 1 のみが許容されます。	
		21	<_CORR_MODE>の 1000 の位が無効です。値 0 と 1 のみが許容されます。	
		30	オフセットベクトルの補正では、少なくとも 1 つの座標における現在値との差分が、セッティングデータ SD41612 \$SN_CORR_TRAFO_LIN_MAX によって指定された最大値より大きくなります。	
	31	書き込み権限がないため、システム変数を書き込もうとして拒否されました。		
<_Corr_Vect>:	補正ベクトル 補正ベクトルの内容は、次のパラメータ<_Corr_Index>および<_Corr_Mode>によって定義されます。 <_Corr_Status> = 30 の場合、ベクトルの内容は上書きされます(前述の通り)。			
	データタイプ:	REAL		
<_Corr_Index>:	補正エレメントの方向ベクトルが補正されるセクションを指定します。			
	データタイプ:	INT		

4.13 工具オフセット

<_Corr_Mode>:	補正モード	
	データタイプ:	INT
	<Corr_Index>パラメータは 10 進数コードです(一から千の位):	
	1 の位:	予備
	10 の位:	<_Corr_Index>の内容が参照する補正エレメントの変更方法を指定します。
		xx0x 補正ベクトルは、補正エレメントにすぐに書き込まれます。 このタイプは、補正エレメントをすぐに書き込むために使用できます。関連するシステムデータ(\$NK_OFF_DIR[<n>, ...])のインデックス<n>が既知である必要はありません。
		xx1x 0 と同様ですが、転送された補正值が世界座標で解釈される点は異なります。 0 と 1 のタイプの違いは、初期状態(すべての回転軸の位置が 0 に等しい)のキネマティックチェーンに他の回転が含まれる場合に必ず発生します。
		xx2x 1 と同様ですが、補正值がセクション全体を指すという点が異なります。すなわち、補正エレメントに値が入力されると、セクション全体が補正值によって定義された長さに達するようになります。
	100 の位:	<_Corr_Vect>パラメータの内容の解釈方法を指定します。
		x0xx 転送された補正ベクトル<_Corr_Vect>には、補正エレメントの新しい長さ全体、または<_Corr_Index>が<_Corr_Mode>の 10 の位と組み合わせて参照するセクションが含まれます(絶対補正)。
		x1xx 転送された補正ベクトル<_Corr_Vect>には、補正エレメントの現在の長さまたは<_Corr_Index>が<_Corr_Mode>の 10 の位と組み合わせて参照するセクションと比べた違いのみが含まれます(増分補正)。
	1000 の位:	最大許容補正がセッティングデータ\$SN_CORR_TOCARR_LIN_MAX で制限されるかどうかを決定します。
		0xxx しきい値の監視が有効です。
		1xxx しきい値の監視がマスクされています。

<_No_Alarm>:	エラー発生時の動作(戻り値 > 0) (オプション)		
	データタイプ:	BOOL	
	値:	FALSE (初期設定)	エラーが発生した場合、プログラムの実行は停止し、アラームが表示されます。
		TRUE	エラーが発生した場合、プログラムの処理は停止せず、アラームは表示されません。 用途:戻り値に対応したユーザー固有の動作

CORRTC に関する詳細情報

工具ホルダのキネマティック構造は、対応するレファレンス点、機械の原点、または工具ホルダのレファレンス点から始まる 1 つ(タイプ T およびタイプ P)または 2 つ(タイプ M)のキネマティックチェーン(サブチェーン)で記述されます。2 つのチェーンの内の 1 つ、工具チェーンは、工具のレファレンス点で終了し、もう 1 つのチェーンであるワークチェーンは、基本座標系の原点で終了します。

CORRTC 機能は、工具ホルダを持つ機械の軸方向を特別な補正エレメントに書き込みます。キネティックチェーンは、\$ NK_TYPE で定義されるタイプ OFFSET のエレメントなどを使用して説明されます。

CORRTC はセクションで動作する

2 つのサブチェーンは、それぞれ最大 4 つのセクションに分けることができます。

- セクション 1 はチェーンの起点で始まり、最初の回転軸で終了します。
- セクション 2 は回転軸 1 と回転軸 2 の間のセクションです。
- セクション 3 は、回転軸 2 とチェーンの終点の間のセクションです。

下の図に、2 つの回転軸を備えた旋回可能な工具ホルダを示します。

4.13 工具オフセット

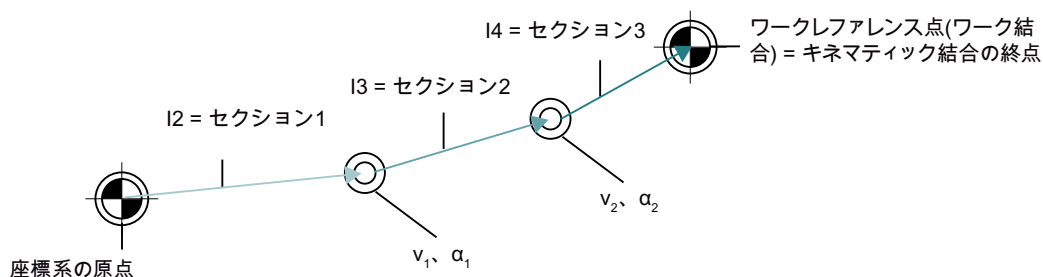


図 4-12 CORRTC の例

セクションは明確に定義されます。キネマティックサブチェーンを起点から終点まで動かした場合、最初のセクションはインデックス 1 となり、次はインデックス 2、などとなります。

補正エレメント

`$TC_CARR_CORR_ELEM [, 0 ... 3]` システム変数を使用して、各セクションの一定のキネマティックチェーンエレメント(タイプ `$NK_TYPE[<n>] = "OFFSET"` のチェーンエレメント)を参照することができます。機械計測中に特定された補正值は、CORRTC 機能を使用して、指定されたエレメントに書き込まれます。

`$TC_CARR_CORR_ELEM[m, 0 ... 3]` での参照の順序は、前述のセクションに対応している必要があります。つまり、オフセットベクトル `I1` に属している `$TC_CARR_CORR_ELEM[m, 0]` のチェーンエレメントは 1 つ、などとなります。

指令値は常に、CORRTC を呼び出したときに動作中の工具ホルダで有効な、対応する値です。工具ホルダを選択すると、保存されていたキネマティックデータの変更された内容は、CORRTC 機能の動作形態に対して無効になります。

4.13.11 オンライン工具長補正(TOFFON、TOFFOF)

システム変数 `$AA_TOFF[<n>]` を使用して、有効な工具長を 3 次元の 3 つの工具方向に応じて、リアルタイムで加算します。

3 個のジオメトリ軸識別子をインデックス `<n>` として使用します。これにより、有効な方向オフセットの数が、そのときに有効なジオメトリ軸で特定されます。

すべての補正を同時に有効にすることができます。

オンライン工具長補正機能を使用して、以下を実行できます。:

- 方向座標変換 TRAORI
- 旋回工具ホルダ TCARR

注記

オンライン工具長補正は**オプション**です。事前に有効にしておいてください。この機能は、動作中の方向座標変換または動作中の旋回工具ホルダと組み合わせた場合にのみ、有効です。

構文

```
TRAORI  
TOFFON (<補正方向>[, <オフセット値>])  
WHEN TRUE DO $AA_TOFF [<補正方向>] ; シンクロナイズドアクションの場合  
... ;  
TOFFOF (<補正方向>)
```

意味

TOFFON:	オンライン工具長補正の 起動	
	<補正方向>:	オンライン工具長補正が有効になる工具方向 (X、Y、Z)
	<オフセット値>:	適用時に、当該の補正方向に対してオフセット値を指定でき、この値が直ちに有効になります。
TOFFOF:	オンライン工具長補正の リセット 指定した補正方向への補正值がリセットされ、先読み停止が開始されます。	

4.13 工具オフセット

例

例 1:工具長補正の選択

プログラムコード	コメント
MD21190 \$MC_TOFF_MODE = 1	; 絶対値へアプローチします。
MD21194 \$MC_TOFF_VELO[0] =1000	
MD21196 \$MC_TOFF_VELO[1] =1000	
MD21194 \$MC_TOFF_VELO[2] =1000	
MD21196 \$MC_TOFF_ACCEL[0] =1	
MD21196 \$MC_TOFF_ACCEL[1] =1	
MD21196 \$MC_TOFF_ACCEL[2] =1	
N5 DEF REAL XOFFSET	
N10 TRAORI (1)	; 座標変換オン
N20 TOFFON (Z)	; Z の工具方向のオンライン工具長補正を有効にします。
N30 WHEN TRUE DO \$AA_TOFF[Z]=10 G4 F5	; 10 の工具長補正を Z 工具方向に補間します。
...	
N100 XOFFSET=\$AA_TOFF_VAL[X]	; 実際の補正を X 方向に割り当てます。
N120 TOFFON (X,-XOFFSET) G4 F5	; X の工具方向で、工具長補正が 0 に戻ります。

例 2:工具長補正の解除

プログラムコード	コメント
N10 TRAORI (1)	; 座標変換オン
N20 TOFFON (X)	; X の工具方向のオンライン工具長補正を有効にします。
N30 WHEN TRUE DO \$AA_TOFF[X] = 10 G4 F5	; 10 の工具長補正を X 工具方向に補間します。
...	
N80 TOFFOF (X)	; X の工具方向の位置オフセットを ...\$AA_TOFF[X]=0 で削除します。 軸は移動しません。 位置オフセットが、実際の向きに対応するワークの実位置に追加 されます。

詳細情報

ブロック解析

先読みのブロック解析のとき、メインランで有効な現在の工具長補正も考慮されます。最大許容軸速度を広範囲に使用できるようにするには、工具補正の設定中に STOPRE 先読み停止でブロック解析を中止してください。

プログラムスタート後に工具長補正を変更しない場合、または、工具長補正の変更後に、先読みとメインランの間に IPO バッファが受け入れ可能な数より多いブロックを処理していた場合は、開始時に常に工具補正は決定されています。

変数\$AA_TOFF_PREP_DIFF

補間器で現在有効な補正と、ブロック解析時に有効であった補正との寸法差は、変数 \$AA_TOFF_PREP_DIFF[<n>]でチェックできます。

マシンデータとセッティングデータの調整

オンライン工具長補正では、以下のシステムデータを使用できます。

- MD20610 \$MC_ADD_MOVE_ACCEL_RESERVE (重畳移動の加速マージン)
- MD21190 \$MC_TOFF_MODE
システム変数の内容\$AA_TOFF[<n>]が、絶対値として移動するか、または統合されます。
- MD21194 \$MC_TOFF_VELO (オンライン工具長補正の速度)
- MD21196 \$MC_TOFF_ACCEL (オンライン工具長補正の加速度)
- 事前設定の制限値のセッティングデータ
:
SD42970 \$SC_TOFF_LIMIT (工具長補正の上限値)

4.13.12 旋回工具のオフセットデータ変更

4.13.12.1 オリエンテーションの計算(ORISOLH)

事前定義された ORISOLH 機能は、ワークを基準に定義されたキネマティックとは無関係な位置に旋削工具を移動するために、ユーザーが機械の回転軸位置を設定するうえで役立ちます。前提条件は、キネマティック結合によりパラメータ設定された 6 軸変換が有効であることです。

次の 2 つの基本機能が使用可能です:

- 工具調整
 β 角と γ 角を指定します。この機能は、このために必要な 3 つの旋回軸の角度を計算します。
- 直接工具調整
2 番目および 3 番目の旋回軸の角度を指定します。この機能は、関連する角度 β および γ と、欠如する第 1 の旋回軸を計算します。

4.13 工具オフセット

注記

旋回軸の順序

ワークから工具への機械の構造を記述するキネマティック結合を実行する場合、6 軸変換の 3 旋回軸の順序で以下の指定が適用されます。

- ワークに最も近い旋回軸が**第 1**の旋回軸です。
- 工具に最も近い旋回軸が**第 3**の旋回軸です。

一般に、第 1 の旋回軸は主軸であり、したがってこの場合には対応する回転は回転フレームを通じて実装されます。

構文

```
<RetVal> = ORISOLH(<Cntrl>,<W1>,<W2>)
```

意味

ORISOLH:	ファンクションコール
----------	------------

4.13 工具オフセット

<RetVal>:	機能の戻り値	
	データタイプ:	INT
	値の範囲:	0, -2, -3, ..., -17
	規格値:	0 機能は異常なく終了しました。
		-2 有効な変換(6 軸オリエンテーション変換)がありません。
		-3 第 1 のパラメータ(<Cntrl>)は負です。
		-4 第 1 のパラメータ(<Cntrl>)のユニットの位は無効です。 値 0 および 1 のみが指定可能です。
		-5 第 1 のパラメータ(<Cntrl>)の 10 の位が無効です。 値 0～3 のみが指定可能です。
		-6 第 1 のパラメータ(<Cntrl>)の 100 の位が無効です。 値 0 および 1 のみが指定可能です。
		-7 第 1 のパラメータ(<Cntrl>)の 1000 の位が無効です。 値 0～3 のみが指定可能です。
		-8 "直接工具調整"機能に対して角度 γ が大きすぎます。
		-9 指定した軸位置の少なくとも 1 つが、"直接工具調整"機能の軸の制限に違反しています。
		-10 工具が有効になっていません。
		-11 要求されたオリエンテーションが設定されていません。
		-12 ハースジョイントの自由な軸角度の調整は、第 1 の解または唯一の解では使用可能ではありません。
		-13 ハースジョイントの自由な軸角度の調整は、第 2 の解では使用可能ではありません。
		-14 ハースジョイントの自由な軸角度の調整は、2 つの解のいずれかで使用可能ではありません。
		-15 第 1 の回転軸はハース軸としてパラメータ設定されています。
		-16 第 2 の回転軸と第 3 の回転軸がハース軸としてパラメータ設定されています。2 軸のうちのいずれかのみをハース軸にできます。

4.13 工具オフセット

		-17	指定した軸位置の少なくとも 1 つが、"直接旋回"機能について、関連するハースジョイントとの互換性がありません。
--	--	-----	--

4.13 工具オフセット

<Cntrl>:	この機能の動作を制御します。	
	データタイプ:	INT
	<Cntrl>パラメータは、10 進コードです(1000 の位に対するユニット):	
	ユニットの位:	ユニットの位は、異常に対する応答を制御します。
	xxx0	異常発生時は(戻り値<0)、アラーム 14106 が出力され、プログラムの処理が中止します。 注記: また、<Cntrl>パラメータが負の場合、ユニットの位の値に関係なくアラームが出力されます。
	xxx1	異常発生時は(戻り値<0)、アラームは出力されません。ユーザーは、プログラムで適切に応答できます。
	10 の位:	ハースジョイントによる旋回軸が存在する場合に、動作を制御します。 注記: このパラメータは、"工具調整"機能について(つまり、100 の位が値"0"である場合に)のみ評価されます。
	xx0x	軸の位置は、最も近い位置に丸み付けされます。
	xx1x	プログラムされた値に対する β 角度の相違が最小限になるように、軸位置が丸み付けされます。
	xx2x	β 角度が、プログラムされた値より小さい最大許容値に一致するように、軸の位置が丸み付けされます(β が切り下げられます)。
100 の位:	実行する機能、または以下の 2 つのパラメータ<W1>および<W2>の内容を指定します。	
	x0xx	"工具調整"機能 パラメータ<W1>および<W2>には以下の意味があります: • <W1> = β • <W2> = γ 旋回軸の関連する角度が計算されます。
	x1xx	"直接工具調整"機能

		<W1>は第 2 の旋回軸の位置指定、<W2>は 6 軸変換の 3 番目の旋回軸の位置指定です。2 つの位置指定と互換性のある第 1 の旋回軸の位置および β と γ の角度を定義します。 異常が発生しなかった場合、2 つの解は常に、 \$P_ORI_POS[<n>, <m>]システム変数に出力します。 第 1 のインデックス(0 または 1)は解を参照し、第 2 のインデックス(0～2)は旋回軸を参照します: • \$P_ORI_POS[0/1, 0]:第 1 の旋回軸の位置 • \$P_ORI_POS[0/1, 1]:角度 β • \$P_ORI_POS[0/1, 2]:角度 γ 位置指定<W1>および<W2>が任意のハースジョイントまたは有効なソフトウェア制限に対して互換性があるかどうかチェックされます。互換性がない場合、対応する異常番号が返されます(<RetVal>パラメータを参照)。 角度<W1>および<W2>を任意に選択した場合、工具の刃先は一般に加工平面にありません。刃先が加工平面から回転する角度 γ は、セッティングデータ SD42999 \$SC_ORISOLH_INCLINE_TOL によって定義される制限値以下であることが必要です。
1000 の位:	100 の位が値"0"である場合、つまり"工具調整"機能では、変更可能な解の位置を指定します。	
	0xxx	計算される軸の位置は、現在の機械軸の位置に可能なかぎり近いことが必要です。
	1xxx	計算されるモジュロ軸の軸位置は、モジュロ範囲の中央に可能なかぎり近いことが必要です。また他の軸では、可能なかぎり 0 に近いことが必要です。非モジュロ軸では、これは軸位置が範囲-180～+180°に削減されることを示します。
	2xxx	計算される軸位置は、軸タイプに関係なく、範囲-180～+180°に削減される必要があります。
<W1>:	第 1 の角度 <Cntrl>パラメータの 100 の位から結果が得られます。	
	データタイプ:	REAL

4.13 工具オフセット

<W1>:	第 2 の角度	
	<Cntrl>パラメータの 100 の位から結果が得られます。	
データタイプ:	REAL	

注記

プログラムされていないパラメータの初期値は"0"です。

詳細情報

検出された解の数と、ORISOLH 機能の実行時の詳細な状態情報は、以下のシステム変数で読み取ることができます:

システム変数	意味		
\$P_ORI_POS [<n>, <m>]	オリエンテーションプログラミングから得られる旋回軸の角度を返します。		
	<n>:	解のインデックス	
		値の範囲:	0, 1
	<m> :	旋回軸のインデックス	
		値の範囲:	0 ... 2
			旋回軸(1〜3)の順序は、\$NT_ROT_AX_NAME の軸の定義に対応します。
ORISOLH 機能の呼び出し時に、"直接工具調整"モード、\$P_ORI_POS[0/1, 1]、および P_ORI_POS[0/1, 2]変数には、2 つの解に属する 2 の角度 β および γ の値が含まれます。 \$P_ORI_POS[<n>, <m>]に入力された第 1 の解(つまりインデックス <n>=0)は常に、要求されたオリエンテーションに直接近づくときに制御によって選択される解です。第 2 のインデックス<m>は、旋回軸、つまり \$NT_ROT_AX_NAME を参照します。 \$P_ORI_POS[<n>, <m>]に入力された軸位置では、\$NK_OFF および \$NK_OFF_FINE に入力されたオフセットが考慮されます。つまり、これらの軸角度は、それ以上変更の必要のないオリエンテーションを設定するために以下のブロックで使用できます。 回転軸がハース軸である場合、解の位置はハースジョイントの最も近い休止位置に丸み付けされます。ハースジョイント回転軸では、正確な解の軸位置と、\$P_ORI_DIFF システム変数のハース増分値に調整した解の軸位置間の差異を読み取ることができます。			

システム変数	意味		
\$P_ORI_DIFF [<n>, <m>]	旋回軸の正確な位置と、オリエンテーションプログラミングから得られる \$P_ORI_POS の位置の差異を返します。		
	<n>:	解のインデックス	
		値の範囲:	0, 1
	<m> :	旋回軸のインデックス	
		値の範囲:	0 ... 2
			旋回軸(1～3)の順序は、\$NT_ROT_AX_NAME の軸の定義に対応します。
位置がインクリメントされる場合(ハースジョイント)、つまり、関連する軸のシステムデータ \$NT_HIRTH_INCR が 0 ではなく、この軸が手動の回転軸である場合にのみ、内容を 0 以外にできます。			

4.13 工具オフセット

システム変数	意味	
\$P_ORI_SOL	複数の旋回軸を含むオリエンテーション変換では、軸角度の計算で指定のオリエンテーションが得られるはずですが、一般に、解は複数存在します。 \$P_ORI_SOL システム変数には、追加の状態情報が含まれる有効な解の番号が格納されます。 \$P_ORI_SOL の内容は、以下のようにコード化されます:	
	値 < 0	一般的な異常状態
		-1 有効な変換に対して、解がまだ計算されていません (ORISOLH が呼び出されていません)。
		-2 変換が有効ではないか、または有効な変換が指定のオリエンテーションプログラミングに対して位置を提供できるオリエンテーション変換(6 軸変換)ではありません。
		-4 適切なオリエンテーションは、現在発生しているキネマティックで設定できません。
		-5 "直接工具調整"モードで ORISOLH 機能が呼び出されたときに、解が見つかりませんでした。
		-6 "直接工具調整"モードでの ORISOLH 機能の呼び出しで、角度 γ が大きすぎます。
		-7 ハースジョイントのために設定できない"直接工具調整"モードで ORISOLH 機能が呼び出されたときに、角度が指定されました。
		-8 第 1 の旋回軸(フレーム軸)は、ハース軸としてパラメータ設定しないでください。
		-9 第 2 の回転軸と第 3 の回転軸がハース軸としてパラメータ設定されています。2 軸のうちのいずれかのみをハース軸にできます。
		-10 ハースジョイントの解の補正が見つかりませんでした。
	値 > 0 ユニット の位	軸の制限および異常条件を考慮せずに数学的に可能な解の数。
		0 解はありません、つまり、要求されたオリエンテーションは設定できません。 この場合、3 つの異なる原因が考えられます: 原則として、要求されたオリエンテーションは、旋回軸の自由な移動範囲であっても、機械キネマ

システム変数	意味
	ティック(直角に配置されていない旋回軸)のために実現できません。この場合、\$P_ORI_SOL の 10 の位および 100 の位はともに 0 であり、旋回軸に割り当てられた \$P_ORI_STAT 状態変数の値は"-4"です。 - 軸の制限に違反するので、計算された解は実現できません。軸の制限なしで得られる旋回軸の位置は、\$P_ORI_POS で読み取ることができます。 - 位置が計算される第 1 の旋回軸に平行に位置合わせされる工具のオリエンテーションベクトルまたはオリエンテーション法線ベクトルのいずれかが得られる"直接工具調整"モードで ORISOLH 機能が呼び出されたときに、軸位置が指定されました。この場合、この軸の位置は定義されません。
	1 解が存在します。 この場合、3 つの異なる原因が考えられます： - 指定のオリエンテーションおよび機械のキネマティックに基づいて、軸の制限を考慮しない場合でも 1 つの解のみが存在します(数学的な観点からは 2 つの一致する解)。この状況は、直角ではないキネマティックのオリエンテーション範囲の刃先で発生します。\$P_ORI_POS には、両方の(同じ)解が含まれます。 - 第 2 の解は軸の制限違反により無効であるために、1 つの解のみが存在します。有効な解は常に、\$P_ORI_POS の第 1 の解です。軸の制限を考慮しない場合に得られる第 2 の解は、\$P_ORI_POS で読み取ることができます。 - ORISOLH 機能が"直接工具調整"モードで呼び出される場合、これは通常の状態です。2 つの旋回軸の指定された軸位置について、欠如する旋回軸に対して計算すべき 1 つの有効な位置のみが一般に存在します。
	2 2 つの解があります。
	8 無限数の解があります。つまり、旋回軸(極軸)の位置は任意です。ただし、他の軸の 2 つの可能な位置から、軸の制限違反のために 1 つは除外されます。
	9 無制限数の解があります。つまり、旋回軸(極軸)の位置は無限です。無制限の軸は、数百の位置または \$P_ORI_STAT システム変数から決定できます。

4.13 工具オフセット

システム変数	意味	
	値 > 0 10 の位:	違反する軸の制限のビット指定による表示。異常の正確な原因は、\$P_ORI_STAT システム変数から特定することができます。
		ビット 0 (値 10): 少なくとも 1 つの解について、第 1 の旋回軸の少なくとも 1 つの軸の制限に違反します。
		ビット 1 (値 20): 少なくとも 1 つの解について、第 2 の旋回軸の少なくとも 1 つの軸の制限に違反します。
		Bit 2 (値 40): 少なくとも 1 つの解について、第 3 の旋回軸の少なくとも 1 つの軸の制限に違反します。
	値 > 0 100 の位	未定義の軸位置のビット指定による表示(無限数の解がある場合、つまりユニット位置が"9"と同じである場合にのみ発生することがあります)。
		ビット 0 (値 100): 第 1 の旋回軸の位置が定義されていません。
		ビット 1 (値 200): 第 2 の旋回軸の位置が定義されていません。
		ビット 2 (値 400): 第 3 の旋回軸の位置が定義されていません。
	第 1 旋回軸、第 2 旋回軸、第 3 旋回軸の名称は、\$NT_ROT_AX_NAME の軸の定義を参照します。	

システム変数	意味	
\$P_ORI_STAT [<n>]	ORISOLH の呼び出し後、最大 3 旋回軸のそれぞれの状態を返します。	
	<n>:	旋回軸のインデックス ($\$NT_ROT_AX_NAME$ の関連する旋回軸のインデックスに対応します)
		値の範囲: 0 ... 2 旋回軸(1~3)の順序は、 $\$NT_ROT_AX_NAME$ の軸の定義に対応します。
	$\$P_ORI_STAT$ の内容は、以下のようにコード化されます:	
	値 < 0	一般的な異常状態
		-1 状態が定義されていません(ORISOLH が呼び出されていません)。
		-2 変換が有効ではないか、または有効な変換が指定のオリエンテーションプログラミングに対して位置を提供できるオリエンテーション変換(6 軸変換)ではありません。
		-3 軸が有効な変換に含まれていません。
		-4 任意に想定される軸の移動範囲を持つ場合でも、要求されたオリエンテーションが現在のキネマティックにより実現できないために、軸の位置は計算できません。
		-5 位置が計算される第 1 の旋回軸に平行に位置合わせされる工具のオリエンテーションベクトルまたはオリエンテーション法線ベクトルのいずれかが得られる"直接工具調整"モードで ORISOLH 機能が呼び出されたときに、軸位置が指定されました。この場合、この軸の位置は定義されません。
		-6 "直接工具調整"モードでの ORISOLH 機能の呼び出しで、角度 γ が大きすぎます。
		-7 ハースジョイントのために設定できない"直接工具調整"モードで ORISOLH 機能が呼び出されたときに、角度が指定されました。
		-8 第 1 の旋回軸(フレーム軸)は、ハース軸としてパラメータ設定しないでください。

4.13 工具オフセット

システム変数	意味	
		-9 第2の回転軸と第3の回転軸がハース軸としてパラメータ設定されています。2軸のうちのいずれかのみをハース軸にできます。
		-10 ハースジョイントの解の補正が見つかりませんでした。
	値 > 0	第1の解の軸の制限違反に対応するビット指定の表示。
	ユニットの位	ビット 0 (値 1): 第1の解が軸の下限に違反します。
		ビット 1 (値 2): 第1の解が軸の上限に違反します。
	値 > 0	第2の解の軸の制限違反に対応するビット指定の表示。
	10 の位:	ビット 0 (値 10): 第2の解が軸の下限に違反します。
		ビット 1 (値 20): 第2の解が軸の上限に違反します。
	値 > 0	未定義の軸位置の表示。
	100 の位	ビット 0 (値 100): 旋回軸の位置が定義されていません。つまり、要求されたオリエンテーションは、回転軸(極位置)の任意の各設定により実現します。この情報は、\$P_ORI_SOL システム変数に含まれています。
軸制限の違反を示す異常番号のうち、いくつかが同時発生する場合があります。軸の制限に違反する場合、360°の倍数を加算または減算することにより、許容可能な軸制限内の位置への到達が試みられます。これが可能でない場合、軸の下限または上限の制限に違反するかどうかは明らかには定義されません。 要求されたオリエンテーションに対応する解がない場合 (\$P_ORI_SOL = 0)、変換時の旋回軸の状態は"0"です。		

注記**\$NT_ROT_AX_NAME**

このシステム変数は、オリエンテーションの設定のために最大 3 軸を参照します。キネマティックトランスフォーメーションから得られるオリエンテーション移動を実行する必要がある機械軸(回転軸)を定義する結合エレメント(\$NK_NAME)の名前が含まれます。これはキネマティック結合の構造から導出されるので、最大 3 つの回転軸がこのシステム変数に含まれる順序は、機械キネマティックには関連しません。ただし、これによって他の変数が回転軸にアクセスする順序が定義されるので、\$NT_ROT_AX_NAME の旋回軸の順序はキネマティックの記述に一致する必要があります。

注記**ステータス情報**

たとえば、オリエンテーションが実現できないか、関連する軸の制限に違反する場合にのみ実現できることを示す状態情報では、NC アラームはトリガされません。ユーザーが自身の責任で指定された条件に適切に応答する必要があります。

4.13.12.2 旋回工具のオフセットデータの変更の起動(CUTMOD、CUTMODK)

旋回工具のオフセットデータの変更は、CUTMOD(旋回工具ホルダと組み合わせた場合)または CUTMODK 言語命令(キネマティック結合によって定義されたオリエンテーション変換の場合)を通じて NC プログラムで起動されます。

注記

キネマティック結合によって定義された旋回工具ホルダおよびオリエンテーション変換は、同時には有効にできず、2 タイプ間で競合は発生しません。

構文

CUTMOD = <値>

または

CUTMODK = <命令>

4.13 工具オフセット

意味

CUTMOD:	旋回工具ホルダと組合わせたファンクションコール
---------	-------------------------

<値>:	割り当て値	
	データタイプ:	INT
	値:	0 機能は解除されます。 システム変数\$P_AD...で提供される値は、対応する工具パラメータと同じです。
		> 0 指定した番号の旋回工具ホルダが有効な場合、この機能が起動します。つまり、この起動は、指定した旋回工具ホルダにリンクされます。 システム変数\$P_AD...で提供される値は、有効な回転に応じて、対応する工具パラメータに対して修正することができます。 指定の旋回工具ホルダを解除すると、一時的にこの機能が解除されます。別の旋回工具ホルダを起動すると、永続的にこの機能が解除されます。このことが、第 1 の状況で同じ旋回工具ホルダを再度選択したときに機能が再起動する理由です。第 2 の状況では、新たな選択が必要です。以降は、指定の番号の旋回工具ホルダが再起動されます。 この機能は、リセットの影響を受けません。
		-1 旋回工具ホルダが有効な場合、この機能は常に起動されます。 工具ホルダを変更する場合、または選択解除後に新たに選択する場合、再度 CUTMOD を設定する必要はありません。
		-2 現在有効な旋回工具ホルダと同じ番号の旋回工具ホルダが有効な場合、この機能は常に有効になります。 旋回工具ホルダが有効でない場合は、CUTMOD=0 と同じ意味になります。 旋回工具ホルダが有効な場合は、現在の工具ホルダ番号を直接指定する場合と同じ意味になります。
	< -2	2 未満の値は無視されます。すなわちこの場合は、CUTMOD がプログラム指令されなかったかのように処理されます。

4.13 工具オフセット

			注: この数値の範囲は、この後の拡張の可能性のために予約されているので使用しないでください。
CUTMODK:	キネマティック結合によって定義されたオリエンテーション変換と組み合わせたファンクションコール		
<命令>:	割り当てられた命令		
	データタイプ:	STRING	
	値:	"NEW"	"オフセットデータの変更"に関連するキネマティック結合により定義された有効な変換の状態、変換の名前、および現在の輪郭フレームが保存されます。 注: この命令は、適切な変換(TRAORI_DYN、TRAORI_STAT、または TRAANG_K)が有効な場合にのみ許容されます。
		「OFF」	有効な"オフセットデータの変更"をオフに切り替えます。以前に"NEW"により格納されたデータが保持されます。 注: またこの命令は、CUTMODK が有効でない場合にのみ許容されます。そうでない場合は起動しません。"オフセットデータの変更"に対してすでに設定されているすべてのデータセットが保持されます。
		"ON"	この命令により、以前に"NEW"命令で保存されたデータセットで"オフセットデータの変更"が再起動されます。 この命令の実行時に、保存されているデータセット名の変換が有効である場合、"オフセットデータの変更"が直ちに有効になります。それ以外の場合は、有効な変換が起動されるまで、起動は遅延します。
		"CLEAR"	"OFF"命令の場合と同様に、"オフセットデータの変更"をオフに切り替え、保存されたデータセットも削除します。 注: またこの命令は、CUTMODK が有効でない場合にのみ許容されます。

注記

SD42984 \$SC_CUTDIRMOD

CUTMOD/CUTMODK 命令では、セッティングデータ SD42984 \$SC_CUTDIRMOD を使用して起動できる機能を置換します。ただし、このセッティングデータで起動した機能は、変更されずそのまま使用可能です。この両機能を並行して使用することは無意味なため、CUTMOD がゼロに等しい場合にのみ、セッティングデータで有効にできます。

詳細情報

変更されたオフセットデータの読み取り

変更されたオフセットデータは、以下のシステム変数および OPI 変数で提供されます:

意味	システム変数	OPI 変数
刃先位置	\$P_AD[2]	cuttEdgeParam2
ホルダ角度	\$P_AD[10]	cuttEdgeParam10
切削方向	\$P_AD[11]	cuttEdgeParam11
逃げ角	\$P_AD[24]	cuttEdgeParam24

"旋回工具のオフセットデータの変更"機能が CUTMOD または CUTMODK 命令により起動され、工具が旋回工具ホルダまたは適切なオリエンテーション変換によって回転したときに、データは常に、対応する工具パラメータ(\$TC_DP2[..., ...] など)に対して変更されます。

その他の機能に関連するシステム変数

システム変数	意味
\$P_CUTMOD_ANG / \$AC_CUTMOD_ANG	有効な加工平面、および CUTMOD 機能と CUTMODK 機能で使用可能な変更された刃先データの工具の回転角度を返します。
\$P_CUTMOD / \$AC_CUTMOD	CUTMOD 命令(オフセットデータの変更を起動する工具ホルダの番号)で最後にプログラムされた現在有効な値を読み取ります。 最後にプログラムされた値が CUTMOD = -2 (現在有効な旋回工具ホルダによる起動)である場合、システム変数に値"-2"は返されず、プログラミング時に有効な旋回工具ホルダの番号が返されます。
\$P_CUTMODK / \$AC_CUTMODK	"オフセットデータの変更"に対して現在有効なデータセットが作成された変換の名前を読み取ります。

4.13 工具オフセット

システム変数	意味
\$P_CUT_INV / \$AC_CUT_INV	工具の回転によって主軸回転方向を反転しなければならない場合、値 TRUE が提供されます。これをおこなうには、読み取り操作の対象となるブロックで、以下の 4 つの条件を満たしてください。 1. 旋削工具または研削工具が有効です (工具タイプ 400~599 や SD42950 \$SC_TOOL_LENGTH_TYPE = 2)。 2. オフセットデータの変更は、CUTMOD 命令または CUTMODK 命令で起動されました。 3. CUTMOD 命令または CUTMODK 命令で選択され、キネマティック結合により定義された旋回工具ホルダまたはオリエンテーション変換が有効です。 4. 工具刃先の得られる法線が 90°を超えて(一般的には 180°)初期位置を基準に回転するように、工具は旋回工具ホルダまたはキネマティックオリエンテーション変換によって回転します。 指定した 4 つの条件のうち少なくとも 1 つが満たされていない場合、この変数は値 FALSE を返します。刃先位置が定義されていない工具の場合、変数の値は常に FALSE です。

システム変数	意味																		
\$P_CUTMOD_ERR	CUTMOD 機能の前の呼び出し後に異常状態 CUTMOD 機能も、工具交換のために自動的に呼び出すことができます。リセット時には、この変数はゼロにリセットされます。工具交換のたびにリセットされ、必要な場合には再度書き込まれます。この変数はビット指定です。このビットの意味は以下のとおりです: <table> <tr> <td>ビット 0:</td><td>有効な工具に対して利用できる切削方向が定義されていません。</td></tr> <tr> <td>ビット 1:</td><td>有効な工具の刃先角度(逃げ角とホルダ角度)が、両方ともゼロです。</td></tr> <tr> <td>ビット 2:</td><td>有効な工具の逃げ角が許容されない値(0°未満または 180°超)です。</td></tr> <tr> <td>ビット 3:</td><td>有効な工具のホルダ角度が許容されない値(0°未満または 90°超)です。</td></tr> <tr> <td>ビット 4:</td><td>有効な工具のプレート角度が許容されない値(0°未満または 90°超)です。</td></tr> <tr> <td>ビット 5:</td><td>有効な工具の刃先位置 - ホルダ角度の組み合わせが許容されません(刃先位置 1~4 のホルダ角度は ≤ 90 度、刃先位置 5~8 のホルダ角度 ≥ 90 度でなければならない)。</td></tr> <tr> <td>ビット 6:</td><td>有効な工具の不正な回転。 工具は有効な加工平面から $\pm 90^\circ$ 回転しました(許容値は約 1°)。したがって、刃先位置はもう加工平面では定義されません。</td></tr> <tr> <td>ビット 7:</td><td>切削プレートが加工平面内になく、切削プレートと加工平面の角度がセッティングデータ SD42998 \$SC_CUTMOD_PLANE_TOL によって指定された上限を超過しています。</td></tr> <tr> <td>ビット 8:</td><td>切削プレートが加工平面内にありません。角度 α が 1° を超えています。角度 α は、角度 β の回転軸および角度 γ の回転軸(G18 の X 軸)と直交する座標軸を中心とした回転角です。</td></tr> </table>	ビット 0:	有効な工具に対して利用できる切削方向が定義されていません。	ビット 1:	有効な工具の刃先角度(逃げ角とホルダ角度)が、両方ともゼロです。	ビット 2:	有効な工具の逃げ角が許容されない値(0°未満または 180°超)です。	ビット 3:	有効な工具のホルダ角度が許容されない値(0°未満または 90°超)です。	ビット 4:	有効な工具のプレート角度が許容されない値(0°未満または 90°超)です。	ビット 5:	有効な工具の刃先位置 - ホルダ角度の組み合わせが許容されません(刃先位置 1~4 のホルダ角度は ≤ 90 度、刃先位置 5~8 のホルダ角度 ≥ 90 度でなければならない)。	ビット 6:	有効な工具の不正な回転。 工具は有効な加工平面から $\pm 90^\circ$ 回転しました(許容値は約 1°)。したがって、刃先位置はもう加工平面では定義されません。	ビット 7:	切削プレートが加工平面内になく、切削プレートと加工平面の角度がセッティングデータ SD42998 \$SC_CUTMOD_PLANE_TOL によって指定された上限を超過しています。	ビット 8:	切削プレートが加工平面内にありません。角度 α が 1° を超えています。角度 α は、角度 β の回転軸および角度 γ の回転軸(G18 の X 軸)と直交する座標軸を中心とした回転角です。
ビット 0:	有効な工具に対して利用できる切削方向が定義されていません。																		
ビット 1:	有効な工具の刃先角度(逃げ角とホルダ角度)が、両方ともゼロです。																		
ビット 2:	有効な工具の逃げ角が許容されない値(0°未満または 180°超)です。																		
ビット 3:	有効な工具のホルダ角度が許容されない値(0°未満または 90°超)です。																		
ビット 4:	有効な工具のプレート角度が許容されない値(0°未満または 90°超)です。																		
ビット 5:	有効な工具の刃先位置 - ホルダ角度の組み合わせが許容されません(刃先位置 1~4 のホルダ角度は ≤ 90 度、刃先位置 5~8 のホルダ角度 ≥ 90 度でなければならない)。																		
ビット 6:	有効な工具の不正な回転。 工具は有効な加工平面から $\pm 90^\circ$ 回転しました(許容値は約 1°)。したがって、刃先位置はもう加工平面では定義されません。																		
ビット 7:	切削プレートが加工平面内になく、切削プレートと加工平面の角度がセッティングデータ SD42998 \$SC_CUTMOD_PLANE_TOL によって指定された上限を超過しています。																		
ビット 8:	切削プレートが加工平面内にありません。角度 α が 1° を超えています。角度 α は、角度 β の回転軸および角度 γ の回転軸(G18 の X 軸)と直交する座標軸を中心とした回転角です。																		

\$P_...:先読み変数

\$AC_...:メインラン変数

すべてのメインラン変数はシンクロナイズドアクションで読み取ることができます。先読みから読み取りアクセス操作をおこなうと、先読み停止が発生します。

4.13 工具オフセット

平面変更

修正された刃先位置、切削方向および工具ホルダまたは逃げ角を求めるために、有効な平面 (G17～G19)の刃先の評価が重要です。

ただし、セッティングデータ SD42940 \$SC_TOOL_LENGTH_CONST (平面選択時の工具長成分の変更)がゼロ以外の有効値(正または負の 17、18、または 19)の場合は、その内容で、当該の数量を使用する平面を定義します。

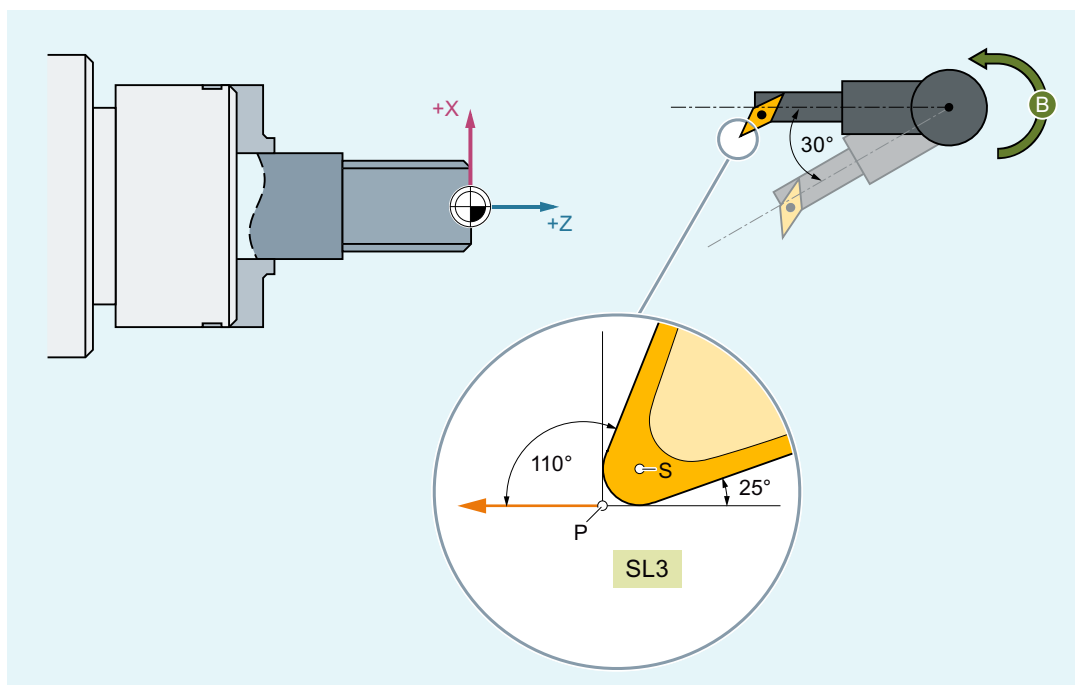
G コードに対するセッティングデータのこの優先規則は、マシンデータ \$MC_TOOL_PARAMETER_DEF_MASK のビット 18 を設定することにより解除できます。これは、このビットが設定された場合に、グループ 6 の G 命令で定義された平面が依然として有効であることを示します。

修正された切削データの効果

修正された刃先位置と修正された刃先基準点は、すでに有効な工具に対しても、プログラミング時に直ちに有効になります。このために、工具を再選択する必要はありません。

例

刃先位置が 3 で、B 軸を中心に工具を回転させる旋回工具ホルダを持つ工具の場合、CUTMOD 命令を使用して、工具の回転後に刃先位置を変更してください。



S: 刃先の中心点

P: 刃先基準点

SL: 刃先位置

プログラムコード	コメント
N10 \$TC_DP1[1,1]=500	
N20 \$TC_DP2[1,1]=3	; 刃先位置
N30 \$TC_DP3[1,1]=12	
N40 \$TC_DP4[1,1]=1	
N50 \$TC_DP6[1,1]=6	
N60 \$TC_DP10[1,1]=110	; ホルダ角度
N70 \$TC_DP11[1,1]=3	; 切削方向
N80 \$TC_DP24[1,1]=25	; 逃げ角
N90 \$TC_CARR7[2]=0 \$TC_CARR8[2]=1 \$TC_CARR9[2]=0	; B 軸
N100 \$TC_CARR10[2]=0 \$TC_CARR11[2]=0	; C 軸
\$TC_CARR12[2]=1	
N110 \$TC_CARR13[2]=0	
N120 \$TC_CARR14[2]=0	
N130 \$TC_CARR21[2]=X	

4.13 工具オフセット

プログラムコード	コメント
N140 \$TC_CARR22[2]=X	
N150 \$TC_CARR23[2]="M"	
N160 TCOABS CUTMOD=0	
N170 G18 T1 D1 TCARR=2	; X Y Z
N180 X0 Y0 Z0 F10000	; 12.000 0.000 1.000
N190 \$TC_CARR13[2]=30	
N200 TCARR=2	
N210 X0 Y0 Z0	; 10.892 0.000 -5.134
N220 G42 Z-10	; 8.696 0.000 -17.330
N230 Z-20	; 8.696 0.000 -21.330
N240 X10	; 12.696 0.000 -21.330
N250 G40 X20 Z0	; 30.892 0.000 -5.134
N260 CUTMOD=2 X0 Y0 Z0	; 8.696 0.000 -7.330
N270 G42 Z-10	; 8.696 0.000 -17.330
N280 Z-20	; 8.696 0.000 -21.330
N290 X10	; 12.696 0.000 -21.330
N300 G40 X20 Z0	; 28.696 0.000 -7.330
N310 M30	

コメントの数値は、機械座標(MCS)でのブロック終点位置を、X→Y→Zの順で示しています。

説明

ブロック N180 では、最初に CUTMOD=0 および回転しない旋回工具ホルダの工具を選択します。旋回工具ホルダのすべてのオフセットベクトルが 0 であるため、\$TC_DP3[1,1]と \$TC_DP4[1,1]で指定した工具の長さに対応する位置へアプローチします。

B 軸を中心に 30°回転する旋回工具ホルダがブロック N200 で有効です。CUTMOD=0 により刃先位置は変化しないため、以前と同様に既存の刃先基準点で決まります。このため、ブロック N210 では、位置へのアプローチにより、以前の刃先基準点がゼロのまま保持されます(つまり、ベクトル(1, 12)が Z/X 平面で 30°回転します)。

ブロック N260 では、ブロック N200 とは逆に、CUTMOD=2 が有効です。旋回工具ホルダの回転の結果、刃先位置は 8 に変更されます。またこれによって、軸位置が逸脱します。

工具径補正(TRC)は、ブロック N220 と N270 の両方またはいずれかで有効です。両方のプログラム区間で刃先位置が異なっても、TRC が有効なブロックの終了位置には影響しません。したがって、対応する位置は同じです。異なった刃先位置は、解除ブロック N260 と N300 の両方またはいずれかでのみ、再度有効になります。

4.13.13 工具環境の使用

機能の概要

- 工具環境の保存(TOOLENV) (ページ 933)
- 工具環境の削除(DELTTOOLENV) (ページ 936)
- T 番号、D 番号、および DL 番号の読み出し (GETTENV) (ページ 937)
- 工具長、工具長成分、またはその両方の読み取り (GETTCOR) (ページ 939)
- 工具成分の変更(SETTCOR) (ページ 947)

システム変数の概要

- 保存した工具環境に関する情報の読み取り (\$P_TOOLENVN、\$P_TOOLENV) (ページ 939)

4.13.13.1 工具環境の保存(TOOLENV)

TOOLENV 機能を使用すれば、メモリに格納されている工具データの評価に必要な現在の状態を保存できます。

個々のデータは、次のとおりです。

- 次のグループで有効な G 命令:
 - 6 (G17, G18, G19)
 - 56 (TOWSTD, TOWMCS, TOWWCS, TOWBCS, TOWTCS, TOWKCS)
- 有効な径方向軸
- マシンデータ:
 - MD18112 \$MN_MM_KIND_OF_SUMCORR (TO 領域内のサムオフセットのプロパティ)
 - MD20360 \$MC_TOOL_PARAMETER_DEF_MASK (工具パラメータの定義)

4.13 工具オフセット

- セッティングデータ:
 - SD42900 \$SC_MIRROR_TOOL_LENGTH (ミラーリング時の工具長符号変更)
 - SD42910 \$SC_MIRROR_TOOL_WEAR (ミラーリング時の工具摩耗の符号変更)
 - SD42920 \$SC_WEAR_SIGN_CUTPOS (刃先位置のある工具の摩耗の符号)
 - SD42930 \$SC_WEAR_SIGN (摩耗の符号)
 - SD42935 \$SC_WEAR_TRANSFORM (工具成分の座標変換)
 - SD42940 \$SC_LENGTH_CONST (平面の変更による工具長成分の変更)
 - SD42942 \$SC_TOOL_LENGTH_CONST_T (平面の変更による工具長成分の変更)
 - SD42950 \$SC_TOOL_LENGTH_TYPE (工具タイプと無関係な工具長成分の割り当て)
 - SD42954 \$SC_TOOL_ORI_CONST_M (平面の変更によるフライス工具の工具方向成分の変更)
 - SD42956 \$SC_TOOL_ORI_CONST_T (平面の変更による旋削工具の工具方向成分の変更)
- 現在のすべてのフレームの向きコンポーネント(回転とミラーリング、作業オフセットまたはスケーリングなし)
- 向きコンポーネント、および有効な旋回工具ホルダの結果として生じた長さ
- 向きコンポーネント、および有効な座標変換の結果として生じた長さ

工具の環境を記述するデータに加えて、有効な工具の T 番号、D 番号、および DL 番号も格納されます。この結果、工具を再度指定しなくても、TOOLENV 呼び出しと同じ環境で、後ほどこの工具にアクセスできます。

構文

<状態> = TOOLENV (<名前>)

意味

TOOLENV (...)	工具環境を保存するための定義済み機能	
:	単独ブロック	あり
	指令:	

<状態>:		機能の戻り値。負の値は異常状態を示します。		
		データタイプ: INT		
		値:	0	動作 OK
			-1	工具環境に予約されたメモリがありません。 MD18116 \$MN_MM_NUM_TOOL_ENV = 0 これは、「工具環境」機能が使用できないことを示します。
			-2	工具環境に対応した空きメモリ場所がこれ以上ありません。
			-3	工具環境名として空文字列は不適切です。
			-4	パラメータ(<名前>)が指定されていません。
パラメータ				
1	<名称>:	現在のデータセットを保存するときに使用する名前。 同じ名前のデータセットがすでにある場合、このデータセットは上書きされます。この場合、Status は 0 になります。		
		データタイプ:	STRING	

詳細情報

基本寸法/アダプタ寸法の工具長補正

工具マガジン管理が有効な場合(「工具管理」オプションのみで使用可能)、次のマシンデータの値で、アダプタ長さまたは工具ベース寸法(刃先別パラメータ\$TC_DP21、\$TC_DP22、および\$TC_DP23)を工具長の計算に組み込むかどうかを定義します。

MD18104 \$MN_MM_NUM_TOOL_ADAPTER (TO 領域内の工具アダプタ)

このマシンデータに加えた変更が効果を現すのはコントロールシステムの電源オン後に限るため、この変更は工具環境に保存されません。

旋回工具ホルダと座標変換の結果として生じる長さ:

注記

方向機能および座標変換を含む両工具ホルダでは、工具の長さの追加成分として、実行される回転に部分的にまたは完全に適用できるシステム変数またはマシンデータを使用できます。得られた工具長さの追加成分も、工具を使用する環境の一部を表すために、TOOLENV の呼び出し時に保存する必要があります。

4.13 工具オフセット

アダプタ座標変換

アダプタ座標変換は、工具アダプタのプロパティであるため、すべての工具のプロパティでもあります。このため、アダプタ座標変換は、工具環境の一部ではなく、別の工具に適用できます。

工具の全長を特定するのに必要な全データを保存すれば、工具が無効になっても、または環境の条件(たとえば、G コードまたはセッティングデータ)が変更されていても、これより後の時点で工具の有効長さを計算できます。同様に、状態を保存した工具と同じ条件下で使用すると仮定して、別の工具の有効長さを計算できます。

工具環境のデータセットの最大数

マシンデータ MD18116 \$MN_MM_NUM_TOOL_ENV は、工具環境を記述するために保存できるデータセットの最大数を定義します。このデータは、TOA 領域に格納されます。また、コントロールシステムをオフにしても、データは維持されます。

データをバックアップすることはできません。つまり、異なるコントローラシステム間でこのデータを伝送することはできません。

4.13.13.2 工具環境の削除(DELTOOLENV)

DELTOOLENV 機能は、工具環境の記述に使用されるデータセットを削除します。削除とは、特定の名前で格納されているデータセットにアクセスできなくなる(アクセスしようとするときアラームが発生する)ことを指します。

注記

データセットを削除できるのは、DELTOOLENV 機能を使用する場合、INITIAL.INI ダウンロードによる場合、またはコールドスタートの(初期設定マシンデータを指定して NC が起動する場合)に限ります。自動削除操作が追加実行されることはありません。

構文

<状態> = DELTOOLENV (<名前>)
<状態> = DELTOOLENV ()

意味

DELTOOLENV (.. .):	工具環境を削除するための定義済み機能	
	単独ブロック 指令:	あり

<状態>:		機能の戻り値。負の値は異常状態を示します。		
		データタイプ: INT		
		値:	0	動作 OK
			-1	工具環境に予約されたメモリがありません。 MD18116 \$MN_MM_NUM_TOOL_ENV = 0 これは、「工具環境」機能が使用できないことを示します。
-2	指定名の工具環境が存在しません。			
パラメータ				
1	<名称>:	削除するデータセットの名前		
		データタイプ:	STRING	
DELTOOLENV ():		DELTOOLENV () は、名前を指定せずに工具環境を記述するデータセットを削除します。		

4.13.13.3 T 番号、D 番号、および DL 番号の読み出し (GETTENV)

GETTENV 機能を使用すれば、工具環境内に格納された T、D、および DL の各番号を読み取ることができます。

構文

<状態> = GETTENV(<名前>, <TDDL>)

意味

GETTENV(...):	工具環境を記述するデータセット内の T 番号、D 番号、および DL 番号を読み取る事前定義された機能	
	単独ブロック 指令:	あり

4.13 工具オフセット

<状態>:		機能の戻り値。負の値は異常状態を示します。		
		データタイプ:	INT	
		値:	0	動作 OK
			-1	工具環境に予約されたメモリがありません。 MD18116 \$MN_MM_NUM_TOOL_ENV = 0 これは、「工具環境」機能が使用できないことを示します。
		-2	指定名の工具環境が存在しません。	
パラメータ				
1	<名称>:	T、D、および DL の各番号を読み取るデータセットの名前		
		データタイプ:	STRING	
2	<TDDL>:	この結果パラメータのフィールドには、工具の T 番号、D 番号、および DL 番号が含まれ、これらの工具環境は次のように指定のデータセットに保存されます。 • <TDDL> [0]:T 番号 • <TDDL> [1]:D 番号 • <TDDL> [2]:DL 番号		
		データタイプ:	INT[3]	
GETTENV (, <TDDL>) , GETTENV (" " , <TDDL>) :		GETTENV 機能を読み出す場合、1 番目のパラメータを省略するか、1 番目のパラメータとして空文字列を送送することができます。この 2 つの特殊な場合のどちらでも、<TDDL> で、有効な工具の T、D、および DL 番号が戻されます。		

4.13.13.4 保存した工具環境に関する情報の読み取り(\$P_TOOLENVN、\$P_TOOLENV)

保存された工具環境に関する情報は、次のシステム変数を使用して読み取ることができます。

\$P_TOOLENVN:	工具環境を記述するために、TOOLENV を使用して定義された(まだ削除されていない)データセットの数を指定します			
	構文:	<n> = \$P_TOOLENVN		
	意味	<n>:	定義されたデータセットの数	
			データタイプ:	INT
			値の範囲:	0 ... MD18116 \$MN_MM_NUM_TOOL_ENV
工具環境が使用できない場合(MD18116 = 0)も、このシステム変数にアクセスできます。この場合、戻り値は「0」になります。				

\$P_TOOLENV:	工具環境を記述する<i>番目のデータセットの名前を指定します			
	構文:	<名前> = \$P_TOOLENV [<i>]		
	意味	<名称>:	番号<i>で指定されたデータセットの名前	
			データタイプ:	STRING
		<i>:	データセットの番号	
			データタイプ:	INT
			値の範囲:	1 ... \$P_TOOLENVN
データセットへの番号の割り当ては、固定されていないため、データセットを削除や作成すると変更されます。データセットは内部的に番号が付けられます。				
<i>が定義されていないデータセットを参照する場合、空文字列が返されます。				
インデックス<i>が有効でない場合、つまり<i>が 1 より小さいか、または工具環境(MD18116 \$MN_MM_NUM_TOOLENV)のデータセットの最大数より大きい場合、次のアラームが出力されます。				
アラーム 17020 「配列インデックス 1 が許可されません」				

4.13.13.5 工具長、工具長成分、またはその両方の読み取り(GETTCOR)

GETTCOR 機能を使用して、工具長または工具長コンポーネントを読み出すことができます。

各パラメータを使用して、対象とするコンポーネントと、工具を使用するときの条件を指定できます。

4.13 工具オフセット

構文

```
<Status> = GETTCOR(<Len>[, <Comp>, <Stat>, <T>, <D>, <DL>])
```

意味

GETTCOR(...):		工具の長さを読み取るか、または工具の長さのコンポーネントを読み取るために事前定義された機能	
単独ブロック 指令:		あり	
<状態>:		機能の戻り値。負の値は異常状態を示します。	
		データタイプ: INT	
		値:	0 動作 OK
			-1 工具環境に予約されたメモリがありません。 MD18116 \$MN_MM_NUM_TOOL_ENV = 0 これは、「工具環境」機能が使用できないことを示します。
			-2 <状態>で指定された名前の工具環境が存在しません。
			-3 パラメータ<Comp>の文字列が無効です。 この異常の発生原因には、文字列が無効な場合または同じ文字列が 2 回プログラム指令されている場合があります。
			-4 T 番号が無効です。
			-5 D 番号が無効です。
			-6 DL 番号が無効です。
			-7 アクセスしようとしたメモリモジュールは存在しません。
			-8 アクセスしようとしたオプションは存在しません(プログラム可能な工具向き、工具管理機能)。
			-9 <Comp>文字列には、コロン(座標系の指定用識別子)が含まれていますが、その後に、座標系を示す有効な文字が指定されていません。
パラメータ			

1	<Len>:	得られるベクトル
		データタイプ: REAL[11]
		ベクトルコンポーネントは、次の順序で配列されます。 • <Len> [0]:工具タイプ • <Len> [1]:刃先位置 • <Len> [2]:横座標 • <Len> [3]:縦座標 • <Len> [4]:垂直座標 • <Len> [5]:工具半径 <Comp>および<Stat>で定義された座標系は、長さコンポーネントの基準座標系として使用されます。<Comp>に座標系が定義されていない場合、工具長は機械座標系で表示されます。 ジオメトリ軸への横軸、縦軸、および垂直軸の割り当ては、工具環境内で有効な平面によって違ってきます。つまり G17 では、縦軸は X に平行であり G18 では、Z 軸に平行になります。 <Len>[6]～<Len>[10]の各コンポーネントには、追加のパラメータが含まれており、これらを使って工具のジオメトリの記述を指定できます(たとえば、ジオメトリの場合は\$TC_DP7～\$TC_DP11、摩耗またはサムオフセットおよびセットアップオフセットの場合は対応するコンポーネント)。 これらの 5 つの追加要素と工具半径が定義されるのは、E、G、S、および W の各コンポーネントの場合に限ります。この評価は、<Stat>に左右されません。このため、<Len>[6]～<Len>[10]の対応する値をゼロ以外の値にできるのは、指定された 4 つのコンポーネントのうち少なくとも 1 つが工具長計算に関係している場合に限ります。残りのコンポーネントは、結果に影響を及ぼしません。各寸法は、制御の基本単位系(インチまたはメートル)を参照します。

4.13 工具オフセット

2	<Comp>:	工具長コンポーネント(オプション)		
		データタイプ:	STRING	
		この文字列は、2 つの 2 次文字列で構成され、各 2 次文字列はコロンで区切られます。		
		一般形式: "<SubStr_1> [: <SubStr_2>]"		
		<SubStr_1>: :	1 番目の 2 次文字列は、工具長の計算時に考慮する工具長コンポーネントを示します。	
			2 次文字列内の文字の順序、およびその表記法(大文字か小文字か)は自由です。ブランクまたはタブは、文字の間に自由に挿入できます。	
注: 2 次文字列内で文字を 2 回プログラムすることはできません。				
文字:	-		マイナス符号(1 番目の文字としてのみ使用できます) すべての工具長コンポーネントから、マイナスに続く文字列のコンポーネントを差し引いて、計算されます。	
	C		アダプタまたは工具ベース寸法(使用中の工具に対して有効な、2 つの選択可能な成分のどちらでも)	
	E		セットアップオフセット	
	G		ジオメトリ	
	K		キネマティックトランスフォーメーション(総合 3、4、および 5 軸座標変換の場合に限り評価されます)	
	S	サムオフセット		
	T	旋回工具ホルダ		
W	摩耗			
1 番目の 2 次文字列が空の場合(ただし、タブは除きます)、すべてのコンポーネントを考慮して、すべての工具長が計算されます。<Comp>パラメータが指定されていなくても、この操作は適用されます。				
<Substr_2>: :	オプションの 2 番目の 2 次文字列は、工具長を出力する座標系を示します。			

4.13 工具オフセット

		2 番目の 2 次文字列は、1 つの当該文字のみから構成されます。												
	文字:	<table><tr><td>A</td><td>設定可能座標系(ACS)</td></tr><tr><td>B</td><td>基本座標系(BCS)</td></tr><tr><td>K</td><td>キネマティックトランスフォーメーションの工具座標系(KCS)</td></tr><tr><td>M</td><td>機械座標系(MCS)</td></tr><tr><td>T</td><td>工具座標系(TCS)</td></tr><tr><td>W</td><td>ワーク座標系(WCS)</td></tr></table>	A	設定可能座標系(ACS)	B	基本座標系(BCS)	K	キネマティックトランスフォーメーションの工具座標系(KCS)	M	機械座標系(MCS)	T	工具座標系(TCS)	W	ワーク座標系(WCS)
A	設定可能座標系(ACS)													
B	基本座標系(BCS)													
K	キネマティックトランスフォーメーションの工具座標系(KCS)													
M	機械座標系(MCS)													
T	工具座標系(TCS)													
W	ワーク座標系(WCS)													
		座標系が指定されていない場合、評価は MCS (機械座標系) で実行されます。回転が考慮される場合、回転は、<Stat> で定義されている工具環境で指定されます。												
3	<_Stat>:	<table><tr><td colspan="2">工具環境を記述するデータセットの名前(オプション)</td></tr><tr><td>データタイプ:</td><td>STRING</td></tr><tr><td colspan="2">このパラメータの値が空文字列("")になっているか、指定されていない場合、現在の状態が使用されます。工具が指定されない場合、現在の工具が使用されます。</td></tr></table>	工具環境を記述するデータセットの名前(オプション)		データタイプ:	STRING	このパラメータの値が空文字列("")になっているか、指定されていない場合、現在の状態が使用されます。工具が指定されない場合、現在の工具が使用されます。							
工具環境を記述するデータセットの名前(オプション)														
データタイプ:	STRING													
このパラメータの値が空文字列("")になっているか、指定されていない場合、現在の状態が使用されます。工具が指定されない場合、現在の工具が使用されます。														
4	<T>:	<table><tr><td colspan="2">工具の内部 T 番号(オプション)</td></tr><tr><td>データタイプ:</td><td>INT</td></tr><tr><td colspan="2">このパラメータが指定されていないか、その値が「0」の場合、<Stat> に設定されている工具が使用されます。</td></tr><tr><td colspan="2">このパラメータの値が「-1」の場合、有効な工具の T 番号が使用されます。有効な工具の番号を明確に指定することもできます。</td></tr><tr><td colspan="2">注: <Stat> が指定されていない場合、工具環境として現在の状態が使用されます。<T> = 0 が、工具環境に保存されている T 番号を参照するため、その環境では、有効な工具が使用されます。つまり、この特殊な場合では、パラメータ<T> = 0 と<T> = -1 には同じ意味があります。</td></tr></table>	工具の内部 T 番号(オプション)		データタイプ:	INT	このパラメータが指定されていないか、その値が「0」の場合、<Stat> に設定されている工具が使用されます。		このパラメータの値が「-1」の場合、有効な工具の T 番号が使用されます。有効な工具の番号を明確に指定することもできます。		注: <Stat> が指定されていない場合、工具環境として現在の状態が使用されます。<T> = 0 が、工具環境に保存されている T 番号を参照するため、その環境では、有効な工具が使用されます。つまり、この特殊な場合では、パラメータ<T> = 0 と<T> = -1 には同じ意味があります。			
工具の内部 T 番号(オプション)														
データタイプ:	INT													
このパラメータが指定されていないか、その値が「0」の場合、<Stat> に設定されている工具が使用されます。														
このパラメータの値が「-1」の場合、有効な工具の T 番号が使用されます。有効な工具の番号を明確に指定することもできます。														
注: <Stat> が指定されていない場合、工具環境として現在の状態が使用されます。<T> = 0 が、工具環境に保存されている T 番号を参照するため、その環境では、有効な工具が使用されます。つまり、この特殊な場合では、パラメータ<T> = 0 と<T> = -1 には同じ意味があります。														
5	<D>:	<table><tr><td colspan="2">工具の刃先(オプション)</td></tr><tr><td>データタイプ:</td><td>INT</td></tr><tr><td colspan="2">このパラメータを指定しないか、その値が「0」になっている場合、使用される D 番号は、T 番号に基づいて決まります。工具環境からの T 番号を使用する場合、工具環境の D 番号も読み取られます。使用しない場合は、現在有効になっている工具の D 番号が読み取られます。</td></tr></table>	工具の刃先(オプション)		データタイプ:	INT	このパラメータを指定しないか、その値が「0」になっている場合、使用される D 番号は、T 番号に基づいて決まります。工具環境からの T 番号を使用する場合、工具環境の D 番号も読み取られます。使用しない場合は、現在有効になっている工具の D 番号が読み取られます。							
工具の刃先(オプション)														
データタイプ:	INT													
このパラメータを指定しないか、その値が「0」になっている場合、使用される D 番号は、T 番号に基づいて決まります。工具環境からの T 番号を使用する場合、工具環境の D 番号も読み取られます。使用しない場合は、現在有効になっている工具の D 番号が読み取られます。														

4.13 工具オフセット

6	<DL>:	ロケーションに依存するオフセットの番号(オプション)	
		データタイプ:	INT
		このパラメータを指定しない場合、使用される DL 番号は、T 番号に基づいて決まります。工具環境からの T 番号を使用する場合、工具環境の D 番号も読み取られます。使用しない場合は、現在有効になっている工具の D 番号が読み取られます。	

例

GETTCOR (_LEN)	すべてのコンポーネントを考慮して、機械座標系内で現在有効になっている工具の工具長を計算します。
GETTCOR (_LEN, "CGW:W")	アダプタまたは工具ベース寸法、ジオメトリ、および摩耗で構成される有効な工具の工具長を計算します。旋回工具ホルダまたはキネマティックトランスフォーメーションなどの他のコンポーネントは考慮されません。ワーク座標系で出力します。
GETTCOR (_LEN, "-K:B")	有効なキネマティックトランスフォーメーションの長さコンポーネントを考慮せずに、有効な工具の工具長全体を計算します。基本座標系で出力します。
GETTCOR (_LEN, ":M", "Testenv1", , 3)	「Testenv1」と命名された工具環境に格納された工具に対して機械座標系で工具長全体を計算します。ただし、格納されている刃先番号に関係なく、刃先番号 D3 に計算を実行します。

他の情報

アダプタ座標変換/旋回工具ホルダ/キネマティックトランスフォーメーション

アダプタ座標変換、旋回工具ホルダ、およびキネマティックトランスフォーメーションにより行われる回転とコンポーネントの入れ替えは、工具環境の一部です。このため、対応する長さコンポーネントが含まれていないと考えられる場合でも必ず実行されます。この動作が望ましくない場合、対応する座標変換が無効である工具環境を定義してください。多くの場合(つまり、機械で、座標変換、または旋回工具ホルダを使用しない場合はいつで

も)、工具環境用に格納されているデータセットは自動的にこれらの条件を満たすので、ユーザーは、特殊な用意をする必要がありません。

旋削工具および研削工具 MD20360 \$MC_TOOL_PARAMETER_DEF_MASK に依存する工具長の計算

次のマシンデータは、旋削工具と研削工具で直径軸が使用される場合に、摩耗および工具長を評価する方法を定義します。

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK (工具パラメータの定義)

ビット	値
0	旋削工具と研削工具では、経方向軸の 摩耗パラメータ は直径値として考慮されます。
	= 0 (初期設定) なし
	= 1 あり
1	旋削工具と研削工具では、経方向軸の 工具長コンポーネント は直径値として考慮されます。
	= 0 (初期設定) なし
	= 1 あり

関連するこの 2 つのビットが設定されている場合、対応する入力値は、係数 0.5 で重み付けされます。この重み付けは、GETTCOR で返される工具長に反映されます。

例:

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK = 3

MD20100 \$MC_DIAMETER_AX_DEF (径方向軸機能によるジオメトリ軸) = "X"

X は径方向軸です(標準的な旋盤構成)

プログラムコード	コメント
N30 \$TC_DP1[1,1]=500	
N40 \$TC_DP2[1,1]=2	
N50 \$TC_DP3[1,1]=3.0	; ジオメトリ L1
N60 \$TC_DP4[1,1]=4.0	
N70 \$TC_DP5[1,1]=5.0	
N80 \$TC_DP12[1,1]=12.0	; 摩耗 L1
N90 \$TC_DP13[1,1]=13.0	
N100 \$TC_DP14[1,1]=14.0	

4.13 工具オフセット

プログラムコード	コメント
N110 T1 D1 G18	
N120 R1=GETTCOR(_LEN,"GW")	
N130 R3=_LEN[2]	; 17.0 (= 4.0 + 13.0)
N140 R4=_LEN[3]	; 7.5 (= 0.5 * 3.0 + 0.5 * 12.0)
N150 R5=_LEN[4]	; 19.0 (= 5.0 + 14.0)
N160 M30	

キネマティックトランスフォーメーションおよび旋回工具ホルダの長さコンポーネント

工具長の計算中、**旋回工具ホルダ**を考慮する場合、この計算には、次の各ベクトルが含まれています。

タイプ	ベクトル
M	I1 および I2
T	I1、I2、および I3
P	工具長は、旋回工具ホルダの影響を受けません。

総合 **5 軸座標変換**では、変換タイプ 24 と 56 の工具長計算に、次のマシンデータが使用されます。

座標変換タイプ	マシンデータ
24	MD24550/24650 \$MC_TRAFO5_BASE_TOOL_1/2 MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2 MD24558/24658 \$MC_TRAFO5_PART_OFFSET_1/2
56	MD24550/24650 \$MC_TRAFO5_BASE_TOOL_1/2 MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2

座標変換タイプ 56(移動工具および移動ワーク)は、旋回工具ホルダのタイプ **M** に対応します。

以前のソフトウェアリリースの場合、この 5 軸座標変換では、ベクトル MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2 (チャネルの 1 番目/2 番目の 5 軸座標変換のキネマティックオフセットのベクトル)は、タイプ **M** の旋回工具ホルダの 2 つのベクトル I₁ および I₃ の合計に対応します。

どちらの場合も、合計のみが座標変換に関連します。2 つの個別のコンポーネントを構成する方法は意味がありません。ただし、工具長を計算する場合、どちらのコンポーネントを工具に割り当て、どちらのコンポーネントを工具台に割り当てるかは重要です。これは、マシンデータ MD24558/24658 \$MC_TRAFO5_JOINT_OFFSET_PART_1/2 (表のベクトルキネマティックオフセット)が導入された理由です。これはベクトル I3 に対応します。マシ

ンデータ:MD24560/24660 \$MC_TRAFO5_JOINT_OFFSET_1/2 は、I1 と I3 の合計に対応しなくなり、ベクトル I1 にのみ対応します。マシンデータ MD24558/24658 \$MC_TRAFO5_JOINT_OFFSET_PART_1/2 がゼロである場合、動作は以前と同じです。

互換性

GETTCOR 機能を、TOOLENV および SETTCOR 機能と一緒に使用すれば、計測サイクルで以前外部で実行されていた機能の一部を置換できます。

有効な工具長を実際に特定するパラメータの一部のみが、計測サイクルで実行されていました。上記各機能を使用して、工具長の計算に関連する計測サイクルの挙動を再現できます。

4.13.13.6 工具成分の変更(SETTCOR)

SETTCOR 機能を使用して、個々の成分を評価するときに、関連するすべての一般条件を考慮して、工具成分を変更することができます。

注記

用語について:以下において、工具長と組み合わせた工具成分が関係する場合、成分はすべての工具長を形成する、ベクトルの観点から考慮される成分、たとえば、ジオメトリや磨耗などを意味します。そのような成分は、3つの個々の値(L1、L2、L3)で構成され、この値を以後、座標値と呼びます。

したがって、工具成分「ジオメトリ」は、3つの座標値\$TC_DP3～\$TC_DP5 で構成されます。

構文

```
<Status> = SETTCOR(<CorVal>, <Comp>, [<CorComp>, <CorMode>,  
<GeoAx>, <Stat>, <T>, <D>, <DL>])
```

意味

SETTCOR(...)):	工具成分を変更するための事前定義された機能	
	単独ブロック 指令:	あり

4.13 工具オフセット

<状態>:	機能の戻り値。負の値は異常状態を示します。		
	データタイプ: INT		
	値:	0	動作 OK
		-1	工具環境に予約されたメモリがありません。 MD18116 \$MN_MM_NUM_TOOL_ENV = 0 これは、「工具環境」機能が使用できないことを示します。
		-2	<状態>で指定された名前の工具環境が存在しません。
		-3	パラメータ<Comp>の文字列が無効です。 この異常の発生原因には、文字列が無効な場合または同じ文字列が 2 回プログラム指令されている場合があります。
		-4	T 番号が無効です。
		-5	D 番号が無効です。
		-6	DL 番号が無効です。
		-7	アクセスしようとしたメモリモジュールは存在しません。
		-8	アクセスしようとしたオプションは存在しません(プログラム可能な工具向き、工具管理機能)。
		-9	パラメータ<CorComp>の数値が不正です。
		-10	パラメータ<CorMode>の数値が不正です。
		-11	パラメータ<Comp>および<CorComp>の内容が矛盾しています。
		-12	パラメータ<Comp>および<CorMode>の内容が矛盾しています。
-13		<GeoAx>パラメータの内容がジオメトリ軸を指定していません。	
-14	書き込もうとしたセットアップオフセットは存在しません。		
パラメータ			

1	<CorVal>:	補正ベクトル <Stat>によって定義されたワーク座標系(WCS)では、次の割り当てが適用されます。 • <CorVal> [0]:横座標 • <CorVal> [1]:縦座標 • <CorVal> [2]:垂直座標 1つの工具成分だけを補正する場合(つまり、ベクトルの補正は行われません。パラメータ<CorMode>を参照してください)、補正值は常に、その成分が作用する軸には関わりなく<CorVal>[0]にあります。残りの2つの成分の内容は、評価されません。 <CorVal>、または<CorVal>の成分が径方向軸を参照する場合、データは半径寸法として評価されます。つまり、たとえば、工具が、指定された寸法だけ「長く」なります。これに応じて、ワークの直径が2倍に変更されます。 各寸法は、コントロールシステムの基本単位系(インチまたはメトリック)を基準にしています。
		データタイプ: REAL[3]

2

<Comp>:

工具成分			
データタイプ:	STRING		
この文字列は、2 つの 2 次文字列で構成され、各 2 次文字列はコロンで区切られます。			
一般形式: "<SubStr_1> [: <SubStr_2>]"			
<SubStr_1>: 	1 番目の 2 次文字列は常に必要であり、1 文字または 2 文字を含めることができます。第 1 の構成部分(Val ₁)の単一の文字、および第 2 の構成部分(Val ₂)の第 2 の文字。後続のパラメータ<CorComp>および<CorMode>に従って処理されます。		
	文字:	C	アダプタまたは工具ベース寸法(使用中の工具に対して有効な、2 つの選択可能な成分のどちらでも)
		E	セットアップオフセット
		G	ジオメトリ
		S	サムオフセット
		W	摩耗
<Substr_2>: 	2 番目の 2 次文字列はオプションです。また、(個別の)文字「W」または「T」を構成できます。		
	文字:	W	2 番目の 2 次文字列が空白であるか、文字「W」を含む場合、あたかもワーク座標系(WCS)で計測されたかのように、オフセット値が考慮されます。
		T	2 番目の 2 次文字列に文字「T」が含まれる場合、あたかも工具座標系(Tool Coordinate System、TCS)で計測されたかのように、オフセット値が考慮されます。
文字列内の文字の表記法(大文字か小文字か)は自由です。空白またはタブ(余白)をいくつでも挿入できます。			

3	<CorComp> :	記述するツールデータセットの成分を指定します(オプション)。		
		データタイプ: INT		
		値:	0	オフセット値<CorVal>[0]は、ワーク座標系または工具座標系で、パラメータ<GeoAx>に伝送されたジオメトリ軸を示します(パラメータ<Comp>の説明も参照)。つまり、オフセット値は、工具長の計算に影響を及ぼすすべてのパラメータを考慮して、指定された工具成分で計算する必要があります。工具長全体が指定された軸方向に指定された値だけ変更されます。 この変更は、<Comp>で指定された成分と<CorMode>で指定されたシンボリックアルゴリズムの修正によって実現してください(以下のパラメータを参照してください)。そのため、結果として得られた修正は、3つの軸成分のすべてに影響を及ぼします。
			1	0の場合と同じ。ただし、ベクトル。ベクトル<CorVal>の内容は、ワーク座標系または工具座標系で横軸、縦軸、および垂直軸を示します(パラメータ<Comp>の説明を参照)。 後続のパラメータ<GeoAx>は評価されません。
			2	ベクトルオフセット、つまり、L1、L2、およびL3は同時に変更することができます。 「0」と「1」のタイプとは違い、<CorVal>に含まれるオフセット値は工具のVal _i 成分の座標(後述のパラメータ<CorMode>を参照してください)を基準にします。 ワーク座標系と比較した、既存の工具の考えられるすべての傾きはオフセットには影響しません。
			3 - 5	工具長 L1~L3 (\$TC_DP3~\$TC_DP5)または対応する磨耗の値の補正、設定オフセットまたは追加オフセット。オフセット値は<CorVal>[0]に含まれています。オフセット値は、工具のVal _i 成分の座標で計測されます(後述のパラメータ<CorMode>を参照してください)。ワーク座標系と比較した、既存の工具の考えられるすべての傾きはオフセットには影響しません。
			6	工具半径(\$TC_DP6)または対応する磨耗値の補正、設定オフセット、または追加オフセットマシンデータ MD20360 \$MC_TOOL_PARAMETER_DEF_MASK でビッ

4.13 工具オフセット

				ト 10 および 11(半径または直径として指定された直径/ 直径摩耗データの評価)が考慮されます。
			7 - 11	\$TC_DP7～\$TC_DP11 または対応する磨耗値の補正、 設定オフセット、または追加オフセットこれらのパラ メータは、工具半径とまったく同じように処理されま す。
			このパラメータを指定しない場合、値は「0」になります。	

4	<CorMode> :	実行する書き込み動作のタイプを指定します(オプション)。		
		データタイプ: INT		
		値:	0	$Val_{1new} = <CorVal>$
			1	$Val_{1new} = Val_{1old} + <CorVal>$
			2	$Val_{1new} = <CorVal>$ $Val_{2new} = 0$
3	$Val_{1new} = Val_{1old} + Val_{2old} + <CorVal>$ $Val_{2new} = 0$			
表記 $Val_{1old} + Val_{2old}$ はシンボル表記です。2つの成分が(<_Stat>の状態のために)別々の方法で評価される場合、つまり、2つの成分間で回転が有効である場合、Val_{2old} は追加の前に座標変換されるため、Val_{2new} の削除後と<CorVal>の追加前に得られる工具長は変わりません。 <CorVal>は常に、Val_1 を参照します。<CorVal>は、パラメータ<Comp>の2番目の部分に依存し、ワーク座標系(WCS)または工具座標系(TCS)で計測される値です。そのため、計算がおこなわれる工具成分に対して既に座標変換済みです。したがって、保存されている値と一緒に直接計算することはできず、Val_1 または Val_2 に追加する前に座標変換し直す必要があります。つまり、このオフセットは、<CorComp>で定義された軸とは異なった軸に作用します—または、複数の軸に作用します。 <CorComp> = 0 の場合、つまり、<CorVal>にベクトルが含まれておらず、個々の値だけが含まれている場合、記述された動作は、<CorVal>が計測された座標(WCS/TCS)で実行されます。特に、これは、タイプ2および3で Val_{2new} を0に設定する場合にも適用されます。そして、その結果は、工具の座標に再変換されます。つまり、0に設定される座標値(L1、L2、L3)のいずれもが0にならないか、以前に0であった座標値が0と等しくなくなります。ただし、対応する動作が3つのジオメトリ軸すべてに対して正常に実行された場合、削除される成分の3つの座標値はすべて常に0になります。また、工具がワーク座標系を基準にして回転しない場合、または、すべての工具成分が座標軸と平行を保つように回転する場合(軸交換動作)、1つの工具座標だけが確実に変更されます。 <CorComp> = 0 を使用して、同じ動作(<CorMode>)を3つの座標軸すべてに対して任意の順序で連続して実行するのは、同じ動作を<CorComp> = 1 で単独で実行するのと同じです。				

4.13 工具オフセット

		パラメータ値「0」および「1」ではパラメータ<Comp>に 1 文字を含める必要があり、パラメータ値「2」および「3」では 2 文字を含める必要があります。 例: <Comp>には文字列「ES」、<CorMode>には値「2」を含めます ⇒ Setup offset_{new} = <CorVal>, summed offset_{new} = 0 パラメータ<CorMode>を指定しない場合、値は「0」になります。				
5	<GeoAx>:	オフセット値[0]を読み取るジオメトリ軸のインデックスを指定します(オプション)。 <table><tr><td>データタイプ:</td><td>INT</td></tr><tr><td>値の範囲:</td><td>0 ... 2</td></tr></table> インデックス 0～2 は、現在の工具環境で有効な平面(G17/G18/G19)の横軸、縦軸、および垂直軸を示します。 パラメータ<CorComp>の値が「0」である場合、このパラメータの内容のみが評価されます。	データタイプ:	INT	値の範囲:	0 ... 2
データタイプ:	INT					
値の範囲:	0 ... 2					
6	<Stat>:	工具環境を記述するデータセットの名前(オプション) <table><tr><td>データタイプ:</td><td>STRING</td></tr></table> このパラメータの値が空文字列("")になっているか、指定されていない場合、現在の状態が使用されます。工具が指定されない場合、現在の工具が使用されます。	データタイプ:	STRING		
データタイプ:	STRING					
7	<T>:	工具の内部 T 番号(オプション) <table><tr><td>データタイプ:</td><td>INT</td></tr></table> このパラメータが指定されていないか、その値が「0」の場合、<Stat>に設定されている工具が使用されます。 このパラメータの値が「-1」の場合、有効な工具の T 番号が使用されます。有効な工具の番号を明確に指定することもできます。 注: <Stat>が指定されていない場合、工具環境として現在の状態が使用されます。<T> = 0 が、工具環境に保存されている T 番号を参照するため、その環境では、有効な工具が使用されます。つまり、この特殊な場合では、パラメータ<T> = 0 と<T> = -1 には同じ意味があります。	データタイプ:	INT		
データタイプ:	INT					

8	<D>:	工具の刃先(オプション)
		データタイプ: INT
		このパラメータを指定しないか、その値が「0」になっている場合、使用される D 番号は、T 番号に基づいて決まります。工具環境からの T 番号を使用する場合、工具環境の D 番号も読み取られます。使用しない場合は、現在有効になっている工具の D 番号が読み取られます。
9	<TL>:	ロケーションに依存するオフセットの番号(オプション)
		データタイプ: INT
		このパラメータを指定しない場合、使用される DL 番号は、T 番号に基づいて決まります。工具環境からの T 番号を使用する場合、工具環境の D 番号も読み取られます。使用しない場合は、現在有効になっている工具の D 番号が読み取られます。T、D、および DL が、ロケーションに依存するオフセットなしで工具を指定している場合、サムオフセットまたは設定オフセットをパラメータ<Comp>で指定することはできません(<Status>のエラーコード)。

注記

3 つのパラメータ<Comp>、<CorComp>、および<CorMode>すべての組合わせに意味があるわけではありません。たとえば、<CorComp>のアルゴリズム 3 では、<Comp>で 2 文字を指定する必要があります。無効なパラメータの組み合わせが指定されている場合、対応するエラーコードが<Status>で返されます。

例

例 1

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1] = 120	; フライス工具
N30 \$TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 \$TC_DP12[1,1]=1.0	; 磨耗 L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"G",0,0,2)	
N80 T1 D1 X0 Y0 Z0	; ==> MCS 位置 X0.000 Y0.000 Z1.333
N90 M30	

<CorComp>は「0」であるため、Z 方向で作用するジオメトリ軸の座標値をオフセット値 0.333 と置き換えてください。

4.13 工具オフセット

その結果、合計工具長は次のようになります。 $L1 = 0.333 + 1.000 = 1.333$

例 2

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1] = 120	; フライス工具
N30 \$TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 \$TC_DP12[1,1]=1.0	; 磨耗 L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"W",0,1,2)	
N80 T1 D1 X0 Y0 Z0	; ==> MCS 位置 X0.000 Y0.000 Z11.333
N90 M30	

<CorComp>は「1」です。これはつまり、Z 軸に作用する 0.333 のオフセット値が磨耗値 1.0 に追加されることを意味します。

その結果、合計工具長は次のようになります。 $L1 = 10.0 + 1.333 = 11.333$

例 3

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1] = 120	; フライス工具
N30 \$TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 \$TC_DP12[1,1]=1.0	; 磨耗 L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"GW",0,2,2)	
N80 T1 D1 X0 Y0 Z0	; ==> MCS 位置 X0.000 Y0.000 Z0.333
N90 M30	

<CorComp>は「2」であるため、Z 軸で有効なオフセットがジオメトリコンポーネントに入力され(前の値は上書きされます)、磨耗値が削除されます。

その結果、合計工具長は次のようになります。 $L1 = 0.333 + 0.0 = 0.333$

例 4

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1] = 120	; フライス工具
N30 \$TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 \$TC_DP12[1,1]=1.0	; 磨耗 L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	

4.13 工具オフセット

プログラムコード	コメント
N70 R1=SETTCOR(_CORVAL,"GW",0,3,2)	
N80 T1 D1 X0 Y0 Z0	;==> MCS 位置 X0.000 Y0.000 Z11.333
N90 M30	

<CorComp>は「3」であるため、磨耗値と補正値がジオメトリコンポーネントに加算され、磨耗コンポーネントが削除されます。

その結果、合計工具長は次のようになります。L1 = 11.333 + 0.0 = 11.333

例 5

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1] = 120	; フライス工具
N30 \$TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 \$TC_DP12[1,1]=1.0	; 磨耗 L1
N50 _CORVAL[0]=0.333	
N60 T1 D1 G17 G0	
N70 R1=SETTCOR(_CORVAL,"GW",0,3,0)	
N80 T1 D1 X0 Y0 Z0	;==> MCS 位置 X0.333 Y0.000 Z11.000
N90 M30	

前述の例では、<CorComp>は「3」ですが、インデックス「0」(X 軸)のジオメトリ軸に対して現在補正が有効であり、これはフライス工具で、G17 により工具成分 L3 に割り当てられます。その結果として、SETTCOR の呼び出し時に、工具パラメータ \$TC_DP3 および \$TC_DP12 は影響を受けません。その代わりに、\$TC_DP5 に補正値が入力されます。

例 6

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=500	; 旋削工具
N30 \$TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 \$TC_DP4[1,1]=15.0	; ジオメトリ L2
N50 \$TC_DP12[1,1]=10.0	; 磨耗 L1
N60 \$TC_DP13[1,1]=0.0	; 磨耗 L2
N70 _CORVAL[0]=5.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=SETTCOR(_CORVAL,"GW",0,3,1)	
N110 T1 D1 X0 Y0 Z0	; ==> MCS 位置 X24.330 Y0.000 Z17.500
N120 M30	

この工具は旋削工具です。N80 でフレーム回転が起動されるため、基本座標系(BCS)がワーク座標系(WCS)に対して回転します。WCS では、G18 が有効なため、インデックス 1 の

4.13 工具オフセット

ジオメトリ軸、つまり X 軸に補正值(N70)が影響を及ぼします。<CorMode> = 3 であるため、N100 が実行された後、WCS の X 軸方向での工具磨耗を 0 でなくてはなりません。

プログラム終了時の関連工具パラメータの内容は、次のとおりです。

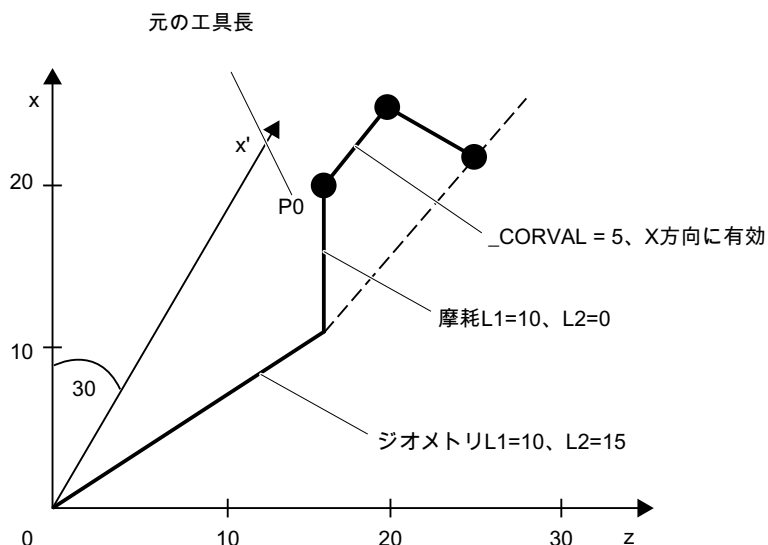
\$TC_DP3[1,1]:21.830 ; ジオメトリ L1

\$TC_DP4[1,1] :21.830 ; ジオメトリ L2

\$TC_DP12[1,1] :2.500 ; 磨耗 L1

\$TC_DP13[1,1] : -4.330 ; 磨耗 L2

次の図に、ジオメトリの関係を示します。_CORVAL を含む磨耗合計が、WCS の X 方向に割り当てられます。これによって点 P2 が生成されます。この点の座標(X/Y 座標で測定)が、工具のジオメトリコンポーネントに入力されます。差分ベクトル $P_2 - P_1$ が引き続き磨耗に含まれます。したがって、磨耗には _CORVAL 方向のコンポーネントがなくなります。



このプログラムの例で、N110 の後、次の命令が続く場合、補正が Z 軸で有効になるため (パラメータ<GeoAx> = 0)、残りの磨耗は完全にジオメトリに含まれます。

```
N120 _CORVAL[0]=0.0
N130 R1=SETTCOR(_CORVAL,"GW",0,3,0)
N140 T1 D1 X0 Y0 Z0 ; ==> MCS 位置 X24.330 Y0.000 Z17.500
```

新しい補正值は「0」であるため、合計工具長と、それに応じて N140 でアプローチする位置は、変更されない場合があります。N120 で _CORVAL が「0」に等しくない場合、新しい合計工具長と、それに応じて N140 でアプローチする新しい位置が生成されますが、工具長の磨耗コンポーネントは常に 0 になります。つまり、合計工具長はその後常に工具のジオメトリコンポーネントに含まれます。

パラメータ<CorComp>=0 を指定して SETTCOR 機能を 2 回呼び出した場合と同じ結果が、<CorComp>=1 (ベクトル補正)を 1 回だけ呼び出してでも得られます。

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=500	; 旋削工具
N30 \$TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 \$TC_DP4[1,1]=15.0	; ジオメトリ L2
N50 \$TC_DP12[1,1]=10.0	; 磨耗 L1
N60 \$TC_DP13[1,1]=0.0	; 磨耗 L2
N70 _CORVAL[0]=0.0	
N71 _CORVAL[1]=5.0	
N72 _CORVAL[2]=0.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=SETTCOR(_CORVAL,"GW",1,3,1)	
N110 T1 D1 X0 Y0 Z0	; ==> MCS 位置 X24.330 Y0.000 Z17.500
N120 M30	

この場合、工具のすべての磨耗コンポーネントが、N100 における SETTCOR の最初の呼び出しの直後に 0 に設定されます。

例 7

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=500	; 旋削工具
N30 \$TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 \$TC_DP4[1,1]=15.0	; ジオメトリ L2
N50 \$TC_DP12[1,1]=10.0	; 磨耗 L1
N60 \$TC_DP13[1,1]=0.0	; 磨耗 L2
N70 _CORVAL[0]=5.0	
N80 ROT Y-30	
N90 T1 D1 G18 G0	
N100 R1=SETTCOR(_CORVAL,"GW",3,3)	
N110 T1 D1 X0 Y0 Z0	; ==> MCS 位置 X25.000 Y0.000 Z15.000
N120 M30	

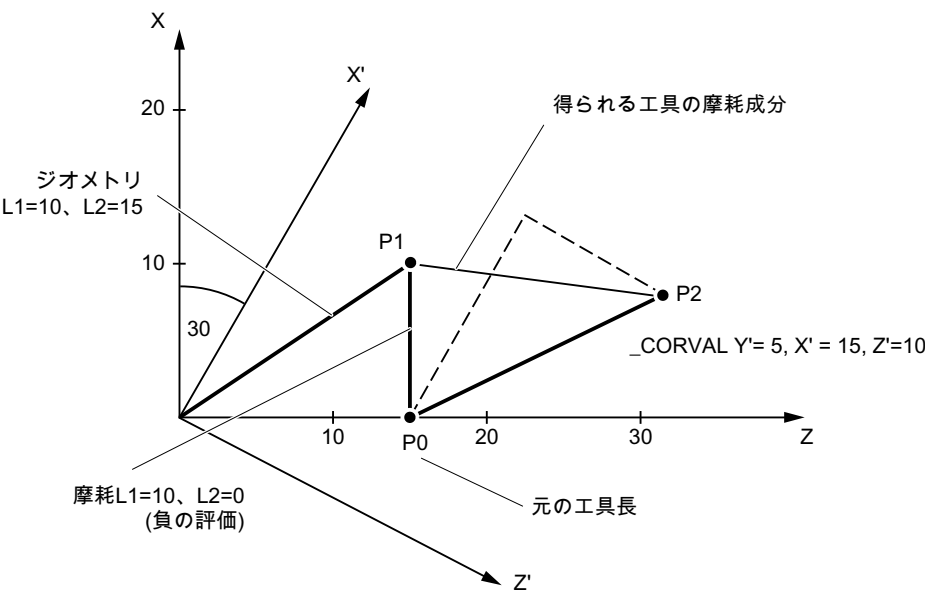
例 6 とは対照的に、パラメータ<CorComp>=3 であり、したがって<GeoAx>パラメータを省略できます。_CORVAL[0]に含まれる値が工具長コンポーネント L1 に即座に影響を及ぼすようになり、N80 での回転は結果に影響せず、\$TC_DP12 の磨耗コンポーネントは _CORVAL[0]とともにジオメトリコンポーネントに含まれ、その結果 N100 における SETTCOR の最初の呼び出し後に\$TC_DP13 により合計工具長がジオメトリコンポーネントに格納されます。

4.13 工具オフセット

例 8

プログラムコード	コメント
N10 DEF REAL _CORVAL[3]	
N20 \$TC_DP1[1,1]=500	; 旋削工具
N30 \$TC_DP3[1,1] = 10.0	; ジオメトリ L1
N40 \$TC_DP4[1,1]=15.0	; ジオメトリ L2
N50 \$TC_DP5[1,1]=20.0	; ジオメトリ L3
N60 \$TC_DP12[1,1]=10.0	; 磨耗 L1
N70 \$TC_DP13[1,1]=0.0	; 磨耗 L2
N80 \$TC_DP14[1,1]=0.0	; 磨耗 L3
N90 \$SC_WEAR_SIGN=TRUE	
N100 _CORVAL[0]=10.0	
N110 _CORVAL[1]=15.0	
N120 _CORVAL[2]=5.0	
N130 ROT Y-30	
N140 T1 D1 G18 G0	
N150 R1=SETTCOR(_CORVAL,"W",1,1)	
N160 T1 D1 X0 Y0 Z0	; ==> MCS 位置 X7.990 Y25.000 Z31.160
N170 M30	

N90 でセッティングデータ:SD42930 \$SC_WEAR_SIGN が有効です。つまり、磨耗は負の符号で評価する必要があります。この補正はベクトル補正(<CorComp> = 1)です。補正ベクトルを磨耗(<CorMode> = 1)に加算してください。Z/X 平面でのジオメトリの関係を次の図に示します。



工具のジオメトリ成分は、<CorMode> = 1 であるために変更されません。図 3 の合計工具長の基準が点 P₂ になるように、WCS で定義された補正ベクトル(Y 軸を中心とする回転)を

磨耗成分に含めてください。したがって、得られる工具の磨耗成分は、2 点 P_1 および P_2 の距離によって示されます。

ただし、セッティングデータ SD42930 \$SC_WEAR_SIGN により、磨耗は負で評価されるので、この方法で決定された補正を負符号とともに補正メモリに入力する必要があります。プログラム終了時の関連工具パラメータの内容は、次のとおりです。

\$TC_DP3[1,1]:10.000 ; ジオメトリ L1 (不変)

\$TC_DP4[1,1]:15.000 ; ジオメトリ L2 (不変)

\$TC_DP5[1,1]:10.000 ; ジオメトリ L3 (不変)

\$TC_DP12[1,1]:2.010 ; 磨耗 L1 ($= 10 - 15 * \cos(30) + 10 * \sin(30)$)

\$TC_DP13[1,1]:-16.160 ; 磨耗 L2 ($= -15 * \sin(30) - 10 * \cos(30)$)

\$TC_DP14[1,1]:-5.000 ; 磨耗 L3

Y 方向の L3 コンポーネントに対するセッティングデータ SD42930 \$SC_WEAR_SIGN の影響は、フレーム回転によってさらに複雑化されることなく認識できます。

他の情報

旋削/研削工具:MD20360 \$MC_TOOL_PARAMETER_DEF_MASK に依存する工具長の計算

次のマシンデータは、旋削/研削工具で直径軸が使用される場合に、磨耗および工具長を評価する方法を定義します。

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK.<ビット> = <値>

<ビット>	<値>	意味
0	0	旋削/研削工具では、経方向軸の 磨耗パラメータ が 半径値 で考慮されます:
	1	旋削/研削工具では、経方向軸の 磨耗パラメータ が 直径値 として考慮されます:
1	0	旋削/研削工具では、経方向軸の 工具長成分 が 半径値 として考慮されます:
	1	旋削/研削工具では、経方向軸の 工具長成分 が 直径値 として考慮されます:

関連するこの 2 つのビットが設定されている場合、対応する入力値は、係数 0.5 で重み付けされます。SETTCOR を使用した補正が、有効な工具長全体の変更が<CorVal>で転送される値と等しくなるよう実行されます。長さを計算する場合、マシンデータ MD20360

4.13 工具オフセット

\$MC_TOOL_PARAMETER_DEF_MASK の結果として、係数 0.5 で長さを評価し、次に伝送された値の 2 倍でこの成分の補正を実行する必要があります。

例

MD20360 \$MC_TOOL_PARAMETER_DEF_MASK=2 (工具長は、係数 0.5 を使用して、直径軸で評価する必要があります)

軸 X が直径軸です

プログラムコード	コメント
N10 DEF REAL _LEN[11]	
N20 DEF REAL _CORVAL[3]	
N30 \$TC_DP1[1,1]=500	; 工具タイプ
N40 \$TC_DP2[1,1]=2	; 刃先位置
N50 \$TC_DP3[1,1]=3.	; ジオメトリ - 長さ 1
N60 \$TC_DP4[1,1]=4.	; ジオメトリ - 長さ 2
N70 \$TC_DP5[1,1]=5.	; ジオメトリ - 長さ 3
N80 _CORVAL[0]=1.	
N90 _CORVAL[1]=1.	
N100 _CORVAL[2]=1.	
N110 T1 D1 G18 G0 X0 Y0 Z0	; ==> MCS 位置 X1.5 Y5 Z4
N120 R1=SETTCOR(_CORVAL,"G",1,1)	
N130 T1 D1 X0 Y0 Z0	; ==> MCS 位置 X2.5 Y6 Z5
N140 R3=\$TC_DP3[1,1]	; = 5. = (3.000 + 2.*1.000)
N150 R4=\$TC_DP4[1,1]	; = 5. = (4.000 + 1.000)
N160 R5=\$TC_DP5[1,1]	; = 6. = (5.000 + 1.000)
N170 M30	

各軸では、工具長の補正は 1 mm (N80～N100)であることが必要です。したがって、長さ L2 および L3 の元の長さに 1 mm が加算されます。要求どおり合計長を 1 mm 変更するために、L1 の元の長さに 2 倍の補正值(2 mm)が加算されます。ブロック N110 および N130 でアプローチする位置を比較すると、各軸位置が 1 mm 変更されたことがわかります。

4.13.14 座標軸への工具長 L1、L2、L3 の割り当ての読み取り(LENTOAX)

「LENTOAX」機能は、横軸、縦軸、および垂直軸への有効な工具の工具長 L1、L2、および L3 の割り当てに関する情報を提供します。ジオメトリ軸への横軸、縦軸、および垂直軸の割り当ては、フレームと有効な平面(G17～G19)の影響を受けます。

工具のジオメトリ成分(\$TC_DP3[<t>,<d>]～\$TC_DP5[<t>,<d>])のみが考慮されます。つまり、他の成分(たとえば、摩耗)に別の軸割り当てを実行しても、結果に影響しません。

構文

```
<Status> = LENTOAX(<AxInd>, <Matrix>[, <Coord>])
```

原理

LENTOAX (...) :	座標軸に対する有効な工具の工具長 L1、L2、および L3 の割り当てを読み取るために事前定義された機能		
	単独ブロック指令:		あり
<状態>:	機能の戻り値。負の値は異常状態を示します。		
	データタイプ:		INT
	値:	0	動作 OK <AxInd>の情報は、記述として十分です(工具長成分はすべて、ジオメトリ軸と平行です)。
		1	機能 OK。ただし、適切な記述かどうか、<Matrix>の内容を評価してください(工具長成分は、ジオメトリ軸に平行ではありません)。
		-1	パラメータ<Coord>の文字列が無効です。
		-2	工具が有効になっていません。
パラメータ			

4.13 工具オフセット

1	<AxInd>:	工具長成分がジオメトリ軸と平行な場合、長さ成分 L1～L3 に割り当てられた軸インデックスは、<AxInd>配列で返されます。 • <AxInd> [0]:横座標 • <AxInd> [1]:縦座標 • <AxInd> [2]:垂直座標	
		データタイプ:	INT[3]
		値:	0 既存の割り当てなし(軸はありません)
		1 ... 3 または -1 ... -3	対応する座標軸で有効な長さの番号。 工具長成分が、負の座標方向を向いている場合には、符号が負になります。
		一部の長さ成分が、ジオメトリ軸に平行または逆平行になっていない場合、工具長成分の最も大きな部分が含まれている軸のインデックスが<AxInd>で返されます。この場合(別の理由で機能が異常を返さない場合)、戻り値は<Status> = 1 です。ジオメトリ軸 1～3 に対する工具長成分 L1～L3 のマッピングは、2 番目のパラメータ<Matrix>の内容によって完全に記述されます。	
2	<Matrix>:	座標軸(横軸、縦軸、垂直軸)のベクトルに対する工具長のベクトル (L1=1、L2=1、L3=1)を表すマトリクスです。つまり工具長成分は、L1,L2,L3 の順で列に割り当てられ、各軸は、横軸、縦軸、垂直軸の順に行に割り当てられます。	
		データタイプ:	REAL
		座標軸に所属するジオメトリ軸を使用できなくとも、つまり、<AxInd>内の対応する入力値が 0 でも、マトリクス内の要素はすべて常に有効です。	

3	<Coord>:	割り当てに使用する座標系(オプション)		
		データタイプ:	STRING	
		文字:	MCS M	工具長が、機械座標系で表されます。
			BCS B	工具長が、基本座標系で表されます。
			WCS W	工具長が、ワーク座標系で表されます(初期設定)。
			KCS K	工具長が、キネマティック座標変換の工具座標系で表されます。
			TCS T	工具長が、工具座標系で表されます。
		文字列内の文字の表記法(大文字か小文字か)は自由です。 パラメータ<Coord>を指定しない場合、WCS が使用されます(初期設定)。		

注記

TCS では、工具長成分はすべて常に軸と平行か逆平行です。

成分を逆平行にできるのは、ミラーリングが有効になっており、次のセッティングデータが有効になっている場合に限りです。

SD42900 \$SC_MIRROR_TOOL_LENGTH (ミラーリング時の工具長符号変更)

例

標準アプリケーション、G17 のフライス工具。

L1 は Z 軸(垂直軸)に適用され、L2 は Y 軸(縦軸)に適用され、L3 は X 軸(横軸)に適用されます。

次の形式で機能呼び出しを実行します。

```
<Status>=LENTOAX(<AxInd>,<Matrix>,"WCS")
```

結果パラメータ<AxInd>には次の値が含まれます。

```
<AxInd>[0] = 3
```

```
<AxInd>[1] = 2
```

```
<AxInd>[2] = 1
```

4.14 軌跡の移動動作

または、省略形では次のようになります。(3, 2, 1)

この場合、当該のマトリクス(<Matrix>)は次のとおりです。

$$\text{<Matrix>} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

G17 から G18 または G19 に変更しても、結果は変わりません。というのも、ジオメトリ軸への長さ成分の割り当ては、横軸、縦軸、垂直軸の割り当てと同じように変化するからです。

ここで、60°の Z 軸フレーム回転が、G17 が有効な状態でプログラム指令されます。たとえば

ROT Z60 です

垂直軸方向(Z 方向)は変更されません。この時点で、L2 の主要成分は、新しい X 軸方向を向いており、L1 の主要成分は、負の Y 軸方向を向いています。その結果として、戻り値(<Status>)は「1」であり、<AxInd>には値(2、-3、1)が含まれます。

この場合、当該のマトリクス(<Matrix>)は次のとおりです。

$$\text{<Matrix>} = \begin{pmatrix} 0 & \sin 60^\circ & \cos 60^\circ \\ 0 & \cos 60^\circ & -\sin 60^\circ \\ 1 & 0 & 0 \end{pmatrix}$$

4.14 軌跡の移動動作

4.14.1 送り速度特性(FNORM、FLIN、FCUB、FPO)

送り速度特性を柔軟に定義できるようにするため、DIN 66025 に従った送り速度指令は、1 次特性と 2 次特性により拡張されます。

3 次特性は、直接、または補間スプラインとしてプログラム指令できます。加工するワークの曲率に応じて、連続した滑らかな速度特性をプログラミングすることができます。

これらの速度特性により、加々速度一定の加速度変更が可能になるため、加工物の表面を均一にできます。

構文

```
F... FNORM
F... FLIN
F... FCUB
F=FPO (... , ... , ...)
```

意味

FNORM	送り速度は DIN 66025 に従って 標準的 (初期設定) この送り速度値はブロックの移動軌跡に応じて指定され、その後、モーダル値として有効になります。
FLIN	軌跡速度パターン 1 次 送り速度値は、ブロック始点の現在値からブロック終点へ、移動軌跡を通して直線的にアプローチして、その後、モーダル値として有効になります。 送り速度特性 FLIN は、G93、G94、G95 で有効ですが、G96/G961 および G97/G971 では 無効 です。 オーバーライドまたはシンクロナイズドアクションに起因する G95 および主軸速度の変化により、同期がわずかに遅れる場合があります。
FCUB	軌跡速度パターン 3 次 ブロックごとにプログラム指令した F 値をブロック終点に対するスプラインで接続します。このスプラインは、直前または次の送り速度機能に応じて接線方向で開始して終了します。F アドレスがブロックにない場合は、最後にプログラム指令した F 値を使用します。 送り速度特性 FCUB は、G93 と G94 で有効ですが、G95、G96/G961、G97/G971 では 無効 です。
F=FPO(.,.,.,.,.)	多項式による軌跡速度パターン F アドレスは、現在の値からブロック終了値まで、多項式で送り速度特性を定義します。終了値はそれ以降、モーダル値として有効となります。

軌跡移動特性のプログラミングの機能は、プログラム指令した送り速度特性にかかわらず、適用されます。

プログラマブル送り速度特性は、G90 と G91 のいずれにも影響されず、常に絶対値です。

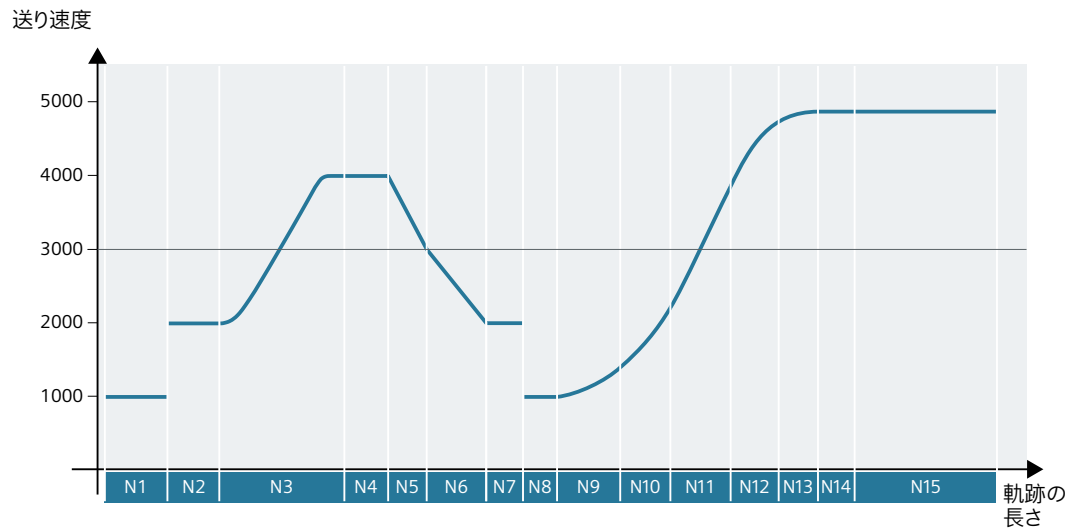
注記

曲線の軌跡区間の送り速度調整

送り速度多項式 F=FPO()および送り速度スプライン FCUB は、常に輪郭 CFC において一定の送り速度で移動させてください。これにより、継続加速目標送り速度パターンを作成できます。

4.14 軌跡の移動動作

例:さまざまな送り速度パターン

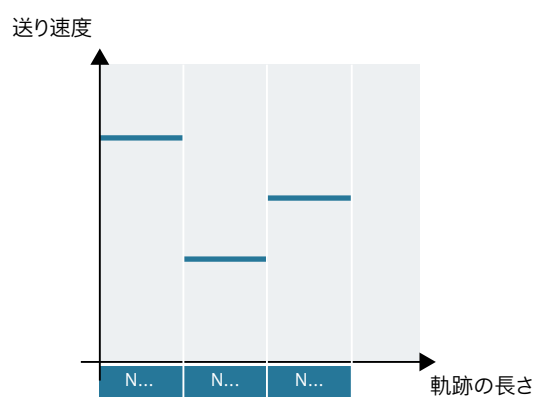


プログラムコード	コメント
N1 F1000 FNORM G1 X8 G91 G64	; 一定送り速度パターン、インクレメンタル指令データ
N2 F2000 X7	; 指令速度のステップ変更
N3 F=FPO(4000, 6000, -4000)	; ブロック終点での送り速度が 4000 となる多項式による、送り速度パターン
N4 X6	; 多項式の送り速度の 4000 がモーダル値として有効です。
N5 F3000 FLIN X5	; リニア送りパターン
N6 F2000 X8	; リニア送りパターン
N7 X5	; 直線の送り速度パターンがモーダル値として有効です
N8 F1000 FNORM X5	; 加減速ステップ変更による一定送り速度パターン。
N9 F1400 FCUB X8	; 複数ブロックでプログラム指令した次のすべての F 値をスプラインで接続します。
N10 F2200 X6	
N11 F3900 X7	
N12 F4600 X7	
N13 F4900 X5	; スプラインパターンをオフに切り替えます。
N14 FNORM X5	
N15 X20	

詳細

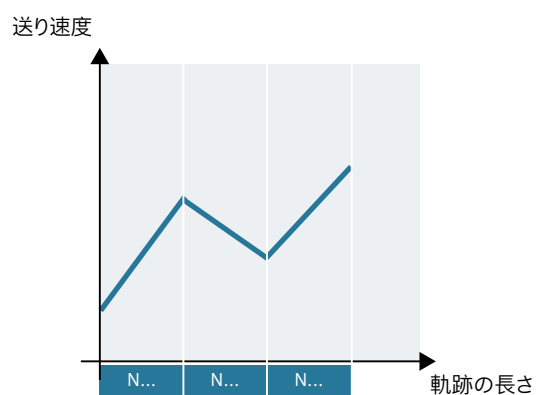
FNORM

送りアドレス F は、DIN 66025 に従い、一定の値として軌跡送り速度を定義します。



FLIN

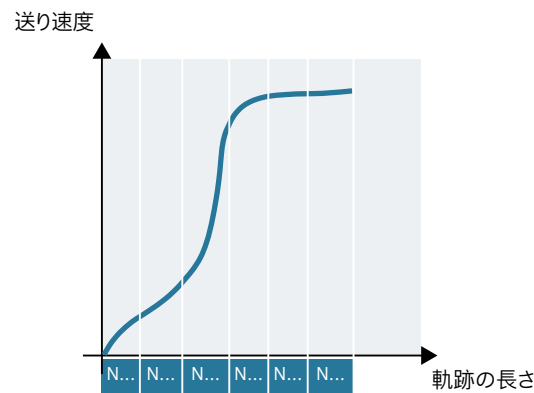
送り速度特性は、現在の送り速度値から、プログラム指令した F 値へと、ブロック終点まで一次特性でアプローチします。



FCUB

送り速度は、現在の送り速度値から、プログラム指令した F 値へと、ブロック終点まで 3 次特性に従ってアプローチします。制御装置はスプラインを使用して、動作中の FCUB で、ノンモーダルでプログラム指令されたすべての送り速度値を接続します。送り速度値はここで、スプライン補間を計算するための補間点として機能します。

4.14 軌跡の移動動作



$F=FPO(...,...,...)$

送り速度特性は、多項式で直接プログラム指令します。多項式係数は、多項式補間を使用するものと同じ方式で指定します。

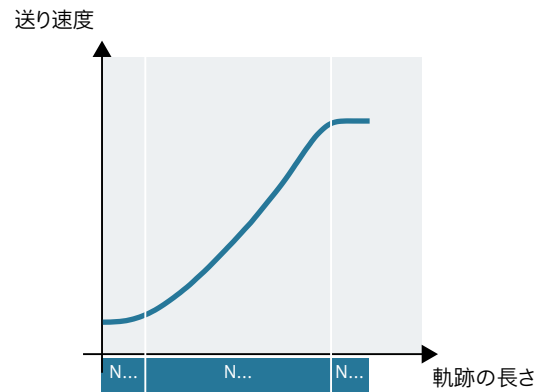
例:

$F=FPO(endfeed, quadf, cubf)$

endfeed、quadf、および cubf は、あらかじめ定義される変数です。

endfeed	ブロック終点の送り速度
quadf	2 次多項式係数
cubf	3 次多項式係数

FCUB が有効な場合は、ブロック始点とブロック終点で FPO を使用して定義した特性へ、スプラインによって接線方向につながります。



コンプレッサ機能が有効な場合の動作

動作中のコンプレッサ機能(COMP...)およびスプラインセグメントを作成するための複数のブロックの組み合わせについては、以下が当てはまります。

FNORM:	グループの最後のブロックの F ワードがスプラインセグメントに適用されます。
FLIN:	グループの最後のブロックの F ワードがスプラインセグメントに適用されます。 プログラム指令した F 値がセグメントの最後まで適用され、その後、直線的にアプローチします。
FCUB:	生成した送り速度スプラインは、プログラム指令終点から、マシンデータ MD20172 \$MC_COMPESS_VELO_TOL で設定した最大距離だけの誤差があります。
F=FPO(...,...,...):	これらのブロックは圧縮されません。

4.14.2 加減速動作

4.14.2.1 加減速モード(BRISK、BRISKA、SOFT、SOFTA、DRIVE、DRIVEA)

次のパートプログラム命令を使用して、現在の加減速モードをプログラム指令できます。

- "BRISK、BRISKA"
単独軸または軌跡軸が、プログラム指令送り速度に達するまで最大加減速度で移動します(加々速度制限のない加減速)。
- "SOFT、SOFTA"
単独軸または軌跡軸が、プログラム指令送り速度に達するまで一定の加減速度で移動します(加々速度一定加減速)。
- "DRIVE、DRIVEA"
単独軸または軌跡軸が、プログラム指令制限速度まで最大加減速度で移動します(マシンデータの設定)。その後、加速度が、プログラム指令送り速度に達するまで減少します(マシンデータの設定)。

4.14 軌跡の移動動作

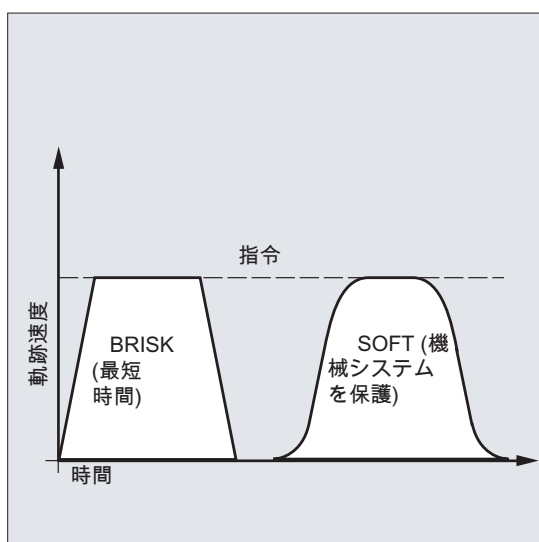


図 4-13 BRISK と SOFT による軌跡速度曲線

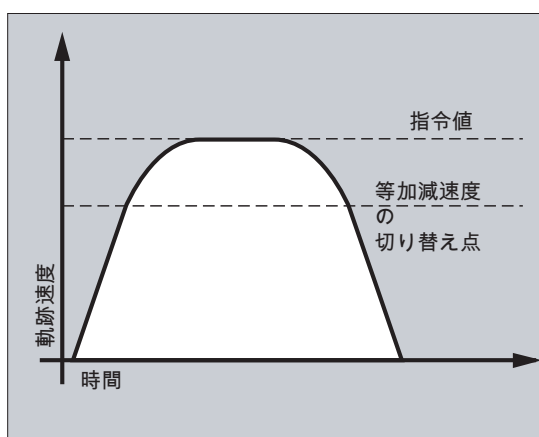


図 4-14 DRIVE による軌跡速度曲線

構文

```
BRISK  
BRISKA(<軸 1>,<軸 2>,...)  
SOFT  
SOFTA(<軸 1>,<軸 2>,...)  
DRIVE  
DRIVEA(<軸 1>,<軸 2>,...)
```


意味

BRISK:	軌跡軸に対して「加々速度制限のない加減速」を有効にする命令です。
BRISKA:	単独軸移動に対して「加々速度制限のない加減速」を有効にする命令です(JOG、JOG/INC、位置決め軸、揺動軸など)。
SOFT:	軌跡軸に対して「加々速度一定加減速」を有効にする命令です。
SOFTA:	単独軸移動(JOG、JOG/INC、位置決め軸、揺動軸など)に対して「加々速度一定加減速」を有効にする命令です。
DRIVE:	軌跡軸に対して、設定制限速度 (MD35220 \$MA_ACCEL_REDUCTION_SPEED_POINT)以上で加減速低減を有効にする命令です。
DRIVEA:	単独軸移動(JOG、JOG/INC、位置決め軸、揺動軸など)に対して、設定制限速度 (MD35220 \$MA_ACCEL_REDUCTION_SPEED_POINT)以上で加減速低減を有効にする命令です。
(<軸 1>, <軸 2>, ...):	呼び出した加減速モードが適用される単独軸です。

必要条件

加工中の加減速モードの変更

加工中にパートプログラムで加減速モードを変更(BRISK ⇄ SOFT)すると、連続軌跡モードであっても、遷移中にブロックの終点でイグザクトストップによりブロックが切り替えられます。

例

例 1:SOFT と BRISKA

プログラムコード

```

N10 G1 X... Y... F900 SOFT
N20 BRISKA(AX5,AX6)
...
```

4.14 軌跡の移動動作

例 2:DRIVE と DRIVEA

プログラムコード

```
N05 DRIVE
N10 G1 X... Y... F1000
N20 DRIVEA (AX4, AX6)
...
```

4.14.2.2 スレーブ軸に対する加減速の動作(VELOLIMA、ACCLIMA、JERKLIMA)

軸連結 (ページ 1039) (接線方向の補正、連結移動、軸間連動機能、電子ギヤ)の場合、スレーブ軸/主軸は 1 本または複数本のマスタ軸/主軸に応じて移動します。

スレーブ軸/主軸の応答性制限は、軸連結がすでに有効な場合でも、パートプログラムまたはシンクロナイズドアクションから VELOLIMA、ACCLIMA、および JERKLIMA 機能を使用して操作できます。

注記

JERKLIMA 機能は、一部のタイプの連結では使用できません。

- ACCLIMA および VELOLIMA は、主軸操作時、すなわち連結を外した場合には、主軸のモーション全体に作用します。変数\$AC_SMAXACC_INFO または\$AC_SMAXVELO_INFO では、制限は識別子 22 で示されます。
- 主軸操作では、VELOLIMA は 100 パーセントに制限されます。
- 以下は、許容速度に関して主軸連結に当てはまります。
連結モーションがすでに最大速度まで使用され、そのためプログラム指令された基本モーションが停止する場合、アラーム 22015「スレーブ主軸は追加のモーションのためのダイナミクスがありません」というメッセージが表示されます。

構文

```
VELOLIMA (<軸>) =<値>
ACCLIMA (<軸>) =<値>
JERKLIMA (<軸>) =<値>
```

意味

VELOLIMA:	最大パラメータ設定 速度 を補正する命令
ACCLIMA:	最大パラメータ設定 加減速度 を補正する命令
JERKLIMA:	最大パラメータ設定 加々速度 を補正する命令

<軸>:	応答性制限の補正が必要なスレーブ軸
<値>:	補正値の割合(%)

例

例 1:スレーブ軸(AX4)の応答性制限の補正

プログラムコード	コメント
...	
VELOLIMA[AX4]=75	; マシンデータに設定された最大軸速度の 75%の制限へ補正
ACCLIMA[AX4]=50	; マシンデータに設定された最大軸加減速度の 50%の制限へ補正
JERKLIMA[AX4]=50	; マシンデータに設定された最大軸加々速度の 50%の制限へ補正
...	

例 2:電子ギヤ

軸 4 を、「電子ギヤ」連結により軸 X に連結します。スレーブ軸の加減速能力を最大加減速度の 70%に制限します。最大許容速度を最大速度の 50%に制限します。連結が正常に動作すると、最大許容速度が 100%に復帰します。

プログラムコード	コメント
...	
N120 ACCLIMA[AX4]=70	; 最大加減速度の制限
N130 VELOLIMA[AX4]=50	; 最大速度の制限
...	
N150 EGON(AX4,"FINE",X,1,2)	; EG 連結の適用
...	
N200 VELOLIMA[AX4]=100	; 最大速度へ復帰
...	

例 3:内部的なシンクロナイズドアクションによる軸間連動機能への効果

軸間連動機能により軸 4 が X に連結されます。加減速動作は、100%の値から、内部的なシンクロナイズドアクション 2 により、80%の値に制限されます。

プログラムコード	コメント
...	
N120 IDS=2 WHENEVER \$AA_IM[AX4] > 100 DO ACCLIMA[AX4]=80	; シンクロナイズドアクション
N130 LEADON(AX4, X, 2)	; 軸間連動機能オン
...	

4.14 軌跡の移動動作

4.14.2.3 テクノロジ用のダイナミック応答値の起動(DYNNORM、DYNPOS、DYNROUGH、DYNSEMIFIN、DYNFINISH、DYNPREC)

G グループ 59「軌跡補間のダイナミック応答モード」の命令で、さまざまなテクノロジカル加工ステップの適切なダイナミック応答を起動できます。

ダイナミック応答値と G 命令が設定できます。したがって、これらはマシンデータ設定に依存します。

詳細情報:機能マニュアル『基本機能』

構文

ダイナミック応答値の起動:

DYNNORM/DYNPOS/DYNROUGH/DYNSEMIFIN/DYNFINISH/DYNPREC

注記

ダイナミック応答値はすでに、関連する G 命令がプログラム指令されたブロックのなかで有効です。加工は停止されません。

特定のフィールド要素の読み取りまたは書き込み:

R<m>=\$MA...[n,X]

\$MA...[n,X]=<値>

意味

DYNNORM:	標準のダイナミック応答の起動
DYNPOS:	位置決めモード、タッピングのダイナミック応答の起動
DYNROUGH:	粗削りのダイナミック応答の起動
DYNSEMIFIN:	荒仕上げのダイナミック応答の起動
DYNFINISH:	仕上げのダイナミック応答の起動
DYNPREC:	滑らかな仕上げのダイナミック応答の起動
R<m>:	番号<m>をもつ R 変数
\$MA...[n,X]:	ダイナミック応答に作用するフィールド要素をもつマシンデータ

<n>:	配列インデックス	
	値の範囲:	0 ... 5
	0	標準のダイナミック応答(DYNNORM)
	1	位置決めモードのダイナミック応答(DYNPOS)
	2	荒削りのダイナミック応答(DYNROUGH)
	3	荒仕上げのダイナミック応答(DYNSEMIFIN)
	4	仕上げのダイナミック応答(DYNFINISH)
	5	滑らかな仕上げのダイナミック応答(DYNPREC)
<X>:	軸アドレス	
<値>:	ダイナミック応答値	

例

例 1:ダイナミック応答値の起動

プログラムコード	コメント
DYNNORM G1 X10	; 初期設定
DYNPOS G1 X10 Y20 Z30 F...	; 位置決めモード、タッピング
DYNROUGH G1 X10 Y20 Z30 F10000	; 荒削り
DYNSEMIFIN G1 X10 Y20 Z30 F2000	; 荒仕上げ
DYNFINISH G1 X10 Y20 Z30 F1000	; 仕上げ
DYNPREC G1 X10 Y20 Z30 F600	; 滑らかな仕上げ

例 2:指定フィールド要素の読み取りまたは書き込み

荒削りの最大加減速度、X 軸

プログラムコード	コメント
R1=\$MA_MAX_AX_ACCEL[2,X]	; 読み取り
\$MA_MAX_AX_ACCEL[2,X]=5	; 書き込み

4.14.3 フィードフォワード制御による移動(FFWON、FFWOF)

フィードフォワード制御は、経路誤差ゼロに向かって輪郭切削をする際の速度による行き過ぎを低減します。フィードフォワード制御の移動により、高精度の軌跡が可能になり、加工レベルが向上します。

4.14 軌跡の移動動作

構文

FFWON

FFWOF

意味

FFWON:	フィードフォワード制御を 有効にする 命令
FFWOF:	フィードフォワード制御を 解除する 命令

注記

フィードフォワード制御のタイプと、フィードフォワード制御で移動する軌跡軸は、マシンデータで指定します。

初期設定:速度のフィードフォワード制御

オプション:加減速度のフィードフォワード制御

例

プログラムコード

N10 FFWON

N20 G1 X... Y... F900 SOFT

4.14.4 輪郭/旋回の許容範囲のプログラミング(CTOL、OTOL、ATOL)

パートプログラムのコンプレッサ機能、スムージング、および旋回のスムージングについて、マシンデータとセッティングデータを使用してパラメータ設定された加工許容範囲を調整するために、アドレス CTOL、OTOL、および ATOL を使用できます。

プログラムされた許容値は、再度プログラムされるか、負の値の割り当てにより削除されるまで有効です。さらにこの許容値は、プログラムの終了時またはリセット時に、削除されます。パラメータ設定された許容値は、削除後に再度有効になります。

構文

CTOL=<Value>

OTOL=<Value>

ATOL[<Axis>]=<Value>

意味

CTOL:	輪郭許容範囲をプログラムするアドレス		
	用途:	- すべてのコンプレッサ機能 - G641 と G644 を除く、すべての丸み付けタイプ	
	先読み停止:	なし	
	効果:	モーダル	
	<Value>:	輪郭の許容範囲の値は長さで指定します。	
		タイプ:	REAL
		単位:	inch/mm (現在の寸法設定で決まります)
		値の範囲:	≥ 0: 許容誤差値 < 0: プログラムされた許容範囲値が削除されます ⇒ マシンデータまたはセッティングデータでパラメータ設定された許容範囲値が再度有効になります。
OTOL:	旋回許容範囲をプログラムするアドレス		
	用途:	- すべてのコンプレッサ機能 - ORISON 旋回スムージング - G641、G644、および OSD を除く、すべてのスムージングタイプ	
	先読み停止:	なし	
	効果:	モーダル	
	<Value>:	旋回の許容範囲は角度で指定します。	
		タイプ:	REAL
		単位:	°
		値の範囲:	≥ 0: 許容誤差値 < 0: プログラムされた許容範囲値が削除されます ⇒ マシンデータまたはセッティングデータでパラメータ設定された許容範囲値が再度有効になります。

4.14 軌跡の移動動作

ATOL:	軸別の許容範囲のプログラミングのアドレス		
用途:	- すべてのコンプレッサ機能 - ORISON 旋回スムージング - G641、G644、および OSD を除く、すべてのスムージングタイプ		
先読み停止:	なし		
効果:	モーダル		
<Axis>:	指令された許容範囲が適用されるチャネル軸の名前		
<Value>:	軸の許容範囲の値は、軸タイプ(直線軸または回転軸)に応じて、長さまたは角度で指定します。		
	タイプ:	REAL	
	単位:	直線軸の場合:	inch/mm (現在の寸法設定で決まります)
		回転軸の場合:	°
	値の範囲:	≥ 0:	許容誤差値
		< 0:	プログラムされた許容範囲値が削除されます ⇒ マシンデータまたはセッティングデータでパラメータ設定された許容範囲値が再度有効になります。

注記

CTOL および OTOL でプログラム指令されたチャネル別許容範囲値が、ATOL プログラム指令された軸別許容範囲値より優先して適用されます。

注記

スケーリングフレーム

スケーリングフレームは、軸位置と同様にプログラム指令の許容範囲に影響します。つまり、相対的な許容範囲はそのまま同じです。

例

プログラムコード	コメント
COMPCAD G645 G1 F10000	; COMPCAD コンプレッサ機能を起動します。
X...Y...Z...	; ここで、マシンデータとセッティングデータが適用されます。
X...Y...Z...	

4.14 軌跡の移動動作

プログラムコード	コメント
X...Y...Z...	
CTOL=0.02	; これ以降、0.02mm の輪郭の許容範囲の適用が開始されます。
X...Y...Z...	
X...Y...Z...	
X...Y...Z...	
ASCALE X0.25 Y0.25 Z0.25	; これ以降、0.005mm の輪郭の許容範囲の適用が開始されます。
X...Y...Z...	
X...Y...Z...	
X...Y...Z...	
CTOL=-1	; これ以降、再度マシンデータとセッティングデータの適用が開始されます。
X...Y...Z...	
X...Y...Z...	
X...Y...Z...	

詳細

システム変数

現在有効な許容範囲は、以下のシステム変数を介して読み取ることができます。

- 先読み停止による読み取り(パートプログラムおよびシンクロナイズドアクション内)
 - **\$AC_CTOL**
現在のメインランブロックが先読みされたときの有効なチャネル別輪郭許容範囲。
有効な輪郭の許容範囲がない場合は、**\$AC_CTOL** が、ジオメトリ軸の許容範囲の二乗の合計の平方根を返します。
 - **\$AC_OTOL**
現在のメインランブロックが先読みされたときの有効なチャネル別旋回許容範囲。
有効な向きの許容範囲がない場合は、**\$AC_OTOL** が、動作中の方向座標変換で、旋回軸の許容範囲の二乗の合計の平方根を返します。それ以外の場合は、値「-1」を返します。
 - **\$AA_ATOL[<軸>]**
現在のメインランブロックが先読みされたときの有効な軸別輪郭許容範囲。
有効な輪郭許容範囲がない場合、**\$AA_ATOL[<ジオメトリ軸>]**は、ジオメトリ軸の数の平方根で除算した輪郭許容範囲を返します。
向きの許容範囲と方向座標変換が有効な場合、**\$AA_ATOL[<旋回軸>]**は、旋回軸の数の平方根で除算した向きの許容範囲を返します。

注記

許容範囲値がプログラム指令されている場合は、個々の機能の許容範囲を区別できるほど、**\$A** 変数が差別化されていません。

このような状況は、マシンデータとセッティングデータで、コンプレッサ機能、スムージング、および旋回スムージングに異なる許容範囲を設定した場合に発生する可能性があります。システム変数は、現在有効な機能で発生する最大の値を返します。たとえば、その向きの許容範囲 0.1°のコンプレッサ機能、および 1°の ORISON 旋回スムージングが有効な場合は、**\$AC_OTOL** 変数は値「1」を返します。向きのスムージングが無効な場合は、**\$AC_OTOL** が値「0.1」を返します。

- 先読み停止を行なわない読み取り(パートプログラム内のみ)
 - **\$P_CTOL**
現在有効なチャネル別の輪郭許容範囲。
 - **\$P_OTOL**
現在有効なチャネル別の向きの許容範囲。
 - **\$PA_ATOL**
現在有効な軸別の輪郭許容範囲。

制約事項

CTOL、OTOL、および ATOL でプログラムされた許容範囲は、それらの許容範囲に間接的に依存している機能にも影響します。

- 指令値演算でのコードエラーの制限
- 自由曲面モードの基本機能

以下のスムージング機能は、CTOL、OTOL、および ATOL のプログラミングには影響されません。

- OSD による旋回のスムージング
OSD は、許容範囲を使用せず、ブロック遷移部からの距離を使用します。
- G644 によるスムージング
G644 は、スムージングには使用されず、工具交換の最適化、および加工を伴わないその他の動作に使用されます。
- G645 によるスムージング
G645 は、実質的に常に G642 と同様に動作し、プログラム指令された許容範囲を使用します。マシンデータ MD33120 \$MA_PATH_TRANS_POS_TOL の許容範囲の値は、曲率の不連続変化を含む、接線変化が均一なブロック遷移部(接線の変化が連続的な円弧/直線遷移部など)でのみ使用します。これらの点での丸み付け軌跡は、プログラム指令輪郭の外にある場合もあり、この場合は、多くの用途で許容範囲が小さくなります。さらに、通常は、NC プログラマには無関係な曲率の変化の種類を補正するために、小さい固定許容範囲が必要となります。

4.14.5 プログラム指令可能な輪郭精度のオン/オフの切り替え(CPRECON、CPRECOF)

「プログラム指令可能な輪郭精度」機能は、速度の自動調整により、輪郭の曲線部での軌跡エラーを低減します。

オンとオフの切り替えは、NC プログラムにおいて G グループ 39 (プログラム指令可能な輪郭精度)のモーダルで有効なコマンドによって行います。

構文

```
CPRECON  
...  
CPRECOF
```

4.14 軌跡の移動動作

意味

CPRECON:	「プログラム指令可能な輪郭精度」機能を オン に切り替えます。
CPRECOF:	「プログラム指令可能な輪郭精度」機能を オフ に切り替えます。

例

プログラムコード	コメント
N10 G0 X0 Y0	
N20 CPRECON	; プログラマブル輪郭精度を有効にします
N30 G1 G64 X100 F10000	; 連続軌跡モードで 10 m/min で加工します。
N40 G3 Y20 J10	; 円弧ブロックの送り速度を自動制限します。
N50 G1 X0	; 再度制限なしの送り速度 (10 m/min) で加工します。
...	
N100 CPRECOF	; 「プログラム指令可能な輪郭精度」を解除します。
N110 G0 ...	

下記も参照

輪郭/旋回の許容範囲のプログラミング(CTOL、OTOL、ATOL) (ページ 978)

詳細

輪郭精度

維持される輪郭精度は、機械の構成に従って(MD20470 \$MC_MC_CPREC_WITH_FFW; 工作機械メーカーの仕様書を参照してください)、設定日 SD42450\$SC_CONTPREC またはプログラム指令輪郭許容範囲 CTOL のいずれかによって指定されます。値が小さいほど、またジオメトリ軸の K_V 係数が小さいほど、輪郭の曲線部での軌跡送り速度は大きく低減されます。

早送り移動の輪郭精度

曲線の輪郭があって「プログラム指令可能な輪郭精度」機能が有効な加工物を機械加工するときは、加工物をバイパスするときのコーナーやコーナーの丸み付けブロックなど、工具の移動を早送りする間でも減速し、指定の輪郭精度を維持します。早送り移動中の軌道速度の低下を最小限にするため、加工物の加工から逸脱する早送り輪郭精度を「プログラム指令可能な輪郭精度」機能に設定できます。

SD42451 \$SC_CONTPREC_G00_ABS (早送りを伴う輪郭精度)

SD42451 = 0 である場合は、\$SC_CONTPREC[DYNNORM]に設定した輪郭精度が早送り移動に適用されます。

最低軌跡送り速度

次のセッティングデータを使用して、「プログラム指令可能な輪郭精度」機能の最低軌跡送り速度を指定することができます。

SD42460 \$SC_MINFEED (CPRECON による最低軌跡送り速度)

より低い F 値がプログラム指令されている場合または軸の動的制限によってより低い軌跡速度が強制される場合を除いて、送り速度がこの値を下回って制限されることはありません。

位置決め軸に影響なし

「プログラム指令可能な輪郭精度」機能は、軌跡のジオメトリ軸のみを考慮します。この機能は、位置決め軸の速度に影響を及ぼしません。

パートプログラム起動時およびチャネルやプログラムの終了リセット後の動作

パートプログラムの起動時およびチャネルやプログラムの終了リセット後には、G 機能グループ 39 に定義した初期制御設定が有効になります(機械メーカーの情報を参照)。

4.14.6 自動フィルタ切り替えの起動/解除(AFISON、AFISOF)

ユーザーは、「自動フィルタ切り替え」機能を使用すると、NC プログラム内のエリアをマークして、この機能用に有効になっているすべての軸を G0 モーションの第 2 フィルタチェーンに自動的に切り替えることもできます。第 2 フィルタチェーンをパラメータ設定して振動を強く減衰すると、G0 モーションに加々速度をさらに大きく設定できます。その結果、コーナーなど、G0 モーション中の軌道速度低下は減り、プログラム実行時間は短縮されます。

構文

この機能のオンとオフを切り替える音声コマンドは、ブロック内でそれぞれ単独にしてください。

```
AFISON  
...  
AFISOF
```

意味

AFISON	「自動フィルタ切り替え」機能を オン に切り替えます。
AFISOF	「自動フィルタ切り替え」機能を オフ に切り替えます。

4.14 軌跡の移動動作

例

プログラムコード	コメント
...	
N390 G1 X1100	; フィルタチェーン 1 有効
	; X=1100 で停止
N400 AFISON	; フィルタチェーン 2 への自動切り替え
N410 G0 X1150	; X=1150 で停止
N420 G1 X1200	; フィルタチェーン 1 への自動切り替え
N430 AFISOF	
N440 G0 X1300	; G0 ブロックでフィルタチェーン 1 も有効
N450 G1 X1400	
N460 AFISON	
N470 G1 X1450	
N480 G1 X1500	
N490 AFISOF	
N500 G1 X1600	
N510 G0 X1700	; G0 ブロックでフィルタチェーン 1 も有効
N520 AFISON	
N530 G1 X1750	; X=1750 で停止
	; フィルタチェーン 2 への自動切り替え
N540 G0 X1800	; X=1800 で停止
	; フィルタチェーン 1 への自動切り替え
N550 AFISOF	
N560 G1 X1900	
N570 G0 X2000	; G0 ブロックでフィルタチェーン 1 も有効
	; X=2100 で停止
N580 AFISON	; フィルタチェーン 2 への自動切り替え
N590 G0 X2050	
N600 G0 X2100	; X=2100 で停止
	; フィルタチェーン 1 への自動切り替え
N610 AFISOF	
N620 G0 X2200	; G0 ブロックでフィルタチェーン 1 も有効
...	

詳細

前提条件

「自動フィルタ切り替え」機能を使用できるようにするには、次の要件を満たしてください。

- 「加々速度の調整」オプションを設定してください。これにはライセンスが必要です。
MD19321 \$ON_TECHNO_FUNCTION_MASK_1, bit 22 = 1
- この機能をチャンネルで有効にしてください。
MD20630 \$MC_AFIS_MODE = 1
- この機能を自動フィルタ切り替え用の各軸で有効にしてください。
MD32332 \$MD_AFIS_ENABLE = 1
- この機能用に有効にしたすべての軸において、次のようにしてください。
 - 加々速度制限を有効にしてください。
MD32400 \$MA_AX_JERK_ENABLE = 1
 - 2つの加々速度フィルタタイプを選択して(→ MD32402 \$MA_AX_JERK_MODE)設定してください。

この要件を満たさないと、アラーム 14782 または 26380 が出力されます。

オン/オフ切り替え時の停止

この機能のオンとオフを切り替えるとき、位置指令フィルタチェーンを切り替える必要がある場合は、軌跡移動のみ停止します。

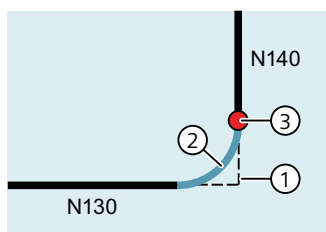
スムージング

フィルタチェーンの停止および自動切り替えをスムーズ化(G64x)が有効な間に実行しても、輪郭のスムーズ化は影響されません。

例:

```
N100 G642 F1000 CTOL=10 CTOLG0=10
N110 G0 X0 Y0
N120 AFISON
N130 G1 X100 Y0
N140 G0 X100 Y100
```

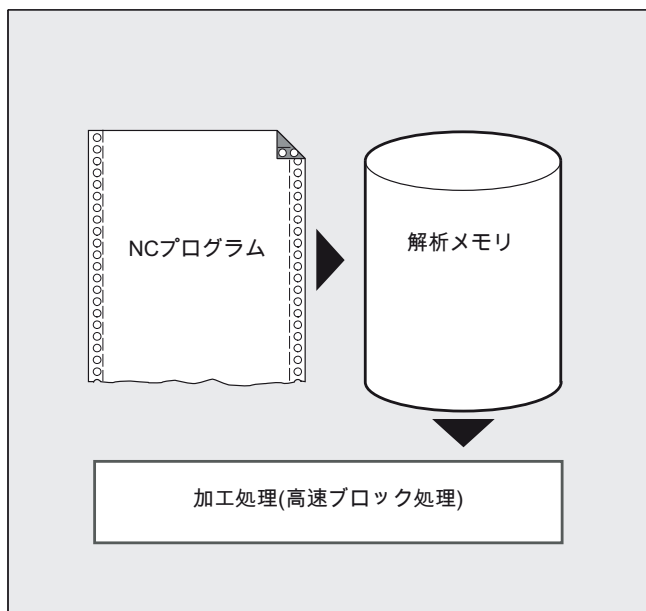
4.14 軌跡の移動動作



- ① プログラム指令軌跡
- ② 輪郭のスムージング
- ③ 停止および第2フィルタチェーンへの切り替え

4.14.7 解析メモリによるプログラム手順(STOPFIFO、STARTFIFO、FIFOCTRL、STOPRE)

コントローラは、拡張レベルに応じて一定の量の解析メモリを備えています。ここに、解析されたブロックがプログラムの実行前に格納され、その後、加工の間に高速ブロック処理として出力されます。これらの処理により、短い軌跡の集合を高速で移動することができます。制御時間が十分に残っていれば、解析メモリは常にいっぱいです。



加工ステップの指定

解析メモリに保持される加工ステップの開始と終了はそれぞれ、パートプログラムで「STOPFIFO」と「STARTFIFO」を使用して指定します。解析して保持したブロックの処理が開始されるのは、「STARTFIFO」命令の後、または解析メモリがいっぱいの場合のみです。

解析メモリの自動制御

解析メモリの自動制御は、「FIFOCTRL」命令で呼び出します。「FIFOCTRL」は最初は「STOPFIFO」とまったく同様に機能します。プログラミングにかかわらず、解析メモリがいっぱいになるまで、処理は開始されません。ただし、解析メモリに空きがある場合の動作は異なります。「FIFOCTRL」を使用した場合は、メモリの使用率が 2/3 に達すると、完全に空になって停止状態まで減速することのないよう、軌跡速度がだんだん減速します。

先読み停止

「STOPRE」命令をブロックでプログラミングすると、ブロックの解析と一時的な記憶を停止します。次のブロックは、先読みがおこなわれて保持されたすべてのブロックが完全に実行されるまで実行されません。先行ブロックは (G9 と同様に) イグザクトストップで停止します。

通知

プログラム中止

工具補正またはスプライン補間が有効な場合は、「STOPRE」命令をプログラム指令しないでください。プログラム指令すると、連続ブロック処理が中断されます。

構文

表 4-3 加工ステップの指定:

STOPFIFO
...
STARTFIFO

表 4-4 解析メモリの自動制御:

...
FIFOCTRL
...

表 4-5 先読み停止:

...
STOPRE
...

4.14 軌跡の移動動作

注記

「STOPFIFO」、「STARTFIFO」、「FIFOCTRL」、および「STOPRE」の命令は、それぞれ個別のブロックにプログラム指令してください。

意味

STOPFIFO:	「STOPFIFO」は、解析メモリに保持する加工ステップ開始を指定します。「STOPFIFO」は処理を停止し、以下の時点まで、解析メモリをいっぱいにします。 「STARTFIFO」または「STOPRE」を認識する または - 解析メモリがいっぱいになる または - プログラム終点に達する
STARTFIFO:	「STARTFIFO」は、加工ステップの高速処理を開始します。これと並行して、解析メモリも満たされます。
FIFOCTRL:	解析メモリの自動制御を適用します
STOPRE:	解析を停止します

注記

バッファリングが不必要な動作(基準点のサーチ、計測機能、など)の命令が加工ステップに含まれる場合は、解析メモリがいっぱいにならないか、またはいっぱいになる前に処理を中断します。

注記

コントローラは、状態データ(\$SA...)にアクセスした場合、内部の先読み停止をおこないます。

例:解析を停止します

プログラムコード	コメント
...	
N30 MEAW=1 G1 F1000 X100 Y100 Z50	; 1 番目の計測点でのプローブによる計測ブロックと直線補間です。
N40 STOPRE	; 先読み停止
...	

4.14.8 停止遅延範囲(DELAYFSTON、DELAYFSTOF)の定義

パートプログラムで条件付き中断可能範囲を定義するために、事前定義された DELAYFSTON および DELAYFSTOF 手順を使用します(停止遅延範囲)。

注記

DELAYFSTON および DELAYFSTOF はシンクロナイズドアクションでは許容されません!

構文

```

DELAYFSTON
...
DELAYFSTOF

```

意味

DELAYFSTON:	停止遅延範囲の先頭の定義	
	単独ブロック指 令:	あり
DELAYFSTOF:	停止遅延領域の終了を定義します	
	単独ブロック指 令:	あり

プログラミング例

次のプログラムブロックは以下のループで繰り返されます。

プログラムコード

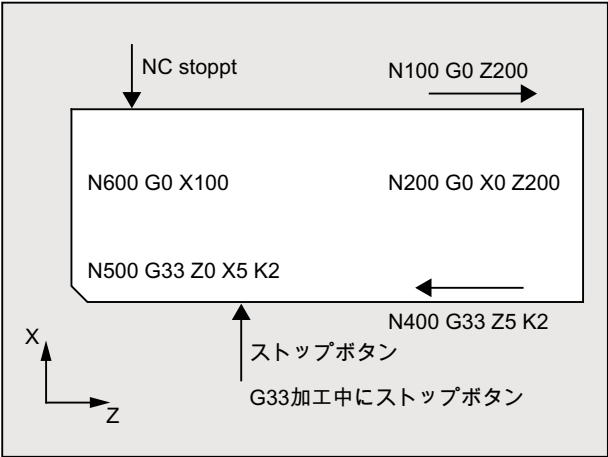
```

...
N99 MY_LOOP:
N100 G0 Z200
N200 G0 X0 Z200
N300 DELAYFSTON
N400 G33 Z5 K2 M3 S1000
N500 G33 Z0 X5 K3
N600 G0 X100
N700 DELAYFSTOF
N800 GOTOB MY_LOOP
...

```

4.14 軌跡の移動動作

次の図では、ユーザーが停止遅延範囲内で「停止」を押すと、停止遅延範囲(つまりブロック N100)の外側で NC が減速を開始することがわかります。これにより、NC が N100 の始点で停止します。



他の情報

サブプログラム終了

DELAYFSTOF は、DELAYFSTON を呼び出しているサブプログラムの終点で、自動的に有効になります。

ネスト

停止遅延領域でサブプログラム 1 がサブプログラム 2 を呼び出した場合は、サブプログラム 2 全体が停止遅延領域になります。特に、DELAYFSTOF はサブプログラム 2 では無効です。

例:

プログラムコード	コメント
N10010 DELAYFSTON	; N10xxx プログラムレベル 1 のブロックです。
N10020 R1 = R1 + 1	
N10030 G4 F1	; 停止遅延領域を開始します。
...	
N10040 subprogram2	
...	
...	; サブプログラム 2 を解釈します。
N20010 DELAYFSTON	; 無効です、停止遅延開始を繰り返します。2 番目のレベルです。
...	
N20020 DELAYFSTOF	; 無効です、別のレベルで終了しています。
N20030 RET	
N10050 DELAYFSTOF	; 停止遅延範囲を同じレベルで終了します。

プログラムコード	コメント
...	
N10060 R2 = R2 + 2	
N10070 G4 F1	; 停止遅延領域が終了します。これ以降は、動作を直ちに停止します。

システム変数

パートプログラム処理が現在、停止遅延領域にあるかどうかを特定するために、以下のシステム変数を確認することができます。

- パートプログラムでは、\$P_DELAYFST
- シンクロナイズドアクションでは\$AC_DELAYFST

値	意味
0	停止遅延範囲が無効です
1	停止遅延領域が有効です

4.14.9 SERUPRO のプログラム位置のマスク(IPTRLOCK、IPTRUNLOCK)

機械のメカニズムが複雑である事例では、停止ブロックサーチの SERUPRO が必要となる場合があります。

プログラム指令可能な中断ポインタを使用すると、「中断点の検索」を使用して検索できない点の直前に介入できます。

また、パートプログラム区間に検索できない区間を定義することもできます。その区間では NC は再介入できなくなります。プログラムが中断すると、NC は、処理した最後のブロックを記録して、それを HMI 操作画面から検索することができます。

構文

```
IPTRLOCK
IPTRUNLOCK
```

これらの命令はパートプログラム行にあり、プログラム指令可能な中断ポインタの使用を可能にします。

意味

IPTRLOCK:	検索不可プログラム区間の開始
IPTRUNLOCK:	検索不可プログラム区間の終了

4.14 軌跡の移動動作

両方の命令はパートプログラムでのみ使用でき、シンクロナイズドアクションでは**使用できません**。

例

自動的な IPTRUNLOCK による 2 つのプログラムレベルでの検索不可プログラム区間の入れ子指令。サブプログラム 1 の自動的な「IPTRUNLOCK」が検索不可区間を終了します。

プログラムコード	コメント
N10010 IPTRLOCK()	
N10020 R1 = R1 + 1	
N10030 G4 F1	; 検索をマスクするプログラム区間の保持ブロックを起動します。
...	
N10040 subprogram2	
...	; サブプログラム 2 を解釈します。
N20010 IPTRLOCK ()	; 無効です、起動を繰り返します。
...	
N20020 IPTRUNLOCK ()	; 無効です、別のレベルで終了しています。
N20030 RET	
...	
N10060 R2 = R2 + 2	
N10070 RET	; サーチを抑制するプログラム区間の終了
N100 G4 F2	; メインプログラムを続行します。

ここで、中断ポインタが再度、100 で中断をおこないます。

詳細情報

検索不可区間の認識と検出

検索できないプログラム区間は、言語命令「IPTRLOCK」と「IPTRUNLOCK」で識別されます。

命令「IPTRLOCK」は、メインランの実行可能なシングルブロックで中断ポインタを解除します(SB1)。これ以降では、このブロックを保持ブロックと呼びます。「IPTRLOCK」の後でプログラムが中止された場合は、この保持ブロックを HMI 操作画面から検索できます。

実行中のブロックからの続行

中断ポインタは「IPTRUNLOCK」で、実行中のブロックに、その後に続くプログラム区間の中断点として配置されます。

検索ターゲットが見つかり、新しい検索ターゲットを保持ブロックによって繰り返すことができます。

ユーザーが編集した中断ポインタは、HMI で再度削除してください。

入れ子の規則

入れ子指令の言語命令「IPTRLOCK」と「IPTRUNLOCK」と、サブプログラム終了との間の連携には、次の点が適用されます。

1. 「IPTRUNLOCK」は、「IPTRLOCK」が呼び出されるサブプログラムの最後に自動的に起動されます。
2. 検索不可区間では、IPTRLOCK は無効です。
3. 検索不可区間でサブプログラム 1 がサブプログラム 2 を呼び出した場合は、サブプログラム 2 全体が検索不可の状態になります。特に「IPTRUNLOCK」は、サブプログラム 2 では効果がありません。

詳細情報:機能マニュアル『Basic Functions』

システム変数

検索不可区間は、パートプログラムで「\$P_IPTRLOCK」を使用して検出できます。

自動中断ポインタ

自動中断ポインタは、以前に定義した連結タイプを検索不可として自動的に定義します。
マシンデータの使用

- 「EGON」の電子ギヤ
- 「LEADON」の軸間連動機能

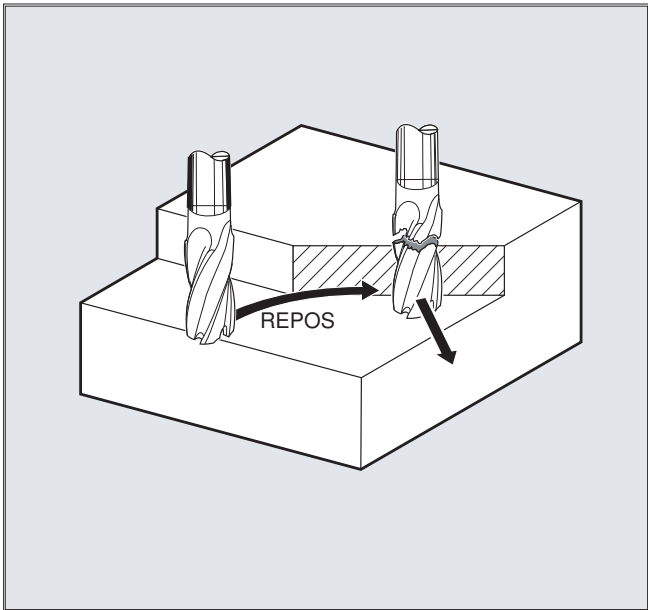
自動中断ポインタが有効です。プログラム指令中断ポインタ、およびマシンデータで起動できる自動中断ポインタが重複する場合は、最大限の検索不可区間が生成されます。

4.14.10 輪郭への再位置決め(REPOSA、REPOSL、REPOSQ、REPOSQA、REPOSH、REPOSHA、DISR、DISPR、RMIBL、RMBBL、RMEBL、RMNBL)

工具の破損、またはワークを計測するためなどの理由で、プログラムの実行を中断して、加工運転中に工具を退避した場合は、プログラムが制御している輪郭上の任意の点を選択して、再度位置決めできます。

REPOS 命令は、ASUB でサブプログラム戻り(M17 など)として機能します。次のブロックは実行されません。プログラム実行の中断については、「割り込みルーチン(ASUB) (ページ 615)」も参照してください。

4.14 軌跡の移動動作



構文

```
REPOSA RMIBL DISPR=...
REPOSA RMBBL
REPOSA RMEBL
REPOSA RMNBL
REPOSL RMIBL DISPR=...
REPOSL RMBBL
REPOSL RMEBL
REPOSL RMNBL
REPOSQ RMIBL DISPR=... DISR=...
REPOSQ RMBBL DISR=...
REPOSQ RMEBL DISR=...
REPOSQA DISR=...
REPOSH RMIBL DISPR=... DISR=...
REPOSH RMBBL DISR=...
REPOSH RMEBL DISR=...
REPOSHA DISR=...
```

意味

アプローチ軌跡の選択

REPOSA:	直線上の輪郭へのジオメトリ軸による再位置決め。 その他のすべての軸も再位置決めされます。
REPOSL:	直線上の輪郭へのジオメトリ軸による再位置決め。 その他の軸は明示的にプログラム指令してください。

REPOSQ DISR=... :	半径 DISR の 4 分円に沿ったジオメトリ軸による輪郭への再位置決め。 その他の軸は明示的にプログラム指令してください。
REPOSQA DISR=... :	半径 DISR の 4 分円に沿ったジオメトリ軸による輪郭への再位置決め。 その他のすべての軸も再位置決めされます。
REPOSH DISR=... :	直径 DISR の半円に沿ったジオメトリ軸による輪郭への再位置決め。 その他の軸は明示的にプログラム指令してください。
REPOSHA DISR=... :	半径 DISR の半円に沿ったジオメトリ軸による輪郭への再位置決め。 その他のすべての軸も再位置決めされます。

再位置決め点の選択

RMIBL:	中断点へアプローチします
RMIBL DISPR=...:	中断点の手前からの距離 DISPR での開始点(mm/inch)
RMBBL:	ブロックの始点へアプローチします
RMEBL:	ブロックの終点へアプローチします
RMEBL DISPR=... :	終点の手前からの距離が DISPR のブロックの終点へアプローチします
RMNBL:	最も近い軌跡点へアプローチします
A0 B0 C0 :	アプローチがおこなわれる軸です

注記

互換性

旧版のソフトウェアとの互換性を残すために、モーダル G 命令 RMI、RMB、RME、および RMN で REPOS アプローチモードをプログラム指令することもできます。ASUB 内で使用する場合は、PROC 命令でこれに SAVE 属性を割り当ててください。これをおこなわない場合、ASUB で使用されるモーダル REPOS アプローチモードが、予め設定されている RMI と異なっている場合に、その後の REPOS 処理でも有効になります。

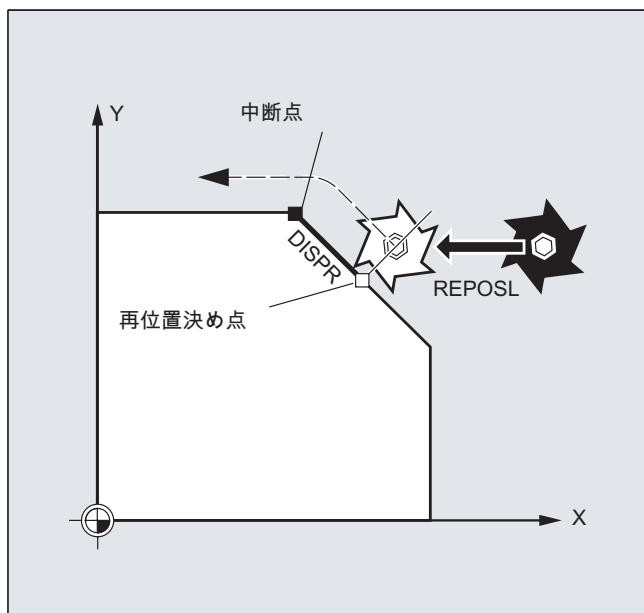
直線上の輪郭への再位置決め REPOSA、REPOSL

工具は、直線上を再位置決め点へアプローチします。

4.14 軌跡の移動動作

例

REPOSL RMIBL DISPR=6 F400

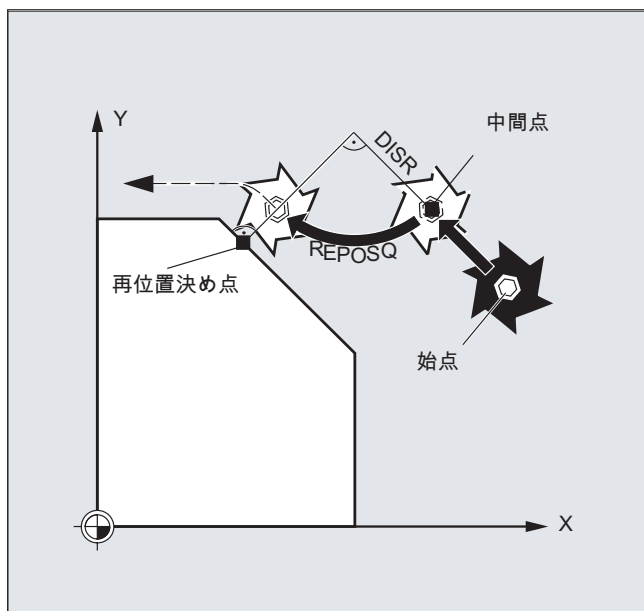


4 分円に沿った輪郭への再位置決め、REPOSQ、REPOSQA

工具は、半径 DISR=... の 4 分円に沿って再位置決め点へアプローチします。制御装置は自動的に、始点と再位置決め点の間の必要な中間点を計算します。

例

REPOSQ RMIBL DISR=10 F400

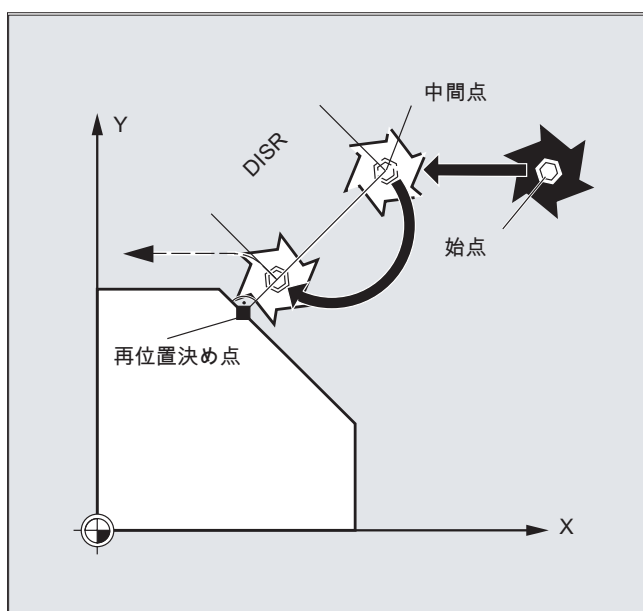


半円に沿った輪郭への再位置決め、REPOSH、REPOSHA

工具は、直径 DISR=...の半円上の再位置決め点へアプローチします。制御装置は自動的に、始点と再位置決め点の間の必要な中間点を計算します。

例

REPOSH RMIBL DISR=20 F400

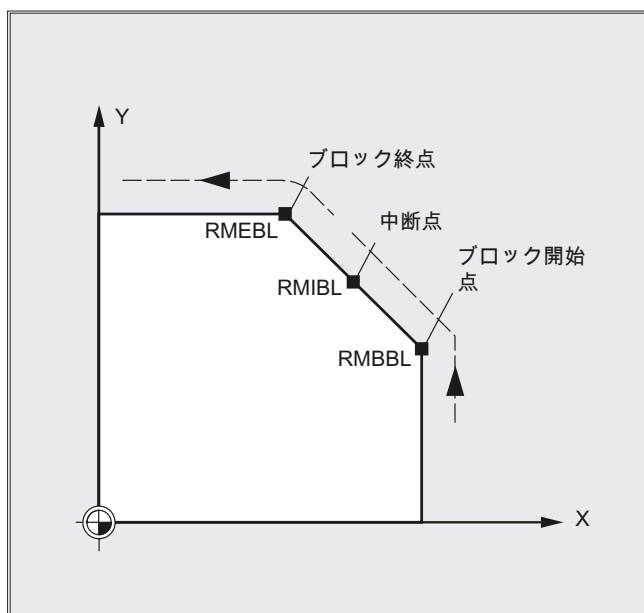


再位置決め点の指定(RMNBL による SERUPRO アプローチではない場合)

プログラムの実行が中断した NC ブロックに対して、3つの異なる再位置決め点から1つを選択できます。

- RMIBL、中断点
- RMBBL、ブロックの始点または最後の終点
- RMEBL、ブロックの終点

4.14 軌跡の移動動作



RMIBL DISPR=...または RME DISPR=...を使用すると、中断点またはブロックの終点の手前にある再位置決め点を選択できます。

DISPR=...を使用すると、再位置決め点と、終点の手前の中断点の間の輪郭距離(mm/inch)を指令できます。値が大きい場合でも、この点が、ブロックの始点を越えることはありません。

DISPR=...命令をプログラム指令しない場合は、DISPR=0 が適用され、これにより、(RMIBL による)中断点または(RMEBL による)ブロックの終点が適用されます。

DISPR 符号

DISPR の符号を評価します。正符号の場合は、以前と同様に動作します。

負符号の場合は、中断点の後ろに、または、RMBBL を使用している場合はブロックの始点の後ろにアプローチします。

中断点とアプローチ点の間の距離は、DISPR の値により異なります。値が大きい場合でも、この点が、ブロックの終点を越えることはありません。

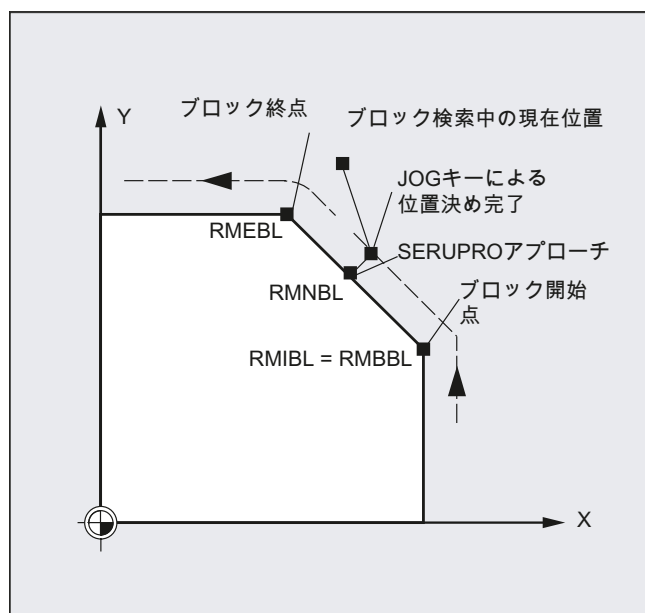
用途例：

クランプへアプローチすると、センサにより認識されます。ASUB が開始され、クランプをバイパスします。

その後、負の DISPR を、クランプの後ろの 1 点に再位置決めし、プログラムを続行します。

RMNBL による SERUPRO アプローチ

加工中に任意の位置で強制的に中止された場合は、SERUPRO アプローチと RMNBL で中止点からの最短軌跡でアプローチし、その後に処理されるのは残移動距離のみとなります。ユーザーは中断ブロックで SERUPRO 処理を開始し、JOG キーを使用して、ターゲットブロックの問題の場所の手前に移動します。



注記

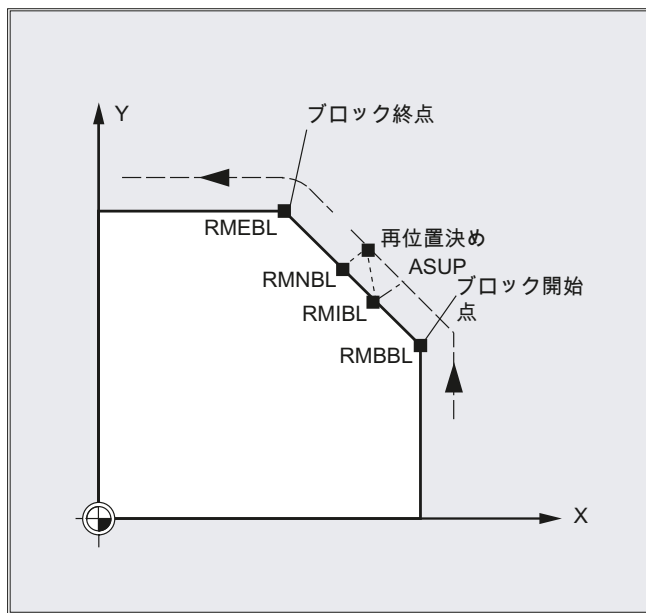
SERUPRO

SERUPRO に対して、RMIBL と RMBBL は同じです。RMNBL は SERUPRO 機能に制限されることなく、一般的に有効です。

4.14 軌跡の移動動作

最も近い軌跡点 RMNBL からのアプローチ

REPOSA を解釈するとき、RMNBL による再位置決めブロック全体は中断後に再起動されませんが、残移動距離のみが処理されます。中断したブロックに最も近い軌跡点へアプローチします。



有効な REPOS モードの状態

中断されたブロックの有効な REPOS モードを、シンクロナイズドアクション、および変数 \$AC_REPOS_PATH_MODE を使用して読み取ることができます。

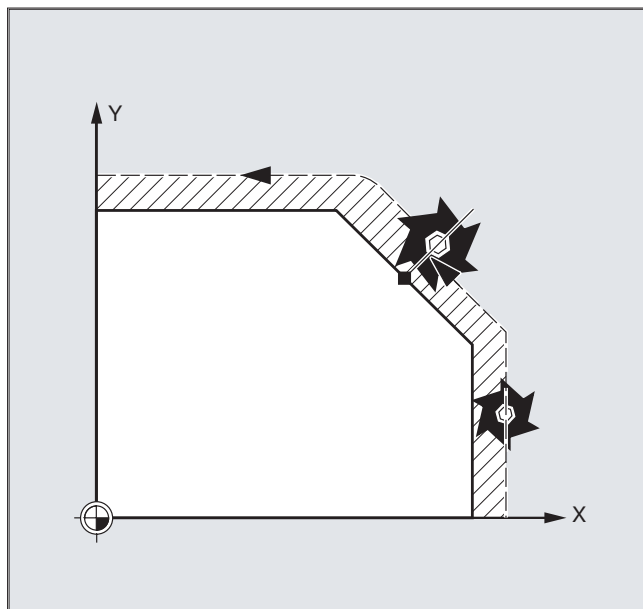
- 0 アプローチが定義されていません
- 1 RMBBL:始点へアプローチします
- 2 RMIBL:中断点へアプローチします
- 3 RMEBL:ブロックの終点へアプローチします
- 4 RMNBL:中断したブロックの次の軌跡点へアプローチします

新しい工具によるアプローチ

工具が破損したためにプログラムの実行を中止した場合は、以下が適用されます。

新しい D 番号をプログラム指令すると、再位置決め点では、加工プログラムは変更後の工具補正值を使用して続行します。

工具補正值を変更した場合は、中断点に再位置決めできない可能性があります。このような場合は、新しい輪郭上で、中断点に最も近い点(DISPR で変更された点の場合もあります)へアプローチします。



輪郭へのアプローチ

工具の輪郭上への再位置決め移動をプログラム指令できます。移動軸のアドレスに対してはゼロを入力してください。

REPOSA、REPOSQA、と REPOSHA の各命令は、すべての軸を自動的に再位置決めします。個々の軸の名称を指定する必要はありません。

REPOSL、REPOSQ、と REPOSH 命令をプログラム指令すると、自動的にすべてのジオメトリ軸が移動します。つまり、命令で軸を指定する必要はありません。他のすべての軸は、命令に指定してください。

REPOSH と REPOSQ の円弧移動には以下が適用されます。

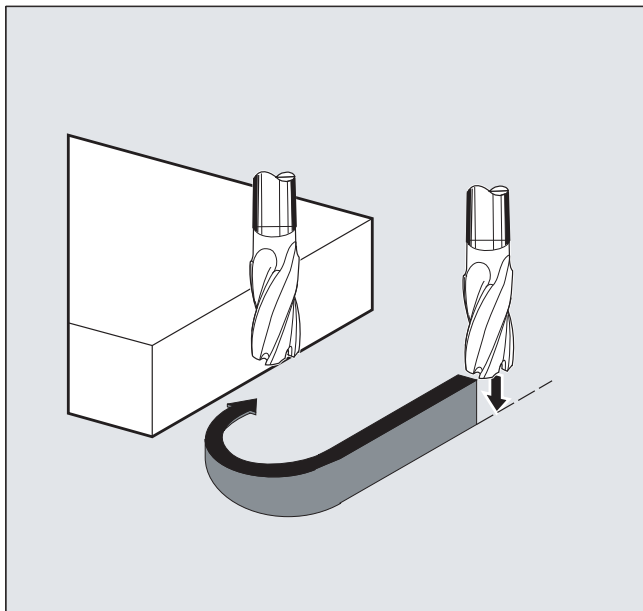
円弧は指定した作業平面 G17～G19 で移動します。

アプローチブロックで 3 番目のジオメトリ軸(切り込み方向)を指定した場合は、切り込み方向のプログラム指令位置と工具位置が一致しない場合、らせんを描いて再位置決め点へアプローチします。

4.14 軌跡の移動動作

以下の場合には、直線アプローチ REPOS L に自動的に切り替わります。

- DISR に値を指定していない。
- 定義されたアプローチ方向がない(移動情報のないブロックのプログラム中断)。
- アプローチ方向が現在の作業平面に垂直。



4.14.11 モーションコントロールへの影響

4.14.11.1 最大軸速度または最大主軸速度の調整(VELOLIM)

軸の最大許容速度、またはマシンデータで設定される主軸の最大許容ギヤ選択依存速度は、パートプログラムで VELOLIM 命令を使用して下げることができます。

有効性

VELOLIM は以下で動作します。

- 運転モードが AUTOMATIC の場合
- 軌跡軸と位置決め軸の場合。
- 主軸運転/軸運転での主軸の場合。

構文

VELOLIM [<軸>] =<値>

意味

VELOLIM:	速度または速度制限値の調整	
<Ax>:	速度または速度制限値の調整が必要な軸または主軸。 MD30455 \$MA_MISC_FUNCTION_MASK、ビット 6 を使用して、VELOLIM が、主軸か軸かに無関係に有効である(ビット 6 = 1)か、または各運転モードで個別にプログラム指令できる(ビット 6 = 0)かを設定できます。個別に有効にする場合は、プログラミング時に次のように識別子で選択します。 - 主軸運転モードの場合は、主軸識別子:S<n> - 軸モードの場合は、軸識別子。例:C	
<値>:	補正值の割合(%) 軸モードの軸または主軸で設定が MD30455 bit 6 = 0 の場合、補正值は設定された最大軸速度(MD32000 \$MA_MAX_AX_VELO)を基準にしています。 主軸または軸モードの主軸で設定が MD30455 bit 6 = 1 の場合、補正值は現在のギヤ選択 (MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT[<n>])を基準にしています。	
	値の範囲:	1 ... 100
	値 100 は速度には影響しません。	

注記

パートプログラムの終了時とチャネルリセット時の応答

パートプログラム終了時とチャネルリセット時の VELOLIM の応答は、マシンデータ MD32320 \$MA_DYN_LIMIT_RESET_MASK のビット 0 の設定によって異なります。

注記

主軸動作での速度制限

「VELOLIM」による速度制限(100 %未満)は、主軸運転で次のシステム変数を使用して検出できます。

- \$AC_SMAXVELO (最大許容主軸速度)
- \$AC_SMAXVELO_INFO (速度制限要因の識別子)

4.14 軌跡の移動動作

例

例 1:速度制限、機械軸

プログラムコード	コメント
...	
N70 VELOLIM[X]=80	; X 方向への軸送りは、軸の許容速度の 80%以下で移動します。
...	

例 2:速度制限、主軸 1 (AX5)

設定:

- MD35130 \$MA_GEAR_STEP_MAX_VELO_LIMIT[1, AX5] = 1000
(ギヤ選択 1 の最大速度= 1000 rpm)
- MD30455 \$MA_MISC_FUNCTION_MASK[AX5]、ビット 6 = 1
(VELOLIM をプログラム指令すると、指定した識別子に関係なく、主軸と軸の運転に同時に機能します)

プログラミング

プログラムコード	コメント
N05 VELOLIM[S1]=90	; 主軸 1 の最大速度を、1000 rpm の 90%に制限します。
...	
N50 VELOLIM[C]=45	; 主軸 1 の最大速度を 1000 rpm の 45%に制限します。C は S1 の軸識別子です。
...	

4.14.11.2 最大軸加々速度の調整(JERKLIM)

パートプログラムで命令 JERKLIM を使用して、軌跡移動に対する軸の最大加々速度(マシンデータで設定)を、問題となるプログラム区間で減少または増加させることができます。

必要条件

加減速モード SOFT を有効にしてください。

有効性

JERKLIM は以下で動作します。

- 運転モードが AUTOMATIC の場合
- 軌跡軸のみの場合

構文

JERKLIM [<軸>] =<値>

意味

JERKLIM:	加々速度制限値の調整	
<Ax>:	加々速度制限値を調整する機械軸	
<値>:	補正值の割合(%) 補正值は、設定された軌跡移動の最大軸加々速度 (MD32431 \$MA_MAX_AX_JERK)を基準としています。	
	値の範囲:	1 ... 200
	値 100 は加々速度に影響しません。	

注記

パートプログラムの終了時とチャネルリセット時の応答

パートプログラム終了時とチャネルリセット時の JERKLIM の応答は、マシンデータ MD32320 \$MA_DYN_LIMIT_RESET_MASK のビット 0 の設定によって異なります。

例

プログラムコード	コメント
N1000 G0 X0 Y0 F10000 SOFT G64	
N1100 G1 X20 RNDM=5 ACC[X]=20 ACC[Y]=30	
N1200 G1 Y20 JERKLIM[Y]=200	; Y 方向への軸送りは、軸の許容加々速度の 200%まで加速/減速できます。
N1300 G1 X0 JERKLIM[X]=2	; X 方向への軸送りは、軸の許容加々速度の 2%以下で加速/減速します。
N1400 G1 Y0	
M30	

4.14.11.3 最大軌跡速度の設定(FLIM)

「最大軌跡速度の調整」機能の場合、軸制限値から得られる軌跡移動の速度は制限可能か、またはパートプログラムの重要なプログラムセクションで下げることができます。最大軌跡速度が制限される速度値は、アドレス FLIM でプログラム指令されます。プログラム指令された値は、次の NC リセットまたはパートプログラムの最後まで有効です。

オーバライドは、FLIM によって制限される送り速度にも影響します。そのため、FLIM はオーバライドスイッチを使用して適度に超過できます。

4.14 軌跡の移動動作

有効性

- 機能は、次の場合に有効です。
- 運転モードが **AUTOMATIC** の場合
 - 軌跡軸のみの場合
 - **G94** のみ
 - 早送りでない場合

注記

[最大軌跡速度の調整]機能のみ、毎分送り **G94** との組み合わせで有効です。この機能は、他の送りタイプと組み合わせることがはできません(**G93**、**G931**、**G95**、**G96**、**G97**、**G971**、**G972**)。

構文

```
...
FLIM=<値>
...
FLIM=-1
...
```

意味

FLIM	最大軌跡速度の調整用アドレス	
<値>	最大軌跡速度に制限すべき速度値	
	データタイプ:	REAL
	値の範囲:	1.0 * 10 ⁻⁶ ... 1.0 * 10 ³⁸
	単位:	mm/分またはインチ/分(使用中の単位系による)
FLIM=-1	FLIM=<値>によってプログラム指令された制限をキャンセルします。	

例

プログラムコード	コメント
...	
N1000 G0 X0 Y0 F10000 G64 G710	
N1100 G1 X20 RNDM=5	

プログラムコード	コメント
N1200 G1 Y20 FLIM=5000	; Y 方向の軸スライドは最大 5 m/分で移動します。
N1300 G1 X0 Y40	; X 方向と Y 方向の軸スライドの軌跡移動は最大 5 m/分で移動します。
N1400 G1 Y0 FLIM=-1	; Y 方向の軸スライドは 10 m/分で移動します。
M30	

4.14.11.4 最大軌跡加速度の設定(PACCLIM)

[最大軌跡速度の調整]機能の場合、軸制限値から得られる軌跡移動の加速度はパートプログラムの重要なプログラムセクションで減じることができます。その加速度値まで最大軌跡加速度を下げるには、アドレス PACCLIM でプログラム指令します。プログラム指令された値は、次の NC リセットまたはパートプログラムの最後まで有効です。

必要条件

ライセンス

このオプション機能にはライセンス(「軌跡加速度制限」、商品番号:6FC5800-0xP26-0YB0)が必要です。ライセンス管理を使用してハードウェアに割り当ててください。

有効性

機能は、次の場合に有効です。

- 運転モードが AUTOMATIC の場合
- 軌跡軸のみの場合

構文

```

...
PACCLIM=<値>
...
PACCLIM=-1
...

```

4.14 軌跡の移動動作

意味

PACCLIM	最大軌跡加速度調整用アドレス	
<値>	最大軌跡加速度を下げるべき加速度値	
	データタイプ:	REAL
	値の範囲:	$1.0 * 10^{-6} \dots 1.0 * 10^{38}$
	単位:	m/s ² または インチ/s ² (使用中の単位系による)
PACCLIM=-1	PACCLIM=<値>によってプログラム指令された制限をキャンセルします。	

注記

PACCLIM の影響は、セッティングデータ SD42500 \$SC_SD_MAX_PATH_ACCEL の影響と同様です(最大軌跡加速度)。ただし、SD42500 と対象的に PACCLIM はブロック同期で動作します。

注記

制限値を計算する場合、SD42500 \$SC_SD_MAX_PATH_ACCEL の値は、SD42502 \$SC_IS_SD_MAX_PATH_ACCEL が「TRUE」に設定されている場合だけ考慮されます。
PACCLIM と SD42500 \$SC_SD_MAX_PATH_ACCEL の両方が有効な場合、2 つの制限値の中で小さな値が有効になります。

例

プログラムコード	コメント
...	
N1000 G0 X0 Y0 F10000 G64 G710	
N1100 G1 X20 RNDM=5	
N1200 G1 Y20 PACCLIM=0.5	; Y 方向の軸スライドは最大 0.5 m/ss で加速/減速します。
N1300 G1 X0 Y40	; X 方向と Y 方向の軸スライドの軌跡移動は、最大 0.5 m/ss で加速/減速します。
N1400 G1 Y0	
M30	

4.14.12 有効な連結によるブロック変更動作(CPBC)

CPBC 命令は、パートプログラムで有効な連結によってブロック変更を実行するために満たさなければならないブロック変更条件を指定します。

構文

CPBC [<従動軸>] = <条件>

意味

CPBC:	有効な連結によるブロック変更条件	
<従動軸>:	従動軸の軸識別子	
<条件>:	ブロック変更条件	
	タイプ:	STRING
	規格値	意味:ブロック変更が実行されます
	"NOC"	連結状態に関わりなく
	"IPOSTOP"	指令同期制御用
	"COARSE"	現在値同期制御「汎用」用
	"FINE"	現在値同期制御「精密」用

例

プログラムコード

```
; 以下の場合、ブロック変更が行われます:
;   - 従動軸 X2 への連結 == 有効
;   - 指令同期制御 == 有効
CPBC[X2]="IPOSTOP"
```

4.15 軸機能

4.15.1 軸入れ替え、主軸入れ替え(RELEASE、GET、GETD)

複数の軸または主軸は、1つのチャンネルでのみ補間できます。1つの軸が2つの異なるチャンネル間で切り替わる(パレット交換など)場合は、最初に軸を現在のチャンネルで有効にして、その後に別のチャンネルに移行してください。軸入れ替えは、複数のチャンネル間で有効です。

軸入れ替えの拡張機能

4.15 軸機能

軸/主軸は、先読みとメインランの間の同期制御で、先読み停止あり、または先読み停止なしでも入れ替えできます。軸の入れ替えは、次の方法によって有効にすることもできます。

- GET/GETD を自動的に使用して、軸コンテナ回転 AXCTSWE または AXCTWED で有効にできます。
- 軸入れ替え処理で、軸を他の軸と結合する場合は、回転フレームで有効にできます。
- シンクロナイズドアクションで有効にできます。これについては、「シンクロナイズドアクション」の「軸入れ替え RELEASE、GET」を参照してください。

工作機械メーカー

工作機械メーカーの取扱説明書を参照してください。軸入れ替えの場合は、1つの軸を、設定可能なマシンデータで、すべてのチャンネルで一義的であるように定義してください。また、軸入れ替えの特性もマシンデータを使用して設定できます。

構文

RELEASE (軸名称、軸名称、....) または RELEASE (S1)

GET (軸名称、軸名称、....) または GET (S2)

GETD (軸名称、軸名称、etc.) または GETD(S3)

GETD(直接の GET 命令)の場合は、軸が直接に、別のチャンネルから取得されます。このため、別チャンネルでは、この GETD 命令用の RELEASE 命令をプログラム指令してはいけません。また、他のチャンネルとの通信も確立してください(待機マークなど)。

意味

RELEASE (軸名称, 軸名称, ...):	軸を解放します。
GET (軸名称, 軸名称, ...):	軸を受け取ります。
GETD (軸名称, 軸名称, ...):	軸を直接受け取ります。
軸名称:	システムでの軸割り当て:AX1、AX2、... または機械軸名称を指定します。
RELEASE (S1):	主軸 S1、S2、...を解放します。
GET (S2):	主軸 S1、S2、...を受け取ります。
GETD (S3):	主軸 S1、S2、...を直接受け取ります。

先読み停止のない GET 要求

先読み停止の**ない** GET 要求の後に、軸が RELEASE (軸) または WAITP (軸) で再度有効になった場合は、その後の GET が、先読み停止**あり**の GET になります。

注意

軸割り当ての変更

GET で受け取られた軸または主軸は、キー RESET またはプログラム RESET の後でも、そのままこのチャンネルに割り当てられます。

プログラムの再起動時に、元のチャンネルで軸が必要となる場合は、入れ替えた軸または主軸をプログラムのなかで、再度割り当ててください。

電源投入時の軸の割り当ては、マシンデータで定義されたチャンネルに割り当てられます。

例

例 1:2 つのチャンネル間の軸入れ替え

6 つの軸のうち、チャンネル 1 で加工に使用するのは、1 番目、2 番目、3 番目、および 4 番目の軸です。

5 番目と 6 番目の軸は、チャンネル 2 でワークの入れ替えに使用します。

軸 2 は、2 つのチャンネル間で入れ替える必要があり、電源投入後は、チャンネル 1 に割り当てることができます。

チャンネル 1 のプログラム「MAIN」:

プログラムコード	コメント
INIT (2,"TRANSFER2")	; チャンネル 2 でプログラム TRANSFER2 を選択します。
N... START (2)	; チャンネル 2 でプログラムを起動します。
N... GET (AX2)	; 軸 AX2 を受け取ります。
...	
N... RELEASE (AX2)	; 軸 AX2 を解放します。
N... WAITM (1,1,2)	; チャンネル 1 と 2 で WAIT マークを待機し、両チャンネルで同期します。
...	; 軸入れ替えの後のプログラムの残り部分です。
N... M30	

チャンネル 2 のプログラム「TRANSFER2」:

プログラミング	コメント
N... RELEASE (AX2)	
N160 WAITM(1,1,2)	; チャンネル 1 と 2 で WAIT マークを待機し、両チャンネルで同期します。
N150 GET (AX2)	; 軸 AX2 を受け取ります。
...	; 軸入れ替えの後のプログラムの残り部分です。

4.15 軸機能

プログラミング	コメント
N... M30	

例 2:同期制御のない軸入れ替え

軸を同期する必要がない場合は、GET によって先読み停止はおこなわれません。

プログラミング	コメント
N01 G0 X0	
N02 RELEASE (AX5)	
N03 G64 X10	
N04 X20	
N05 GET (AX5)	; 同期制御が不要な場合は、このブロックの実行はできません。
N06 G01 F5000	; このブロックは実行できません。
N07 X20	; N04 の X 軸の位置決めと同じであるため、このブロックは実行できません。
N08 X30	; N05 の後に実行可能な最初のブロックです。
...	

例 3:先読み停止のない軸入れ替えの起動

必要条件:先読み停止のない軸入れ替えは、マシンデータで設定してください。

プログラミング	コメント
N010 M4 S100	
N011 G4 F2	
N020 M5	
N021 SPOS=0	
N022 POS[B]=1	
N023 WAITP (B)	; 軸 B は中立軸になります。
N030 X1 F10	
N031 X100 F500	
N032 X200	
N040 M3 S500	; 軸は先読み停止/REORG をおこないません。
N041 G4 F2	
N050 M5	
N099 M30	

たとえば主軸または軸 B が 180°の角度だけ移動した後に、PLC 軸として N023 ブロックの直後に 1°の角度だけ戻った場合、この軸は中立状態に戻ります、このため、N40 ブロックで先読み停止をおこないません。

詳細情報

軸入れ替えの必要条件

- 軸を使用するすべてのチャンネルで、マシンデータを使用して軸を定義してください。
- 電源投入後に軸をどのチャンネルに割り当てるかを、**軸**マシンデータで定義してください。

説明

軸の解放:RELEASE

軸を有効にするときの注意事項:

1. 軸は座標変換に使用しないでください。
2. 軸リンク(法線方向制御)に使用されるすべての軸を有効にしてください。
3. 同時位置決め軸は、この事例では入れ替えることができません。
4. ガントリマスタ軸のスレーブ軸はすべて、マスタ軸とともに移行します。
5. 連結軸の場合(連結移動、軸間連動機能、電子ギヤ)は、そのグループのマスタ軸のみを有効にすることができます。

軸の受け取り:GET

実際の軸の入れ替えはこの命令でおこなわれます。プログラム指令した命令の対象であるチャンネルが、軸を受け取ります。

GET の動作:

同期制御による軸入れ替え:

軸は、入れ替えのときに別のチャンネルまたは PLC に割り当てたが、「WAITP」また G74 を実行して原点同期を行っていない、または GET の前に残移動距離を削除した場合は、常に原点同期を行ってください。

- (STOPRE として)先読み停止が実行されます。
- 実行は、入れ替えが完了するまで中断されます。

自動の「GET」

軸が、原則としてチャンネルで使用でき、現在「チャンネル軸」として定義されている場合は、GET が自動的に実行されます。軸がすでに同期している場合は、先読み停止は行われません。

4.15 軸機能

軸入れ替え動作の応用設定

軸の移行タイミングは、マシンデータを使用して、次のように設定できます。

- WAITP で軸が中立状態になると、2 つのチャンネル間で自動的に軸入れ替えもおこなわれます(以前と同じ動作です)。
- 軸コンテナ回転を要求すると、実行チャンネルへの割り当てが可能な軸コンテナのすべての軸が、GET または GETD を自動的に使用して、このチャンネルに入ります。軸コンテナ回転が完了すると、その後の軸入れ替えが、再び可能になります。
- 中間ブロックがメインランに挿入されると、確認がおこなわれ、再解析が必要かどうか特定されます。再解析は、このブロックの軸ステータスが現在の軸ステータスに一致しない場合にのみ必要です。
- 先読みとメインランの間の、先読み停止と同期制御のある GET ブロックの代わりに、軸を先読み停止なしで入れ替えることができます。この場合は、単に中間ブロックが GET 要求により生成されます。メインランでこのブロックが実行されると、システムにより、このブロックの各軸のステータスが実際の軸ステータスに一致するかどうかを確認します。

4.15.2 別のチャンネルへの軸の移行(AXTOCHAN)

AXTOCHAN 言語命令を使用すると、別のチャンネルに移動する軸を要求できます。軸は、NC パートプログラムとシンクロナイズドアクションの両方から、対応するチャンネルに移動できます。

構文

AXTOCHAN (軸名称, チャンネル番号[, 軸名称, チャンネル番号[, ...]])

意味

要素	説明
AXTOCHAN:	特定のチャンネル用の軸を要求します
軸名称:	システムでの軸割り当て:X、Y、…または、関連する機械軸名称の入力です。実行チャンネルは、同じチャンネルとは限りません。また、軸に対して現在の補間権限のあるチャンネルとも限りません。
チャンネル番号:	軸の割り当て先のチャンネルの名称です。

注記**競合位置決め軸と PLC のみの制御軸**

PLC 軸は、競合位置決め軸のようにチャンネルを入れ替えることはできません。PLC のみが制御する軸を、NC プログラムに割り当てることはできません。

詳細情報:『機能マニュアル、軸と主軸』

例**NC プログラムの AXTOCHAN**

軸 X と Y は、1 番目と 2 番目のチャンネルで宣言されています。現在は、チャンネル 1 に補間権限があり、次のプログラムはこのチャンネルで起動されます。

プログラムコード	コメント
N110 AXTOCHAN (Y, 2)	; Y 軸を 2 番目のチャンネルに移動します。
N111 M0	
N120 AXTOCHAN (Y, 1)	; Y 軸を元に戻します (中立)。
N121 M0	
N130 AXTOCHAN (Y, 2, X, 2)	; Y 軸と X 軸を 2 番目のチャンネルに移動します (両軸は中立です)。
N131 M0	
N140 AXTOCHAN (Y, 2)	; Y 軸を 2 番目のチャンネルに移動します (NC プログラム)。
N141 M0	

詳細情報**NC プログラムの AXTOCHAN**

GET は、同じチャンネルの NC プログラムに対してその軸が要求された場合にのみ、実行されます(つまり、システムは状態が実際に変わるまで待機します)。その軸が別のチャンネルに要求されるか、同じチャンネルで中立軸になる場合は、それに従って要求が送られます。

シンクロナイズドアクションからの AXTOCHAN

同じチャンネルに対して軸が要求される場合は、シンクロナイズドアクションの AXTOCHAN がシンクロナイズドアクションの GET に割り当てられます。この場合、軸は、同じチャンネルに対する 1 回目の要求時に中立軸になります。2 回目の要求時には、NC プログラムの GET 要求と同様に、軸が NC プログラムに割り当てられます。シンクロナイズドアクションからの GET 要求について詳しくは、「シンクロナイズドアクション」を参照してください。

4.15.3 軸機能(AXNAME、AX、SPI、AXTOSPI、ISAXIS、AXSTRING、MODAXVAL)

「AXNAME」は、たとえば軸の名称が不明な場合に、一般的に有効なサイクルを生成するために使用します。

「AX」は、ジオメトリ軸と同期軸を間接的にプログラム指令する場合に使用します。軸識別子は、**AXIS** タイプの変数で保存されるか、「AXNAME」や「SPI」などの命令により提供されます。

「SPI」は、主軸同期などの主軸に軸機能をプログラム指令している場合に使用します。

「AXTOSPI」を使用して、軸識別子を主軸インデックスへ変換します(SPI の逆の機能です)。

「AXSTRING」を使用して、軸識別子(データタイプ **AXIS**)を文字列に変換します(「AXNAME」の逆の機能です)。

「ISAXIS」を汎用サイクルで使用して、特定のジオメトリ軸が存在できるようにして、後続の\$P_AXNX 呼び出しがエラーメッセージで中止されないようにします。

「MODAXVAL」は、モジュロ回転軸のモジュロ位置を特定するために使用します。

構文

```
AXNAME ("文字列")
AX[AXNAME ("文字列")]
SPI (n)

AXTOSPI (A) または AXTOSPI (B) または AXTOSPI (C)
AXSTRING ( SPI (n) )
ISAXIS (<ジオメトリ軸番号>)
<モジュロ位置>=MODAXVAL (<軸>,<軸位置>)
```

意味

AXNAME:	入力文字列を軸識別子に変換します。入力文字列には、有効な軸名称を含めてください。
AX:	可変軸識別子
SPI:	主軸番号を軸識別子に変換します。転送パラメータには、有効な主軸番号を含めてください。
n:	主軸番号
AXTOSPI:	軸識別子を整数の主軸インデックスへ変換します。「AXTOSPI」は、SPI の逆の機能になります。
X, Y, Z:	変数または定数としての AXIS タイプの軸識別子

AXSTRING:	関連する主軸番号で文字列を出力します。
ISAXIS:	指定したジオメトリ軸が存在するかどうかを確認します。
MODAXVAL:	モジュロ回転軸に対して、モジュロ位置を特定します。これは、パラメータで設定したモジュロ範囲を基準とするモジュロ余りに対応します(初期設定では、0～360°です。モジュロ範囲の開始位置とサイズは、MD30340 MODULO_RANGE_START と MD30330 \$MA_MODULO_RANGE で変更できます)。

注記

SPI の拡張

軸機能 SPI(n)を使用すると、フレーム成分の読み取りと書き込みも実行できます。つまり、\$P_PFRAME[SPI(1),TR]=2.22 構文などでフレームを記述できます。

さらにアドレス AX[SPI(1)]=<軸位置>で軸の位置決めをプログラム指令して、軸を移動できます。必要条件は、主軸が位置決めモードまたは軸モードであることです。

例

例 1:AXNAME、AX、ISAXIS

プログラムコード	コメント
OVRA[AXNAME("径方向軸")]=10	; 径方向軸のオーバーライド
AX[AXNAME("径方向軸")]=50.2	; 径方向軸の終了位置
OVRA[SPI(1)]=70	; 主軸 1 のオーバーライド
AX[SPI(1)]=180	; 主軸 1 の終了位置
IF ISAXIS(1) == FALSE GOTOF CONTINUE	; 横軸が使用可能?
AX[\$P_AXN1]=100	; 横軸を移動します
CONTINUE:	

例 2 :AXSTRING

AXSTRING[SPI(n)]でプログラム指令すると、主軸に割り当てた軸の軸インデックスが主軸番号として出力されなくなり、代わりに文字列「Sn」が出力されます。

プログラムコード	コメント
AXSTRING[SPI(2)]	; 文字列「S2」が出力されます。

例 3 :MODAXVAL

モジュロ回転軸 A のモジュロ位置を特定します。

軸位置 372.55 は、計算の開始値です。

パラメータ設定したモジュロ範囲は次のように 0～360°です。

4.15 軸機能

```
MD30340 MODULO_RANGE_START = 0
```

```
MD30330 $MA_MODULO_RANGE = 360
```

プログラムコード	コメント
R10=MODAXVAL (A, 372.55)	; 算出したモジュロ位置は R10 = 12.55 です。

例 4 :MODAXVAL

プログラム指令した軸識別子がモジュロ回転軸を表わしていない場合は、変換される値(<軸位置>)を、変換せずに返します。

プログラムコード	コメント
R11=MODAXVAL (X, 372.55)	; X は直線軸です; R11 = 372.55

4.15.4 置換可能なジオメトリ軸(GEOAX)

「切り替え可能なジオメトリ軸」機能により、マシンデータによって設定されたジオメトリ軸を他のチャネル軸に置き換えることができます。

構文

```
GEOAX (<n>, <チャネル軸>, <n>, <チャネル軸>, <n>, <チャネル軸>)
GEOAX ()
```

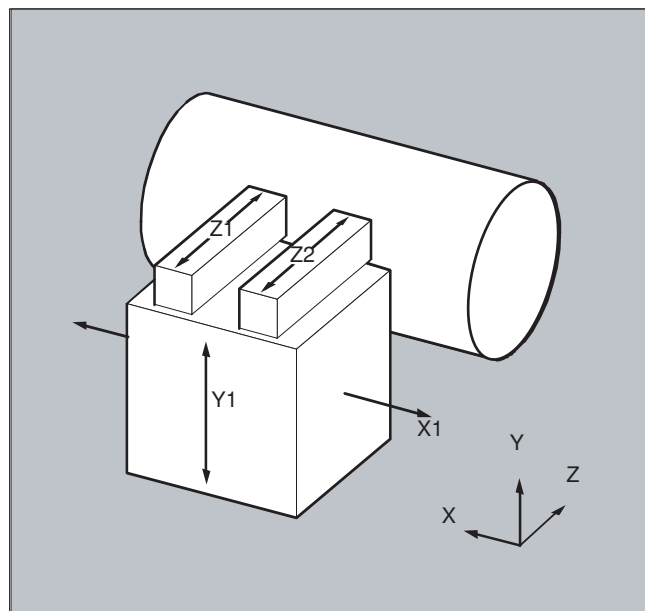

意味

GEOAX (...)	切り替え可能なジオメトリ軸の機能 注: パラメータ指定のない GEOAX () では、マシンデータでパラメータ設定されたジオメトリ軸の基本設定が再度有効になります。
<n>	指定されたチャネル軸と置換されるジオメトリ軸の番号 値の範囲: 0、1、2、3 注: 0:指定されたチャネル軸は、置換されずにジオメトリ軸グループから削除されます。 1:1 番目のジオメトリ軸 ≡ WCS の座標軸 X (横軸) 2:2 番目のジオメトリ軸 ≡ WCS の座標軸 Y (縦軸) 3:3 番目のジオメトリ軸 ≡ WCS の座標軸 Z (垂直軸)
<チャネル軸>	ジオメトリ軸グループに追加されるチャネル軸の名称

例

例 1:交互にジオメトリ軸となる 2 つの軸の切り替え

チャネル軸 X1、Y1、Z1、Z2 を使用して、次のように工具スライドを移動できます。



4.15 軸機能

電源投入後にまず、3 番目のジオメトリ軸として Z1 が、「Z」というジオメトリ軸名称で有効になり、X1 と Y1 とともにジオメトリ軸グループを構成するように、ジオメトリ軸を設定します。

これで軸 Z1 と Z2 は、パートプログラムでジオメトリ軸 Z として交互に使用されるようになります。

プログラムコード	コメント
...	
N100 GEOAX (3, Z2)	; チャネル軸 Z2 が 3 番目のジオメトリ軸 (Z) として機能します。
N110 G1 ...	
N120 GEOAX (3, Z1)	; チャネル軸 Z1 が 3 番目のジオメトリ軸 (Z) として機能します。
...	

例 2:チャネル軸が 6 つの場合のジオメトリ軸の変更

機械には、名称が XX、YY、ZZ、U、V、W の 6 つのチャネル軸があります。

マシンデータによるジオメトリ軸構成の基本設定は、次のとおりです。

チャネル軸 XX =1 番目のジオメトリ軸(X 軸)

チャネル軸 YY =2 番目のジオメトリ軸(Y 軸)

チャネル軸 ZZ =3 番目のジオメトリ軸(Z 軸)

プログラムコード	コメント
N10 GEOAX ()	; ジオメトリ軸の基本構成が有効です。
N20 G0 X0 Y0 Z0 U0 V0 W0	; すべての軸を位置 0 に早送りします。
N30 GEOAX (1, U, 2, V, 3, W)	; チャネル軸 U が 1 番目の (X)、V が 2 番目の (Y) ; W が 3 番目のジオメトリ軸 (Z) になります。
N40 GEOAX (1, XX, 3, ZZ)	; チャネル軸 XX が 1 番目の (X)、ZZ が 3 番目の ; ジオメトリ軸 (Z) になります。チャネル軸 V は 2 番目の ; ジオメトリ軸 (Y) のままです。
N50 G17 G2 X20 I10 F1000	; X/Y 平面上の一周円です。チャネル軸 ; XX と V が移動します。
N60 GEOAX (2, W)	; チャネル軸 W が 2 番目のジオメトリ軸 (Y) になります。
N80 G17 G2 X20 I10 F1000	; X/Y 平面上の一周円です。チャネル軸 ; XX と W が移動します。
N90 GEOAX ()	; 初期状態にリセットします。
N100 GEOAX (1, U, 2, V, 3, W)	; チャネル軸 U が 1 番目の (X)、V が 2 番目の ; (Y)、W が 3 番目のジオメトリ軸 (Z) になります。
N110 G1 X10 Y10 Z10 XX=25	; チャネル軸 U、V、W をそれぞれ ; 位置 10 へ移動します。XX が付加軸として位置 25 に移動します。

プログラムコード	コメント
N120 GEOAX(0,V)	; V をジオメトリ軸グループから解除します。 ; U は 1 番目のジオメトリ軸 (X)、 ; W は 3 番目のジオメトリ軸 (Z) のままです。 ; 2 番目のジオメトリ軸 (Y) にはそのまま、何も割り当てられません。
N130 GEOAX(1,U,2,V,3,W)	; チャネル軸 U は 1 番目の (X) のまま、V は ; 2 番目の (Y) になり、W は 3 番目のジオメトリ軸 (Z) のままです。
N140 GEOAX(3,V)	; V は 3 番目のジオメトリ軸 (Z) になります。 ; ここで、W は上書きされるため、ジオメトリ軸グループ ; から解除されます。2 番目のジオメトリ軸 (Y) は ; 何も割り当てられないままです。

マシンデータ

軸構成

ジオメトリ軸、付加軸、機械軸のチャネル軸への割り当て:

- MD10000 \$MN_AXCONF_MACHAX_NAME_TAB
- MD20050 \$MC_AXCONF_GEOAX_ASSIGN_TAB
- MD20060 \$MC_AXCONF_GEOAX_NAME_TAB
- MD20070 \$MC_AXCONF_MACHAX_USED
- MD20080 \$MC_AXCONF_CHANAX_NAME_TAB
- MD35000 \$MA_SPIND_ASSIGN_TO_MACHAX

リセット応答

変更されたジオメトリ軸割り当てのリセット応答:

- MD20110 \$MC_RESET_MODE_MASK、ビット 12
- MD20118 \$MC_GEOAX_CHANGE_RESET

NC スタートの動作

- MD20112 \$MC_START_MODE_MASK、ビット 12

PLC ユーザープログラムへの通知

ジオメトリ軸が変更されたときに PLC インタフェースで出力される M 命令のパラメータ設定オプション。

- MD22532 \$MC_GEOAX_CHANGE_M_CODE

必要条件

ジオメトリ軸の切り替えなし

- 次のいずれかの機能が動作中の場合、ジオメトリ軸の切り替えは実行できません。
 - 座標変換
 - スプライン補間
 - 工具径補正
 - 工具仕上げオフセット
- ジオメトリ軸ともう 1 つのチャネル軸が同じ名称の場合。
- ジオメトリ軸の切り替えに参加する軸のいずれかが、ブロック全体での位置決め軸や従動軸の軸連結などのブロックの境界を越えて実行される操作に関与する場合。

回転軸

回転軸はジオメトリ軸としてプログラムできません。

置換後の軸の状態

ジオメトリ軸グループ内で切り替えで置換した軸は、そのチャネル軸名称による切り替え操作後に、追加軸としてプログラム指令できます。

フレーム、プロテクションゾーン、作業領域リミット

すべてのフレーム、プロテクションゾーン、および作業領域リミットは、ジオメトリ軸の切り替え後に解除されます。

極座標

GEOAX でジオメトリ軸を置換すると、G17～G19 による平面変更と同様に、モーダル極座標を 0 の値に設定します。

DRF、WO

ハンドルオフセット(DRF)または外部ゼロオフセット(WO)が使用可能で、切り替え後もそのまま有効です。

ジオメトリ軸の基本構成

GEOAX () 命令は、ジオメトリ軸グループの基本構成を呼び出します。

システムは、POWER ON 後、および「レファレンス点復帰」モードへの切り替え時に、自動的に基本構成に戻ります。

工具長補正

動作中の工具長補正は切り替え操作後も有効です。ただし、新しく追加した、または位置を置換したジオメトリ軸の場合は、移動が完了していないと見なされます。これらのジオメ

トリ軸の最初の動作命令では、それに対応して、得られる移動距離が、工具長補正とプログラム指令した移動距離の和になります。

置換操作後も軸グループの位置を保持するジオメトリ軸は、工具長補正に対する状態も保持します。

動作中の座標変換のジオメトリ軸構成

- 動作中の座標変換に対して座標変換マシンデータによってパラメータ設定されるジオメトリ軸構成は、「切り替え可能なジオメトリ軸」機能を使用して変更できません。
- 座標変換に対する異なるジオメトリ軸構成には、座標変換マシンデータで異なるデータセットをパラメータ設定してください。
- GEOAX で変更したジオメトリ軸構成は、座標変換を起動すると解除されます。
- ジオメトリ軸に関しては、動作中の座標変換の座標変換固有のジオメトリ軸パラメータ設定が、ジオメトリ軸の切り替えに関連するパラメータ設定よりも優先されます。
例:座標変換が動作中です。マシンデータに従って、座標変換はチャンネルリセットで保持されます。同時に、ジオメトリ軸の基本設定がチャンネルリセットで復元されます。座標変換に対して指定されたジオメトリ軸構成が保持されます。
- 座標変換をオフにすると、パラメータ設定されたジオメトリ軸構成の基本設定が再度有効になります。

JOG モード、REF 機械機能

JOG モード、REF 機械機能(レファレンス点復帰)に切り替えると、マシンデータでパラメータ設定されたジオメトリ軸構成が有効になります。

4.15.5 軸コンテナ(AXCTSWE、AXCTSWED、AXCTSWEC)

「AXCTSWED」命令または「AXCTSWE」命令を使用して、指定した軸コンテナの回転を有効にします。

「AXCTSWEC」命令により、以前に設定された軸コンテナ回転のイネーブルはキャンセルされます。

構文

```
AXCTSWE (<ID>)  
AXCTSWED (<ID>)  
AXCTSWEC (<ID>)
```

4.15 軸機能

意味

AXCTSWE:	軸コンテナの回転の有効化 プログラムの処理は「AXCTSWE」で停止しません。 回転は、軸コンテナ関連のすべてのチャンネルが有効になると直ちに実行されます。	
AXCTSWED:	軸コンテナ関連の他のチャンネルを考慮せずに軸コンテナの回転を有効化 注 - パートプログラムまたはシンクロナイズドアクションのセットアップを簡素化する命令タイプ - 軸コンテナ関連の他のチャンネルに関する動作は、以下で指定できます。 MD12760 \$MN_AXCT_FUNCTION_MASK、ビット 0	
AXCTSWEC:	軸コンテナの回転の有効化の取り消し 注 回転がまだ起動していない場合にのみ、軸コンテナの回転のイネーブルはキャンセルできます。 \$AN_AXCTSWA[<軸コンテナ>] == 0 システム変数については、「軸コンテナ(AXCTSWE、AXCTSWED、AXCTSWEC) (ページ 1025)」を参照してください。	
<ID>:	軸コンテナまたはコンテナ軸の識別子	
	CT<番号>:	軸コンテナの初期設定の識別子: MD12750 \$MN_AXCT_NAME_TAB 例: 「CT1」
	<コンテナ>:	軸コンテナのユーザー用識別子 MD12750 \$MN_AXCT_NAME_TAB 例: 「CONTAINER_1」
	<軸>:	チャンネル内の既知のコンテナ軸の識別子

注記

移動量

軸コンテナ回転の移動量は、次のセッティングデータで設定します。

SD41700 \$SN_AXCT_SWWIDTH

詳細情報

診断

軸コンテナの現在の状態は、次のシステム変数で読み取ることができます。

システム変数	タイプ	説明
\$AC_AXCTSWA[<名称>]	BOOL	軸コンテナのチャンネル別の状態です。
\$AN_AXCTSWA[<軸コンテナ>]	BOOL	軸コンテナの NCU 別の状態です。
\$AN_AXCTSWE[<軸コンテナ>]	INT	軸コンテナの回転のスロット別の状態です。 システム変数で、軸コンテナスロットビット 単位の状態が提供されます。各ビットはスロ ットに対応しています。
\$AN_AXCTAS[<軸コンテナ>]	INT	軸コンテナの切り替えに使用されるロケーシ ョン(スロット)の数

自動 GET/GETD による軸コンテナの回転

次のマシンデータを使用して、「AXCTSWE」命令での自動「GET/GETD」により、チャンネルのすべてのコンテナ軸がチャンネルに入るように設定できます。軸入れ替えは、コンテナ回転後にのみ再度可能になります。

MD10722 \$MN_AXCHANGE_MASK、ビット 1 = 1

注記

自動「GET/GETD」による軸コンテナ回転は、「メイン軸」状態の軸(PLC 軸など)では実行されません。これは、軸コンテナ回転のためには軸がその状態を終了する必要があるためです。

4.15.6 有効な軸位置を待機します(WAITENC)

言語命令「WAITENC」を使用して、NC プログラムは、同期または同期解除された軸位置が、MD34800 \$MA_WAIT_ENC_VALID = 1 で設定された軸で使用可能になるまで待機します。

4.15 軸機能

ASUB を開始したり、運転を JOG モードに変更したりすると、待機状態が中断することがあります。プログラムが続行されると、該当する場合は、再び待機状態になります。

注記

操作画面では、保留状態の「検出を待機」を使用して、待機状態が表示されます。

構文

「WAITENC」は、任意の NC プログラムのプログラム区間でプログラム指令できます。
専用ブロックでプログラム指令してください。

```
...  
WAITENC  
...
```

例

「WAITENC」は、以下の用途例で示すように、イベント起動のユーザープログラム.../_N_CMA_DIR/_N_PROG_EVENT_SPF などを使用します。

用途例 :方向座標変換による電源切断後の工具の引き戻し

工具オリエンテーションによる加工が、停電により中断されました。
停電から復旧すると、イベント起動のユーザープログラム.../_N_CMA_DIR/_N_PROG_EVENT_SPF が呼び出されます。

イベント起動のユーザープログラムでは、システムが、「WAITENC」によって同期または同期解除された位置決め軸を待機し、その後に、ワークを工具方向に配置するフレームを計算できるようになります。

プログラムコード	コメント
...	
IF \$P_PROG_EVENT == 4	; 起動しています。
IF \$P_TRAFO <> 0	; 座標変換が選択されています。
WAITENC	; 旋回軸の有効な軸位置を待機します。
TOROTZ	; WCS の Z 軸を工具軸の方向に回転します。
ENDIF	
M17	
ENDIF	
...	

その後、JOG モードで、工具軸の方へ後退移動によって工具を後退させることができます。

4.15.7 プログラム指令可能なパラメータセット切り替え(SCPARA)

SCPARA 命令を使用して、軸の特定のパラメータセットへの切り替えを要求できます。

注記

ねじ切り時のパラメータセット切り替えなし

ねじ切り G33 およびタッピング G331/G332 時には、パラメータセットは制御装置によって選択されるため、切り替えできません。

パラメータセット切り替えの無効化

パラメータセット切り替えは、NC/PLC インタフェース経由で要求することもできます。切り替えの競合を避けるために、次のように NC/PLC インタフェース経由で NC (SCPARA) のパラメータセット切り替えをロックできます。

DB31, ... DBX9.3 (NC によって無効にするパラメータセットの指定)

注記

NC/PLC インタフェース経由でパラメータセット切り替えがロックされているときに SCPARA がパラメータセット切り替えを要求すると、切り替えはエラーメッセージを表示せずに拒否されます。

構文

SCPARA [<軸>]=<値>

意味

SCPARA:	命令:パラメータセットの切り替え	
<軸>:	軸識別子(チャネル軸)	
	タイプ:	AXIS
<値>:	パラメータセット番号:1, 2, 3, ... 最大パラメータブロック番号	

4.15 軸機能

例

プログラムコード	コメント
...	
N110 SCPARA[X]= 3	; 選択:軸 X、3 パラメータセット
...	

詳細情報

パラメータセット切り替えの有効化

軸のパラメータセット切り替えは明示的に有効にしてください。

MD35590 \$MA_PARAMSET_CHANGE_ENABLE[<軸>]

パラメータセット番号の読み取り

選択したパラメータセット(指定したパラメータセット)の番号は、システム変数\$AA_SCPARを介して読み取ることができます。

4.15.8 調整の有効化/無効化(CADAPTON、CADAPTOF)

事前定義された CADAPTON()および CADAPTOF()プロシージャを使用すると、ダイナミック応答または制御パラメータを調整する事前定義済みの補正をパートプログラムから有効化、更新および無効化できます。これらは CYCLE782 (ページ 1289)でも使用されます。

構文

```

...
CADAPTON(<結果>,<軸>,<InVar>[,<InVal>])
...
CADAPTOF(<結果>,<軸>,<InVar>)
...

```

意味

CADAPTON () :	補正関係を有効化します			
CADAPTOF () :	補正関係を無効化します 注記: 有効な補正関係(MD16501 = 1) CADAPTOF () を無視し、以前にプログラム指令された入力値(<InVal>)が有効なままです。			
<結果>:	結果変数:ステータスの戻り値(参照パラメータによる呼び出し)			
	データタイプ:	INT		
	値:	0	エラーなし	
		1	パラメータ設定された有効な調整テーブルがありません	
		2	<軸>パラメータ無効	
		3	<InVar>パラメータ無効	
		4	予備	
5	<InVal>パラメータ無効			
<軸>:	補正関係の入力軸の機械軸名称			
	データタイプ:	AXIS		
	値の範囲:	チャンネルで定義された機械軸の名称		
	注記: このパラメータを使用すると、これらの補正はアドレス指定され、MD16504 \$MN_CADAPT_INPUT_AX にパラメータ<軸>に対応する値を入力します。その他の<InVar>および<InVal>パラメータがこの軸に割り当てられます。			
<InVar>:	補正関係の入力変数			
	データタイプ:	INT		
	値:	1	軸の慣性	
		2	軸の位置	
3		軸速度		

4.15 軸機能

<InVal>:	補正関係の入力値 インテリジェント負荷調整のオプションパラメータ(<InVar>=1)。		
	データタイプ:	REAL	
	値:	> 0	実際の慣性モーメント
		= 0	負荷は定義されていないか、または未知です。特定のケースでは、割り当てられた補正が出力変数として置換値 1.0 を提供します。
		< 0	この呼び出しは<ステータス>=5 (パラメータ<InVal>無効)で終了します。有効化された最後の入力変数は有効なままです。
注記: 入力値がプログラム指令されていない場合、コントロールシステムの起動後に最後にプログラム指令された値またはデフォルト値(= 0)が有効です。			

例

プログラムコード	コメント
DEF INT RESULT	; 結果変数の定義
...	
CADAPTON (RESULT,MX1,2)	; 補正の有効化
IF RESULT <> 0	; 結果変数の評価 (これは各 CADAPTON/CADAPTOF 命令後に必要です)
MSG ("CADAPT-RESULT=" << RESULT)	
STOPRE	
SETAL (61000)	
ENDIF	
...	
CADAPTOF (RESULT,MX1,2)	; 補正の無効化
...	; 結果変数の評価
...	

詳細情報

リンク軸

機能をリンク軸に適用できません(NCU リンク機能)。この場合、結果変数が戻り値 2 (「パラメータ<軸>が無効」)を提供します。

ブロック検索

- 計算なしのブロック検索

プログラムの開始とターゲットブロックの間にプログラムされた CADAPTON/CADAPTOF 命令が無視されます。ユーザーは、加工部分に関連する調整が選択されるよう、検索ターゲットを選択する必要があります。

- 輪郭またはブロック終点における計算ありのブロックサーチ

有効化または無効化命令、ならびに考えられるプログラムされた入力値が収集され、検索終了時の実際の状態が検索アクションブロックで出力されます。

- プログラムテスト(SERUPRO)モードにおける計算ありのブロック検索

SERUPRO が無効の場合、CADAPTON/CADAPTOF 命令を使用して実現される調整の状態が、実際の操作に移行されます。

4.15.9 FIR 加々速度フィルタのダイナミックモードへの設定(CALCFIR)

ダイナミックモードを変更した後は、FIR ローパス加々速度フィルタが有効なチャンネル内のすべての軸について同じ減衰動作と輪郭精度を実現するために、ダイナミック応答関連の FIR フィルタ設定を計算し、NC 音声命令 CALCFIR を使用して有効にできます。

用途

CALCFIR は CUST_832 メーカサイクルで使用します。

標準ワークフロー

CYCLE832 は、自由曲面加工用の NC プログラムの最初にプログラム指令されます。プログラムを実行すると、CYCLE832 がメーカサイクル CUST_832 を呼び出し、工作機械メーカが CUST_832 に実装した NC 言語命令 CALCFIR が実行されます。

有効性

CALCFIR によって上書きされたフィルタ設定は、次の CALCFIR 呼び出しで再度上書きされるまで有効なままとなります。G グループ 59(軌跡補間のダイナミック応答モード)の命令を使用して、実際のダイナミック応答モードの変更をプログラム指令する場合は、これが必ず必要です。

コントロールシステムの電源オン/チャンネルリセット/プログラム終了リセット時の動作

コントロールシステムの電源がオンになるか、チャンネル/プログラム終了リセット時には、FIR フィルタ設定が有効になり、G グループ 59 の初期設定に合わせられます。

4.15 軸機能

必要条件

- 「Top Speed」オプションが設定されています。

注記

「Top Speed」で最適な結果を実現するには、「Top Surface」オプションも使用することをお勧めします。

- 補間グループの軸では以下の条件を満たしてください。
 - FIR ローパス加々速度フィルタが有効です。
 - ダイナミック応答に依存する FIR フィルタ設定が行われています。
 - CALCFIR によるフィルタ設定の上書きが有効になります。
- 変化率制限 SOFT/SOFTA が有効です。

構文

CALCFIR が別のブロックにプログラム指令されています。メーカサイクル CUST_832 の呼び出しは、ダイナミック応答モードをプログラム指令した直後に実行されます。

```
...
DYN...
CALCFIR
...
```

意味

DYN...:	ダイナミック応答モードを選択するための G グループ 59 のコマンド	
	DYNNORM:	標準のダイナミック応答の起動
	DYNPOS:	位置決めモード、タッピングのダイナミック応答の起動
	DYNROUGH:	粗削りのダイナミック応答の起動
	DYNSEMIFIN:	荒仕上げのダイナミック応答の起動
	DYNFINISH:	仕上げのダイナミック応答の起動
	DYNPREC:	滑らかな仕上げのダイナミック応答の起動
CALCFIR:	あらかじめ設定されている FIR ローパス加々速度フィルタをダイナミックに設定する手順	

注記

CALCFIR は、自動的な NEWCONFIG を実行し、FIR ローパス加々速度フィルタのパラメータを割り付けるための軸マシンデータと CPRECON (ページ 983) の両方で FIR フィルタ演算の結果を有効にします。自動的な NEWCONFIG は、追加の NEWCONFIG 関連のマシンデータが有効になることを意味します。

境界条件

シンクロナイズドアクションでの使用

CALCFIR は、シンクロナイズドアクションではプログラム指令できません。

4.15.10 パートプログラムにおけるドライブパラメータの読み書き(DRVPRD、DRVPRW)

機械メーカーは、音声コマンド DRVPRD と DRVPRW により、ドライブパラメータを軸固有にパートプログラムレベルで読み書きできます。

DRVPRD と DRVPRW は定義済みプロシージャであり、NC ブロックのみに存在する必要があるため、いつでもチャンネルごとに単一のドライブパラメータを読み取るか、または書き込むことしかできません。

DRVPRD と DRVPRW は、機械メーカーが作成したサイクルで使用します。

前提条件

以下の要件が適用されます。

- NC 制御ドライブのセットアップが完了している。
- CST ディレクトリ(Siemens)または CMA ディレクトリ(メーカー)に、読み取り権限と書き込み権限が常にある。

構文

DRVPRD (<結果>, <軸>, <DrvParNo>, <DrvParIdx>, <値>)
DRVPRW (<結果>, <軸>, <DrvParNo>, <DrvParIdx>, <値>)

4.15 軸機能

意味

DRVPRD () :	ドライブパラメータを 読み取る ための定義済みプロシージャ		
DRVPWR () :	ドライブパラメータを 書き込む ための定義済みプロシージャ		
<結果>:	結果変数:ステータスの戻り値		
	データタイプ:	INT	
	値:	= 0	機能が実行され、エラーは発生しませんでした。
		> 0	機能が実行され、エラーが発生しました。 詳しくは、「詳細」>「戻り値による診断」を参照してください。
<軸>:	機械軸名称		
	データタイプ:	AXIS	
	値の範囲:	チャンネルで定義された機械軸の名称	
<DrvParNo> :	ドライブパラメータ番号		
	データタイプ:	INT	
<DrvParIdx> :	ドライブパラメータインデックス		
	データタイプ:	INT	
<値>:	ドライブパラメータの値 DRVPRD:読み取り値 DRVPWR:書き込む値		
	データタイプ:	REAL	

例

X 軸のパラメータ p1460[0]「速度コントローラ P ゲイン補正速度」を読み取ります。

プログラムコード	コメント
N100 DEF INT _STATUS	; 音声コマンドの戻り値用の変数定義
N110 DEF REAL _Result	; 読み取ったパラメータ値用の変数定義
N120 DRVPRD(_Status,AX1,1460,0,_Result)	; パラメータ p1460 の読み取り

詳細

戻り値による診断

ドライブパラメータへのアクセスが無効であるのは、音声コマンドの戻り値がゼロ以外の値を返す場合です。

戻り値には 2 つの数値範囲が予約されています。

音声コマンド呼び出し時の戻り値:

- 0 アクセスエラーはありません。
- 1 許可されない値です。
- 2 軸を使用できません。
- 3 ドライブを使用できません。
- 4 アクセス権がありません。
- 5 ブロック検索/プログラムテストでは不可能です。
- 6 ドライブにアクセスできません。

ドライブパラメータの読み取り/書き込み時の戻り値:

- 1000 許可されないパラメータ番号です。
- 1001 パラメータ値の変更ができません。
- 1002 下限違反または上限違反です。
- 1003 サブインデックスが正しくありません。
- 1004 配列がなく、サブインデックスがありません。
- ...
- 1107 コントローラが有効であるとき、書き込みアクセスが許可されません。
- 1110 試運転ステータスの場合にかぎり、書き込みアクセスが許可されます。モータ (p0010 = 3)
- ...
- 1204 書き込みアクセス権がありません。

音声コマンドの戻り値をドライブの戻り値から区別するため、値 1000 が加算されます。

詳細情報:SINUMERIK 診断マニュアル、アラーム 201042「プロジェクトダウンロード中のパラメータエラー」

プログラムシミュレーション中の動作

使用可能なドライブパラメータは、使用するドライブシミュレーションのタイプと範囲によって決まります。ドライブシミュレーションが見つからない場合、音声コマンドは戻り値 3 を返します。

機械メーカは、変数\$P_SIM を使用して、プログラムシミュレーションを特別に処理できます。

他の機能との相互作用

- 並行書き込み/読み取りアクセス
HMI またはその他のクライアントからの並行読み取り/書き込みアクセスは、音声コマンドの実行時間に影響する場合があります。
- REPOS
音声コマンドの実行が REORG イベントによって中断された場合、コマンドは反復されます。
- ブロック検索/プログラムテスト
ブロック検索およびプログラムテスト(SERUPRO)の場合、ドライブパラメータの読み取りおよび書き込みを行う音声コマンドはサポートされません。結果変数はアクセスエラー 5 を返します。
システム変数\$P_SEARCH と\$P_ISTEST をパートプログラムで使用して、ブロック検索またはプログラムテストが有効であるかどうかについて確認すると、DRVPRD と DRVPWR をスキップして置換方法を実装することなどができます。

制約事項

プログラミング中には、次の条件に従ってください。

- シンクロナイズドアクションでの使用
音声コマンドは、シンクロナイズドアクションでプログラムできません。シンクロナイズドアクションは、周期的でないアクセスの実行を待機しません。
音声コマンドをシンクロナイズドアクションで使用すると、アラーム 12571 が出力されます。
- 読み取り可能/書き込み可能なドライブパラメータ
SERVO DO または HLA DO の軸固有のドライブパラメータ、および NC の軸に割り当てられた PROFIdrive 標準ドライブのパラメータのみを読み書きできます。
次の DO のパラメータの読み取り/書き込みはできません。CU_I、CU_NX、CU3x、TM、HUB、INFEED。

4.16 軸連結

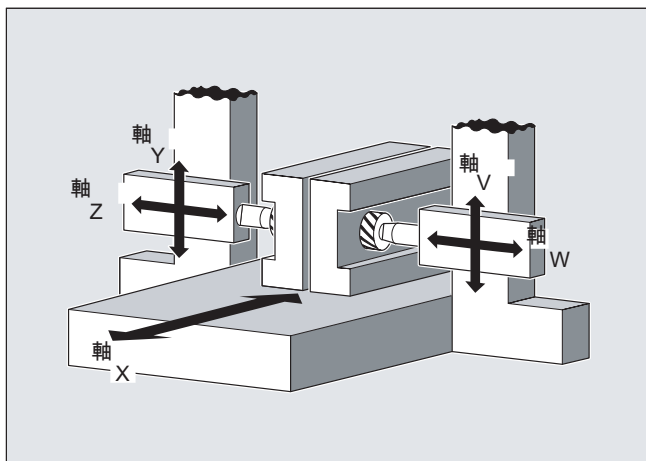
4.16.1 連結移動(TRAILON、TRAILOF)

定義したマスタ軸が移動すると、それに割り当てた連結移動軸(=スレーブ軸)は、連結係数を考慮に入れて、マスタ軸が指令する距離だけ移動します。

このようなマスタ軸とスレーブ軸を合わせて、連結軸と呼びます。

用途

- シミュレーション軸を使用した軸の移動。マスタ軸はシミュレーション軸で、連結軸は実軸です。このようにして、連結係数に応じて、実軸を移動できます。
- 2つの連結移動グループによる両面加工:
 - 1番目のマスタ軸 Y、連結移動軸 V
 - 2番目のマスタ軸 Z、連結移動軸 W



構文

TRAILON (<スレーブ軸>, <マスタ軸>, <連結係数>)
 TRAILOF (<スレーブ軸>, <マスタ軸>, <マスタ軸 2>)
 TRAILOF (<スレーブ軸>)

意味

TRAILON:	連結軸グループを起動、および定義する命令です。	
	効果:	モーダル

4.16 軸連結

<スレーブ軸>:	パラメータ 1:スレーブ軸の軸名称		
	注: 連結移動軸は、他の連結移動軸のマスタ軸としても機能できません。このようには、各種さまざまな連結軸グループを作成できます。		
<マスタ軸>:	パラメータ 2:スレーブ軸の軸名称		
<連結係数>:	パラメータ 3:連結係数		
	連結係数は、連結移動軸とマスタ軸の軌跡間に必要な関係を、次のように指定します。		
	<連結係数> =連結移動軸の移動量/マスタ軸の移動量		
	タイプ:	REAL	
	初期設定:	1	
	負の値を入力すると、マスタ軸と連結軸が逆方向に移動します。 連結係数をプログラム指令しない場合は、連結係数 1 が自動的に適用されます。		
TRAILOF:	連結軸グループを解除する命令です。		
	効果:	モーダル	
	2 個のパラメータを持つ TRAILOF は、以下のように指定したマスタ軸との連結のみを解除します。 TRAILOF (<スレーブ軸>,<マスタ軸>)		
	連結移動軸にマスタ軸が 2 つ含まれる場合は、3 個のパラメータで TRAILOF を呼び出して、両方の連結を解除できます。 TRAILOF (<スレーブ軸>,<マスタ軸>,<マスタ軸 2>)		
	次のように、マスタ軸を指定せずに TRAILOF をプログラミングしても同じ結果となります。 TRAILOF (<スレーブ軸>)		

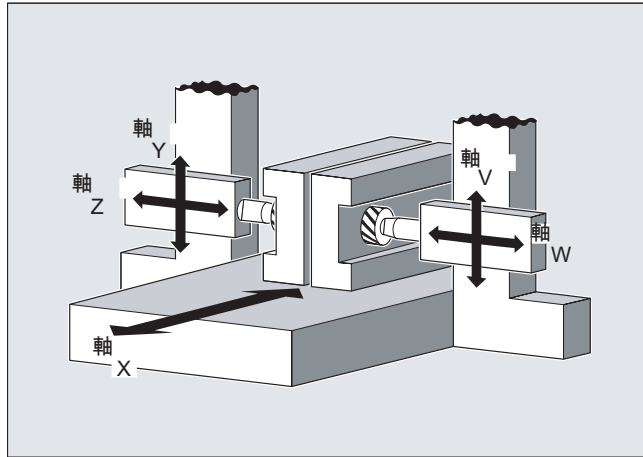
注記

連結軸の移動は常に、基本座標系(BCS)で実行されます。

同時に起動できる連結軸グループの数は、機械上の軸の組み合わせ数の最大値によってのみ制限されます。

例

ワークは、図に示す軸構成により両面加工されます。これをおこなうには、連結軸の組み合わせを2つ作成してください。



プログラムコード	コメント
...	
N100 TRAILON (V,Y)	; 1 番目の連結軸グループを適用します。
N110 TRAILON (W,Z,-1)	; 2 番目の連結軸グループを適用します。負の連結係数です。連結移動軸は、マスタ軸から反対方向に移動します。
N120 G0 Z10	; それぞれ反対の軸方向へ Z 軸と W 軸が切り込みます。
N130 G0 Y20	; 同じ軸方向へ Y 軸と V 軸が切り込みます。
...	
N200 G1 Y22 V25 F200	; 連結移動軸 V の連結移動と独立移動を重畳します。
...	
TRAILOF (V,Y)	; 1 番目の連結軸グループを解除します。
TRAILOF (W,Z)	; 2 番目の連結軸グループを解除します。

詳細情報

軸タイプ

連結軸グループは、任意の直線軸と回転軸を組み合わせで作成できます。シミュレーション軸は、マスタ軸としても定義できます。

連結移動軸

2つまでのマスタ軸を同時に、1つのスレーブ軸に割り当てることができます。連結軸の組み合わせが異なるように割り当てます。

4.16 軸連結

連結移動軸は、使用可能なすべての範囲の動作命令(G0、G1、G2、G3 など)でプログラム指令できます。連結軸は、独立して定義された軌跡だけでなく、連結係数に基づく連結軸のマスタ軸から計算された軌跡も移動します。

応答性制限

応答性制限は、連結軸グループの適用タイプによって、次のように異なります。

- パートプログラムでの起動

起動がパートプログラムで実行され、起動されたチャンネルですべてのマスタ軸がプログラム軸として有効である場合は、マスタ軸の移動中に、すべての連結移動軸のダイナミック応答が考慮され、連結移動軸の過負荷が防止されます。

起動中のチャンネルで、プログラム軸として有効でないマスタ軸によりパートプログラムで起動が実行された場合(\$AA_TYP ≠ 1)は、マスタ軸の移動中に、連結移動軸のダイナミック応答は考慮されません。これにより、連結移動軸が過負荷になり、連結に必要なダイナミック応答が不足する場合があります。

- シンクロナイズドアクションでの起動

起動がシンクロナイズドアクションで実行された場合は、マスタ軸の移動中に、連結移動軸のダイナミック応答は考慮されません。これにより、連結移動軸が過負荷になり、連結に必要なダイナミック応答が不足する場合があります。

**注意****軸の過負荷**

連結軸グループが有効な場合は、以下のようになります。

- このため、シンクロナイズドアクションの場合
- パートプログラムが、連結移動軸のチャンネルにプログラム軸ではないマスタ軸を含む場合は、

マスタ軸上の移動により連結移動軸の過負荷が発生しないよう、ユーザー/工作機械メーカーが適切な処置をおこなってください。

連結状態

軸の連結状態は、パートプログラムで次のシステム変数を使用して確認できます。

\$AA_COUP_ACT[<軸>]

規格値	意味
0	連結は無効
8	連結移動が有効

モジュロ回転軸用の連結移動軸の残移動距離の表示

マスタ軸と連結移動軸がモジュロ回転軸の場合、マスタ軸での $n * 360^\circ$ ($n=1, 2, 3 \dots$) からの移動動作は、連結をオフにするまで連結移動軸の残移動距離表示で加算されます。

例: TRAILON でマスタ軸 B とスレーブ軸 C を定義するプログラム区間

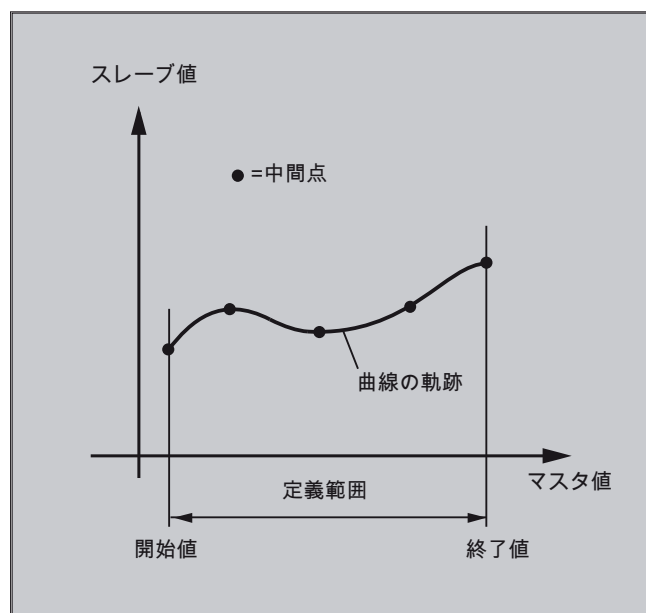
プログラムコード	コメント
TRAILON (C,B,1)	; 連結を起動します。
G0 B0	; 開始位置
	; ブロック始点の残移動距離表示:
G91 B360	; B=360, C=360
G91 B720	; B=720, C=1080
G91 B360	; B=360, C=1440

4.16.2 カーブテーブル(CTAB)

4.16.2.1 カーブテーブルの定義(CTABDEF、CATBEND)

カーブテーブルとは、パートプログラムまたは、先頭の CTABDEF と末尾の CTABEND で囲まれたパートプログラム区間です。

このパートプログラム区間内で、移動命令で固有のスレーブ軸位置を、マスタ軸の個々の位置に割り当てます。これらのスレーブ軸位置は、5 次までの多項式で曲線定義を計算するときの中間点として使用します。



4.16 軸連結

必要条件

マシンデータは、カーブテーブルの定義に十分なメモリ空間が確保されるよう設定してください(→工作機械メーカー)。

構文

```
CTABDEF (<スレーブ軸>,<マスタ軸>,<n>,<周期性>[,<メモリロケーション>])
...
CTABEND
```

意味

CTABDEF ():	カーブテーブル定義の開始	
CTABEND:	カーブテーブル定義の終了	
<スレーブ軸>:	カーブテーブルを使用して移動が計算される軸	
<マスタ軸>:	スレーブ軸移動の計算用のマスタ値を提供する軸	
<n>:	カーブテーブルの番号(ID) カーブテーブルの番号は一義的であり、メモリロケーションには影響されません。スタティックとダイナミック NC メモリの両方に、同じ番号のカーブテーブルは作成できません。	
<周期性>:	カーブテーブルの周期性	
	0	カーブテーブルは周期的ではありません(カーブテーブルが処理されるのは、回転軸の場合でも 1 回のみです)
	1	カーブテーブルはマスタ軸に関して周期的です
	2	カーブテーブルはマスタ軸とスレーブ軸に関して周期的です
<メモリロケーション>:	メモリロケーションの指定(任意選択)	
	"SRAM"	カーブテーブルをスタティック NC メモリに作成します。
	"DRAM"	カーブテーブルをダイナミック NC メモリに作成します。
	注: このパラメータに値をプログラム指令しない場合は、 MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE で設定した既定のメモリロケーションを使用します。	

注記**上書き**

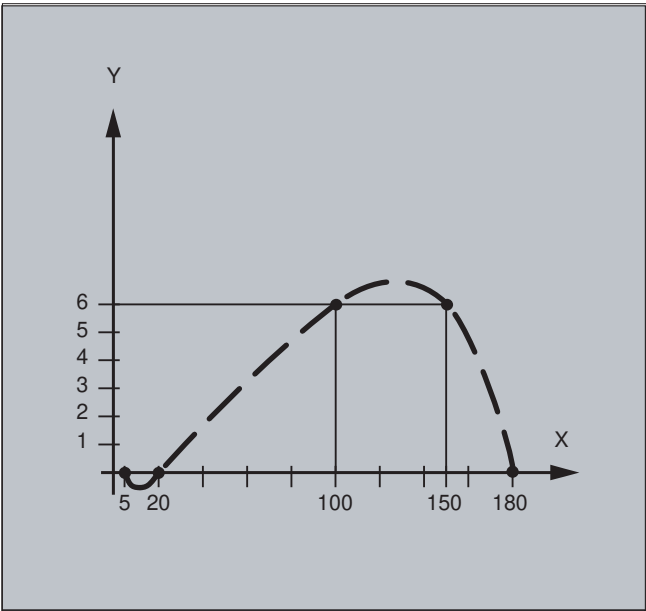
カーブテーブルは、その番号(<n>)が別のカーブテーブル定義で使用されると、直ちに上書きされます(例外:カーブテーブルが軸連結で有効な場合、または CTABLOCK でロックされている場合)。カーブテーブルが上書きされるときに、警告は出力されません。

例**例 1:カーブテーブル定義のプログラム区間**

プログラム区間をそのまま使用して、カーブテーブルを定義します。先読み停止の STOPRE 命令はそのまま変更されずに残すことができます。その後、プログラム区間がカーブテーブル定義に使用されなくなり、CTABDEF と CTABEND が削除されると、再びすぐに有効になります。

プログラムコード	コメント
...	
CTABDEF (Y,X,1,1)	; カーブテーブルの定義です。
...	
IF NOT (\$P_CTABDEF)	
Stopre	
ENDIF	
...	
CTABEND	

例 2 :周期的でないカーブテーブルの定義



プログラムコード	コメント
N100 CTABDEF (Y,X,3,0)	; 周期的でないカーブテーブル (番号 3) の定義の開始。
N110 X0 Y0	; 1 番目の移動命令。開始値と 1 番目の中間点を定義します。 マスタ値:0、スレーブ値:0
N120 X20 Y0	; 2 番目の補間点 マスタ値:0...20、スレーブ値:開始値...0
N130 X100 Y6	; 3 番目の補間点 マスタ値:20...100、スレーブ値:0...6
N140 X150 Y6	; 4 番目の補間点 マスタ値:100...150、スレーブ値:6...6
N150 X180 Y0	; 5 番目の補間点 マスタ値:150...180、スレーブ値:6...0
N200 CTABEND	; 定義の終了カーブテーブルは、5 次以下の多項式としての内部形式で生成されます。指定した中間点による曲線定義の計算は、モーダルで選択された補間タイプ (円弧補間、直線補間、スプライン補間) によって異なります。定義を開始する前のパートプログラムの状態に復帰します。

例 3 :周期的なカーブテーブルの定義

番号 2 の周期的なカーブテーブル、マスタ値の範囲 0～360、0 から 45 に移動して 0 に戻るスレーブ軸で定義します。

プログラムコード	コメント
N10 DEF REAL DEPPPOS	
N20 DEF REAL GRADIENT	
N30 CTABDEF (Y,X,2,1)	; 定義の開始

プログラムコード	コメント
N40 G1 X=0 Y=0	
N50 POLY	
N60 PO[X]=(45.0)	
N70 PO[X]=(90.0) PO[Y]=(45.0,135.0,-90)	
N80 PO[X]=(270.0)	
N90 PO[X]=(315.0) PO[Y]=(0.0,-135.0,90)	
N100 PO[X]=(360.0)	
N110 CTABEND	; 定義の終了
; Y と X の連結による曲線のテスト:	
N120 G1 F1000 X0	
N130 LEADON(Y,X,2)	
N140 X360	
N150 X0	
N160 LEADOF(Y,X)	
N170 DEPPPOS=CTAB(75.0,2,GRADIENT)	; カーブテーブル機能でマスタ値 75.0 を読み取ります。
N180 G0 X75 Y=DEPPPOS	; マスタ軸とスレーブ軸を位置決めします。
; 連結の起動後は、スレーブ軸の同期制御は不要です。	
N190 LEADON(Y,X,2)	
N200 G1 X110 F1000	
N210 LEADOF(Y,X)	
N220 M30	

詳細情報

カーブテーブルの開始値と終了値

カーブテーブルの定義範囲の始まりを示す開始値は、カーブテーブル定義内で指定された、最初の当該の軸位置(最初の移動命令)です。カーブテーブルの定義範囲の終了値は、最後の移動指令に特定されます。

使用言語

カーブテーブル定義内では、すべての NC 言語を使用できます。

注記

以下の入力は、カーブテーブル定義で使用できません。

- 先読み停止
- マスタ軸移動の解除(座標変換を変更した場合など)
- スレーブ軸のみの移動命令
- マスタ軸の反転(つまり、マスタ軸の位置は常に一定であることが必要です)
- さまざまなプログラムレベルでの CTABDEF 命令と CTABEND 命令

4.16 軸連結

モーダル命令の有効性

カーブテーブル定義内で作成したモーダル命令はすべて、カーブテーブル定義が完了すると、無効になります。したがって、カーブテーブル定義されたパートプログラムの前後では同じ状態になります。

R パラメータへの割り当て

カーブテーブル定義での R パラメータへの割り当ては、CTABEND 後にリセットされます。

例:

プログラムコード	コメント
...	
R10=5 R11=20	;R10=5
...	
CTABDEF	
G1 X=10 Y=20 F1000	
R10=R11+5	;R10=25
X=R10	
CTABEND	
...	;R10=5

ASPLINE、BSPLINE、CSPLINE の起動

ASPLINE、BSPLINE、または CSPLINE をカーブテーブル定義 CTABDEF～CTABEND 内で有効にした場合は、このスプラインの適用前に 1 個以上の始点をプログラム指令してください。CTABDEF の直後には適用しないでください。適用した場合は、カーブテーブル定義の前の現在の軸位置によって、スプラインが異なります。

例:

プログラムコード
...
CTABDEF (Y,X,1,0)
X0 Y0
ASPLINE
X=5 Y=10
X10 Y40
...
CTABEND

カーブテーブルの反復使用

カーブテーブルをスタティック NC メモリ (SRAM) に保存した場合は、カーブテーブルを使用して計算したマスタ軸とスレーブ軸の関数関係が、パートプログラムの終了、および POWER OFF の後も、選択したテーブル番号で保持されます。

ダイナミックメモリ (DRAM) で作成されたカーブテーブルは POWER ON 時に削除されるため、再作成が必要となる場合があります。

一度作成したカーブテーブルは、マスタ軸とスレーブ軸のあらゆる組み合わせに適用でき、作成に使用した軸には影響されません。

カーブテーブルの上書き

カーブテーブルは、その番号が別のカーブテーブル定義で使用されると、直ちに上書きされます。

例外:カーブテーブルが軸連結で有効な場合、または CTABLOCK でロックされている場合は例外です。

注記

カーブテーブルが上書きされるときに、警告は出力されません。

カーブテーブル定義の有効性

\$P_CTABDEF システム変数は、いつでもパートプログラムで使用して、カーブテーブル定義が有効かどうかをチェックできます。

カーブテーブル定義の無効化

カーブテーブル定義に関連する命令を解除した後は、そのパートプログラム区間を、実際のパートプログラムとして再度使用できます。

「外部メモリからの実行」によるカーブテーブルのロード

カーブテーブルを外部メモリから実行する場合は、MD18360 \$MN_MM_EXT_PROG_BUFFER_SIZE での再ロードバッファ (DRAM) サイズを選択して、カーブテーブル定義全体を再ロードバッファに同時に格納できるようにしてください。これをおこなわない場合は、アラームが発生して、パートプログラムの処理がキャンセルされます。

スレーブ軸のスキップ

マシンデータ

MD20900 \$MC_CTAB_ENABLE_NO_LEADMOTION

の設定に応じて、マスタ軸で移動が生じない場合のスレーブ軸のスキップが可能となります。

4.16.2.2 カーブテーブルの有無の確認 (CTABEXISTS)

CTABEXISTS 命令を使用すると、特定の番号のカーブテーブルが NC メモリにあるかどうかを確認できます。

4.16 軸連結

構文

CTABEXISTS (<n>)

意味

CTABEXISTS:	スタティックまたはダイナミック NC メモリのカーブテーブル番号<n>の有無を確認します。	
	0	カーブテーブルがありません
	1	カーブテーブルがあります
<n>:	カーブテーブルの番号(ID)	

4.16.2.3 カーブテーブルの削除(CTABDEL)

CTABDEL を使用すると、カーブテーブルを削除できます。

注記

軸連結で有効なカーブテーブルは削除できません。

構文

CTABDEL (<n>)
CTABDEL (<n>, <m>)
CTABDEL (<n>, <m>, <メモリロケーション>)
CTABDEL ()
CTABDEL (, , <メモリロケーション>)

意味

CTABDEL:	カーブテーブルを削除する命令
<n>:	削除するカーブテーブルの番号(ID) カーブテーブルの範囲 CTABDEL (<n>, <m>) を削除する場合、<n>は、その範囲内の最初のカーブテーブルの番号を指定します。
<m>:	カーブテーブルの範囲 CTABDEL (<n>, <m>) を削除する場合、<m>は、その範囲内の最後のカーブテーブルの番号を指定します。 <m>には、<n>より大きい値を指定してください。

<メモリロケーション>:	メモリロケーションの指定(任意選択)	
	メモリロケーションを 指定せず に削除すると、指定したカーブテーブルをスタティックとダイナミック NC メモリで削除します。	
	メモリロケーションを 指定して 削除すると、指定したカーブテーブルのうち、指定したメモリロケーションのカーブテーブルのみを削除します。残りのカーブテーブルは保持されます。	
	"SRAM"	スタティック NC メモリの削除
	"DRAM"	ダイナミック NC メモリの削除

削除するカーブテーブルをプログラム指令せずに CTABDEL をプログラム指令した場合は、**すべての**カーブテーブル、または指定したメモリのすべてのカーブテーブルを、次のように削除します。

CTABDEL () :	スタティックとダイナミック NC メモリのすべてのカーブテーブルを削除します
CTABDEL (, , "SRAM") :	スタティック NC メモリのすべてのカーブテーブルを削除します
CTABDEL (, , "DRAM") :	ダイナミック NC メモリのすべてのカーブテーブルを削除します

注記

CTABDEL (<n> , <m>) または CTABDEL () で複数のカーブテーブルを削除するときに、それらのカーブテーブルの 1 つ以上が連結で有効な場合は、削除命令が実行されません。つまり、指定したカーブテーブルは、**すべて**削除されません。

4.16.2.4 カーブテーブルの削除、および上書きを防止するロック(CTABLOCK、CTABUNLOCK)

ロックを設定すると、不用意な削除と上書きからカーブテーブルを保護できます。ロックは、設定後はいつでも解除できます。

構文

ロック:
CTABLOCK (<n>)
CTABLOCK (<n> , <m>)
CTABLOCK (<n> , <m> , <メモリロケーション>)
CTABLOCK ()
CTABLOCK (, , <メモリロケーション>)

4.16 軸連結

ロック解除:

CTABUNLOCK (<n>)

CTABUNLOCK (<n>,<m>)

CTABUNLOCK (<n>,<m>,<メモリロケーション>)

CTABUNLOCK ()

CTABUNLOCK (,,<メモリロケーション>)

意味

CTABLOCK:	削除/上書きを防止するロックを 設定 する命令	
CTABUNLOCK:	削除/上書きを防止するロックを 解除 する命令 CTABUNLOCK は、CTABLOCK でロックしたカーブテーブルのロックを解除します。動作中の連結で使用されるカーブテーブルのロックは解除されず、削除できません。CTABLOCK によるロックは、連結が解除されたタイミングで動作中の連結に適用されていたロックが解除されると同時に、解除されます。これによって、このカーブテーブルは削除できます。CTABUNLOCK を再度呼び出す必要はありません。	
<n>:	ロック/ロック解除するカーブテーブルの番号(ID) カーブテーブルの範囲 CTABLOCK (<n>,<m>)/ CTABUNLOCK (<n>,<m>) をロック/ロック解除する場合、<n>を使用してその範囲内の最初のカーブテーブルの番号を指定します。	
<m>:	カーブテーブルの範囲 CTABLOCK (<n>,<m>)/ CTABUNLOCK (<n>,<m>) をロック/ロック解除する場合、<m>を使用してその範囲内の最後のカーブテーブルの番号を指定します。 <m>には、<n>より大きい値を指定してください。	
<メモリロケーション>:	メモリロケーションの指定(任意選択) メモリロケーションを 指定せずに ロック/ロック解除すると、指定したカーブテーブルをスタティックとダイナミック NC メモリでロック/ロック解除します。 メモリロケーションを 指定して ロック/ロック解除すると、指定したカーブテーブルのうち、指定したメモリロケーションのカーブテーブルのみをロック/ロック解除します。残りのカーブテーブルはロック/ロック解除されません。	
	"SRAM"	スタティック NC メモリでのロック/ロック解除
	"DRAM"	ダイナミック NC メモリでのロック/ロック解除

ロック/ロック解除するカーブテーブルをプログラム指令せずに CTABLOCK/CTABUNLOCK をプログラム指令した場合は、**すべての**カーブテーブル、または指定したメモリのすべてのカーブテーブルを、次のようにロック/ロック解除します。

CTABLOCK():	スタティックとダイナミック NC メモリのすべてのカーブテーブルをロックします
CTABLOCK(, , "SRAM"):	スタティック NC メモリのすべてのカーブテーブルをロックします
CTABLOCK(, , "DRAM"):	ダイナミック NC メモリのすべてのカーブテーブルをロックします
CTABUNLOCK():	スタティックとダイナミック NC メモリのすべてのカーブテーブルをロック解除します
CTABUNLOCK(, , "SRAM"):	スタティック NC メモリのすべてのカーブテーブルをロック解除します
CTABUNLOCK(, , "DRAM"):	ダイナミック NC メモリのすべてのカーブテーブルをロック解除します

4.16.2.5 カーブテーブル: カーブテーブル機能の特定(CTABID、CTABISLOCK、CTABMEMTYP、CTABPERIOD)

これらの命令を使用すると、カーブテーブルの重要な機能(テーブル番号、ロック状態、メモリロケーション、周期性)を調べることができます。

構文

```
CTABID(<p>)
CTABID(<p>,<メモリロケーション>)
CTABISLOCK(<n>)
CTABMEMTYP(<n>)
TABPERIOD(<n>)
```

意味

CTABID:	指定したメモリの<p>番目のカーブテーブルとして入力したテーブル番号を返します。 例: CTABID(1,"SRAM")は、スタティック NC メモリの 1 番目のカーブテーブルの番号を返します。この場合、1 番目のカーブテーブルとは、最も大きいテーブル番号のカーブテーブルです。 注: 連続する CTABID の呼び出しの間に、CTABDEL によるカーブテーブルの削除などの理由で、メモリのカーブテーブルの順序が変わった場合は、CTABID(<p>,...)が、同じ<p>番目の別のカーブテーブルを返す場合があります。	
CTABISLOCK:	番号<n>のカーブテーブルの ロック状態 を、次のように返します。	
	0	カーブテーブルはロックされていません
	1	CTABLOCK によりカーブテーブルがロックされています
	2	動作中の連結によりカーブテーブルがロックされています
	3	CTABLOCK、および動作中の連結によりカーブテーブルがロックされています
	-1	カーブテーブルがありません
CTABMEMTYP:	番号<n>のカーブテーブルの メモリロケーション を、次のように返します。	
	0	スタティック NC メモリのカーブテーブル
	1	ダイナミック NC メモリのカーブテーブル
	-1	カーブテーブルがありません
CTABPERIOD:	番号<n>のカーブテーブルの 周期性 を、次のように返します。	
	0	カーブテーブルは周期的ではありません
	1	カーブテーブルはマスタ軸で周期的です
	2	カーブテーブルはマスタ軸とスレーブ軸で周期的です
	-1	カーブテーブルがありません
<p>:	メモリの登録番号	
<n>:	カーブテーブルの番号(ID)	

<メモリロケーション>:	メモリロケーションの指定(任意選択)	
	"SRAM"	スタティック NC メモリ
	"DRAM"	ダイナミック NC メモリ
	注: このパラメータに値をプログラム指令しない場合は、 MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE で設定した既定の メモリロケーションを使用します。	

4.16.2.6 カーブテーブル値の読み取り(CTABTSV、CTABTEV、CTABTSP、CTABTEP、CTABSSV、CTABSEV、CTAB、CTABINV、CTABTMIN、CTABTMAX)

パートプログラムで、次のカーブテーブル値を読み取ることができます。

- カーブテーブルの開始と終了のスレーブ軸値とマスタ軸値
- カーブセグメントの開始と終了のスレーブ軸値
- マスタ軸値に対するスレーブ軸値
- スレーブ軸値に対するマスタ軸値
- 以下のスレーブ軸の最小値と最大値
 - カーブテーブルの定義範囲全体
または
 - 定義したカーブテーブル区間

構文

```

CTABTSV(<n>,<傾き>[,<スレーブ軸>])
CTABTEV(<n>,<傾き>[,<スレーブ軸>])
CTABTSP(<n>,<傾き>[,<マスタ軸>])
CTABTEP(<n>,<傾き>[,<マスタ軸>])
CTABSSV(<マスタ値>,<n>,<傾き>[,<スレーブ軸>])
CTABSEV(<マスタ値>,<n>,<傾き>[,<スレーブ軸>])
CTABSEV(<マスタ値>,<n>,<傾き>[,<スレーブ軸>])
CTABINV(<スレーブ値>,<近似値>,<n>,<傾き>[,<スレーブ軸>,<マスタ軸>])
CTABTMIN(<n>[,<スレーブ軸>])
CTABTMAX(<n>[,<スレーブ軸>])
CTABTMIN(<n>,<a>,<b>[,<スレーブ軸>,<マスタ軸>])
CTABTMAX(<n>,<a>,<b>[,<スレーブ軸>,<マスタ軸>])

```

4.16 軸連結

意味

CTABTSV:	番号<n>のカーブテーブルの 開始位置 でスレーブ軸値を読み取ります
CTABTEV:	番号<n>のカーブテーブルの 終了位置 でスレーブ軸値を読み取ります
CTABTSP:	番号<n>のカーブテーブルの 開始位置 でマスタ軸値を読み取ります
CTABTEP:	番号<n>のカーブテーブルの 終了位置 でマスタ軸値を読み取ります
CTABSSV:	指定したマスタ軸値(マスタ値>)のカーブセグメントの 開始位置 で、スレーブ軸値を読み取ります
CTABSEV:	指定したマスタ軸値(マスタ値>)のカーブセグメントの 終了位置 で、スレーブ軸値を読み取ります
CTAB:	指定したマスタ軸値(マスタ値>)に対するスレーブ軸値を読み取ります
CTABINV:	指定したスレーブ軸値(スレーブ値>)に対するマスタ軸値を読み取ります
CTABTMIN:	以下のスレーブ軸の 最小値 を定義します - カーブテーブルの定義範囲全体 または <a> ... までの、定義した区間
CTABTMAX:	以下のスレーブ軸の 最大値 を定義します - カーブテーブルの定義範囲全体 または <a> ... までの、定義した区間
<n>:	カーブテーブルの番号(ID)
<傾き>:	傾き>パラメータは、計算した位置でのカーブテーブル機能の 傾斜 を返します。
<スレーブ軸>:	移動がカーブテーブルを使用して計算される軸(任意選択)
<マスタ軸>:	スレーブ軸移動の計算のマスタ値を提供する軸(任意選択)
<スレーブ値>:	CTABINV の関連するマスタ軸値を読み取るためのスレーブ軸値
<マスタ値>:	次の用途のマスタ軸値です。 - CTAB による、関連するスレーブ軸値の読み取り または - CTABSSV/CTABSEV によるカーブセグメントの選択

<近似値>:	CTABINV によるマスタ軸値のスレーブ軸値への割り当ては、常に一義的である必要はありません。したがって、CTABINV にはパラメータとして、予想されるマスタ軸値の近似値が必要となります。
<a>:	CTABTMIN/CTABTMAX によるマスタ値区間の下限
:	CTABTMIN/CTABTMAX によるマスタ値区間の上限
	注: a>から b>までのマスタ値区間は常に、カーブテーブルの定義範囲内にしてください。

例

例 1:

カーブテーブルの定義範囲全体で、スレーブ軸の最小値と最大値により、カーブテーブルの開始位置と終了位置のスレーブ軸値とマスタ軸値を、定義します。

プログラムコード	コメント
N10 DEF REAL STARTPOS	
N20 DEF REAL ENDPOS	
N30 DEF REAL STARTPARA	
N40 DEF REAL ENDPARA	
N50 DEF REAL MINVAL	
N60 DEF REAL MAXVAL	
N70 DEF REAL GRADIENT	
...	
N100 CTABDEF (Y,X,1,0)	; カーブテーブル定義の開始。
N110 X0 Y10	; 1 番目のカーブテーブルセグメントの開始位置。
N120 X30 Y40	; 1 番目のカーブテーブルセグメントの終了位置= 2 番目のカーブテーブルセグメントの開始位置
N130 X60 Y5	; 2 番目のカーブテーブルセグメントの終了位置= ...
N140 X70 Y30	
N150 X80 Y20	
N160 CTABEND	; カーブテーブル定義の終了。
...	
N200 STARTPOS=CTABTSV (1,GRADIENT)	; カーブテーブル開始位置のスレーブ軸値= 10
N210 ENDPOS=CTABTEV (1,GRADIENT)	; カーブテーブル終了位置のスレーブ軸値= 20
N220 STARTPARA=CTABTSP (1,GRADIENT)	; カーブテーブル開始位置のマスタ軸値= 0
N230 ENDPARA=CTABTEP (1,GRADIENT)	; カーブテーブル終了位置のマスタ軸値= 80
N240 MINVAL=CTABTMIN (1)	; Y=5 のスレーブ軸の最小値
N250 MAXVAL=CTABTMAX (1)	; Y=40 のスレーブ軸の最大値

4.16 軸連結

例 2:

マスタ軸値 X=30 に関連するカーブセグメントの開始位置と終了位置のスレーブ軸値を特定します。

プログラムコード	コメント
N10 DEF REAL STARTPOS	
N20 DEF REAL ENDPOS	
N30 DEF REAL GRADIENT	
...	
N100 CTABDEF(Y,X,1,0)	; カーブテーブル定義の開始。
N110 X0 Y0	; 1 番目のカーブテーブルセグメントの開始位置。
N120 X20 Y10	; 1 番目のカーブテーブルセグメントの終了位置= 2 番目のカーブテーブルセグメントの開始位置
N130 X40 Y40	; 2 番目のカーブテーブルセグメントの終了位置= ...
N140 X60 Y10	
N150 X80 Y0	
N160 CTABEND	; カーブテーブル定義の終了。
...	
N200 STARTPOS=CTABSSV(30.0,1,GRADIENT)	; 2 番目のセグメントの開始位置 Y = 10
N210 ENDPOS=CTABSEV(30.0,1,GRADIENT)	; 2 番目のセグメントの終了位置 Y = 40

詳細情報

シンクロナイズドアクションでの使用

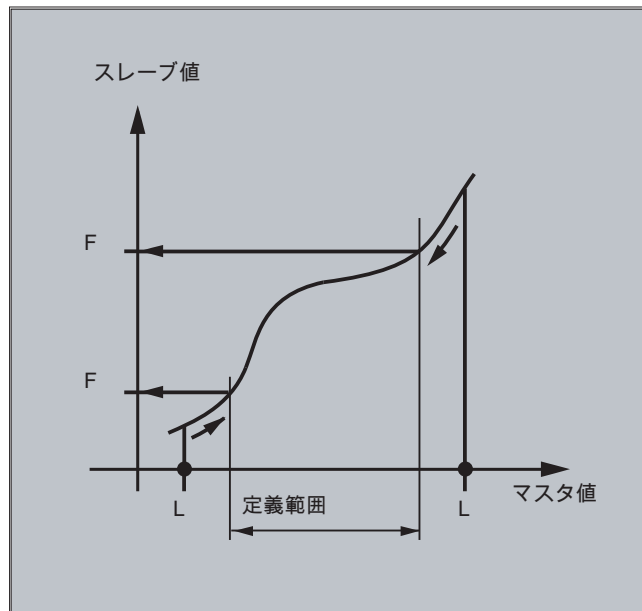
カーブテーブル値を読み取るための命令はすべて、シンクロナイズドアクションでも使用できます(章「シンクロナイズドアクション」も参照してください)。

CTABINV、CTABTMIN、および CTABTMAX の各命令の使用時には、以下を確認してください。

- 実行時には十分な NC 処理能力が使用できる。
または
- カーブテーブルのセグメントの数を呼び出し前に確認しており、当該カーブテーブルが、必要に応じて分割可能である。

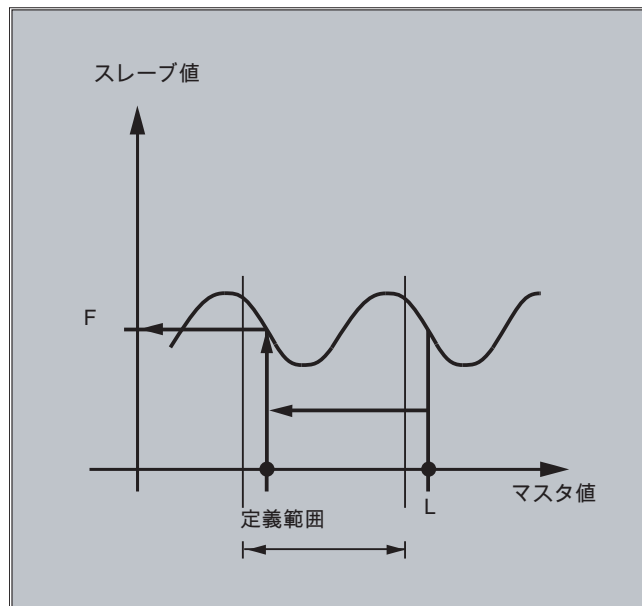
周期的でないカーブテーブルによる CTAB

指定した<マスタ値>が定義範囲外にある場合は、上限または下限が以下の値として出力されます。



周期的なカーブテーブルによる CTAB

指定した<マスタ値>が定義範囲外にある場合は、マスタ値は定義範囲で使用されたモジュールとなり、対応するスレーブ値が出力されます。



CTABINV の近似値

したがって、CTABINV 命令には、予想されるマスタ値の近似値が必要となります。CTABINV は、近似値に最も近いマスタ値を返します。近似値には、以前の補間クロックサイクルのマスタ値などがあります。

4.16 軸連結

カーブテーブル機能の傾斜

出力された傾斜(<傾き>)により、対応する位置のマスタ軸またはスレーブ軸の速度の計算が可能となります。

マスタ軸またはスレーブ軸の指定

マスタ軸とスレーブ軸を異なる長さの単位で設定している場合は、マスタ軸とスレーブ軸のいずれか、または両方の選択指定が必要です。

CTABSSV、CTABSEV

CTABSSV と CTABSEV 命令は、以下の場合は、プログラム指令されたセグメントを確認するためには**不適切**です。

- 円弧またはインボリュート曲線をプログラム指令している。
- CHF/ RND による面取りまたは丸み付けが有効である。
- G643 によるスムージングが有効である。
- NC ブロック圧縮(COMP...)が有効である。

4.16.2.7 カーブテーブル: メモリ使用の確認(CTABNO、CTABNOMEM、CTABFNO、CTABSEGID、CTABSEG、CTABFSEG、CTABMSEG、CTABPOLID、CTABPOL、CTABFPOL、CTABMPOL)

プログラマは、これらの命令を使用して、カーブテーブル、そのセグメント、および多項式のメモリ使用に関する最新情報を得ることができます。

構文

```
CTABNO
CTABNOMEM (<メモリ ロケーション>)
CTABFNO (<メモリ ロケーション>)
CTABSEGID (<n>,<メモリ ロケーション>)
CTABSEG (<メモリ ロケーション>,<セグメントタイプ>)
CTABFSEG (<メモリ ロケーション>,<セグメントタイプ>)
CTABMSEG (<メモリ ロケーション>,<セグメントタイプ>)
CTABPOLID (<n>)
CTABPOL (<メモリ ロケーション>)
CTABFPOL (<メモリ ロケーション>)
CTABMPOL (<メモリ ロケーション>)
```


意味

CTABNO:	定義した(スタティック、およびダイナミック NC メモリの)カーブテーブルの合計数を特定します。	
CTABNOMEM:	指定した<メモリロケーション>の、定義したカーブテーブルの数を特定します。	
CTABFNO:	指定した<メモリロケーション>で引き続き有効なカーブテーブルの数を特定します。	
CTABSEGID:	番号<n>のカーブテーブルに使用する、指定した<セグメントタイプ>のカーブセグメントの数を特定します。	
CTABSEG:	指定した<メモリロケーション>で使用される、指定した<セグメントタイプ>のカーブセグメントの数を特定します。	
CTABFSEG:	指定した<メモリロケーション>で引き続き有効な、指定した<セグメントタイプ>のカーブセグメントの数を特定します。	
CTABMSEG:	指定した<メモリロケーション>での指定した<セグメントタイプ>のカーブセグメントの最大可能数を特定します。	
CTABPOLID:	番号<n>のカーブテーブルが使用するカーブ多項式の数を特定します	
CTABPOL:	指定した<メモリロケーション>で使用されるカーブ多項式の数を特定します	
CTABFPOL:	指定した<メモリロケーション>で引き続き有効なカーブ多項式の数を特定します	
CTABMPOL:	指定した<メモリロケーション>で最大のカーブ多項式数を特定します	
<n>:	カーブテーブルの番号(ID)	
<メモリロケーション>:	メモリロケーションの指定(任意選択)	
	"SRAM"	スタティック NC メモリ
	"DRAM"	ダイナミック NC メモリ
	注: このパラメータに値をプログラム指令しない場合は、 MD20905 \$MC_CTAB_DEFAULT_MEMORY_TYPE で設定した既定のメモリロケーションを使用します。	

4.16 軸連結

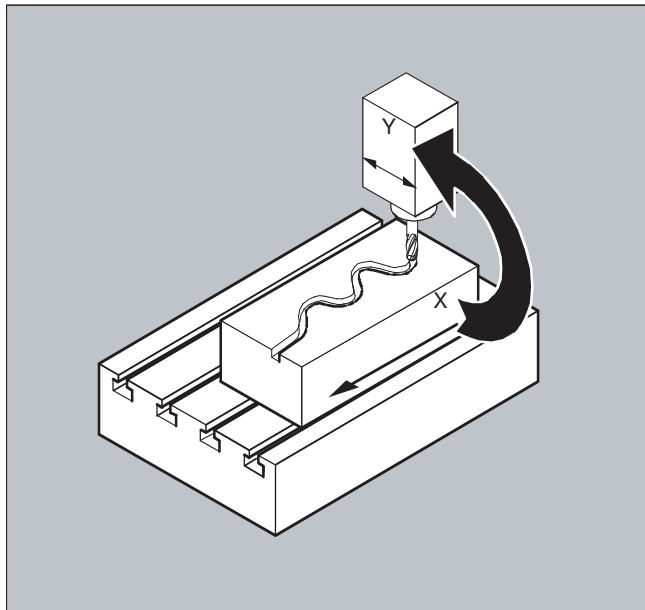
<セグメントタイプ >:	セグメントタイプの指定(任意選択)	
	"L"	直線セグメント
	"P"	多項式セグメント
	注: このパラメータに値をプログラム指令しない場合は、直線セグメントと多項式セグメントの合計が出力されます。	

4.16.3 軸間連動機能(LEADON、LEADOF)

注記

この機能は SINUMERIK 828D では使用できません。

軸間連動機能を使用すると、マスタ軸とスレーブ軸が同期して移動します。カーブテーブルを使用して、スレーブ軸の位置、または得られる多項式を、マスタ軸の位置に一義的に割り当てることができます。また、必要に応じて、そのシミュレーションも実行できます。



マスタ軸とは、カーブテーブルの入力値を与える軸です。**スレーブ軸**は、カーブテーブルを使用して計算した位置を使用する軸です。

現在値連結と指令値連結

スレーブ軸の位置計算の出力値として、以下をマスタ軸として使用できます。

- マスタ軸位置の現在値:現在値連結
- マスタ軸位置の指令値:指令値連結

軸間連動機能は常に、基本座標系で適用されます。

カーブテーブルの作成については、「カーブテーブル」の章を参照してください。

構文

LEADON (<スレーブ軸>,<マスタ軸>,<n>)
LEADOF (<スレーブ軸>,<マスタ軸>)

または、マスタ軸を指定せずに解除する場合は:
LEADOF (<スレーブ軸>)

軸間連動機能は、パートプログラムから、および動作中のシンクロナイズドアクションからのいずれでも起動/解除ができます。

意味

LEADON:	軸間連動機能を起動します
LEADOF:	軸間連動機能を解除します
<スレーブ軸>:	スレーブ軸
<マスタ軸>:	マスタ軸
<n>:	カーブテーブル番号
\$SA_LEAD_TYPE:	指令値連結と現在値連結の切り替え

軸間連動機能の解除、LEADOF

軸間連動機能を解除すると、スレーブ軸が再び、通常のコマンド軸となります。

軸間連動機能、および様々な運転状態、RESET

マシンデータの設定によっては、軸間連動機能が RESET で解除されます。

シンクロナイズドアクションによる軸間連動機能の例

プレス工場では、マスタ軸(駆動軸)、および搬送軸と補助軸から構成される搬送システムの軸間の通常の機械的連結に代わって、電子連結システムに置き換えられています。

ここでは、機械的搬送システムに代わって電子的搬送システムを使用する方法を示します。連結とその解除の処理は、**内部的なシンクロナイズドアクション**として実行されます。

4.16 軸連結

マスタ軸 LV (駆動軸)から、カーブテーブルで定義したスレーブ軸として搬送軸と補助軸を制御します。

スレーブ軸

- X 送りまたは長手方向軸
- YL 閉軸または径方向軸
- ZL 上昇軸
- U ロール送り、補助軸
- V ガイドヘッド、補助軸
- W 注油、補助軸

動作

実行される動作には、以下のシンクロナイズドアクションなどがあります。

- 連結の起動、LEADON (<スレーブ軸>,<マスタ軸>,<カーブテーブル番号>)
- 連結の解除、LEADOF (<スレーブ軸>,<マスタ軸>)
- 現在値の設定、PRESETON (<軸>,<値>)
- マークの設定、\$AC_MARKER[i]=<値>
- 連結タイプ:実/仮想マスタ値
- 軸位置へのアプローチ、POS [<軸>]=<値>

条件

高速デジタル入力部、リアルタイム変数\$AC_MARKER および位置比較演算をブール演算子 AND で連結し、条件として使用します。

注記

以下の例では、プログラムを読み取りやすくするためにのみ、改行、インデント、および太字タイプを使用しています。制御装置の場合、行番号の後はすべて 1 行であると見なされます。

コメント

プログラムコード	コメント
	; すべての内部的なシンクロナイズドアクションを定義します。
	; ****マーカのリセット
N2 \$AC_MARKER[0]=0 \$AC_MARKER[1]=0 \$AC_MARKER[2]=0 \$AC_MARKER[3]=0 \$AC_MARKER[4]=0 \$AC_MARKER[5]=0 \$AC_MARKER[6]=0 \$AC_MARKER[7]=0	
	; **** E1 0=>1 転送オン
N10 IDS=1 EVERY (\$A_IN[1]==1) AND (\$A_IN[16]==1) AND (\$AC_MARKER[0]==0) DO LEADON(X,LW,1) LEADON(YL,LW,2) LEADON(ZL,LW,3) \$AC_MARKER[0]=1	

プログラムコード	コメント
	; **** E1 0=>1 連結ローラ送りオン
N20 IDS=11 EVERY (\$A_IN[1]==1) AND (\$A_IN[5]==0) AND (\$AC_MARKER[5]==0)	
DO LEADON(U,LW,4) PRESETON(U,0) \$AC_MARKER[5]=1	
	; **** E1 0=>1 ガイドヘッド連結オン
N21 IDS=12 EVERY (\$A_IN[1]==1) AND (\$A_IN[5]==0) AND (\$AC_MARKER[6]==0)	
DO LEADON(V,LW,4) PRESETON(V,0) \$AC_MARKER[6]=1	
	; **** E1 0=>1 注油連結オン
N22 IDS=13 EVERY (\$A_IN[1]==1) AND (\$A_IN[5]==0) AND (\$AC_MARKER[7]==0)	
DO LEADON(W,LW,4) PRESETON(W,0) \$AC_MARKER[7]=1	
	; **** E2 0=>1 連結オフ
N30 IDS=3 EVERY (\$A_IN[2]==1)	
DO LEADOF(X,LW) LEADOF(YL,LW) LEADOF(ZL,LW) LEADOF(U,LW) LEADOF(V,LW) LEADOF(W,LW) \$AC_MARKER[0]=0	
\$AC_MARKER[1]=0 \$AC_MARKER[3]=0 \$AC_MARKER[4]=0 \$AC_MARKER[5]=0 \$AC_MARKER[6]=0 \$AC_MARKER[7]=0	
....	
N110 G04 F01	
N120 M30	

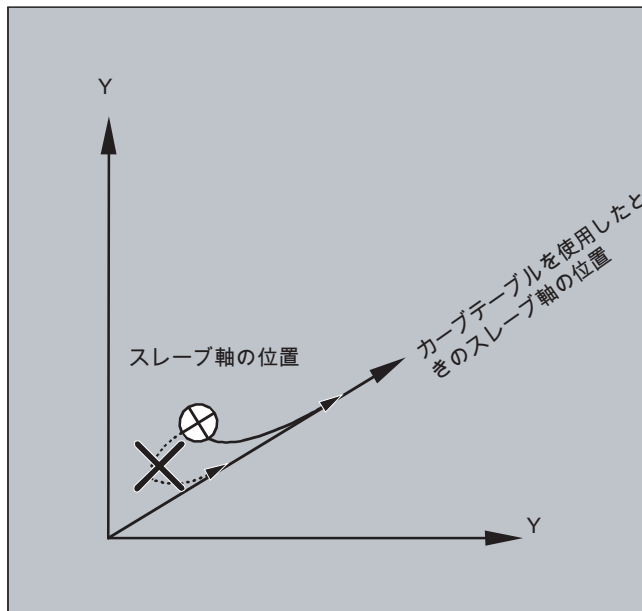
説明

軸間連動機能には、マスタ軸とスレーブ軸の同期制御が必要です。この同期制御は、軸間連動機能が有効なときに、スレーブ軸がカーブテーブルから計算されたカーブ定義の許容範囲内にある場合にのみ可能となります。

スレーブ軸位置の許容範囲は、マシンデータ MD 37200:COUPLE_POS_POL_COARSE A_LEAD_TYPE で定義します。

軸間連動機能が有効なときに、スレーブ軸がまだ正しい位置にない場合は、スレーブ軸に対して計算された位置指令値が実際のスレーブ軸位置とほぼ同じになると、直ちに同期制御の実行が自動的に開始されます。同期制御処理中に、スレーブ軸の指令速度(CTAB カーブテーブルを使用してメイン主軸から計算されます)で定義された方向にスレーブ軸を移動します。

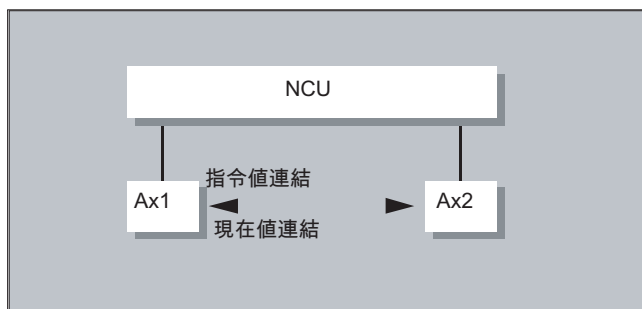
4.16 軸連結

**同期制御なし**

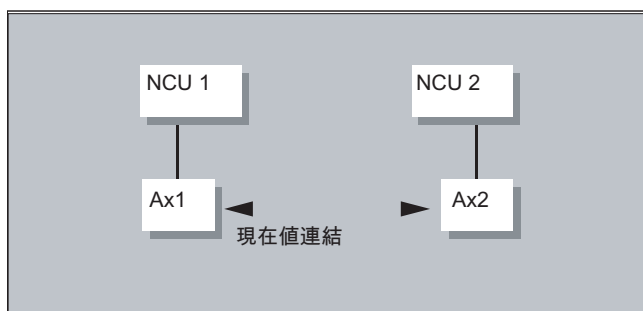
軸間連動機能が有効なときに、計算されたスレーブ軸位置が現在のスレーブ軸位置から離れていく場合は、同期制御を確立できなくなります。

現在値連結と指令値連結

マスタ軸とスレーブ軸の同期制御は、指令値連結の方が現在値連結より優れているため、初期設定は指令値連結となっています。



指令値連結は、マスタ軸とスレーブ軸が同じ NCU によって補間される場合にのみ可能です。外部のマスタ軸を使用する場合は、現在値でのみ、スレーブ軸をマスタ軸に連結できます。



切り替えは、セッティングデータ\$SA_LEAD_TYPE を使用してプログラム指令できます。

現在値連結と指令値連結間の切り替えは、必ずスレーブ軸の停止時におこなってください。

切り替え後の再原点同期は、軸が停止状態になったときのみ可能です。

用途例

機械が大きく振動しているときは、現在値には誤差を含みます。軸間連動機能をプレス搬送で使用する場合は、最も振動が大きい作業ステップでは、現在値連結から指令値連結への切り替えが必要となる場合があります。

指令値連結によるマスタ値シミュレーション

マシンデータを使用して、マスタ軸の補間器とサーボの接続を切断できます。この方法で、実際にはマスタ軸を移動せずに、指令値連結の指令値を生成できます。

指定値連結から生成したマスタ値は、以下の変数から読み取って、シンクロナイズドアクションなどで使用できます。

- \$AA_LEAD_P	マスタ値の位置
- \$AA_LEAD_V	マスタ値の速度

マスタ値の作成

マスタ値を他の自動プログラム指令方式で生成できるオプションもあります。このようにして生成したマスタ値は、以下の変数から読み取ったり、これらの変数に書き込んだりできます。

- \$AA_LEAD_SP	マスタ値の位置
- \$AA_LEAD_SV	マスタ値の速度

これらの変数を使用する前に、セッティングデータ\$SA_LEAD_TYPE = 2 を設定してください。

連結の状態

NC プログラムで次のシステム変数を使用すると、連結の状態を確認できます。

4.16 軸連結

\$AA_COUP_ACT[[軸]]
0:動作中の連結はありません
16:軸間連動機能が動作中です

シンクロナイズドアクションの状態管理

事象の切り替えと連結は、以下のリアルタイム変数で管理します。

\$AC_MARKER[i] = n
以下を使用して管理します。
i フラグ番号
n 状態値

4.16.4 電子ギヤ(EG)

「電子ギヤ」機能を使用すると、**スレーブ軸**の移動を、5 つまでの**マスタ軸**に応じて、移動ブロックに従って制御できます。各マスタ軸とスレーブ軸の関係は連結係数で定義します。

スレーブ軸移動部分は、個々のマスタ軸移動部分をそれぞれの連結係数で乗算し、それらを加算して算出します。EG 軸グループが有効になると、定義位置に対してスレーブ軸を同期できます。ギヤグループには、次の動作を実行できます。

- 定義
- 有効化
- 無効化
- 削除
-

任意選択で、スレーブ軸移動を以下から計算することもできます。

- マスタ軸の指令値、および
- マスタ軸の現在値

各マスタ軸とスレーブ軸の非直線の関係は、**カーブテーブル**を使用した拡張機能として実現することもできます(「軌跡移動動作」の章を参照してください)。電子ギヤは多段にできます。つまり、電子ギヤのスレーブ軸が、その他の電子ギヤのマスタ軸となる場合があります。

4.16.4.1 電子ギヤの定義(EGDEF)

EG 軸グループは、スレーブ軸および、当該の連結タイプで少なくとも 1 つ以上、最大 5 つのそれぞれのマスタ軸を指定して定義します。

必要条件

EG 軸グループの定義の必要条件は次のとおりです。

スレーブ軸の軸連結は定義できません(または、既存の軸連結は最初に EGDEL で解除してください)。

構文

EGDEF (スレーブ軸, マスタ軸 1, 連結タイプ 1, マスタ軸 2, 連結タイプ 2, ...)

意味

EGDEF:	電子ギヤの定義	
スレーブ軸:	マスタ軸に影響される軸	
マスタ軸 1 マスタ軸 5	スレーブ軸に影響する軸	
連結タイプ 1 連結タイプ 5	連結タイプ 連結タイプは、すべてのマスタ軸で同じである必要はありません。また、それぞれのマスタ軸毎に個々にプログラム指令してください。	
	値:	意味
	0	スレーブ軸は、対応するマスタ軸の 現在値 に影響されます。
	1	スレーブ軸は、対応するマスタ軸の 指令値 に影響されます。

注記

連結係数は、EG 軸グループを定義すると、ゼロに設定されます。

4.16 軸連結

注記

EGDEF は、先読み停止をおこないます。また、システムで、複数のマスタ軸が**カーブテーブル**によってスレーブ軸に影響を及ぼす場合には、EGDEF によるギヤの定義をそのまま使用してください。

例

プログラムコード	コメント
EGDEF (C,B,1,Z,1,Y,1)	; EG 軸グループの定義です。マスタ軸 B、Z、Y は指令値によって、スレーブ軸 C に影響を及ぼします。

4.16.4.2 電子ギヤの起動(EGON、EGONSYN、EGONSYNE)

EG 軸グループを起動する方法は 3 つあります。

構文

タイプ 1:

以下を使用して、選択した EG 軸グループを同期せずに起動します。

```
EGON (FA, "ブロック切り替えモード", LA1, Z1, N1, LA2, Z2, N2, ..., LA5, Z5, N5)
```

タイプ 2:

以下を使用して、選択した EG 軸グループを同期して起動します。

```
EGONSYN (FA, "ブロック切り替えモード", SynPosFA, [, LAi, SynPosLAI, Zi, Ni])
```

タイプ 3:

以下で指定したアプローチモードで、選択した EG 軸グループを同期して起動します。

```
EGONSYNE (FA, "ブロック切り替えモード", SynPosFA, アプローチモード  
[, LAi, SynPosLAI, Zi, Ni])
```

意味

タイプ 1:

FA	スレーブ軸	
ブロック切り替えモード:	以下のモードを使用できます。	
	"NOC"	ブロック切り替えは直ぐに実行されます
	"FINE"	ブロック切り替えは「精密同期」で実行されます
	"COARSE"	ブロック切り替え「汎用同期」で実行されます
	"IPOSTOP"	ブロック切り替えが、指令値に基づいた同期制御に対して実行されます
LA1, ... LA5	マスタ軸	
Z1, ... Z5	連結係数 i の分子	
N1, ... N5	連結係数 i の分母 連結係数 $i = \text{分子 } i / \text{分母 } i$	

起動行にプログラム指令できるのは、以前に EGDEF 命令で指定したマスタ軸のみです。少なくとも 1 つのマスタ軸をプログラム指令してください。

タイプ 2:

FA	スレーブ軸	
ブロック切り替えモード:	以下のモードを使用できます。	
	"NOC"	ブロック切り替えは直ぐに実行されます
	"FINE"	ブロック切り替えは「精密同期」で実行されます
	"COARSE"	ブロック切り替え「汎用同期」で実行されます
	"IPOSTOP"	ブロック切り替えが、指令値に基づいた同期制御に対して実行されます
[, LAi, SynPosLAi, Zi, Ni]	(角括弧は記述不要です) 以下を最小 1 個、最大 5 個、順に記述します。	
LA1, ... LA5	マスタ軸	
SynPosLAi	i 番目のマスタ軸の同期位置	

4.16 軸連結

Z1, ... Z5	連結係数 i の分子
N1, ... N5	連結係数 i の分母
	連結係数 i = 分子 i / 分母 i

起動行にプログラム指令できるのは、以前に EGDEF 命令で指定したマスタ軸のみです。スレーブ軸とマスタ軸に対してプログラム指令した「同期位置」(SynPosFA と SynPosLA)を使用して位置が定義され、この位置に対して軸グループが同期していると解釈されます。グループの起動時に電子ギヤが同期状態でない場合は、スレーブ軸が、定義されたその同期位置に移動します。

タイプ 3:

タイプ 2 のパラメータに対応するパラメータ、および以下のパラメータがあります。

アプローチモード:	以下のモードを使用できます。	
	"NTGT"	時間を優先して、次の歯間へアプローチします
	"NTGP"	距離を優先して、次の歯間へアプローチします
	"ACN"	負の方向にアブソリュートで回転軸を移動します
	"ACP"	正の方向にアブソリュートで回転軸を移動します
	"DCT"	プログラム指令の同期位置に対して時間を優先します
	"DCP"	プログラム指令した同期位置に対して距離を優先します

タイプ 3 は、モジュロマスタ軸に連結されたモジュロスレーブ軸にのみ有効です。時間を優先する場合は、スレーブ軸の速度制限が考慮されます。

詳細情報

起動タイプの説明**タイプ 1:**

グループが起動した瞬間のマスタ軸とスレーブ軸の位置が、「同期位置」として格納されます。「同期位置」はシステム変数\$AA_EG_SYNで読み取ることができます。

タイプ 2:

連結グループにモジュロ軸が含まれる場合は、その位置の値に対してモジュロ演算がおこなわれます。これにより、次の同期位置(次の歯間などへアプローチすることを*相対同期*と呼びます)に移動できます。同期位置へアプローチするのは、「スレーブ軸オーバーライド有効」インタフェース信号 DB(30 + 軸番号)、DBX 26 ビット 4 がスレーブ軸に対して設定さ

れた場合のみです。その代わりに、プログラムは EGONSYN ブロックで停止して、上記の信号が設定されるまで自己クリアアラーム 16771 が発生します。

タイプ 3:

歯間距離(°)は、「 $360 * Zi/Ni$ 」のように計算されます。スレーブ軸が呼び出し時に停止した場合は、距離優先の場合と、時間優先の場合は同じ応答を返します。

スレーブ軸がすでに移動中の場合は、NTGP は、スレーブ軸の現在の速度にかかわらず、次の歯間で同期をおこないます。スレーブ軸がすでに移動中の場合は、NTGT は、スレーブ軸の現在の速度に応じて、次の歯間で同期が異なります。必要に応じて、軸の減速もおこなわれます。

カーブテーブル

マスタ軸のいずれかにカーブテーブルを使用する場合は、次のようになります。

Ni	直線連結では、連結係数の分母を 0 に設定してください(直線連結では分母 0 は別の意味を持ちます)。分母がゼロの場合は、
Zi	がカーブテーブルの番号であることを指令します。指定した番号のカーブテーブルは常に、POWER ON 時に定義されます。
LAI	指定したマスタ軸は、連結係数によって連結用に指定したマスタ軸に対応します(直線連結)。

カーブテーブルの使用、および電子ギヤの多段と同期制御について詳しくは、以下を参照してください。

補助機能:

『機能マニュアル』、「軸と主軸」

Power On、RESET、操作モード変更、ブロックサーチでの電子ギヤの動作

- POWER ON 後には、連結は無効です。
- 動作中の連結の状態は、RESET にも操作モードの切り替えにも影響されません。
- ブロックサーチのときは、電子ギヤの切り替え、解除、および定義をおこなう命令は、実行も記録もされずにスキップされます。

電子ギヤのシステム変数

電子ギヤのシステム変数を使用して、パートプログラムで EG 軸グループの現在の状態を特定し、必要に応じて、それに対応できます。

電子ギヤのシステム変数は、次のように指定します。

\$AA_EG_ ...

または

4.16 軸連結

\$VA_EG_ ...

追加情報

『機能マニュアル、システム変数』

4.16.4.3 電子ギヤの起動(EGOFS、EGOFC)

動作中の EG 軸グループを解除する方法は 3 つあります。

プログラミング

タイプ 1:

構文	意味
EGOFS (スレーブ軸)	電子ギヤが解除されます。スレーブ軸が減速し、停止状態となります。この呼び出しは、先読み停止をおこないます。

タイプ 2:

構文	意味
EGOFS (従動軸、マスタ軸 1、 ..., マスタ軸 5)	この命令パラメータを設定すると、スレーブ軸移動への個々のマスタ軸の影響を 選択 して、解除できます。

少なくとも 1 つのマスタ軸を指定してください。指定したマスタ軸のスレーブ軸への影響は、選択して禁止されます。この呼び出しは、先読み停止をおこないます。呼び出しにまだ動作中のマスタ軸が含まれている場合、スレーブ軸はその影響で動作し続けます。すべてのマスタ軸の影響が、この方法により取り除かれた場合は、スレーブ軸が減速して、停止状態となります。

タイプ 3:

構文	意味
EGOFC (スレーブ主軸 1)	電子ギヤが解除されます。スレーブ主軸が、解除時に適用される速度で移動し続けます。この呼び出しは、先読み停止をおこないます。

注記

このタイプは、主軸にのみ使用できます。

4.16.4.4 電子ギヤの定義の解除(EGDEL)

EG 軸グループは、その定義が解除される前に解除してください。

プログラミング

構文	意味
EGDEL (スレーブ軸)	軸グループの連結定義を解除します。EGDEF を使用して、同時に起動する軸グループの最大数に達するまで、追加の軸グループを定義できます。この呼び出しは、先読み停止をおこないません。

4.16.4.5 毎回転送り速度(G95) / 電子ギヤ(FPR)

FPR 命令を使用すると、電子ギヤのスレーブ軸を、毎回転送り速度を特定する軸として指定できます。この命令に関しては、以下の点に注意してください。

- 送り速度は、電子ギヤのスレーブ軸の指令速度で特定されます。
- 指令速度は、マスタ主軸とモジュロ軸(軌跡軸でない)の速度、およびそれらに関連する連結係数から計算されます。
- 直線または非モジュロのマスタ軸の速度部分、およびスレーブ軸の重畳移動は考慮されません。

4.16.5 主軸同期

同期運転には、スレーブ主軸(FS)とマスタ主軸(LS)が関連し、これは**主軸同期ペア**と呼ばれます。連結が有効であるとき(同期運転)、定義されている機能の相互関係に従って、スレーブ主軸はマスタ主軸の移動をコピーします。

各機械の主軸同期ペアは、チャンネルマシンデータで固定設定として割り当てるか、または CNC パートプログラムで特定の用途のために定義することができます。各 NC チャンネルで同時に運転できる主軸同期ペアは 2 組までです。

パートプログラムからの連結動作は、以下のとおりです。

- 定義または変更
- 有効化
- 無効化
- 解除

4.16 軸連結

また、ソフトウェア条件に応じて、以下が適用されます。

- 同期条件が満たされるまで待機できます。
- ブロック切り替え方式を変更できます。
- 指令値連結または現在値連結のいずれかのタイプを選択するか、またはメイン主軸とスレーブ主軸の間の角度オフセットを指定します。
- 連結を起動すると、それ以前にプログラム指令したスレーブ軸で連結をおこないます。
- 検出された同期の不一致、または既知の同期の不一致は補正されます。

4.16.5.1 主軸同期: プログラミング(COUPDEF、COUPDEL、COUPON、COUPONC、COUPOF、COUPOFS、COUPRES、WAITC)

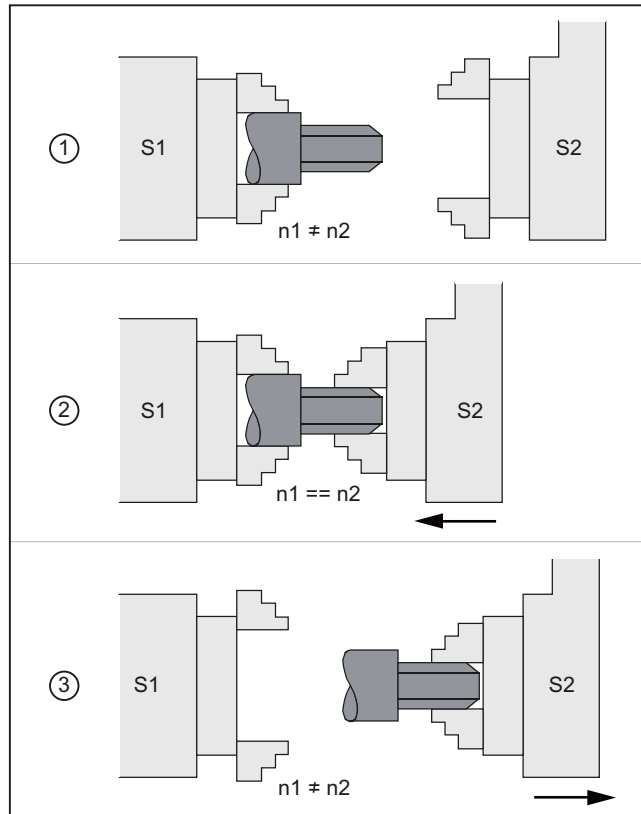
「主軸同期」では、スレーブ主軸(FS)とマスタ主軸 (LS)の速度同期移動をプログラム指令可能な伝達比でおこなうことができます。

この機能は次のモードをサポートしています。

- 速度同期制御($n_{FS} = n_{LS}$)
- 位置同期制御($\phi_{FS} = \phi_{LS}$)
- 角度オフセットによる位置同期制御($\phi_{FS} = \phi_{LS} + \Delta\phi$)

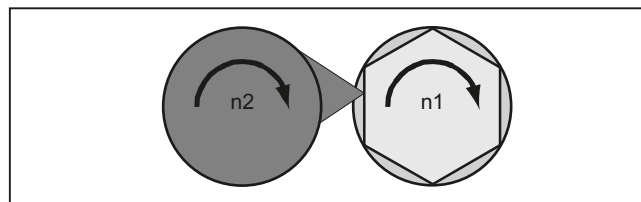
用途例:

- 背面加工などのための高速ワーク移動。伝達比:1:1



- ① 速度の同期化
- ② ワークの移動
- ③ 背面の加工

- 多角形加工(ポリゴン加工)、速度同期制御。伝達比: $n_1:n_2$



構文

COUPDEF(<FS>,<LS>,<ZFS>,<NLS>,<ブロック切り替え>,<連結タイプ>)
 COUPON(<FS>,<LS>,<POSFS>)
 COUPONC(<FS>,<LS>)
 COUPOF(<FS>,<LS>,<POSFS>,<POSLS>)
 COUPOFS(<FS>,<LS>)
 COUPOFS(<FS>,<LS>,<POSFS>)

4.16 軸連結

COUPRES (<FS>, <LS>)
 COUPDEL (<FS>, <LS>)
 WAITC (<FS>, <ブロック切り替え>, <LS>, <ブロック切り替え>)

注記

略式表記

COUPOF、COUPOFS、COUPRES、および COUPDEL の各命令には、マスタ主軸を指定しない省略表記を使用できます。

意味

COUPDEF:	ユーザー用に連結を定義/変更します
COUPON:	連結を起動します。スレーブ主軸が、現在の速度に基づいてマスタ主軸に同期します
COUPONC:	以前にプログラム指令した M3 S...または M4 S...による起動時に連結をおこないます。 スレーブ主軸の速度の差は、直ちに処理されます。
COUPOF:	連結を解除します。 - 即時のブロック切り替え: COUPOF (<S2>, <S1>) <POSFS>または<POSLS>解除位置を通過後にのみブロック切り替え: COUPOF (<S2>, <S1>, <POSFS>) COUPOF (<S2>, <S1>, <POSFS>, <POSLS>)
COUPOFS:	スレーブ主軸の停止で連結を解除します。 即時のブロック切り替えにより、高速でブロック切り替えをおこないます: COUPOFS (<S2>, <S1>) 解除位置の通過後にのみブロック切り替えをおこないます: COUPOFS (<S2>, <S1>, <POSFS>)
COUPRES:	連結パラメータを、構成のマシンデータとセッティングデータに再設定します。
COUPDEL:	ユーザー定義の連結を解除します
WAITC:	同期実行条件 (ブロック切り替え時に NOC が IPO に復帰します)が満たされるまで待機します

<FS>:	スレーブ主軸を指定します	
任意選択パラメータ:		
<LS>:	マスタ主軸を指定します 主軸番号で指定します例: S2, S1	
<ZFS>, <NLS>:	FS と LS の間の伝達比。 <ZFS>/<NLS> = 分子/分母 初期設定: <ZFS>/<NLS> = 1.0 ; 任意選択の分母の指定	
<ブロック切り替え>:	ブロック切り替え動作 ブロック切り替えは次のようにおこなわれます。	
	"NOC"	即時に実行する
	"FINE"	「精密同期」へ到達時に実行する
	"COARSE"	「汎用同期」へ到達時に実行する
	"IPOSTOP"	IPOSTOP への到達時; つまり、指令値に基づいた同期制御後に実行する(初期設定)
	ブロック切り替え動作はモーダルに有効です。	
<連結タイプ>:	連結タイプ:FS と LS の連結	
	"DV"	指令値連結(初期設定)
	"AV"	現在値連結
	"VV"	速度連結
	連結タイプはモーダルです。	
<POSFS>:	マスタ主軸とスレーブ主軸の間の角度オフセット	
	値の範囲:	0°... 359.999°
<POSFS>, <POSLS>:	スレーブ主軸とマスタ主軸の解除位置 「ブロック切り替えは、POS _{FS} , POS _{LS} を通過すると有効になります」	
	値の範囲:	0°... 359.999°

例

マスタ主軸とスレーブ主軸による加工

プログラムコード	コメント
	マスタ主軸=メイン主軸=主軸 1
	スレーブ主軸=主軸 2

4.16 軸連結

プログラムコード	コメント
N05 M3 S3000 M2=4 S2=500	マスタ主軸は 3000 rpm で、 スレーブ主軸は 500 rpm で回転します。
N10 COUPDEF(S2,S1,1,1,"NOC","Dv")	連結の定義(設定も可能です)。
...	
N70 SPCON	マスタ主軸を閉ループ位置制御にします(指令値連結)。
N75 SPCON(2)	スレーブ主軸を閉ループ位置制御にします。
N80 COUPON(S2,S1,45)	オフセット位置への高速連結= 45°。
...	
N200 FA[S2] = 100	位置決め速度= 100°/min
N205 SPOS[2] = IC(-90)	負方向に 90°重畳して移動します。
N210 WAITC(S2,"Fine")	「精密」同期を待機します。
N212 G1 X...Y...F...	加工
...	
N215 SPOS[2] = IC(180)	正方向に 180°重畳して移動します。
N220 G4 S50	ドウェル時間=メイン主軸の 50 回転に相当します
N225 FA[S2] = 0	設定速度を有効にします(マシンデータ)。
N230 SPOS[2] = IC(-7200)	20 回転。設定速度で負方向に移動します。
...	
N350 COUPOF(S2,S1)	高速連結解除、S=S2=3000
N355 SPOSA[2] = 0	0°で FS を停止します。
N360 G0 X0 Y0	
N365 WAIT(2)	主軸 2 を待機します。
N370 M5	FS を停止します。
N375 M30	

速度差のプログラミング

プログラムコード	コメント
	マスタ主軸=メイン主軸=主軸 1 スレーブ主軸=主軸 2
N01 M3 S500	マスタ主軸が 500 rpm で回転します。
N02 M2=3 S2=300	スレーブ主軸が 300 rpm で回転します。
...	
N10 G4 F1	メイン主軸のドウェル時間です。
N15 COUPDEF(S2,S1,-1)	連結係数の割合 -1:1
N20 COUPON(S2,S1)	連結を起動します。スレーブ主軸の速度が、マスタ主軸の速度と連結係数から得られます。
...	
N26 M2=3 S2=100	速度差のプログラミング。

速度差に対する移動処理の例

1.COUPON を使用して、これ以前にプログラム指令したスレーブ主軸の連結を起動

プログラムコード	コメント
	マスタ主軸=メイン主軸=主軸 1
	スレーブ主軸=主軸 2
N05 M3 S100 M2=3 S2=200	マスタ主軸は 100 rpm で、スレーブ主軸は 200 rpm で回転します。
N10 G4 F5	ドウェル時間=メイン主軸の 5 秒に相当します。
N15 COUPDEF (S2,S1,1)	FS から LS への伝達比は 1.0 です (初期設定)。
N20 COUPON (S2,S1)	マスタ主軸へ高速連結します。
N10 G4 F5	スレーブ主軸が 100 rpm で回転します。

2.COUPONC を使用して、これ以前にプログラム指令したスレーブ主軸の連結の起動

プログラムコード	コメント
	マスタ主軸=メイン主軸=主軸 1
	スレーブ主軸=主軸 2
N05 M3 S100 M2=3 S2=200	マスタ主軸は 100 rpm で、スレーブ主軸は 200 rpm で回転します。
N10 G4 F5	ドウェル時間=メイン主軸の 5 秒に相当します。
N15 COUPDEF (S2,S1,1)	FS から LS への伝達比は 1.0 です (初期設定)。
N20 COUPONC (S2,S1)	マスタ主軸に高速連結し、以前の速度を S2 に転送します。
N10 G4 F5	S2 が 100 rpm + 200 rpm = 300 rpm で回転します

3.COUPON による、停止状態のスレーブ主軸との連結の起動

プログラムコード	コメント
	マスタ主軸=メイン主軸=主軸 1
	スレーブ主軸=主軸 2
N05 SPOS=10 SPOS[2]=20	スレーブ主軸 S2 を位置決めモードにします。
N15 COUPDEF (S2,S1,1)	FS から LS への伝達比は 1.0 です (初期設定)。
N20 COUPON (S2,S1)	マスタ主軸へ高速連結します。
N10 G4 F1	連結を起動し、S2 が 20°で停止します。

4.COUPONC による、停止状態のスレーブ主軸との連結の起動

注記

位置決めモードまたは軸モード

スレーブ主軸が連結前に位置決めモードまたは軸モードの場合、スレーブ主軸は、COUPON (<FS>, <LS>) および COUPONC (<FS>, <LS>) の場合と同様に動作します。

4.16 軸連結

注記

マスタ主軸と軸運転

連結を定義する前にマスタ主軸が軸運転をおこなっている場合は、マシンデータ MD32000 \$MA_MAX_AX_VELO (最大軸速度)による速度制限値が、連結が有効になった後にも適用されます。

この動作を回避するには、連結を定義する前に、軸を主軸モード(M3 S...または M4 S...)に切り替えてください。

詳細情報

連結設定

連結設定の場合、LS と FS をマシンデータで定義します。主軸設定は、パートプログラムでは変更できません。連結はパートプログラムで、COUPDEF を使用して設定できます(ただし、書き込み保護は無効にしてください)。

ユーザー定義連結

COUPDEF を使用すると、パートプログラムで連結を再定義または変更できます。連結がすでに有効な場合は、新しい連結を定義する前に、最初に COUPDEL で有効な連結を解除してください。

連結は、以下を使用して全体を定義します。

COUPDEF (<FS>,<LS>,<TFS>,<TLS>, ブロック切り替え動作, 連結タイプ)

スレーブ主軸(FS)とマスタ主軸(LS)

連結は、FS と LS の軸名称を使用して一意に定義します。軸名称は、COUPDEF 命令毎にプログラム指令してください。他の連結パラメータはモーダルであるため、それらを変更する場合にのみプログラム指令してください。

例:

```
COUPDEF (S2,S1)
```

伝達比

伝達比は、FS と LS 間の速度比として定義します。

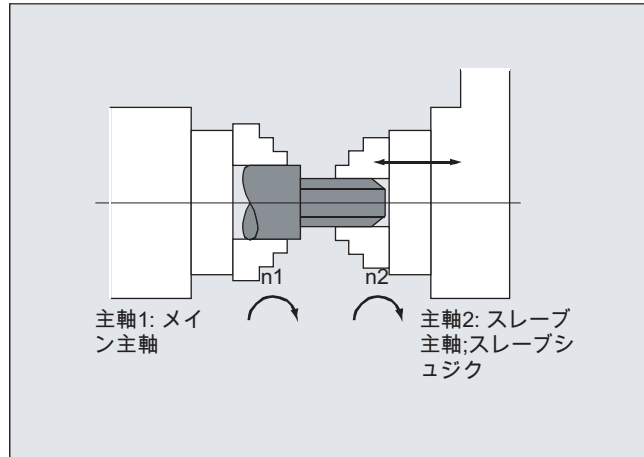
スレーブ主軸/マスタ主軸 = 分子/分母

分子をプログラム指令し、分母はプログラム指令しないでください。初期値 1.0 が分母に設定されます。

例:

スレーブ主軸 S2 とマスタ主軸 S1、伝達比= 1/1 の場合の例です。

COUPDEF (S2, S1, 1.0)



注記

伝達比は、(連結の動作中で各主軸が回転中の場合でも)、変更できます。

ブロック切り替え動作 NOC、FINE、COARSE、IPOSTOP

ブロック切り替え動作のプログラミング時には、以下の略式表記を使用できます。

- "NO":即時に実行する(初期設定)
- "FI":「精密同期」へ到達時に実行する
- "CO":「汎用同期」へ到達時に実行する
- "IP":IPOSTOP への到達時; つまり、指令値に基づいた同期制御後に実行する

連結のタイプ

注記

連結タイプは、連結を解除したときのみ変更できます。

4.16 軸連結

同期モードの起動 COUPON, <POSFS>

- LS と FS の間の任意の角度オフセットでの連結の起動:
 - COUPON (S2, S1)
 - COUPON (S2)
- 角度オフセット<POSFS>による連結の起動
 <POSFS>、正方向へ回転ではマスタ主軸の 0°を基準とします。
 <POSFS>値の範囲:0°... 359.999°
 - COUPON (S2, S1, 30)

注記

角度オフセットは、連結が有効なときにも変更できます。

スレーブ主軸の位置決め

主軸同期連結が有効の場合でも、FS を LS とは無関係に±180°の範囲に位置決めすることができます。

- SPOS による FS の主軸位置決め
 例:SPOS[2] = IC(-90)
 SPOS についての追加情報は、以下を参照してください。

追加情報

主軸の位置決め(SPOS、SPOSA、M19、M70、WAITS) (ページ 133)

速度差

速度差の結果、速度制御モードとなり、LS の移動による FS 速度と主軸のプログラム指令による FS 速度との符号付き重畳により、主軸同期結合が有効になります。

- COUPONC による主軸同期連結
- S<FS>=<速度> [M<FS>=<回転方向>]

注記**必要条件**

- 速度 S... も、回転方向 M3/M4 でプログラム指令します。
- 主軸同期結合 COUPONC による、LS の移動を介した主軸速度の重畳(M <回転方向> S<FS>)は、重畳が有効になっている場合にだけ有効になります。
- マスタ主軸のダイナミック応答は、スレーブ主軸への重畳の適用時に、その応答性制限値を超えないような範囲に制限してください。

速度差についての追加情報は、以下を参照してください。

追加情報

『機能マニュアル、軸と主軸』

速度、加速度:FA、ACC、OVRA、VELOLIMA

スレーブ主軸の軸速度と軸加速度は、以下を使用してプログラム指令できます。

- FA[SPI(S<n>)] または FA[S<n>] (軸の速度)
- ACC[SPI(S<n>)] または ACC[S<n>] (軸の加速度)
- OVRA[SPI(S<n>)] および OVRA[S<n>] (軸のオーバーライド)
- VELOLIMA[SPI(S<n>)] および VELOLIMA[S<n>] (軸の速度の加速と減速)

ここでは n> = 1、2、3、... (スレーブ主軸の主軸番号) の場合です

詳細情報

送り速度制御 (ページ 119)

注記

最大軸加々速度の減速または加速は、主軸には無効です。

軸のダイナミックパフォーマンスについての追加情報は、以下を参照してください。

追加情報

『機能マニュアル、軸と主軸』

プログラム指令可能なブロック切り替え動作 WAITC

WAITC を使用すると、連結パラメータまたは位置決め動作への変更後などに、さまざまな同期条件(汎用、精密、IPOSTOP)によってブロック切り替え動作を定義できます。同期条件を指定しない場合は、COUPDEF 定義で指定したブロック切り替え動作が適用されます。

例

- スレーブ主軸 S2 に対する同期条件 FINE、およびスレーブ主軸 S4 に対する同期条件 COARSE が満たされるまで待機する場合:WAITC(S2,"FINE",S4,"COARSE")
- COUPDEF に従って同期条件が満たされるまで待機する場合 WAITC()

連結の解除 COUPOF

COUPOF を使用すると、連結の解除動作を、以下のように定義できます。

- 即時のブロック切り替えにより連結の解除をおこなう:
 - COUPOF(S2,S1) (マスタ主軸を指定する)
 - COUPOF(S2) (マスタ主軸を指定しない)
- 解除位置通過後に連結を解除します。ブロック切り替えは、解除位置を通過した後に実行されます。
 - COUPOF(S2,S1,150) (解除位置 FS:150°)
 - COUPOF(S2,S1,150,30) (解除位置 FS:150°、LS:30°)

4.16 軸連結

スレーブ主軸停止による連結の解除 COUPOFS

COUPOFS を使用すると、スレーブ主軸停止による連結の解除動作を、以下のように定義できます。

- スレーブ主軸停止と即時のブロック切り替えにより連結の解除をおこなう：
 - COUPOFS (S2, S1) (マスタ主軸を指定する)
 - COUPOFS (S2) (マスタ主軸を指定しない)
- 解除位置通過後に、スレーブ主軸停止により連結を解除します。ブロック切り替えは、解除位置を通過した後に実行されます。
 - COUPOFS (S2, S1, 150) (解除位置 FS:150°)

連結の解除 COUPDEL

COUPDEL は以下のように、連結を解除します。

- COUPDEL (S2, S1) (マスタ主軸を指定する)
- COUPDEL (S2) (マスタ主軸を指定しない)

連結パラメータのリセット、COUPRES

COUPRES は以下のように、マシンデータとセッティングデータでパラメータ設定した連結値を有効にします。

- COUPRES (S2, S1) (マスタ主軸を指定する)
- COUPRES (S2) (マスタ主軸を指定しない)

システム変数

- スレーブ主軸の現在の連結状態
スレーブ主軸の現在の結合状態は、以下のビット指定値で読み取ることができます。
<値> = \$AA_COUP_ACT[<FS>]

ビット	<値>	意味
-	0	連結は無効
2	4	主軸同期連結が有効
注 • 上記以外のすべての値は、軸モードを表しています。 • 主軸がスレーブ主軸または複数の連結の場合、すべての連結の結合状態の値が全体の状態として返されます。		

- 現在の角度オフセット
スレーブ主軸とマスタ主軸との間の現在の角度オフセットは、以下で読み取ることができます。
 - \$AA_COUP_OFFS[<FS>] (指令値側の角度オフセット)
 - \$VA_COUP_OFFS[<FS>] (現在値側の角度オフセット)
- 用途例
フォローアップモードのキャンセル後に、NC プログラムで角度オフセットの差分を修正
角度オフセットの差分 = プログラム指令された角度オフセット - システム変数

詳細情報

システム変数に関する詳細情報は、次の関連資料にあります。

リストマニュアル、システム変数

4.16.6 汎用連結(CP...)

「汎用連結」は、既存の結合タイプ(連結移動、軸間連動機能、電子ギヤ、主軸同期)のすべての連結特性を組み合わせる汎用連結機能です。

この機能を使用して、フレキシブルなプログラム指令がおこなえます。

- ユーザーは、自分のアプリケーションに必要な結合特性を選択できます(ブロック構築原理)。
- それぞれの結合特性を個別にプログラム指令できます。
- 定義された結合の結合特性(結合係数など)を変更できます。
- 追加の結合特性を後から使用できます。
- スレーブ軸の座標系基準システム(基本座標系または機械座標系)をプログラム指令できます。
- 特定の結合特性をシンクロナイズドアクションでもプログラム指令できます。

詳細情報:機能マニュアル シンクロナイズドアクション

注記

結合移動(TRAIL*)、軸間連動機能(LEAD*)、電子ギヤ(EG*)、主軸同期(COUP*)の以前の結合の呼び出しは適応サイクルでサポートします。

4.16 軸連結

すべてのキーワードと結合特性の一覧

下の表に、汎用結合のすべてのキーワードとプログラム指令可能な結合特性の一覧を示します。

キーワード	連結特性/意味	構文
CPDEF	連結モジュールの作成	CPDEF=($\langle Fx \rangle$)
CPDEL	連結モジュールの削除	CPDEL=($\langle Fx \rangle$)
CPLA	マスタ軸の定義	CPLA [$\langle Fx \rangle$] = ($\langle Lx \rangle$)
CPLDEF	マスタ軸の定義と連結モジュールの作成 (CPDEF + CPLA でも可能)	CPLDEF [$\langle Fx \rangle$] = ($\langle Lx \rangle$) または CPDEF=($\langle Fx \rangle$) CPLA [$\langle Fx \rangle$] = ($\langle Lx \rangle$)
CPLDEL	連結モジュールのマスタ軸の削除 (CPDEF + CPLA でも可能)	CPLDEL [$\langle Fx \rangle$] = ($\langle Lx \rangle$) または CPDEL=($\langle Fx \rangle$) CPLA [$\langle Fx \rangle$] = ($\langle Lx \rangle$)
CPON	連結モジュールのスイッチオン	CPON=($\langle Fx \rangle$)
CPOF	連結モジュールのスイッチオフ	CPOF=($\langle Fx \rangle$)
CPLON	連結モジュールのマスタ軸のスイッチオン	CPLON [$\langle Fx \rangle$] = $\langle Lx \rangle$
CPLOF	連結モジュールのマスタ軸のスイッチオフ	CPLOF [$\langle Fx \rangle$] = $\langle Lx \rangle$
CPLNUM	連結係数の分子	CPLNUM [Fx, Lx] = $\langle \text{値} \rangle$
CPLDEN	連結係数の分母	CPLDEN [Fx, Lx] = $\langle \text{値} \rangle$
CPLCTID	カーブテーブルの番号	CPLCTID [Fx, Lx] = $\langle \text{値} \rangle$
CPLSETVAL	連結基準	CPLSETVAL [Fx, Lx] = " $\langle \text{連結基準} \rangle$ "
		" $\langle \text{連結基準} \rangle$ ":
		"CMDPOS" 指令値連結
		"CMDVEL" 速度連結
		"ACTPOS" 現在値連結

キーワード	連結特性/意味	構文		
CPFRS	座標系基準システム	CPFRS [FAx] ="<座標系基準>"		
		"<座標系基準>":	"BCS"	基本座標系
			"MCS"	機械座標系
CPBC	ブロック切り替え条件	CPBC [FAx] ="<ブロック切り替え条件>"		
		"<ブロック切り替え条件>":	"NOC"	ブロック切り替えは、連結状態とは無関係に実行されます。
			"IPOSTOP"	ブロック切り替えは、指令値同期制御で実行されます。
			"COARSE"	ブロック切り替えは現在値同期制御「汎用」で実行されます。
			"FINE"	ブロック切り替えは現在値同期制御「精密」で実行されます。
CPFPOS + CPON	連結をオンにしたときのスレーブ軸の同期位置	CPON=FAx CPFPOS [FAx] =<値>		
CPLPOS + CPON	連結をオンにしたときのマスター軸の同期位置	CPLPOS [FAx, LAx] =<値>		

4.16 軸連結

キーワード	連結特性/意味	構文		
CPFMSON	同期モード	CPFMSON[FAx]="<同期モード>"		
		"<同期モード>":	"CFAST"	連結は時間を優先して接続されます。
			"CCOARSE"	連結は、連結規則に従って必要なスレーブ軸位置が、現在のスレーブ軸位置の範囲内にある場合にのみ接続されます。
			"NTGT"	時間を優先して、次の歯間へアプローチします。
			"NTGP"	軌跡を優先して、次の歯間へアプローチします。
			"NRGT"	歯数に対するギア数の比率に従い、時間を優先して次のセグメントにアプローチします。
			"NRGP"	歯数に対するギア数の比率に従い、軌跡を優先して次のセグメントにアプローチします。
			"ACN"	回転軸の場合のみ 回転軸が、負の軸方向の同期位置に移動します。すぐに同期制御がおこなわれます。
			"ACP"	回転軸の場合のみ 回転軸が、正の軸方向の同期位置に移動します。すぐに同期制御がおこなわれます。
			"DCT"	回転軸の場合のみ

キーワード	連結特性/意味	構文		
				回転軸が、時間を優先してプログラム指令された同期位置に移動します。すぐに同期制御がおこなわれます。
			"DCP"	回転軸の場合のみ 回転軸が、軌跡を優先してプログラム指令された同期位置に移動します。すぐに同期制御がおこなわれます。
CPFMON	スイッチオンでのスレーブ軸の動作	CPFMON[FAx]= "<スイッチオンの動作>"		
		<スイッチオンの動作>":	"STOP"	主軸の場合のみ スレーブ主軸の動作中の動作がスイッチオンの前に停止します。
			"CONT"	主軸とメイン移動軸の場合のみ スレーブ軸/主軸の現在の動作が、開始動作として連結に引き継がれます。
			"ADD"	主軸の場合のみ 現在の重畳移動に加えられた連結動作の動作要素、つまり、スレーブ軸/主軸の現在の動作が現在の 重畳移動として保持されます。

4.16 軸連結

キーワード	連結特性/意味	構文		
CPFMOF	すべてスイッチオフでのスレーブ軸の動作	CPFMOF[FAx]="<スイッチオフの動作>"		
		<スイッチオフの動作>":	"STOP"	スレーブ軸/主軸の停止。 動作中の重畳動作も停止状態まで減速します。その後、連結が解除されます。
			"CONT"	主軸とメイン移動軸の場合のみ スレーブ主軸が、解除時に適用される回転数/速度で移動し続けます。
CPFPOS + CPOF	スイッチオフ時のスレーブ軸のスイッチオフ位置	CPOF=(FAx) CPFPOS[FAx]=<値>		

キーワード	連結特性/意味	構文		
CPMRESET	RESET に対する連結動作	CPMRESET[FAx]="<リセット応答>"		
		<リセット応答>:	"NONE"	連結の現在の状態が保持されます。
			"ON"	適切な連結モジュールが作成されると、連結がスイッチオンされます。定義されているすべてのマスタ軸の関係が有効になります。これは、マスタ軸の関係のすべてまたは一部が有効な場合も実行されます。つまり、連結が完全に有効な状態でも再原点同期が実行されます。
			"OF"	動作中の重畳動作も停止状態まで減速します。その後、連結が解除されます。関連する連結モジュールが明示的な定義(CPDEF)なしで作成されると、連結モジュールは削除されます。そうでない場合は、保持されます。つまり、引き続き使用できます。
			"OFC"	主軸でのみ可能 スレーブ主軸が、解除時に適用される回転数/速度で移動し続けます。連結がスイッチオフされます。関連する連結モジュールが明示的な定義(CPDEF)なしで作成されると、連結モ

4.16 軸連結

キーワード	連結特性/意味	構文		
				ジュールは削除されます。そうでない場合は、保持されます。つまり、引き続き使用できます。
			"DEL"	動作中の重畳動作も停止状態まで減速します。連結が解除された後、削除されます。
			"DELC"	主軸でのみ可能 スレーブ主軸が、解除時に適用される回転数/速度で移動し続けます。連結が解除された後、削除されます。

キーワード	連結特性/意味	構文		
CPMSTART	パートプログラム開始時の 連結動作	CPMSTART [FAx] ="<開始動作>"		
		"<開始動作>":	"NONE"	連結の現在の状態が保持されます。
			"ON"	連結のスイッチオン。 定義されているすべての マスタ軸の関係が有効 になります。これは、 マスタ軸の関係の すべてまたは一部が有 効な場合も実行されま す。つまり、連結が完 全に有効な状態でも再 原点同期が実行されま す。
			"OF"	連結がスイッチオフさ れます。関連する連結 モジュールが明示的な 定義(CPDEF)なしで作成 されると、連結モジュ ールは削除されます。 そうでない場合は、保 持されます。つまり、 引き続き使用できます。
			"DEL"	連結が解除された後、 削除されます。
CPMPRT	プログラムテストによるブ ロック検索実行での、パ ートプログラム開始時の連結 動作	CPMPRT [FAx] ="<開始動作>"		
		"<開始動作>":	CPMSTART を参照ください	
CPLINTR	マスタ軸の入力値のオフ セット値	CPLINTR [FAx, LAx] =<値>		
CPLINSC	マスタ軸の入力値の単位変 換係数	CPLINSC [FAx, LAx] =<値>		
CPLOUTTR	連結の出力値のオフセッ ト値	CPLOUTTR [FAx, LAx] =<値>		

4.16 軸連結

キーワード	連結特性/意味	構文															
CPLOUTSC	連結の出力値の単位変換係数	CPLOUTSC [FAx, LAx] =<値>															
CPSYNCOF	位置同期制御「汎用」の検出値	CPSYNCOF [FAx] =<値>															
CPSYNFIP	位置同期制御「精密」の検出値	CPSYNFIP [FAx] =<値>															
CPSYNCOF2	「汎用」位置同期制御の2番目の検出値	CPSYNCOF2 [FAx] =<値>															
CPSYNFIP2	「精密」位置同期制御の2番目の検出値	CPSYNFIP2 [FAx] =<値>															
CPSYNCOV	速度同期制御「汎用」の検出値	CPSYNCOV [FAx] =<値>															
CPSYNFIV	速度同期制御「精密」の検出値	CPSYNFIV [FAx] =<値>															
CPMBRAKE	特定の停止信号と停止命令に対するスレーブ軸の応答	CPMBRAKE [FAx] =<ビット指定値>															
CPMVDI	特定の NC/PLC インタフェース信号に対する、スレーブ軸の応答	CPMVDI [FAx] =<ビット指定値>															
CPMALARM	特別な連結関連アラーム出力のマスク	CPMALARM [FAx] =<ビット指定値>															
CPSETTYPE	連結タイプ	CPSETTYPE [FAx] ="<連結タイプ>" <table border="1"> <tr> <td>"<連結タイプ>":</td><td>"CP"</td><td>自由にプログラム可能</td></tr> <tr> <td></td><td>"TRAIL"</td><td>連結タイプ「連結移動」</td></tr> <tr> <td></td><td>"LEAD"</td><td>連結タイプ「軸間連動機能」</td></tr> <tr> <td></td><td>"EG"</td><td>連結タイプ「電子ギヤ」</td></tr> <tr> <td></td><td>"COUP"</td><td>連結タイプ「同期制御主軸」</td></tr> </table>	"<連結タイプ>":	"CP"	自由にプログラム可能		"TRAIL"	連結タイプ「連結移動」		"LEAD"	連結タイプ「軸間連動機能」		"EG"	連結タイプ「電子ギヤ」		"COUP"	連結タイプ「同期制御主軸」
"<連結タイプ>":	"CP"	自由にプログラム可能															
	"TRAIL"	連結タイプ「連結移動」															
	"LEAD"	連結タイプ「軸間連動機能」															
	"EG"	連結タイプ「電子ギヤ」															
	"COUP"	連結タイプ「同期制御主軸」															

FAx:スレーブ軸/主軸

LAX:マスタ軸/主軸

注記

(シンクロナイズドアクションのパートプログラムで)明示的にプログラム指令されていない連結特性は、初期設定で有効になります。

プリセットされた連結特性は、初期設定(CPSETTYPE="CP")の代わりに、キーワード CPSETTYPE の設定に従って有効になります。

詳細情報

汎用連結について詳しくは、以下を参照してください。

- 『機能マニュアル、軸と主軸』

4.16.7 法線方向制御

4.16.7.1 連結の定義(TANG)

予約手順 TANG(...)を使用して、回転軸との間の法線方向連結がスレーブ軸として定義され、2本のジオメトリ軸がマスタ軸として定義されます。スレーブ軸はマスタ軸の軌跡タンジェントと連続的に調整されます。

注記

連結係数

1 という連結係数は、明示的にプログラムする必要はありません。
接線軸の方向が、連結係数-1 を使用して回転されます。

構文

TANG (<スレーブ軸>, <マスタ軸_1>, <マスタ軸_2>, <連結係数>, <座標系>, <調整>)

4.16 軸連結

意味

TANG (...):	法線方向連結の定義	
<スレーブ軸>:	スレーブ軸(回転軸)の軸名称	
	データタイプ:	AXIS
	値の範囲:	チャンネル軸名称
<マスタ軸_1>	マスタ軸(ジオメトリ軸)の軸名称 ¹⁾	
<マスタ軸_2>:	データタイプ:	AXIS
	値の範囲:	チャンネルのジオメトリ軸の名称
<連結係数>:	マスタ軸の軌跡タンジェントの変更のための、スレーブ軸の角度変化の係数 n: 角度変化 _{スレーブ軸} = 角度変化 _{軌跡タンジェント} * n	
	データタイプ:	REAL
	初期値:	1.0
<座標系>:	有効な座標系 ²⁾	
	データタイプ:	CHAR
	値:	"B" 基本座標系(デフォルト値) :
		"W": ワーク座標系(使用不可)

<調整>:	調整タイプ		
	データタイプ: CHAR		
	値:	"S":	Standard (デフォルト値) 回転軸のダイナミック応答はマスタ軸に影響を及ぼしません。回転軸のダイナミック応答が実速度追従に必要とされるよりも大きい場合、この方法は十分に正確です。回転軸のダイナミック応答が軌跡タンジェントの変化に従うのに十分な大きさが無い場合、回転軸の向きは未定義の丸み付け隙間に沿ったターゲット方向からはずれます。
		"P":	回転軸のダイナミック応答はマスタ軸の軌跡演算で考慮されます。 このために、TANGON()による法線方向連結の起動時に、2つの追加パラメータを指定してください。 - 丸み付け隙間 - 角度許容範囲 セクション「連結を起動(TANGON) (ページ 1101)」を参照 注記 キネマティックトランスフォーメーションの場合は、調整方法「P」の使用をお奨めします。
注 デフォルト値は明示的にプログラムする必要はありません。			

4.16 軸連結

1) 注記

法線方向連結用のマスタ軸として、機械の初期位置を基準にした機械座標系(MCS)内をプログラムされた軌跡に沿って移動するジオメトリ軸を使用してください。たとえば、旋回ヘッドを持つフライス盤で旋回サイクル CYCLE800 を使用する場合、サイクルがどのように設定されているかに従って、補間が WCS で、たとえばジオメトリ軸 X および Y によって行われます。ただし、法線方向連結は、プログラムされた軌跡に沿って MCS 内で移動するマスタ軸としてのジオメトリ軸で定義してください。このために、機械の**非旋回**条件のジオメトリ軸をマスタ軸として使用してください。

2) 注記

基本座標系(BCS)を MCS を基準にして回転させないでください。たとえば、BCS を ROT 命令または旋回サイクル CYCLE800 で回転させると、法線方向制御が正しくなくなります。

4.16.7.2 挿入ブロックの生成の有効化(TLIFT)

プログラムされた軌跡上の任意の位置でのスレーブ軸のタンジェント変化が、マシンデータ MD37400 \$MA_EPS_TLIFT_TANG_STEP でパラメータ設定されている制限値を超えると、それ以降の軌跡演算はコーナでの設定した動作によって決まります。予約手順 TLIFT(...)を使用しない場合、軌跡は TANG(...) (ページ 1097)および TANGON(...) (ページ 1101)に関連してプログラムされた丸め動作に従って移動されます。

挿入ブロックの生成の有効化

TLIFT(...)が TANG(...)の後にプログラムされている場合、制御装置によって自動的に生成された挿入ブロックが、先読み中にコーナが検出されたときに当該ポイントに挿入されます。

プログラムが実行されると、マスタ軸は挿入ブロックに達したときに停止されます。挿入ブロックで、スレーブ軸は後続のブロックの軌跡タンジェント方向に向かって最大の軸応答性で回転されます。その後、マスタ軸はプログラムされた軌跡上をさらに移動します。

挿入ブロックの生成の無効化

挿入ブロックの生成を無効化するには、TANG(...)を使用して(ただし、その後で TLIFT(...)を使用して挿入ブロックの生成を有効にせずに)法線方向連結をもう一度定義してください。

構文

TLIFT (<スレーブ軸>)

意味

TLIFT(...):	挿入ブロックの計算によるコーナ検出の有効化		
<スレーブ軸>:	スレーブ軸(回転軸)の軸名称		
	データタイプ:	AXIS	
	値の範囲:	チャンネル軸名称	

スレーブ軸の回転速度

軌跡軸

法線方向連結が有効になる前にスレーブ軸が軌跡軸としてすでに移動している場合、回転動作は挿入ブロックで軌跡軸として実行されます。

FGREF[<axis>]=0.001 を使用して基準半径を指定する場合、回転動作はパラメータ設定された最大軸速度で実行されます。

MD32000 \$MA_MAX_AX_VELO[<スレーブ軸>]

軸の位置決め

法線方向連結が有効になる前にスレーブ軸が軌跡軸としてまだ移動していない場合、回転は挿入ブロックで位置決め軸として実行されます。

回転動作はパラメータ設定された位置決め軸速度で実行されます。

MD32060 \$MA_POS_AX_VELO[<スレーブ軸>]

4.16.7.3 連結を起動(TANGON)

予約手順 TANGON(...)を使用して、TANG(...) (ページ 1097) で以前に定義した法線方向連結を有効化します。これで、スレーブ軸はその後の移動中に、軌跡タンジェントと連続的に調整されます。

スレーブ軸の角度

軌跡タンジェントを基準にしたスレーブ軸の角度は、TANG(...) で指定された座標変換比、マシンデータ MD37402 \$MA_TANG_OFFSET でパラメータ設定されたオフセット角度、追加で適用される TANGON(...) に対して指定されたオフセット角度によって決まります。

調整「P」

値「P」が法線方向連結の定義(TANG(...))で調整パラメータとして指定されている場合、パラメータ「丸み付け隙間」とオプションでパラメータ「角度許容範囲」を連結の起動時に設定してください。

4.16 軸連結

角度許容範囲に値 0 が指定されている場合、パラメータ「丸み付け隙間」のみが有効になります。

角度許容範囲に 0 よりも大きい値が指定されている場合、有効な丸み付け隙間は、パラメータ設定された丸み付け隙間とパラメータ設定された角度許容範囲に基づいた丸み付け隙間の最小の値になります。

スレーブ軸のダイナミック応答がパラメータ設定された条件に従うのに十分でない場合は、マスタ軸の軌跡速度がそれに応じて小さくなります。

構文

TANGON (<スレーブ軸>, <オフセット角度>, <丸み付け隙間>, <角度許容範囲>)

意味

TANGON (...):	法線方向連結の有効化	
<スレーブ軸>:	スレーブ軸(回転軸)の軸名称	
	データタイプ:	AXIS
	値の範囲:	チャンネル軸名称
<オフセット角度>:	軌跡タンジェントを基準にしたスレーブ軸のオフセット角度 レファレンス点は回転軸の原点になります。	
	データタイプ:	REAL
<丸み付け隙間>:	最大許容丸み付け隙間 丸み付け隙間がダイナミック条件によって大きくなる場合は、 マスタ軸の軌跡速度が小さくなります。	
	データタイプ:	REAL
<角度許容範囲>:	スレーブ軸のゼロ設定と軌跡タンジェントの間の指定された 角度を基準にした最大許容範囲	
	データタイプ:	REAL

4.16.7.4 連結の無効化(TANGOF)

予約手順 TANGOF(...)を使用して、TANG(...) (ページ 1097)で定義され、TANGON(...) (ページ 1101)で有効になった法線方向連結を無効にします。これで、スレーブ軸はマスタ軸の軌跡タンジェントと連続的に調整されなくなります。ただし、スレーブ軸のマスタ軸への連結は無効化後も保持され、これによって、たとえば以下の機能が防止されます。

- 平面変更
- ジオメトリ軸の切り替え
- スレーブ軸の新しい法線方向連結の定義

スレーブ軸のマスタ軸への連結の最終的な取り消しは、連結が TANGDEL(...) (ページ 1103)で解除されるまで完了しません。

プログラミング

TANGOF (<スレーブ軸>)

意味

TANGOF (...):	法線方向連結の無効化	
<スレーブ軸>:	スレーブ軸(回転軸)の軸名称	
	データタイプ:	AXIS
	値の範囲:	チャンネル軸名称

4.16.7.5 連結の解除(TANGDEL)

TANG(...) (ページ 1097)で定義された法線方向連結は、TANGOF(...) (ページ 1103)による法線方向連結の無効化後も保持されます。このため、既存の法線方向連結は引き続き、たとえば以下の機能を防止します。

- 平面変更
- ジオメトリ軸の切り替え
- スレーブ軸の新しい法線方向連結の定義

予約手順 TANGDEL(...)を使用して、法線方向連結が TANGOF(...)で無効化された後で、既存の法線方向連結を解除します。

構文

TANGDEL (<スレーブ軸>)

4.16 軸連結

意味

TANGDEL (...):	TANG()で定義された法線結合連結の解除	
	効果:	ノンモーダル
<スレーブ軸>:	法線方向連結を解除するスレーブ軸の名称	
	データタイプ:	AXIS
	値の範囲:	チャンネル軸名称

例

マスタ軸の変更

スレーブ軸に対して新しい法線方向連結を別のマスタ軸で定義するには、まず既存の法線方向連結を解除してください。

プログラムコード	コメント
N10 TANG (A, X, Y, 1)	; スレーブ軸 A の法線方向連結の定義 A と X および Y 間
N20 TANGON (A)	; スレーブ軸 A の法線方向連結の有効化
N30 X10 Y20	
...	
N80 TANGOF (A)	; スレーブ軸 A の法線方向連結の無効化
N90 TANGDEL (A)	; スレーブ軸 A の法線方向連結の解除
...	
N120 TANG (A, X, Z)	; スレーブ軸 A の新しい法線方向連結の定義
N130 TANGON (A)	; スレーブ軸 A の新しい法線方向連結の有効化
...	

ジオメトリ軸の切り替え

既存の連結に対してジオメトリ軸の切り替えを実行するには、まず連結を解除してください。

プログラムコード	コメント
N10 GEOAX (2, Y1)	; 2 番目のジオメトリ軸 = 機械軸 Y1
N20 TANG (A, X, Y)	; スレーブ軸 A の法線方向連結の定義
N30 TANGON (A, 90)	; スレーブ軸 A の法線方向連結の有効化
N40 G2 F8000 X0 Y0 I0 J50	; 回転のブロック
N50 TANGOF (A)	; スレーブ軸 A の法線方向連結の無効化
N60 TANGDEL (A)	; スレーブ軸 A の法線方向連結の解除
N70 GEOAX (2, Y2)	; 2 番目のジオメトリ軸 = 機械軸 Y2
N80 TANG (A, X, Y)	; スレーブ軸 A の新しい法線方向連結の定義
N90 TANGON (A, 90)	; スレーブ軸 A の新しい法線方向連結の有効化
...	

4.16.8 マスタ/スレーブ連結(MASLDEF、MASLDEL、MASLON、MASLOF、MASLOFS)

「マスタ/スレーブ連結」により、以下のことが可能になります。

- 関与する軸が停止状態の場合は、スレーブ軸をマスタ軸に連結できます。
- 回転中の速度制御された主軸を連結/連結解除できます。
- ダイナミック設定。

注記

位置決めモード

位置決めモードの軸と主軸の場合は、連結は停止状態のときにのみ、起動と解除がおこなわれます。

構文

```
MASLON (<スレーブ_1>,<スレーブ_2>,...)
MASLOF (<スレーブ_1>,<スレーブ_2>,...)
MASLOFS (<スレーブ_1>,<スレーブ_2>,...)
```

ダイナミック設定

```
MASLDEF (<スレーブ_1>,<スレーブ_2>,...,<マスタ>)
MASLDEL (<スレーブ_1>,<スレーブ_2>,...)
```

意味

MASLON:	一時的なマスタ/スレーブ連結の有効化	
	<スレーブ_x>,...:	スレーブ軸 1 ... n
MASLOF:	動作中のマスタ/スレーブ連結の連結解除	
	<スレーブ_1>,...:	スレーブ軸 1 ... n
MASLOFS:	マスタ/スレーブ連結の連結解除とスレーブ主軸の自動的な減速(注「速度制御モードでの主軸の連結動作」を参照してください)	
	<スレーブ_1>,...:	スレーブ軸 1 ... n
MASLDEF:	パートプログラムからのマスタ/スレーブグループの作成/変更	
	<スレーブ_1>,...:	スレーブ軸 1 ... n
	<マスタ>:	マスタ軸

4.16 軸連結

MASLDEL:	マスタ/スレーブ連結の解除とグループの定義の削除	
	<スレーブ_1>, ...:	スレーブ軸 1 ... n
	注: マシンデータで設定されたマスタ/スレーブ定義は保持されます。	

注記**速度制御モードでの主軸の連結動作**

速度制御モードの主軸の場合 MASLON、MASLOF、MASLOFS、および MASLDEL の連結動作は、次のマシンデータで明示的に指定します。

MD37263 \$MA_MS_SPIND_COUPLING_MODE

MD37263 = 0 の初期設定の場合は、使用される軸が停止状態のときにのみ、スレーブ軸が連結、および連結解除されます。MASLOFS は MASLOF に対応します。

MD37263 = 1 の場合は、連結命令が直ちに実行され、移動も実行されます。MASLON の場合は、連結が直ちに起動され、MASLOFS または MASLOF の場合は、直ちに解除されます。MASLOF では、このとき回転中のスレーブ主軸は、新しい速度がプログラム指令されるまで、その速度を保持します。ただし、MASLOFS では、自動的に減速します。

注記

MASLOF/MASLOFS の場合は、自動先読み停止はおこないません。先読み停止をおこなわないため、次のプログラミングまで、スレーブ軸の \$P システム変数は値が更新されません。

注記

スレーブ軸の場合は、PRESETON を使用して、現在値をマスタ軸の同一値に同期できません。これを行うには、電源投入によって原点確立されていないスレーブ軸の現在値をマスタ/軸の値に設定するために、固定の/スレーブ連結を一時的に解除してください。これで、連結が固定して、再確立されます。

固定のマスタ/スレーブ連結は、次のマシンデータの設定:

MD37262 \$MA_MS_COUPLING_ALWAYS_ACTIVE = 1

を使用して起動します。この設定は、一時的な連結のための言語命令には無効です。

例

例 1: マスタ/スレーブ連結のスレーブ軸に現在値を設定します。

固定のマスタ/スレーブ連結をおこなうために、PRESETON でスレーブ軸の現在値をマスタ軸の値に設定します。

プログラムコード	コメント
\$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=0	; スレーブ軸の固定連結を解除します。
NEWCONF	; マシンデータの変更を有効にします。
STOPRE	
MASLOF(Y1)	; 一時的な連結を解除します
PRESETON(AX2,\$VA_IM(M_AX))	; スレーブ軸の現在値 = マスタ軸の現在値
\$MA_MS_COUPLING_ALWAYS_ACTIVE[AX2]=1	; スレーブ軸の固定連結を有効にします。
NEWCONF	; マシンデータの変更を有効にします。

例 2: マスタ/スレーブ連結のダイナミック設定

軸コンテナ回転後に別の主軸との連結を有効にするには、以前の連結を解除し、設定を消去して、新しい連結を設定してください。

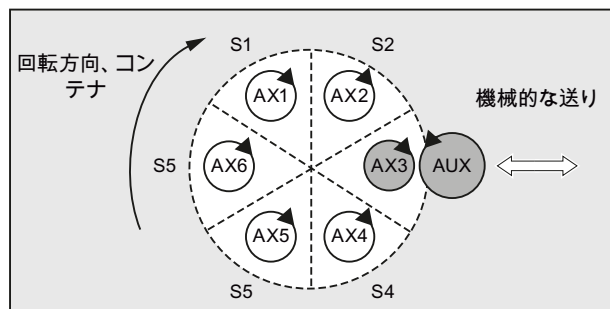


図 4-15 軸コンテナ回転前

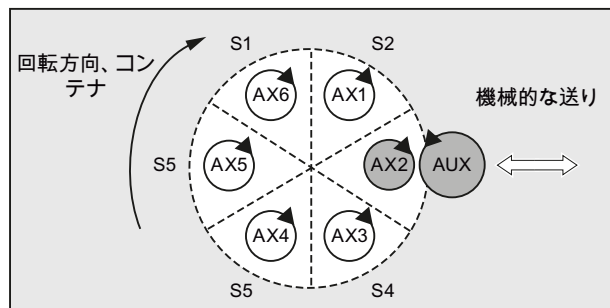


図 4-16 軸コンテナを1スロットだけ回転後

プログラムコード	コメント
MASLDEF(AUX,S3)	; AUX:スレーブ、S3:マスタ = AX3
MASLON(AUX)	; 連結のオン

4.17 シンクロナイズドアクション

プログラムコード	コメント
M3=3 S3=4000	; マスタを回転します。
MASLDEL (AUX)	; 切断して連結を解除します。
AXCTSWE (CT1)	; 軸コンテナ回転を有効にします。
MASLDEF (AUX, S3)	; AUX:スレーブ、S3:マスタ = AX2

詳細情報

- 機能マニュアル、用途
- 機能マニュアル、基本機能

4.17 シンクロナイズドアクション

4.17.1 シンクロナイズドアクションの定義

シンクロナイズドアクションはパートプログラムのブロック内で定義されます。シンクロナイズドアクションの一部でないその他のすべての命令は、このブロック内でプログラミングできません。

シンクロナイズドアクションは次の要素で構成されます。

適用範囲、ID 番号 (オプション)	条件部分 (オプション)			条件が満たされている動作部分			条件が満たされていない動作部分 (オプション)		
	周波数	G 命令 (オプション)	条件	キーワード	G 命令 (オプション)	動作	キーワード	G 命令 (オプション)	動作
--- ¹⁾ ID=<番号> IDS=<番号>	--- ¹⁾ WHENEVER R FROM WHEN EVERY	G...	論理式	DO	G...	動作 1 ... 動作 n	ELSE	G...	動作 1 ... 動作 n

¹⁾ プログラム指令なし

構文

```
DO <アクション 1> ... <アクション n>
<頻度> [<G 機能>] <条件> DO <アクション 1> ... <アクション n>
ID=<番号> <頻度> [<G 機能>] <条件> DO <アクション 1> ... <アクション n>
IDS=<番号> <頻度> [<G 機能>] <条件> DO <アクション 1> ... <アクション n>
IDS=<番号> <頻度> [<G 機能>] <条件> DO <動作...n> ELSE <動作 1...n>
```

詳細情報

『機能説明書』シンクロナイズドアクション

4.18 揺動

4.18.1 非同期揺動(OS、OSP1、OSP2、OST1、OST2、OSCTRL、OSNSC、OSE、OSB)

揺動軸は、揺動動作を解除するまで、2つの反転点1と2の間を、定義した送り速度で前後に移動します。

その他の軸は、揺動動作中に自由に補間できます。連続切り込みは、軌跡移動または位置決め軸を使用して実行できますが、揺動移動と切り込み移動は**無関係**です。

非同期揺動の機能

- 非同期揺動は、軸毎にブロック境界を越えて有効です。
- パートプログラムを使用して、揺動移動をブロック単位で起動できます。
- 複数の軸の補間の組合せと揺動軌跡の重畳は実行できません。

プログラミング

以下の命令を使用すると、パートプログラムから非同期揺動の起動と制御ができます。

プログラム指令値は、メインラン時のブロック同期制御により、対応するセッティングデータに入力され、再度変更されるまでそのまま有効です。

構文

```
OSP1 [<軸>] = <値> OSP2 [<軸>] = <値>
OST1 [<軸>] = <値> OST2 [<軸>] = <値>
FA [<軸>] = <値>
OSCTRL [<軸>] = (<設定オプション>, <リセットオプション>)
OSNSC [<軸>] = <値>
OSE [<軸>] = <値>
OSB [<軸>] = <値>
OS [<軸>] = 1
```

4.18 揺動

OS[<軸>] = 0

意味

<軸>:	揺動軸の名称		
OS:	揺動を起動/解除します		
	値:	1	揺動を オン にします
		0	揺動を オフ にします
OSP1:	反転点 1 の位置を定義します		
OSP2:	反転点 2 の位置を定義します 注: インクレメンタル移動が有効の場合は、NC プログラムでプログラム指令した最新の対応する反転位置まで、位置をインクレメンタルで計算します。		
OST1:	反転点 1 の停止時間を秒単位で定義します		
OST2:	反転点 2 の停止時間を秒単位で定義します		
	<値>:	-2	イグザクトストップを待機せずに補間が継続します
		-1	汎用イグザクトストップを待機します
		0	精密イグザクトストップを待機します
		>0	精密イグザクトストップを待機した後、指定した停止時間だけ待機します
			注: 停止時間の単位は、G4 でプログラム指令した停止時間と同じです。
FA:	送り速度を定義します 送り速度は、位置決め軸の定義送り速度です。送り速度を定義しない場合は、マシンデータに格納された値が適用されます。		

OSCTRL:	設定オプションとリセットオプションを指定します	
	オプション値 0～3 は、解除時の反転点での動作コードです。0～3 のいずれかのタイプを選択できます。その他の設定内容は、選択したタイプと自由に組み合わせることができます。複数のオプションをプラス記号(+)で付加します。	
	<値>:	0 揺動の解除時に次の反転点で停止します(初期設定) 注: 値 1 と 2 をリセットした場合にのみ可能です。
		1 揺動を解除すると、反転点 1 で停止します
		2 揺動を解除すると、反転点 2 で停止します
		3 スパークアウトストロークをプログラム指令していない場合は、揺動を解除するときに反転点へアプローチしません
		4 スパークアウト処理後に終了位置へアプローチします
		8 揺動移動が残移動距離削除によって取り消されたときに、スパークアウトストロークの実行が必要になり、必要に応じて、終了位置へアプローチします。
		16 揺動移動が残移動距離削除によって取り消されたときに、停止の場合と同様に、対応する反転点へのアプローチが必要になります。
		32 新しい切り込みは、次の反転点の後にのみ有効です
		64 FA が 0 と等しい、つまり $FA = 0$ の場合:軌跡の重畳が有効です FA が 0 と等しくない、つまり $FA \neq 0$ の場合:速度の重畳が有効です
		128 回転軸の DC 指令(最短軌跡)
OSNSC:	256	スパークアウトストロークがデュアルストロークです(初期設定)。1=シングルストローク。
	512	最初に開始位置へアプローチします
OSNSC:		スパークアウトストローク数を定義します

4.18 揺動

OSE:	(ワーク座標系の)終了位置を定義し、揺動の解除後にそこへアプローチします。 注: 「OSE」をプログラム指令すると、OSCTRLに対してオプション4が自動的に有効になります。
OSB:	(ワーク座標系の)開始位置を定義し、揺動の起動前にそこへアプローチします。 反転点1の前に開始位置にアプローチします。開始位置が反転点1と同じである場合は、次に反転点2にアプローチします。開始位置に到達したとき、この位置が反転点1と同一であっても停止時間は適用されず、軸は精密イグザクトストップ信号を待機します。設定されているすべてのイグザクトストップ条件が満たされます。 注: セッティングデータ SD43770 \$SA_OSCILL_CTRL_MASK のビット9は、開始位置へのアプローチを開始するように設定してください。

例

例 1:2 つの反転点間で揺動する揺動軸

揺動軸 Z は、位置 10 と 100 の間で揺動します。反転点 1 は精密イグザクトストップにより、反転点 2 は汎用イグザクトストップにより実行されます。揺動軸の送り速度は 250 です。加工運転の終了時には 3 つのスパークアウトストロークを実行し、揺動軸は終了位置 200 へアプローチします。切り込み軸の送り速度は 1 で、X 方向の切り込みは位置 15 で終了します。

プログラムコード	コメント
WAITP (X,Y,Z)	; 初期設定:
G0 X100 Y100 Z100	; 軸の位置決め運転へ切り替え。
WAITP (X,Z)	
OSP1[Z]=10 OSP2[Z]=100	; 反転点 1、反転点 2。
OSE[Z]=200	; 終了位置:
OST1[Z]=0 OST2[Z]=-1	; U1 の停止時間:精密イグザクトストップ ; U2 の停止時間:汎用イグザクトストップ
FA[Z]=250 FA[X]=1	; 揺動軸と切り込み軸の速度
OSCTRL[Z]=(4,0)	; オプションの設定
OSNSC[Z]=3	; 3 つのスパークアウトストローク。
OS[Z]=1	; 揺動を起動します。
WHEN \$A_IN[3]==TRUE DO DELDTG(X)	; 残移動距離削除。
POS[X]=15	; X 軸の開始位置。

プログラムコード	コメント
POS[X]=50	; X 軸の終了位置。
OS[Z]=0	; 揺動を停止します。
M30	

注記

「OSP1[Z]=...」～「OSNCS[Z]=...」命令処理は、1 つのブロックでプログラム指令することもできます。

例 2:反転位置のオンライン変更による揺動

非同期の揺動に必要なセッティングデータは、パートプログラムに設定できます。

セッティングデータをプログラムに直接記述した場合、先読みのときに、この変更が有効になります。同期制御動作は、先読み停止(STOPRE)を使用すると、実現できます。

プログラムコード	コメント
\$SA_OSCILL_REVERSE_POS1[Z]=-10	
\$SA_OSCILL_REVERSE_POS2[Z]=10	
G0 X0 Z0	
WAITP(Z)	
ID=1 WHENEVER \$AA_IM[Z] < \$\$AA_OSCILL_REVERSE_POS1[Z] DO \$AA_OVR[X]=0	; 揺動軸の現在位置が反転点を超えた場合は、切り込み軸が停止します。
ID=2 WHENEVER \$AA_IM[Z] < \$\$AA_OSCILL_REVERSE_POS2[Z] DO \$AA_OVR[X]=0	
OS[Z]=1 FA[X]=1000 POS[X]=40	; 揺動を起動します。
OS[Z]=0	; 揺動を解除します。
M30	

詳細情報

揺動軸

揺動軸には以下が適用されます。

- すべての軸を揺動軸として使用できます。
- 複数の揺動軸を同時に有効にすることができます(最大値:位置決め軸数)。
- 揺動軸には、現在プログラムで有効な G 命令とは無関係に、直線補間 G1 が常に有効です。

4.18 揺動

揺動軸では、以下が可能です。

- ダイナミック座標変換の入力軸として機能します
- ガントリ軸と連結移動軸のガイド軸として機能します
- 以下のように移動します。
 - 加々速度制限なし「BRISK」
または
 - 加々速度制限あり「SOFT」
または
 - 膝形加減速カーブで(位置決め軸として)

揺動の反転点

揺動位置を次のように定義する場合は、現在のオフセットを考慮してください。

- アブソリュート指定
"OSP1[Z]=<値>"
反転点の位置=オフセットの合計+プログラム指令値
- 相対指定
"OSP1[Z]=IC(<値>)"
反転点の位置=反転点 1 +プログラム指令値

例:

プログラムコード

```
N10 OSP1[Z]=100 OSP2[Z]=110
...
N40 OSP1[Z]=IC(3)
```

WAITP

揺動をジオメトリ軸で実行する場合は、「WAITP」でこの軸を揺動のために有効にしてください。

揺動が終了すると、「WAITP」を使用して、揺動軸を位置決め軸として再度入力します。これで、通常の使用を再開できます。

シンクロナイズドアクションと停止時間による揺動

設定停止時間を過ぎると、揺動中に内部ブロック切り替えが実行されます(軸の新しい残移動距離によって示されます)。ブロック切り替え時には、解除機能がチェックされます。解除機能は、動作処理 OSCTRL の制御設定に従って定義されます。このダイナミック応答は送りオーバーライドの作用を受けます。

スパークアウトストロークが開始されるか、終了位置にアプローチ前に、揺動ストロークが実行されることがあります。解除動作が変更されたようにみえますが、そうではありません。

4.18.2 シンクロナイズドアクションによる揺動制御(OSCILL)

この揺動モードでは、切り込み移動を実行できるのは、反転点または、定義した反転領域内のみです。

揺動移動は、必要条件に応じて、次のように動作します。

- 続行する、または
- 切り込みの実行が終了するまで停止する。

構文

1. 揺動のパラメータを定義する
2. シンクロナイズドアクションを定義する
3. 軸を割り当てて、切り込みを定義する

意味

OSP1 [<揺動軸>]=	反転点 1 の位置
OSP2 [<揺動軸>]=	反転点 2 の位置
OST1 [<揺動軸>]=	反転点 1 の停止時間(秒単位)
OST2 [<揺動軸>]=	反転点 2 の停止時間(秒単位)
FA [<揺動軸>]=	揺動軸の送り速度
OSCTRL [<揺動軸>]=	オプションの設定またはリセット
OSNSC [<揺動軸>]=	スパークアウトストローク回数
OSE [<揺動軸>]=	終了位置
WAITP (<揺動軸>)	揺動のための軸を有効にします

軸割り当て、切り込み

OSCILL [<揺動軸>]=(<切り込み軸 1>,<切り込み軸 2>,<切り込み軸 3>)

POSP [<切り込み軸>]=(<終了位置>,<区間長>,<モード>)

4.18 揺動

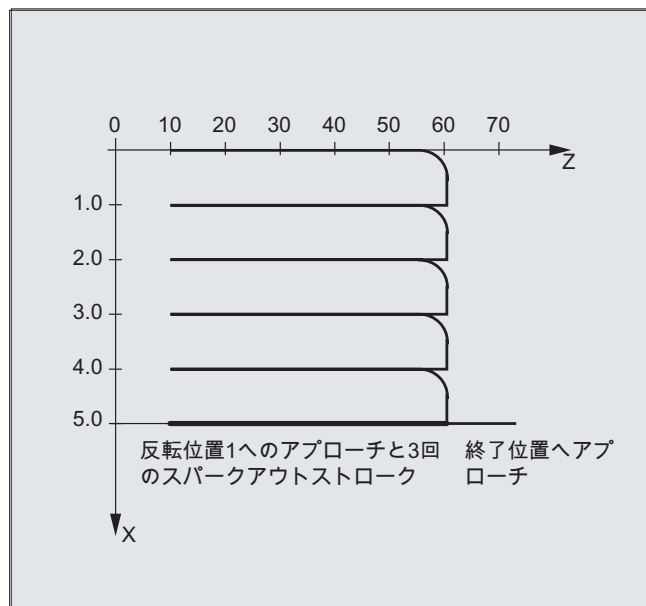
OSCILL:	切り込み軸または軸を揺動軸に割り当てます
POSP:	全体または区間切り込みを定義します(「ファイルとプログラム管理」の章を参照してください)
終了位置:	すべての区間切り込みが移動後の切り込み軸の終了位置
区間長:	反転点/反転領域での区間切り込みの長さ
モード:	切り込み全体を区間切り込みに分割 =残りを同じサイズの 2 区間に分割(初期設定) =すべての区間切り込みを同じサイズに分割

シンクロナイズドアクション

WHEN... .. DO	…のときに、…を実行します
WHENEVER ... DO	…のときは常に、…を実行します

例

反転点 1 で切り込みを実行しません。反転点 2 では、反転点 2 より距離 ii2 だけ前で切り込みを開始します。揺動軸はこの反転点では区間切り込みの終了を待機しません。Z 軸が揺動軸で、X 軸が切り込み軸です



1.揺動のパラメータ

プログラムコード	コメント
DEF INT ii2	; 反転領域 2 の変数を定義します
OSP1[Z]=10 OSP2[Z]=60	; 反転点 1 と 2 を定義します
OST1[Z]=0 OST2[Z]=0	; 反転点 1:精密イグザクトストップ 反転点 2: 精密イグザクトストップ
FA[Z]=150 FA[X]=0.5	; 揺動軸 Z の送り速度、切り込み軸 X の送り速度
OSCTRL[Z]=(2+8+16.1)	; 反転点 2 で揺動動作を解除します; 残移動距離削除後にスパークアウトして 終了位置へアプローチします; 残移動距離削除後に反転位置へアプローチしま す
OSNC[Z]=3	; スパークアウトストローク
OSE[Z]=70	; 終了位置 = 70
ii2=2	; 反転点範囲を設定します
WAITP(Z)	; Z 軸の揺動を有効にします

2.シンクロナイズドアクション

プログラムコード	コメント
WHENEVER \$AA_IM[Z]<\$SA_OSCILL_REVERSE_POS2[Z] DO -> \$AA_OVR[X]=0 \$AC_MARKER[0]=0	; MCS の揺動軸 Z の実位置が反転範囲 2 の開始位 置より小さい場合は、切り込み軸 X の軸オーバ ライドを常に 0%に、インデックス 0 のビットメモ リを常に値 0 に設定します。
WHENEVER \$AA_IM[Z]>=\$SA_OSCILL_REVERSE_POS2[Z] DO \$AA_OVR[Z]=0	; MCS の揺動軸 Z の実位置が反転位置 2 より大 きい場合は、揺動軸 Z の軸オーバライドを常に 0%に 設定します。
WHENEVER \$AA_DTEPW[X] == 0 DO \$AC_MARKER[0]=1	; 区間切り込みの残移動距離が 0 の場合は、イン デックス 0 のビットメモリを常に値 1 に設定しま す。
WHENEVER \$AC_MARKER[0]==1 DO \$AA_OVR[X]=0 \$AA_OVR[Z]=100	; インデックス 0 のビットメモリが 1 に等しい場 合は必ず、切り込み軸 X の軸オーバライドを 0%に 設定します。その結果、早すぎる切り込みが防止 されます(振動軸 Z はまだ、反転領域 2 にありま すが、切り込み軸 X で新規切り込みが可能で す)。振動軸 Z の軸オーバライドの設定を 0% (2 番目のシンクロナイズドアクションの動作) から 100%に戻して、移動します。

->単独ブロックにプログラム指令してください。

3.揺動の開始

プログラムコード	コメント
OSCILL[Z]=(X) POSP[X]=(5,1,1)	; 軸を起動します 揺動軸 Z を切り込み軸として軸 X に割り当てま す。 終了位置 5 まで、軸 X が 1 区間ずつ移動しま す。
M30	; プログラム終了

4.18 揺動

詳細情報

1. 揺動パラメータの定義

揺動のパラメータは、切り込み軸と揺動軸の割り当て、および切り込みの定義を含む移動ブロックの前に定義してください(「非同期揺動」を参照してください)。

2. シンクロナイズドアクションの定義

次の同期制御条件を定義できます。

切り込みを、揺動軸が反転領域内

(ii1、ii2)、または反転点(U1、U2)に到達するまでマスクします。

反転点での切り込みのときに**揺動動作を停止**します。

区間切り込みの完了と同時に**揺動移動を再起動**します。

次の区間切り込みの起動を定義します。

3. 揺動軸と切り込み軸の他に、区間切り込みと全体切り込みを割り当てます。

揺動パラメータの定義

揺動軸と切り込み軸の割り当て **OSCILL;OSCILL**

OSCILL[<揺動軸>]=(<切り込み軸 1>,<切り込み軸 2>,<切り込み軸 3>)

軸割り当てと揺動移動の起動は、「OSCILL」命令で定義します。

3 軸までの切り込み軸を、1 つの揺動軸に割り当てることができます。

注記

揺動を開始する前に、軸の動作の同期制御条件を定義してください。

切り込みの定義:**POSP;POSP**

POSP[<切り込み軸>]=(<終了位置>,<区間長>,<モード>)

「POSP」命令で制御装置に以下を宣言します。

- 全体切り込み(終了位置を基準とする)
- 反転点または反転領域での区間切り込みの長さ
- 終了位置への到達時の区間切り込み動作(モードを基準として)

モード= 0	最後の 2 つの区間切り込みの終点までの残移動距離を、2 つの等しい区間に分割します(初期設定)。
モード= 1	すべての区間切り込みは、同じサイズです。これらは、全体切り込みから算出します。

シンクロナイズドアクションの定義

一般的な揺動には、次の一覧に示す、シンクロナイズドアクションを使用します。

個々の処理の解決方法の例は、ユーザー用の揺動移動を作成するモジュールとして使用できます。

注記

それぞれの場合について、異なる同期制御条件をプログラム指令できます。

キーワード

WHEN ... DO ...	…のときに、…を実行します
WHENEVER ... DO	…のときは常に、…を実行します

機能

以下に説明する言語リソースで、次の機能を実行できます。

1. 反転点での切り込み。
2. 反転領域での切り込み。
3. 両方の反転点の切り込み。
4. 反転点で揺動移動の停止。
5. 揺動移動の再起動。
6. 早すぎる区間切り込みの起動の防止。

ここに示すシンクロナイズドアクションのすべての例は、以下を前提としています。

- 反転点 1 < 反転点 2
- Z = 揺動軸
- X = 切り込み軸

注記

詳しくは、「シンクロナイズドアクション」の章を参照してください。

揺動軸と切り込み軸、および区間切り込みと全体切り込みの割り当て

反転点範囲の切り込み

切り込み移動は、反転点に達する前に反転領域内で開始してください。

4.18 揺動

これらのシンクロナイズドアクションでは、揺動軸が反転領域内に入るまで、切り込み移動を禁止します。

前述の前提に従って、次の命令を使用します。

反転範囲 1:

```
WHENEVER $AA_IM[Z]>$SA_OSCILL_RESERVE_POS1[Z]+ii1 DO $AA_OVR[X] = 0
```

MCS の揺動軸の実位置が反転範囲の開始位置 1 より大きい場合は常に、切り込み軸の軸オーバーライドを 0%に設定してください。

反転範囲 2:

```
WHENEVER $AA_IM[Z]<$SA_OSCILL_RESERVE_POS2[Z]+ii2 DO $AA_OVR[X] = 0
```

MCS での揺動軸の実位置が反転範囲の開始位置 2 より小さい場合は常に、切り込み軸の軸オーバーライドを 0%に設定してください。

反転点での切り込み

揺動軸が反転点に達していない限り、切り込み軸は移動しません。

前述の前提に従って、次の命令を使用します。

反転範囲 1:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

MCS での揺動軸 Z の実位置が反転点 1 の位置より大きい、または小さい場合は常に、切り込み軸 X の軸オーバーライドを 0%に、揺動軸 Z の軸オーバーライドを 100%に設定してください。

反転範囲 2:

反転点 2 の場合:

```
WHENEVER $AA_IM[Z]<>$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0
$AA_OVR[Z]=100
```

MCS での揺動軸 Z の実位置が反転点 2 の位置より大きい、または小さい場合は常に、切り込み軸 X の軸オーバーライドを 0%に、揺動軸 Z の軸オーバーライドを 100%に設定してください。

反転点で揺動移動を停止

揺動軸は反転点で停止し、同時に切り込み移動を開始します。揺動動作が続行され、切り込み移動が完了します。

まだ動作中の以前のシンクロナイズドアクションが切り込み移動を停止している場合は、同時に、このシンクロナイズドアクションを使用して切り込み移動を開始できます。

前述の前提に従って、次の命令を使用します。

反転範囲 1:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS1[Z] DO $AA_OVR[X]=0  
$AA_OVR[Z]=100
```

MCS での揺動軸の実位置が反転位置 1 と同じである場合は常に、揺動軸の軸オーバーライドを 0%に、切り込み軸の軸オーバーライドを 100%に設定してください。

反転範囲 2:

```
WHENEVER $SA_IM[Z]==$SA_OSCILL_RESERVE_POS2[Z] DO $AA_OVR[X]=0  
$AA_OVR[Z]=100
```

MCS での揺動軸 Z の実位置が反転位置 2 と同じである場合は常に、揺動軸 X の軸オーバーライドを 0%に、切り込み軸の軸オーバーライドを 100%に設定してください。

反転点のオンライン評価

比較演算の右側に \$\$ で割り付られたメインラン変数がある場合は、2 個の変数を相互に、IPO サイクルで連続的に評価と比較をおこないます。

注記

詳細については、「シンクロナイズドアクション」の章を参照してください。

揺動移動の再開

このシンクロナイズドアクションの目的は、区間切り込み移動の完了時に揺動軸の移動を続けることです。

前述の前提に従って、次の命令を使用します。

```
WHENEVER $AA_DTEPW[X]==0 DO $AA_OVR[Z]= 100
```

WCS の切り込み軸 X での区間切り込みの残りの距離が 0 に等しい場合は常に、揺動軸の軸オーバーライドを 100%に設定してください。

次の区間切り込み

切り込みの完了時には、次の区間切り込みの起動が早すぎないようにします。

このためには、チャンネル別マーク(\$AC_MARKER[インデックス])を使用します。このマークは、区間切り込みの終了(区間残移動距離=0)時に有効になり、軸が反転領域を出たときに解除されます。次の切り込み移動はシンクロナイズドアクションで防止します。

4.19 パンチングとニブリング

所定の前提に基づいて、以下の命令が反転点 1 に適用されます。

1. マークの設定:

```
WHENEVER $AA_DTEPW[X] == 0 DO $AC_MARKER[1]=1
```

WCS の切り込み軸 X での区間切り込みの残りの距離が 0 に等しい場合は常に、インデックス 1 のビットメモリを 1 に設定してください。

2. マークの削除

```
WHENEVER $AA_IM[Z]<> $SA_OSCILL_RESERVE_POS1[Z] DO $AC_MARKER[1] = 0
```

MCS の揺動軸 Z の実位置が反転点 1 の位置より大きい、または小さい場合は常に、ビットメモリ 1 を 0 に設定してください。

3. 切り込みの禁止

```
WHENEVER $AC_MARKER[1]==1 DO $AA_OVR[X]=0
```

ビットメモリ 1 が同じである場合は常に、切り込み軸 X の軸オーバーライドを 0% に設定してください。

4.19 パンチングとニブリング

4.19.1 起動と解除

4.19.1.1 パンチングとニブリングの起動または解除(SPOF、SON、PON、SONS、PONS、PDELAYON、PDELAYOF、PUNCHACC):

パンチングとニブリングの起動/解除

PON と SON を使用して、パンチングとニブリング機能を起動します。SPOF は、すべてのパンチングとニブリング専用の機能を終了します。モーダル命令 PON と SON は、同時には使用できません。つまり、PON は SON を解除し、SON は PON を解除します。

開始制御のあるパンチング/ニブリング

SONS と PONS 機能も、パンチングまたはニブリング機能を起動します。

SON/PON (補間レベルのストローク制御)とは異なり、これらの機能を使用すると、信号に対応したストローク開始制御をサーボレベルで実行します。これにより、ストローク回数が増加し、その結果、パンチング能力が向上します。

開始制御で信号を使用している間は、ニブリング軸またはパンチング軸の位置変更を伴うすべての機能(手動ハンドル移動、PLC によるフレームの変更、計測機能など)が禁止されます。

遅延のあるパンチング

PDELAYON は、パンチングストロークの出力を遅延します。このモジュールの有効命令には準備機能があり、このために、通常は PON の前に位置します。PDELAYOF の後は、通常のパンチングが再開されます。

注記

遅延時間は、セッティングデータ SD42400 \$SC_PUNCH_DWELLTIME で設定します。

軌跡依存の加速

PUNCHACC を使用すると、穴の間隔に応じて異なる加速度を定義する加減速特性を指定できます。

2 番目のパンチングインタフェース

2 番目のパンチングインタフェース(もう 1 台のパンチングユニットまたは類似のもの)の使用が必要な機械の場合は、制御装置高速デジタル入/出力部の 2 番目のペア(I/O ペア)へ、交互に切り替えることができます。どちらのインタフェースでも、パンチング/ニブリング機能をすべて使用できます。SPIF1 と SPIF2 命令を使用して、1 番目と 2 番目のパンチングインタフェースを切り替えます。

注記

必要条件:マシンデータで、2 番目の I/O ペアをパンチング機能に定義してください(→工作機械メーカーの仕様書を参照してください)。

構文

```
PON G...X...Y...Z...
SON G...X...Y...Z...
SONS G...X...Y...Z...
PONS G...X...Y...Z...
PDELAYON
PDELAYOF
PUNCHACC (<Smin>,<Amin>,<Smax>,<Amax>)
SPIF1/SPIF2
SPOF
```

4.19 パンチングとニブリング

意味

PON:	パンチングを起動します。
SON:	ニブリングの起動
PONS:	開始制御のあるパンチングを起動します。
SONS:	開始制御のあるニブリングを起動します。
SPOF:	パンチング/ニブリングを解除します。
PDELAYON:	遅延のあるパンチングを起動します。
PDELAYOF:	遅延のあるパンチングを解除します。
PUNCHACC:	移動に依存した加減速度を起動します。 パラメータ:
	<Smin> 穴の間隔の最小値
	<Amin> 初期加減速度 <Amin>には、<Amax>より大きい値を指定できます。
	<Smax> 穴の間隔の最大値
	<Amax> 最終加速度 <Amax>には、<Amin>より大きい値を指定できます。
SPIF1:	1 番目のパンチングインタフェースを起動します。 高速 I/O の 1 番目のペアを使用してストロークを制御します。
SPIF2:	2 番目のパンチングインタフェースを起動します。 高速 I/O の 2 番目のペアを使用してストロークを制御します。
	注: 1 番目のパンチングインタフェースは常に、RESET またはコントロールシステムの電源投入の後に有効になります。1 つのパンチングインタフェースのみを使用する場合は、プログラム指令する必要はありません。

例

例 1:ニブリングの起動

プログラムコード	コメント
...	
N70 X50 SPOF	; パンチングを開始せずに位置決めします。
N80 X100 SON	; ニブリングを起動し、移動 (X=50) の前、およびプログラム指令した移動の完了時 (X=100) にストロークを開始します。
...	

例 2:遅延のあるパンチング

プログラムコード	コメント
...	
N170 PDELAYON X100 SPOF	; パンチングを開始せずに位置決めし、遅延したパンチング開始を起動します。
N180 X800 PON	; パンチングを起動します。パンチのストロークは、終了位置に達すると、遅延されて出力されます。
N190 PDELAYOF X700	; 遅延のあるパンチングを解除し、プログラム指令した移動の完了時に通常のパンチングを開始します。
...	

例 3:2 つのパンチングインタフェースによるパンチング

プログラムコード	コメント
...	
N170 SPIF1 X100 PON	; ブロックの終点で、1 番目の高速出力でストロークを開始します。[ストローク有効]信号が、1 番目の入力で監視されます。
N180 X800 SPIF2	; 2 番目のストロークを、2 番目の高速出力で開始します。[ストローク有効]信号が、2 番目の入力で監視されます。
N190 SPIF1 X700	; その後のストロークはすべて、1 番目のインタフェースで制御されます。
...	

詳細情報

開始制御のあるパンチングとニブリング(PONS/SONS)

開始制御のあるパンチングとニブリングは、複数のチャンネルで同時には実行できません。PONS または SONS は、一度に 1 つのチャンネルでのみ起動できます。

軌跡依存の加速(PUNCHACC)

例:

PUNCHACC (2, 50, 10, 100)

2 mm 未満の穴間隔:

軸は、最大加速度の 50%のレートで加速します。

2～10 mm の穴間隔:

加速度は間隔に比例して、100%に増加します。

10 mm を超える穴間隔:

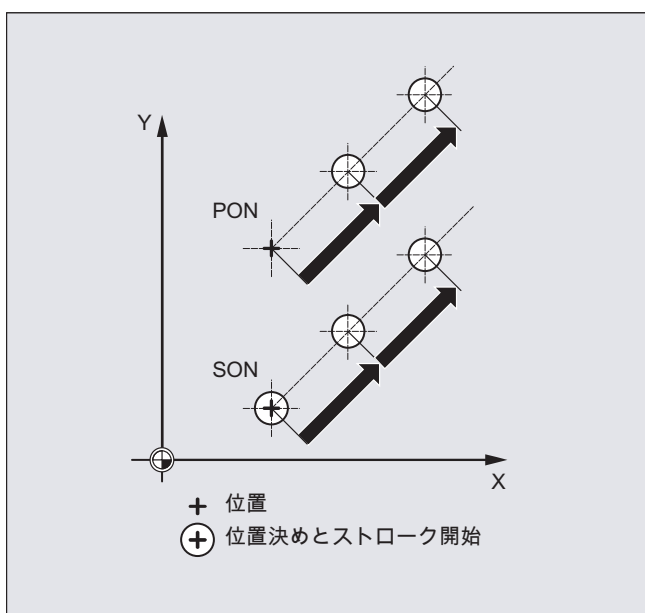
100%の加速度で移動します。

4.19 パンチングとニブリング

最初のストロークの開始

機能の起動後に最初のストロークを開始するタイミングは、ニブリングまたはパンチングの選択に応じて、次のように異なります。

- PON/PONS:
 - すべてのストローク(起動後の最初のブロックのストロークも含む)をブロックの終点で実行します。
- SON/SONS:
 - ニブリング機能の起動後の最初のストロークを、ブロックの始点で実行します。
 - それ以降の各ストロークは、ブロックの終点で開始します。



その時のパンチングとニブリング

パンチング軸またはニブリング軸(有効平面上の軸)の移動情報がブロックに含まれる場合にのみ、ストロークを開始します。

ただし、ストロークを同じ位置で開始するためには、パンチング軸/ニブリング軸の1つを移動軌跡0でプログラム指令できます。

旋回工具による加工

注記

旋回工具を、プログラム指令軌跡に接して位置決めする場合は、法線方向制御機能を使用してください。

M 命令の使用

以前のバージョンのように、マクロ指令により、言語命令の代わりに特殊な M 機能を使用できます(互換性)。以前のシステムで使用されていた M 機能と同等の言語命令は、次のとおりです。

M20、M23	≡	SPOF
M22	≡	SON
M25	≡	PON
M26	≡	PDELAYON

マクロファイルの例:

プログラムコード	コメント
DEFINE M25 AS PON	; パンチング ON
DEFINE M125 AS PONS	; 開始制御のあるパンチングのオン
DEFINE M22 AS SON	; ニブリング ON
DEFINE M122 AS SONS	; 開始制御のあるニブリングのオン
DEFINE M26 AS PDELAYON	; 遅延のあるパンチングのオン
DEFINE M20 AS SPOF	; パンチング、ニブリングのオフ
DEFINE M23 AS SPOF	; パンチング、ニブリングのオフ

プログラミング例:

プログラムコード	コメント
...	
N100 X100 M20	; パンチングを開始せずに位置決めします。
N110 X120 M22	; ニブリングを起動し、移動の前後でストロークを開始します。
N120 X150 Y150 M25	; パンチングを起動し、動作終了時にストロークを開始します。
...	

4.19.2 自動軌跡分割

軌跡セグメントへの分割

パンチングまたはニブリングを有効にすると、SPP と SPN の両方で、軌跡軸に対してプログラム指令した合計移動区間を、同じ長さの軌跡セグメント数に分割します(等間隔の軌跡分割)。内部的には、各軌跡セグメントがブロックに対応します。

4.19 パンチングとニブリング

ストローク数

パンチング時には、最初のストロークを最初の軌跡セグメントの終点で実行します。これに対し、ニブリング時には、最初の軌跡セグメントの始点で実行します。したがって、移動区間全体で、以下の数を取得します。

パンチング:ストローク数=軌跡セグメント数

ニブリング:ストローク数=軌跡セグメント数+1

補助機能

補助機能は、生成したブロックの最初で実行されます。

構文

SPP=

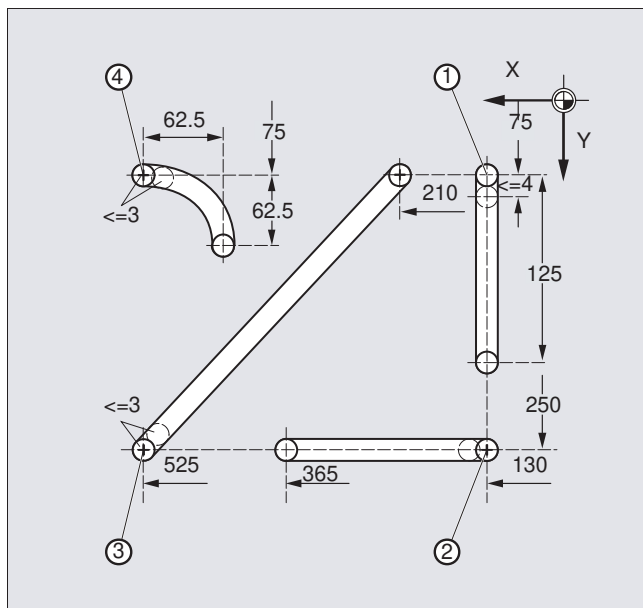
SPN=

意味

SPP:	軌跡セグメントのサイズ(ストロークの最大間隔); モーダル
SPN:	ブロック毎の軌跡セグメント数; モーダルで有効

例 1

プログラム指令したニブリングセグメントを、自動的に軌跡セグメントに分割します。

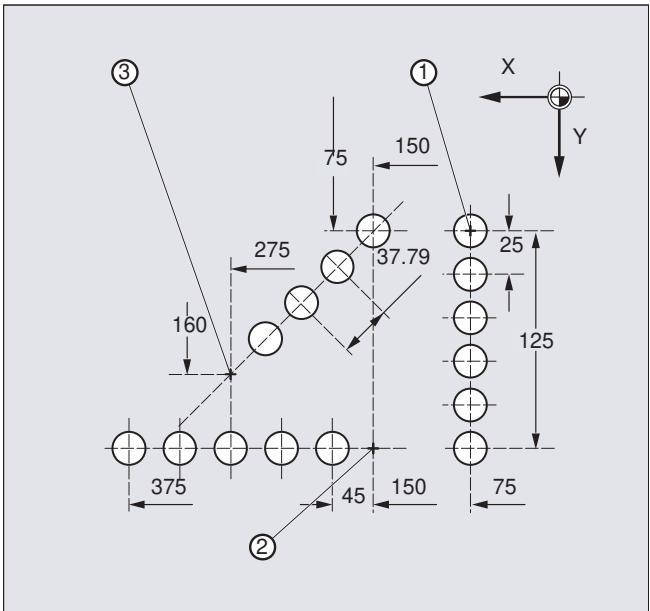


プログラムコード	コメント
N100 G90 X130 Y75 F60 SPOF	; 始点 1 での位置決め
N110 G91 Y125 SPP=4 SON	; ニブリングのオン; 自動軌跡分割の 最大軌跡セグメント長さ: 4 mm
N120 G90 Y250 SPOF	; ニブリングのオフ; 始点 2 への 位置決め
N130 X365 SON	; ニブリングのオン; 自動軌跡分割の 最大軌跡セグメント長さ: 4 mm
N140 X525 SPOF	; ニブリングのオフ; 始点 3 への 位置決め
N150 X210 Y75 SPP=3 SON	; ニブリングのオン; 自動軌跡分割の 最大軌跡セグメント長さ: 3 mm
N160 X525 SPOF	; ニブリングのオフ; 始点 4 への 位置決め
N170 G02 X-62.5 Y62.5 I J62.5 SPP=3 SON	; ニブリングのオン; 自動軌跡分割の 最大軌跡セグメント長さ: 3 mm
N180 G00 G90 Y300 SPOF	; ニブリングのオフ

4.19 パンチングとニブリング

例 2

自動軌跡分割を、それぞれ連続穴に対しておこないます。最大軌跡セグメント長さ(SPP 値)を分割に対して指定します。



プログラムコード	コメント
N100 G90 X75 Y75 F60 PON	; 始点 1 への位置決め; 穴を 1 つあけます
N110 G91 Y125 SPP=25	; 自動軌跡分割の最大軌跡 セグメント長さ:25 mm
N120 G90 X150 SPOF	; パンチングのオフ; 始点 2 への 位置決め
N130 X375 SPP=45 PON	; パンチングのオン;自動軌跡分割の 最大軌跡セグメント長さ:45 mm
N140 X275 Y160 SPOF	; パンチングのオフ; 始点 3 への 位置決め
N150 X150 Y75 SPP=40 PON	; パンチングのオン、プログラム指令した 40 mm の軌跡セグメント長さ ではなく、算出した 37.79 mm の軌跡セグメント長さを使用します。
N160 G00 Y300 SPOF	; パンチングのオフ; 位置決め

4.19.2.1 軌跡軸の軌跡分割

SPP 軌跡セグメントの長さ

SPP を使用して、各ストロークの最大間隔を指定します。これにより、軌跡セグメントの最大長が指定され、合計移動距離がこの長さに分割されます。この命令は、SPOF または SPP=0 で解除します。

例:

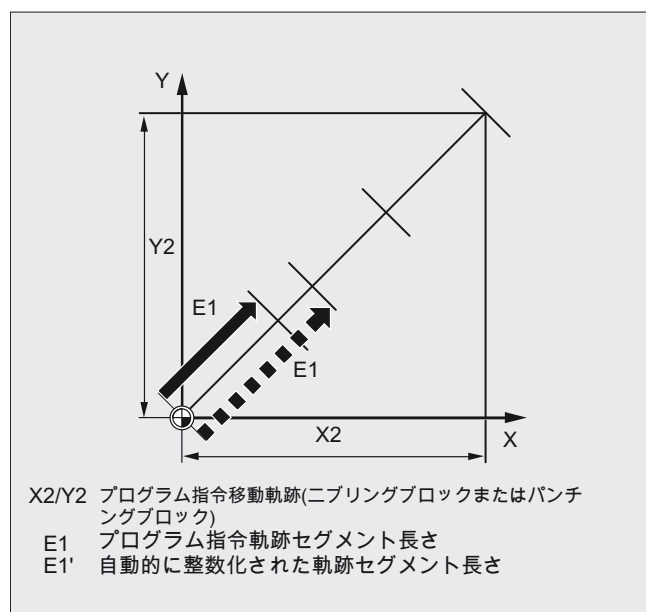
```
N10 SON X0 Y0
```

```
N20 SPP=2 X10
```

合計移動距離 10 mm を、それぞれ 2 mm の 5 つの軌跡区間に分割します(SPP=2)。

注記

SPP による軌跡セグメントは常に等間隔です。つまり、すべてのセグメントは等しい長さです。言い換えると、プログラム指令した軌跡セグメントのサイズ(SPP の設定)は、合計移動距離と SPP 値の商が整数の場合にのみ有効です。これに該当しない場合は、商が整数になるように、軌跡セグメントのサイズを内部的に縮小します。



例:

```
N10 G1 G91 SON X10 Y10
```

```
N20 SPP=3.5 X15 Y15
```

4.19 パンチングとニブリング

合計移動距離が 15 mm で軌跡セグメント長さが 3.5 mm の場合は、商が整数値ではありません(4.28)。この場合は、SPP 値を、次の整数の商に減らします。この例の結果は、3 mm の軌跡セグメント長さです。

SPN の軌跡セグメントの数

SPN は、合計移動距離から生成される軌跡セグメントの数を定義します。セグメントの長さは自動的に計算されます。SPN はノンモーダルであるため、パンチングまたはニブリングはそれぞれ、PON または SON で事前に有効にしておいてください。

4.19.2.2 単独軸の軌跡分割

軌跡軸の他に、単独軸をパンチング軸/ニブリング軸として定義した場合は、自動軌跡分割機能を単独軸に対して有効にできます。

SPP の単一軸の動作

プログラム指令した軌跡セグメント長さ(SPP)は基本的に、軌跡軸を基準とします。したがって、単一軸運動と SPP 値に加えて軌跡軸がプログラムされているブロックでは、SPP 値は無視されます。

単独軸と軌跡軸の両方をブロックにプログラムした場合、単独軸の動作は、当該のマシンのデータの設定によって異なります。

1. 初期設定

単独軸が移動する軌跡は、SPP で生成された中間ブロックの間で均等に配分されます。

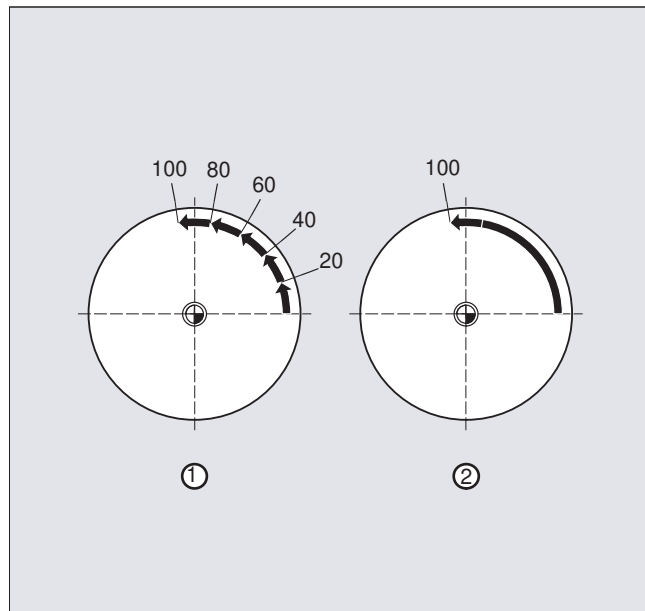
例:

```
N10 G1 SON X10 A0
```

```
N20 SPP=3 X25 A100
```

ストローク間隔を 3 mm にした結果として、X 軸(軌跡軸)の合計移動距離 15 mm に対して 5 つのブロックが生成されます。

このため、A 軸がすべてのブロックで 20°回転します。



1. 軌跡分割をおこなわない単独軸
単独軸は、生成されたブロックのうち、最初のブロックで全距離を移動します。
2. 異なる軌跡分割
単独軸の動作は、軌跡軸の補間に応じて、次のように異なります。
 - 円弧補間: 軌跡分割
 - 直線補間: 軌跡分割を行いません

SPN の動作

プログラムした軌跡セグメント数は、軌跡軸を同時にプログラムしていない場合でも適用されます。

必要条件単独軸をパンチング軸/ニブリング軸として定義します。

4.20 研削加工

4.20.1 研削用工具監視の有効化/無効化(TMON、TMOF)

予約手順 TMON(...)および TMOF(...)を使用して、研削用工具監視を有効化または無効化します(ジオメトリおよび速度監視)。

4.20 研削加工

必要条件

工具固有のパラメータ\$TC_TPG1～\$TC_TPG9 を設定してください。

構文

```
TMON (<TNo>)  
...  
TMOF (<TNo>)
```

意味

TMON (...) :	研削用工具監視の有効化 命令は、研削用工具監視が有効になるチャンネルでプログラミングしてください。
TMOF (...) :	研削用工具監視の無効化 命令は、研削用工具監視が無効になるチャンネルでプログラミングしてください。
<TNo>:	T 番号 注記: 監視を現在使用されている有効なといしではなく、無効なといしに対してオンまたはオフにする場合にのみ必要です。
TMOF (0):	すべての工具の監視を解除します

4.21 停止延長と退避(ESR)

停止延長と退避機能(以降、ESR と呼びます)は、障害状況が発生した場合に、処理に応じてフレキシブルな対応ができるようにします。

- **停止延長**

特定の障害状況で可能であれば、停止延長を有効にしたすべての軸を正しい手順で停止させます。

- **後退**

現在使用中の工具をできるだけ素早くワークから退避させます。

- **ジェネレータ運転(SINAMICS ドライブ機能「Vdc 制御」)**

電源電圧の故障などによって、パラメータ設定可能な DC リンク電圧値が降下した場合、この目的(ジェネレータ運転)のためにドライブの制動エネルギーを回生して退避に必要な電力エネルギーを生成します。

トリガソース

一般ソース(NC 外部/グローバルまたはモードグループ/チャンネル別の):

- デジタル入力(例: NCU モジュール)または読み返しが可能な制御装置内部のデジタル出力イメージ(\$A_IN、\$A_OUT)
- チャンネル状態(\$AC_STAT)
- VDI 信号(\$A_DBB)
- 多数のアラームを含むグループメッセージ(\$AC_ALARM_STAT)

軸ソース

- スレーブ軸の緊急退避のしきい値(電子連結の同期制御、\$VA_EG_SYNCDIFF[<スレーブ軸>])
- ドライブ:DC リンクの警告レベル(不足電圧寸前)、\$AA_ESR_STAT[<軸>]
- ドライブ:ジェネレータの最低速度レベル(使用可能な回生回転エネルギーがこれ以上ない)、\$AA_ESR_STAT[<軸>]

静的シンクロナイズドアクションのゲート論理:ソース/応答リンク

静的シンクロナイズドアクションのフレキシブルなゲート機能を使用すると、ソースに応じた特定の応答を比較的素早く返すことができます。

静的シンクロナイズドアクションを使用するすべての関連ソースのリンクは、ユーザーの責任でおこないます。これにより、ソースのシステム変数を全体としてまたはビット入力

4.21 停止延長と退避(ESR)

によつて的確に評価し、必要な応答を返す論理演算をおこなうことができます。静的シンクロナイズドアクションは、すべての運転モードで有効です。

起動

機能の有効化

ジェネレータ運転、停止、退避の機能は、対応する制御信号\$AA_ESR_ENABLE の設定によって解除されます。この制御信号は、シンクロナイズドアクションで変更できます。

機能の起動

システム変数\$AC_ESR_TRIGGER を設定すると、有効なすべての軸でまとめて ESR が起動されます。

ジェネレータ運転は、DC リンクの不足電圧寸前の状態が検出されたときに、ドライブで「自動的に」開始されます。

ドライブに依存しない停止や退避は、通信障害(NC とドライブ間)が検出されてドライブで DC リンクの不足電圧が検出されたときに有効になります(設定と有効化が必要)。

また、ドライブに依存しない停止や退避は、適切な制御信号\$AN_ESR_TRIGGER(すべてのドライブに対するブロードキャスト命令)を設定して NC によって起動することもできます。

詳細情報

『機能マニュアル、軸と主軸』

4.21.1 NC 制御 ESR

4.21.1.1 NC 制御による退避(POLF、POLFA、POLFMASK、POLFMLIN)

NC 制御による後退には、特定の初期条件が必要です(「NC 制御による退避(POLF、POLFA、POLFMASK、POLFMLIN) (ページ 1136)」を参照してください)。この必要条件を満たしている場合は、システム変数\$AC_ESR_TRIGGER(または一軸用の\$AA_ESR_TRIGGER)を設定すると、チャンネル内の 1 つまたは複数の後退軸に対して設定された速リトラクト(LIFTFAST)が起動されます。

構文

POLF (<軸>) =<位置>
POLFA (<軸>,<タイプ>,<位置>)

4.21 停止延長と退避(ESR)

POLFMASK(<軸_1>,<軸_2>,...)
 POLFMLIN(<軸_1>,<軸_2>,...)

POLFA には以下の省略形式を使用できます。
 POLFA(<軸>,<タイプ>) ; 単独軸の後退の省略形
 POLFA(軸,0/1/2) : 簡単な無効化/有効化
 POLFA(軸,0,\$AA_POLFA[軸]) : 先読み停止をおこなう
 POLFA(軸,0) : 先読み停止をおこなわない

意味

POLF:	後退軸の目標位置を指定するアドレス		
	POLF はモーダルです。		
	<軸>:	後退するジオメトリ軸またはチャネル/機械軸の名称	
	位置>:	後退位置	
		タイプ:	REAL
		ジオメトリ軸には WCS、その他の場合は MCS が使用されます。 ジオメトリ軸とチャネル/機械軸が同じ識別子を持つ場合は、後退は WCS でおこなわれます。	

4.21 停止延長と退避(ESR)

POLFA:	一軸の後退位置を指定する予約サブプログラム呼び出し			
	<軸>:	チャンネル軸識別子		
	<タイプ>:	位置指定モード		
		タイプ:	INT	
		値:	0:	位置値を無効としてマーク
			1:	位置値はアブソリュート
			2:	位置値はインクレメンタル(距離)
	注: 軸が一軸でないか、タイプが欠落している、またはタイプ=0 の場合、対応するアラームが出力されます。			
位置>:	後退位置(上記を参照してください)			
	注: 位置値はタイプ=0 でも確定できます。この値は無効としてマークするだけで、後退させるには位置値を再プログラム指令してください。			
POLFMASK:	高速リトラクトの始動後に 互いに無関係 に後退する軸を選択する予約サブプログラム呼び出し			
	<軸_1>, ...:	高速リトラクト時に POLF で定義した位置に移動する軸の名称 指定する軸はすべて同じ座標系に属するものにしてください。		
	POLFMASK () で軸を指定しない場合は、互いに無関係に後退するすべての軸の高速リトラクトが解除されます。			
POLFMLIN:	高速リトラクトの始動後に 直線補間 で後退する軸を選択する予約サブプログラム呼び出し			
	<軸_1>, ...:	上記を参照してください。		
	POLFMLIN () で軸を指定しない場合は、直線補間で後退するすべての軸の高速リトラクトが解除されます。			

注記

POLFMASK または POLFMLIN で固定位置への高速リトラクトを有効にする前に、選択した軸に対して POLF で位置をプログラム指令してください。

注記

軸を 1 つずつ POLFMASK、POLFMLIN または POLFMLIN、POLFMASK の順に有効にした場合、それぞれの軸に対して常に最後の定義が適用されます。

注記

POLF でプログラム指令した位置と、POLFMASK または POLFMLIN による有効化は、パートプログラムの起動時に解除されます。つまり、パートプログラム毎に POLF の値と POLFMASK または POLFMLIN で選択する軸を再プログラム指令してください。

注記

省略形式の POLFA を使用する場合は、タイプのみが変更されます。そのため、後退位置か後退軌跡のどちらかに実際の意味のある値が含まれていることを確認してください。特に、後退位置と後退軌跡は、電源投入後に再設定してください。

例

個々の軸の後退:

プログラムコード	コメント
MD37500 \$MA_ESR_REACTION[AX1]=21	; NC 制御による後退
...	
\$AA_ESR_ENABLE[AX1] = 1	
POLFA (AX1,1,20.0)	; AX1 に軸の後退位置 20.0 (アブソリュート) を割り当てます。
\$AA_ESR_TRIGGER[AX1]=1	; 後退はここから開始します。

詳細情報**NC 制御による退避の必要条件**

- チャンネルで NC 制御による退避のための後退軸を設定します。
MD37500 \$MA_ESR_REACTION = 21
- この軸に対して ESR を有効にしてください。
\$AA_ESR_ENABLE = 1
- 遅延時間を定義します。
MD21380 \$MC_ESR_DELAY_TIME1
MD21381 \$MC_ESR_DELAY_TIME2
- パートプログラムで POLF を使用して軸別の後退位置を設定しています。

4.21 停止延長と退避(ESR)

- POLFMASK/POLFMLIN を使用して NC 制御による後退のための軸を選択します。
- 後退移動のための有効な信号を設定し、設定したままにしてください。

NC 制御による退避の有効化と起動

システム変数\$AC_ESR_TRIGGER = 1 を設定し、このチャンネルに後退軸を設定し(MD37500 \$MA_ESR_REACTION = 21 など)、この軸に対して\$AA_ESR_ENABLE = 1 を設定すると、このチャンネルで高速リトラクト(LIFTFAST)が有効になります。

POLFMASK または POLFMLIN で選択した軸に対して、POLF(または POLFA)で設定した後退移動が、パートプログラムでこれらの軸に対して定義された軌跡移動の代わりに使用されます。

MD21380 \$MC_ESR_DELAY_TIME1 と MD21381 \$MC_ESR_DELAY_TIME2 の合計時間が後退に使用できる最大時間です。この時間が経過すると、後退軸に対してフォローアップを伴う急減速も開始されます。

注記

拡張後退(\$AC_ESR_TRIGGER によって起動された LIFTFAST/LFPOS など)は**中断できず**、非常停止でのみ、すぐに終了できます。

注記

\$AC_ESR_TRIGGER によって起動された後退は、複数の後退を防ぐためにロックされています。

一軸の後退

一軸の後退では、POLFA で一軸の後退位置をプログラム指令し、次の条件を満たしてください。

- \$AA_ESR_ENABLE = 1
- 起動時(\$AAAA_ESR_TRIGGER = 1)には<軸>を一軸にしてください。
- <タイプ>は 1 または 2 にしてください。

高速リトラクト時の後退方向

高速リトラクトを開始したときに有効なフレームが考慮されます。

注記

回転有りのフレームも POLF によるリトラクトの方向に影響します。

軸入れ替え

後退軸は常に 1 つの NC チャンネルのみに割り当てて、チャンネル間で切り替えないでください。後退軸を別のチャンネルに入れ替えようとすると、アラームが出力されます。

\$AA_ESR_ENABLE[AX]=0 を使用してこの軸を無効にした後でのみ、新しいチャンネルに入れ替えることができます。軸の入れ替え後、\$AA_ESR_ENABLE[AX]=1 によって軸を再度有効にできます。

中立軸

中立軸は NC 制御による ESR を実行できません。

4.21.1.2 NC 制御による停止

NC 制御による停止は、チャンネルでシステム変数\$AC_ESR_TRIGGER(または一軸用の\$AA_ESR_TRIGGER)によって設定した停止軸に対して適用されます。

必要条件

- チャンネルで NC 制御による停止を適用する停止軸を設定します。
MD37500 \$MA_ESR_REACTION = 22
- この軸に対して ESR を有効にしてください。
\$AA_ESR_ENABLE = 1
- 遅延時間を定義します。
MD21380 \$MC_ESR_DELAY_TIME1(遅延時間、ESR 軸)
MD21381 \$MC_ESR_DELAY_TIME2(補間減速のための ESR 時間)

実行

この軸は、MD21380 で設定した時間の間、プログラム指令に従って補間を続けます。MD21380 で指定した遅延時間の経過後、制御減速(減速停止)が開始されます。MD21381 の時間は、補間制御減速に使用できる最大時間です。この時間の経過後、高速制動とその後の実速度追従が開始されます。

例

一軸の停止:

プログラムコード	コメント
MD37500 \$MC_ESR_REACTION[AX1] = 22	; NC 制御による停止
MD21380 \$MC_ESR_DELAY_TIME1[AX1] = 0.3	
MD21381 \$MC_ESR_DELAY_TIME2[AX1] = 0.06	

4.21 停止延長と退避(ESR)

プログラムコード	コメント
...	
\$AA_ESR_ENABLE[AX1] = 1	
\$AA_ESR_TRIGGER[AX1]=1	; ここから停止が開始します。

4.21.2 ドライブ統合 ESR

4.21.2.1 ドライブ内蔵の停止の設定(ESRS)

ドライブ内蔵の ESR 機能の「停止」のためのドライブパラメータは、ESRS (...) 機能を使用して設定されます。

構文

ESRS (<アクセス_1>,<停止時間_1>[, ..., <軸_n>,<停止時間_n>])

意味

ESRS (...) :	ESR 機能「停止」のためにドライブパラメータに書き込む機能 この機能は: • ブロック内では単独にします。 • 先読み停止をおこないます。 • シンクロナイズドアクションでは使用できません。	
<軸_1>, ..., <軸_n>:	ドライブ内蔵の停止を設定する軸 この軸に対して、ドライブパラメータ p0888 (設定)がドライブに書き込まれます。 p0888 = 1	
	タイプ:	AXIS
	値の範囲:	チャネル軸識別子

4.21 停止延長と退避(ESR)

<停止時間_1>, ... , <停止時間_n>:	故障発生後にドライブが実速度指令で移動を続ける時間 指定された軸に対して、ドライブパラメータ p0892 (タイマ) がドライブに書き込まれます。 p0892 = <停止時間>	
	単位:	s
	タイプ:	REAL
	値の範囲:	0.00 - 20.00
ファンクションコールで最大 5 軸までプログラム指令できます; n = 5		

4.21.2.2 ドライブ内蔵の後退の設定(ESRS)

ドライブ内蔵の ESR 機能の「後退」のためのドライブパラメータは、ESRR (...) 機能を使用して設定されます。

構文

ESRR (<軸_1>, <後退距離_1>, <後退速度_1> [, ... , <軸_n>, <後退距離_n>, <後退速度_n>])

意味

ESRR (...):	ESR 機能「後退」のためにドライブパラメータに書き込む機能 この機能は: • ブロック内では単独にします。 • 先読み停止をおこないます。 • シンクロナイズドアクションでは使用できません。	
<軸_1>, ... , <軸_n>:	ドライブ内蔵の後退を設定する軸 この軸に対して、ドライブパラメータ p0888 (設定)がドライブに書き込まれます。 p0888 = 2	
	タイプ:	AXIS
	値の範囲:	チャネル軸識別子

4.22 プログラム実行時間/ワークカウンタ

<後退量_1>, ..., <後退量_n>:	ドライブの場合、後退量が後退速度に変換されます。指定された軸の場合、値はドライブパラメータ p0893 (速度)に書き込まれます。 $p0893 = (<後退量_n> \text{後退速度に変換})$	
	単位:	mm/min、inch/min、°/min(軸の単位による)
	タイプ:	REAL
	値の範囲:	MIN - MAX
<後退速度_1>, ..., <後退速度_n>:	ドライブの場合、後退速度が時間に変換されます。指定された軸の場合、値はドライブパラメータ p0892 (タイマ)に書き込まれます。 $p0892 = <後退量_n> / <後退速度_n>$	
	単位:	mm/min、inch/min、°/min(軸の単位による)
	タイプ:	REAL
	値の範囲:	0.00 - MAX
ファンクションコールで最大 5 軸までプログラム指令できます; n = 5		

4.22 プログラム実行時間/ワークカウンタ

プログラム実行時間とワークカウンタに関する情報が、工作機械のオペレータをサポートするために提供されています。

この情報は、NC プログラムまたは PLC プログラム、またはその両方で、システム変数として処理できます。また、この情報を操作画面に表示することもできます。

4.22.1 プログラム実行時間

「プログラム実行時間」機能には、技術的な処理を監視する内部 NC タイマが装備されています。この技術的な処理は、NC 別とチャンネル別システム変数を使用すれば、パートプログラムとシンクロナイズドアクション内に読み込むことができます。

実行時間計測の起動(\$AC_PROG_NET_TIME_TRIGGER)は、この機能で唯一の書き込み可能なシステム変数で、これを使用して、プログラム区間を選択して計測します。つまり、NC

4.22 プログラム実行時間ワークカウンタ

プログラムに\$AC_PROG_NET_TIME_TRIGGERを書き込んで、時間計測を再度有効化および無効化できます。

システム変数	意味	動作
NC 別		
\$AN_SETUP_TIME	初期値による、コントローラの最後の電源投入(「コールド再始動」)後の経過時間(分単位)です。 初期値により、コントローラの電源投入のたびに、自動的に「0」にリセットされます。	• 常に有効です
\$AN_POWERON_TIME	コントローラの最後の通常の電源投入(「ウォーム再始動」)後の経過時間(分単位)です。 コントローラの通常の電源投入のたびに、自動的に「0」にリセットされます。	
チャンネル別		
\$AC_OPERATING_TIME	自動モードの NC プログラムの合計実行時間(秒単位)です。 コントローラの電源投入のたびに、自動的に値が「0」にリセットされます。	• MD27860 により有効になります。 • 自動モードの場合のみ有効です。
\$AC_CYCLE_TIME	選択した NC プログラムの実行時間(秒単位)です。 新しい NC プログラムを起動するたびに、自動的に値が「0」にリセットされます。	
\$AC_CUTTING_TIME	処理時間(秒単位)です。 (1 軸以上が動作中)の軌跡軸の実行時間は、早送りが動作していないすべての NC プログラムで、NC スタートとエンドオブプログラム、または NC リセットの間に計測します。計測は、ドウェル時間が有効なときは中断されます。 初期値により、コントローラの電源投入のたびに、自動的に値が「0」にリセットされます。	

4.22 プログラム実行時間ワークカウンタ

システム変数	意味	動作
\$AC_ACT_PROG_NET_TIME	現在の NC プログラムの実際の正味実行時間(秒単位)です。 新しい NC プログラムが起動すると、自動的に「0」にリセットされます。	- 常に有効です - 自動モードの場合のみ有効です。
\$AC_OLD_PROG_NET_TIME	M30 で正常に終了した直後のプログラムの正味実行時間(秒単位)です。	
\$AC_OLD_PROG_NET_TIME_COUNT	\$AC_OLD_PROG_NET_TIME の変化 POWER ON 後の \$AC_OLD_PROG_NET_TIME_COUNT は「0」です。 コントローラが\$AC_OLD_PROG_NET_TIME に新しく書き込んだ場合は常に、 \$AC_OLD_PROG_NET_TIME_COUNT が増加します。	

4.22 プログラム実行時間 ワークカウンタ

システム変数	意味		動作
\$AC_PROG_NET_TIME_TRIGGER	実行時間計測の起動:		自動モードの場合のみ有効です。
	0	中立状態 起動は無効です。	
	1	終了 測定を停止し、\$ \$AC_ACT_PROG_NET_TIME から \$AC_OLD_PROG_NET_TIME に値をコピー します。\$AC_ACT_PROG_NET_TIME は 「0」に設定された後、引き続き実行されま す。	
	2	開始 測定を開始し、その際に \$AC_ACT_PROG_NET_TIME が「0」に設定 されます。\$AC_OLD_PROG_NET_TIME は 変更されません。	
	3	停止 計測を停止します。 \$AC_OLD_PROG_NET_TIME を変更せず、 \$AC_ACT_PROG_NET_TIME を、再開するま で一定に保ちます。	
	4	再開 測定が再開されます。つまり、前に停止し た測定が継続されます。 \$AC_ACT_PROG_NET_TIME は続行されま す。\$AC_OLD_PROG_NET_TIME は変更さ れません。	
POWER ON の結果、すべてのシステム変数が「0」にリセットされます。			

4.22 プログラム実行時間 ワークカウンタ

注記

工作機械メーカー

マシンデータ MD27860 \$MC_PROCESSTIMER_MODE を使用して、使用可能なタイマーを起動します。

一部の機能(GOTOS、オーバライド= 0%、動作中の試運転送り速度、プログラムテスト、ASUB、PROG_EVENT、…など)の有効時間の計測の特性は、マシンデータ MD27850 \$MC_PROG_NET_TIMER_MODE と MD27860 \$MC_PROCESSTIMER_MODE を使用して設定します。

詳細情報:機能マニュアル、基本機能

注記

ワーク加工の残り時間

次のタイマー値を使用して同じワークを連続で加工する場合、ワークの加工残り時間が特定できます。

- 最後に生産したワークの加工処理時間です(\$AC_OLD_PROG_NET_TIME を参照してください)。
- 現在の処理時間です(\$AC_ACT_PROG_NET_TIME を参照してください)。

操作画面には、現在の処理時間の他に、残り時間も表示されます。

注記

STOPRE の使用

システム変数\$AC_OLD_PROG_NET_TIME および\$AC_OLD_PROG_NET_TIME_COUNT は、先読み停止処理を自動的に実行しません。これは、パートプログラムで使用する時に、前回のプログラム実行時のシステム変数の値を使用する場合は問題はありません。ただし、実行時間計測のトリガ(\$AC_PROG_NET_TIME_TRIGGER)を書き込む頻度が非常に高く、その結果、\$AC_OLD_PROG_NET_TIME が頻繁に変更される場合は、パートプログラムで STOPRE を明示的に使用してください。

必要条件

• ブロック検索

プログラム実行時間は、ブロックサーチでは特定されません。

• REPOS

REPOS の処理時間を、現在の処理時間に加算します(\$AC_ACT_PROG_NET_TIME)。

例

例 1: 「mySubProgrammA」の時間測定

プログラムコード

```
...  
N50 DO $AC_PROG_NET_TIME_TRIGGER=2  
N60 FOR ii= 0 TO 300  
N70 mySubProgrammA  
N80 DO $AC_PROG_NET_TIME_TRIGGER=1  
N95 ENDFOR  
N97 mySubProgrammB  
N98 M30
```

プログラムが N80 行を処理したら、「mySubProgrammA」の正味実行時間が \$AC_OLD_PROG_NET_TIME 内に記録されます。

\$AC_OLD_PROG_NET_TIME の値は、次のようになります。

- M30 の後も保持されます。
- ループを通過するたびに更新されます。

例 2: 「mySubProgrammA」と「mySubProgrammC」の時間測定

プログラムコード

```
...  
N10 DO $AC_PROG_NET_TIME_TRIGGER=2  
N20 mySubProgrammA  
N30 DO $AC_PROG_NET_TIME_TRIGGER=3  
N40 mySubProgrammB  
N50 DO $AC_PROG_NET_TIME_TRIGGER=4  
N60 mySubProgrammC  
N70 DO $AC_PROG_NET_TIME_TRIGGER=1  
N80 mySubProgrammD  
N90 M30
```

4.22.2 ワークカウンタ

「ワークカウンタ」機能により、制御装置内部でさまざまなカウンタを特別に使用して、ワークを数えることができます。

4.22 プログラム実行時間 ワークカウンタ

このカウンタは、0～999 999 999 の値の範囲で読み取りと書き込みのアクセスができる、チャンネル別システム変数です。

システム変数	意味
\$AC_REQUIRED_PARTS	生産ワーク数(ワーク数の指令値) このカウンタで定義したワーク数に達すると、実ワークカウンタ数(\$AC_ACTUAL_PARTS)を「0」にリセットします。
\$AC_TOTAL_PARTS	完成したワークの総数(実際のワークの総数) このカウンタは、起動時から加工されるワークの総数を指定します。この値は、初期値により、制御装置の電源投入時にのみ、自動的に「0」にリセットされます。
\$AC_ACTUAL_PARTS	完成したワーク数(実際のワークの総数) このカウンタは、起動時から加工されるワークの総数を登録します。このカウンタは、\$AC_REQUIRED_PARTS>0 の場合に限り、必要なワーク数(\$AC_REQUIRED_PARTS)に達すると、自動的に「0」にリセットします。
\$AC_SPECIAL_PARTS	ユーザーが選択したワーク数 このカウンタで、ユーザー用のワークカウントを指定できます。指令ワーク数の値(\$AC_REQUIRED_PARTS)に達したときに出力されるアラームを定義できます。ユーザーが、ユーザ自身でカウンタをリセットしてください。

注記

初期値により、制御装置の電源投入時に、すべてのワークカウンタは「0」に設定され、その起動とは無関係に、読み取りと書き込みができます。

注記

チャンネル別マシンデータを使用すると、カウンタの起動タイミング、カウンタのリセットのタイミング、およびカウントアルゴリズムを制御できます。

注記**ユーザー定義の M 命令によるワークのカウント**

さまざまなワークカウンタのカウントパルスが、エンドオブプログラム(M2/M30)ではなく、ユーザー定義の M 命令で起動されるように、マシンデータを設定できます。

4.23 プログラムシミュレーション

4.23.1 機能

最新のパートプログラムがプログラムシミュレーションで完全に計算され、その結果がユーザーインターフェースにグラフで表示されます。プログラミングの結果を、機械軸を移動せずに確認することができます。間違っ​​てプログラム指令された加工ステップが早期に検出され、ワークが誤って加工されるのを防止できます。

シミュレーション NC

シミュレーションでは独自の NC インスタンス(シミュレーション NC)を使用します。そのため、シミュレーションを開始する前に、実際の NC をシミュレーション NC に合わせる必要があります。この調整により、すべての有効なマシンデータが NC から読み出され、シミュレーション NC に読み込まれます。NC とサイクルマシンデータは、有効なマシンデータに含まれています。

4.23.2 コンパイルサイクルありのプログラムシミュレーション

いくつかのコンパイルサイクル(CC)はプログラムシミュレーションでサポートされています。サポートされているコンパイルサイクルのマシンデータは、コントローラの電源投入後に調整されます。「シミュレーションの開始」による調整はおこなわれません。

注記

パートプログラムでは、CC 用の言語命令とサポートされていない CC のマシンデータは使用できません(「パートプログラム内の CC 命令」の節も参照)。

サポートされていない CC の特殊な移動(OEM 座標変換)は - 特定の状況下では - 正しく表示されません。

パートプログラム内の CC 命令

したがって、サポートされていないコンパイルサイクルのパートプログラム内の言語命令(OMA1 ... OMA5、OEMIP01/2、G810 ... G829、ユーザー自身の手順と機能)は、その結果アラームメッセージが発生し、個々の処理をおこなわずにシミュレーションがキャンセルされます。

対策:

パートプログラム(\$P_SIM query)で、欠落している CC 用の言語要素を個別に処理します。

4.24 その他の機能

例:

プログラムコード	コメント
N1 G01 X200 F500	
IF (1== \$P_SIM)	
N5 X300	; CC がシミュレーションに対して有効ではありません。
ELSE	
N5 X300 OMA1=10	
ENDIF	

4.24 その他の機能

4.24.1 マシンデータの起動(NEWCONF)

NEWCONF 命令はすべてのマシンデータを有効化します。この機能は、HMI 操作画面で「マシンデータ有効化」ソフトキーを押して有効にすることもできます。

「NEWCONF」機能を実行すると、自動的に先読み停止が実行されます。つまり、軌跡移動が中断されます。

構文

NEWCONF

意味

NEWCONF:	「NEW_CONFIG」有効レベルのすべてのマシンデータを有効にする命令
----------	--------------------------------------

パートプログラムからの、複数チャンネルにわたる NEWCONF の実行

パートプログラムから軸のマシンデータが変更され、NEWCONF で起動された場合は、パートプログラムチャンネルに影響する変更を含むマシンデータのみが NEWCONF で有効になります。

注記

すべての変更が確実に適用されるよう、NEWCONF 命令を、マシンデータの変更が影響する軸、または機能が計算される、すべてのチャンネルで実行してください。

NEWCONF は軸マシンデータには影響しません。

PLC 制御軸には軸 RESET をおこなってください。

例

フライス加工:さまざまなテクノロジーによるドリル位置決め加工

プログラムコード	コメント
N10 \$MA_CONTOUR_TOL[AX]=1.0	;マシンデータを変更します。
N20 NEWCONF	;マシンデータを有効にします。
...	

4.24.2 現在の NC 言語の適用範囲の確認(Stringis)

機能「Stringis(...)」を使用すると、指定した文字列が、実際の言語の適用範囲で NC プログラミング言語の要素として使用可能かどうかを確認できます。

定義

INT Stringis (String <名称>)

構文

Stringis (<名称>)

4.24 その他の機能

意味

STRINGIS	戻り値を伴う機能
<名称>:	確認する NC プログラミング言語要素の名称
戻り値:	戻り値の書式は yxx (10 進数)です。

NC プログラミング言語の要素

確認できる NC プログラミング言語の要素は以下のとおりです。

- 既存のすべての G 機能グループの G 命令、例:G0、INVCW、POLY、ROT、KONT、SOFT、CUT2D、CDON、RMBBL、SPATH
- DIN または NC のアドレス、ADIS、RNDM、SPN、SR、MEAS など
- 関数、例:TANG(...)または GETMDACT
- 処理、例:SBLOF
- キーワード、例:ACN、DEFINE、SETMS
- システムデータ(マシンデータ\$M...、セッティングデータ\$S...、オプションデータ\$O... など)
- システム変数\$A...、\$V...、\$P...
- 算術変数 R...
- 有効なサイクルのサイクル名称
- GUD 変数と LUD 変数
- マクロ名称
- ラベル名称

戻り値

戻り値の先頭の桁のみが有効です。戻り値の書式は yxx で、y =基本情報、および xx =詳細情報です。

戻り値	意味
000	'名称'文字列は、このシステムでは不明な文字列です ¹⁾
100	'名称'文字列は、NC プログラミング言語の要素ですが、現在はプログラミングできません(オプション/機能が無効です)
2xx	'名称'文字列は、NC プログラミング言語のプログラム指令可能な要素です(オプション/機能が有効です)。詳細情報 xx には、以下の要素タイプに関する詳細情報が含まれます。

戻り値	意味	
	xx	意味
	01	DIN アドレスまたは NC アドレス ²⁾
	02	G 命令(G04、INVCW など)
	03	戻り値を伴う機能
	04	戻り値を伴わない機能
	05	キーワード、例: DEFINE
	06	マシンデータ(\$M...)、セッティングデータ(\$S...)、またはオプションデータ(\$O...)
	07	システム変数(\$...)や算術変数(R...)などのシステムパラメータ
	08	サイクル(サイクルを NC にロードし、サイクルプログラムを起動してください ³⁾)
	09	GUD 変数(GUD 変数を GUD 定義ファイルで定義して、GUD 変数を有効にしてください)
	10	マクロ名称(マクロをマクロ定義ファイルで定義して、マクロを起動してください) ⁴⁾
	11	現在のパートプログラムの LUD 変数
	12	ISO G 命令(ISO 言語モードを有効にしてください)
400	'名称'文字列は NC アドレスです。ただし、その名称は xx == 01 でも xx == 10 でもなく、また、G でも R でもありません ²⁾	
y00	何も割り当てることはできません	

1)コントローラによっては、特定の状況の場合に、例えば SINUMERIK 802D sl などの場合、当社の NC 言語命令のサブセットのみが認識されます。このようなコントローラの場合は、主要な当社 NC 言語命令に対して、値 0 が返されます。この動作は、MD10711 \$MN_NC_LANGUAGE_CONFIGURATION を使用して変更できます。MD10711 = 1 の場合は、当社 NC 言語命令に対して、常に値 100 が返されます。

2)NC アドレスに使用できる文字は、A、B、C、E、I、J、K、Q、U、V、W、X、Y、Z です。これらの NC アドレスは、アドレス拡張子を付けてプログラミングすることもできます。アドレス拡張子は、STRINGIS で確認するときに指定できます。例:201 == STRINGIS("A1").

文字 D、F、H、L、M、N、O、P、S、T は NC アドレス、またはユーザー定義の補助機能です。これらに対しては、常に値 400 が返されます。例:400 == STRINGIS("D").STRINGIS で確認するときは、これらの NC アドレスにアドレス拡張子を付けて指定することはできません。

例:000 == STRINGIS("M02")、ただし、400 == STRINGIS("M").

3)サイクルパラメータの名称は、STRINGIS では確認できません。

4)マクロとして定義されたアドレス。G、H、M、L はマクロと見なされます。

4.24 その他の機能

例

以下の例では、文字列として指定した NC 言語要素(特に注記がない限り)を、原則としてコントローラにプログラミングできます。

1. 文字列「T」を補助機能として定義する場合:
400 == STRINGIS("T")
000 == STRINGIS("T3")
2. 文字列「X」を軸として定義する場合:
201 == STRINGIS("X")
201 == STRINGIS("X1")
3. 文字列「A2」を拡張子付きアドレスとして定義する場合:
201 == STRINGIS("A")
201 == STRINGIS("A2")
4. 文字列「INVCW」を名称付き G 命令として定義する場合:
202 == STRINGIS("INVCW")
5. 文字列「\$MC_GCODE_RESET_VALUES」をマシンデータとして定義する場合:
206 == STRINGIS("\$MC_GCODE_RESET_VALUES")
6. 文字列「GETMDACT」が NC 言語機能である場合:
203 == STRINGIS("GETMDACT")
7. 文字列「DEFINE」がキーワードの場合:
205 == STRINGIS("DEFINE")
8. 文字列「\$TC_DP3」がシステムパラメータ(工具長成分)である場合:
207 == STRINGIS("\$TC_DP3")
9. 文字列「\$TC_TP4」がシステムパラメータ(工具サイズ)である場合:
207 == STRINGIS("\$TC_TP4")
10. 文字列「\$TC_MPP4」がシステムパラメータ(マガジンロケーションの状態)である場合:
 - 工具マガジン管理が動作中の場合: 207 == STRINGIS("\$TC_MPP4") ;
 - 工具マガジン管理が動作中でない場合: 000 == STRINGIS("\$TC_MPP4")また、下記の「工具マガジン管理」の節も参照してください。
11. 文字列「MACHINERY_NAME」を GUD 変数として定義する場合:
209 == STRINGIS("MACHINERY_NAME")
12. 文字列「LONGMACRO」をマクロとして定義する場合:
210 == STRINGIS("LONGMACRO")
13. 文字列「MYVAR」を LUD 変数として定義する場合:
211 == STRINGIS("MYVAR")
14. 文字列「XYZ」が、NC では不明な命令、GUD 変数、マクロ名称、またはサイクル名称である場合:
000 == STRINGIS("XYZ")

工具マガジン管理

工具マガジン管理機能が動作中でない場合は、以下のマシンデータとは無関係に、工具マガジン管理のシステムパラメータが STRINGIS で指定されます。

- MD10711 \$MN_NC_LANGUAGE_CONFIGURATION

値は常に 000 となります。

ISO モード

「ISO モード」機能が動作中の場合は、次のようになります。

- MD18800 \$MN_MM_EXTERN_LANGUAGE (適用、外部 NC 言語)
- MD10880 \$MN_MM_EXTERN_CNC_SYSTEM (設定するコントロールシステム)

STRINGIS は、指定した文字列を最初に SINUMERIK G 命令として確認します。この文字列が SINUMERIK G 命令でない場合は、次に、ISO G 命令として確認します。

プログラム指令切り換え(G290 (SINUMERIK モード)、G291 (ISO モード))は、STRINGIS には影響しません。

例

機能 STRINGIS(...)に有効なマシンデータの値は、次のとおりです。

- MD10711 \$MN_NC_LANGUAGE_CONFIGURATION = 2 (オプションを設定した NC 言語命令のみを、認識します)
- MD19410 \$ON_TRAFO_TYPE_MASK = 'H0' (オプション:座標変換)
- MD10700 \$MN_PREPROCESSING_LEVEL='H43' (サイクルの先読みが有効)

次のサンプルプログラムが、エラーメッセージを表示せずに実行されます。

プログラムコード	コメント
N1 R1=STRINGIS("TRACYL")	; R1 == 0、座標変換機能がないため、TRACYL が「不明」と見なされるためです。 オプション
N2 IF STRINGIS("TRACYL") ==204	
N3 TRACYL(1,2,3)	; N3 をスキップします
N4 ELSE	
N5 G00	; また、代わりに N5 を実行します
N6 ENDIF	
N7 M30	

4.24 その他の機能

4.24.3 対話形式によるパートプログラムからのウィンドウの呼び出し(MMC)

NC プログラムのユーザー用ダイアログは、予約サブプログラム MMC(...)を使用して、ユーザーインタフェースに表示できます。

次のタイプのダイアログを設定できます。

- Run MyScreens
- Easy XML
- User XML

詳細情報:

- プログラミングマニュアル Run MyScreens
- プログラミングマニュアル Easy XML

構文

MMC ("<ADDRESS>,<COMMAND>,<FILE>,<DIALOG>", "<QUIT>")

意味

MMC (...):	サブプログラム識別子 パラメータは位置コードを使用して、命令文字列と応答文字列という 2 つのコンマ区切り文字列で指定されます。	
コマンド内のパラメータ:		
<ADDRESS>:	設定したユーザー対話ボックスを実行する操作エリア	
	機能	操作エリア
	「Run MyScreens」 ユーザーダイアログ	CYCLES
	「Easy XML」 ユーザーダイアログ	CYCLES
	User XML	XML
	ポップアップウィンドウ 「Run MyScreens」	POPUPDLG
	ポップアップウィンドウ 「EasyXML」	POPUPDLG

<COMMAND>:	実行する命令	
	機能	命令
	「Run MyScreens」 ユーザーダイアログ	PICTURE_ON、 PICTURE_OFF
	「Easy XML」 ユーザーダイアログ	PICTURE_ON、 PICTURE_OFF
	User XML	XML_ON、XML_OFF
	ポップアップウィンドウ 「Run MyScreens」	PICTURE_ON、 PICTURE_OFF
	ポップアップウィンドウ 「EasyXML」	PICTURE_ON、 PICTURE_OFF
<FILE>:	表示するダイアログをプログラムするファイルの名前	
	機能	ファイル
	「Run MyScreens」 ユーザーダイアログ	<name>.com
	「Easy XML」 ユーザーダイアログ	<name>.xml
	User XML	<name>.xml
	ポップアップウィンドウ 「Run MyScreens」	<name>.com
	ポップアップウィンドウ 「EasyXML」	<name>.xml
<DIALOG>:	表示する対話画面の名前	
	機能	ダイアログの名前
	NC プログラム内に直接設定するポップアップウィンドウ 「Easy XML」を除くすべての機能	<FILE>ファイルで設定するダイアログの名前
	NC プログラム内に直接設定するポップアップウィンドウ 「Easy XML」 (例 3 を参照)	main
応答文字列内のパラメータ:		

4.24 その他の機能

<QUIT>:	確認タイプ	
	N:	応答なし 命令の送信が終わると、プログラムの実行を続行します。命令を正常に実行できない場合でも、フィードバックはありません。 注 NC プログラム内に表示時間(ドウェル時間)がプログラムされている場合、応答タイプ「N」を使用してください(後述の例 2 を参照)
	A:	非同期応答 命令の発行後にプログラムの実行を続行します。戻り値はユーザー固有の応答変数(GUD 変数)に保存されます。これは、ダイアログ設定の範囲内で定義され、NC プログラム内に読み込むことができます。

必要条件

- ダイアログの定義ファイル*.com は、「proj」フォルダに保存してください。
- ダイアログの Easy XML 定義ファイル*.xml は、「appl」フォルダに保存してください。定義ファイルが別のディレクトリに保存されている場合、パスを「appl」ディレクトリから始まる相対パスで指定してください。
- 異なるチャネルのユーザー定義ダイアログを、同時に表示することはできません。
- MMC 機能はシミュレーションでサポートされてません。

例

例 1

NC プログラム内でのダイアログの表示とユーザー操作への応答。

プログラムコード	コメント
; 応答変数 QUIT は、グローバルユーザー変数 (GUD) として作成済みです	
; ダイアログが設定されたときのタイプ STRING:	
; DEF NCK STRING[20] QUIT	
QUIT = "XXX"	; 応答変数の初期化
G4 F5	

プログラムコード	コメント
MMC("CYCLES,PICTURE_ON,test.com,test1","A")	; ダイアログの表示 ; - 操作エリア:CYCLES ; - 画像ステータス:PICTURE_ON (表示) ; - ダイアログ画面ファイル:test.com ; - ダイアログ画面:test1
INPUT:	; ユーザー入力を待機
STOPRE	; 先読み停止
IF MATCH (QUIT,"RUN") >= 0 GOTOF WORK	; ソフトキー「RUN」
IF MATCH (QUIT,"CHK") >= 0 GOTOF CHECK	; ソフトキー「CHK」
GOTOB INPUT	; => 待機
WORK:	; ソフトキー「RUN」が押された
MSG("Continue with processing -> NC start")	; 出力メッセージ
MMC("CYCLES,PICTURE_OFF","N")	; ダイアログを閉じる
M0	; NC スタートを待機
GOTOF END	; => プログラム終了
CHECK:	; ソフトキー「CHK」が押された
MSG("Approach position -> NC start")	; 出力メッセージ
MMC("CYCLES,PICTURE_OFF","N")	; ダイアログを閉じる
M0	; NC スタートを待機
GOTOF END	; => プログラム終了
END:	
...	

例 2

ダイアログの表示時間は、NC プログラムでドウェル時間などを使用して定義されます。

プログラムコード	コメント
F1000 G94	
...	
MMC("POPUPDLG,PICTURE_ON,xmldial_emb.xml,main","N")	; ダイアログの表示
X200	
Z40	
MMC("POPUPDLG,PICTURE_OFF","N")	; ダイアログを閉じる

例 3

NC プログラムへのポップアップスクリプトの埋め込みとその用法。

プログラムコード

```
PROC POPUP_TEST
```

4.24 その他の機能

プログラムコード

```
; ----- スクリプト -----  
; <main_dialog entry="rpara_main">  
;   <let name="xpos" />  
;   <let name="ypos" />  
;   <let name="field_name" type="string" />  
;   <let name="num" />  
;   <menu name="rpara_main">  
;     <open_form name="rpara_form"/>  
;     <softkey_back>  
;       <close_form />  
;     </softkey_back>  
;   </menu>  
;   <form name="rpara_form">  
;     <init>  
;       <caption>NC パートプログラムからのマスク</caption>  
;       <let name="count" >0</let>  
;       <op>  
;         xpos = 120;  
;         ypos = 34;  
;         "nck/Channel/Parameter/R[10]" = 10;  
;       </op>  
;       <!-- load the number of controls -->  
;       <op>  
;         num = "nck/Channel/Parameter/R[10]";  
;       </op>  
;       <while>  
;         <condition> count < num</condition>  
;         <print name="field_name" text="edit%d">count</print>  
;         <op>  
;           ypos = ypos + 24;  
;           count = count + 1;  
;         </op>  
;       </while>  
;     </init>  
;     <paint>  
;       <op>  
;         xpos = 8;  
;         ypos = 36;  
;         count = 0;  
;       </op>  
;     <while>
```

プログラムコード

```

;          <condition>count < num</condition>
;          <print name="field_name" text="R-Parameter%d">count</print>
;          <text xpos = "$xpos" ypos = "$ypos" >$$$field_name</text>
;          <op>
;              ypos = ypos + 24;
;              count = count + 1;
;          </op>
;      </while>
;  </paint>
;  </form>
; </main_dialog>
; ===== プログラムセクション =====
...
G94 F100
MMC ("POPUDDL, PICTURE_ON, xml_dial_emb.xml, main", "N")
G4 F4
X200
MMC ("POPUDDL, PICTURE_OFF", "N")
G4 F2
X0
...

```

4.24.4 Process DataShare - 外部機器/ファイルへの出力(EXTOPEN、WRITE、EXTCLOSE)

パートプログラムから外部デバイス/ファイルへのデータの書き込みは、3つのステップで実行されます。

1. 外部機器/ファイルを開きます。
外部機器/ファイルが、EXTOPEN 命令を使用して書き込みをおこなうために、そのチャンネルに対して開かれます。
2. データの書き込み
出力データは、NC 言語の文字列機能、たとえば SPRINT を使用して処理できます。書き込みには WRITE 命令が使用されます。
3. 外部機器/ファイルを閉じます。
プログラムの終わりに達した場合(M30)、またはチャンネルのリセット時に、EXTCLOSE 命令を使用して外部機器/ファイルがもう一度解放されます。

構文

```
DEF INT <結果>
```

4.24 その他の機能

```
DEF STRING [<n>] <出力>
...
EXTOPEN (<結果>, <ExtDev>, <SyncMode>, <AccessMode>, <WriteMode>)
...
<Output>="データ出力"
WRITE (<結果>, <ExtDev>, <出力>)
...
EXTCLOSE (<結果>, <ExtDev>)
```

意味

EXTOPEN:	外部機器/ファイルを開くための予約手順命令です		
<結果>:	パラメータ 1:結果変数		
	結果変数値を使用して、操作が正常におこなわれたかどうかをプログラムで評価し、処理を適切に続行することができます。		
	タイプ:	INT	
	値:	0	エラーなし
		1	外部機器を開けません。
		2	外部機器が設定されていません。
		3	無効なパスを持つ外部機器が設定されています。
		4	外部機器のアクセス権限がありません。
		5	使用モード:外部機器がすでに「占有」されています。
		6	使用モード:外部機器がすでに「共有」されています。
		7	フィールド長が LOCAL_DRIVE_MAX_FILESIZE を超えています。
		8	外部機器の最大数を超えています。
		9	LOCAL_DRIVE のオプションが設定されていません。
		11	予備
		12	書き込みモード:データが extdev.ini と矛盾しています。
		16	無効な外部パスがプログラム指令されています。
		22	外部機器が搭載されていません。

<ExtDev>:	パラメータ 2:開こうとしている外部機器/ファイルのシンボルの識別子	
	タイプ:	STRING
	シンボル識別子は以下で構成されます。 1. 論理機器名称 2. 必要に応じて、その後にファイルパスが続きます(「/」を使用して付加します)。 以下の 論理機器名称 が定義されています。	
	"LOCAL_DRIVE":	ローカル CF カード(定義済み)
	"CYC_DRIVE":	SIEMENS サイクルで使用するために予約されたドライブ名称(定義済み)
	"/dev/ext/1",... "/dev/ext/9":	使用可能なネットワークドライブ 注: extdev.ini ファイルで設定する必要があります。
	"/dev/cyc/1", "/dev/cyc/2":	SIEMENS サイクルで使用するために予約されているドライブ名称 注: extdev.ini ファイルで設定する必要があります。
	ファイルパス: "LOCAL_DRIVE"や"CYC_DRIVE"などに対してファイルパスを指定してください。 "LOCAL_DRIVE/my_dir/my_file.txt" - 論理機器名称"/dev/ext/1...9"と"/dev/cyc/1...2"を以下のように設定できます。 - すでに参照されたファイルに対して設定する場合は、論理機器名称だけを指定できます。例: "/dev/ext/4" - ディレクトリの参照用に設定する場合は、ファイルパスを指定してください。例: "/dev/ext/5/my_dir/my_file.txt"	
	注: 論理機器名称"/dev/ext/1...9"、"/dev/v24"、"/dev/cyc/1...2"の場合、大文字/小文字は無視されます。大文字/小文字は、ファイルへのパスを指定する場合に意味を持ちます。"LOCAL_DRIVE"と"CYC_DRIVE"では、大文字だけが使用できます。	

4.24 その他の機能

<SyncMode>:	パラメータ 3:当該機器/ファイルへの WRITE 命令の処理モード		
	タイプ:	STRING	
	値:	"SYN":	同期書き込み 書き込み操作が終了するまでプログラムの実行が停止します。 同期書き込み操作が正常に完了したかどうかは、WRITE 命令のエラー変数を評価することで確認できます。
		"ASYN":	非同期書き込み プログラムの実行は WRITE 命令で中断されません。 注: このモードでは、WRITE 命令の結果変数で情報は提供されず、結果変数は常に値 0(エラーなし)です。この特別なモードでは、WRITE 命令が正常に実行されたかどうかは確認できません。
<AccessMode>:	パラメータ 4:当該機器/ファイルの使用モード		
	タイプ:	STRING	
	値:	"SHARED":	機器/ファイルが「共有」モードで要求されます。このモードでは、他のチャネルも機器を使用することができます。つまり、機器を開くことができます。
		"EXCL":	機器/ファイルが当該チャネルだけで使用されます。他のチャネルは機器を使用できません。

<WriteMode>:	パラメータ 5:当該ファイル/機器への WRITE 命令の書き込みモード(任意選択)	
	タイプ:	STRING
	値:	"APP": 付加 ファイルの内容は常に保持されます。書き込み呼び出しは最後に付加されます。
		「OVR」 : 上書き ファイルの内容は削除され、その後の書き込み呼び出しを使用して再生成されます。
注: extdev.ini ファイルで設定された書き込みモードは、このパラメータを使用して上書き できません。干渉が発生した場合、EXTOPEN 呼び出しがエラーで通知されます。		

WRITE:	出力データを書き込むための予約手順
--------	-------------------

EXTCLOSE:	開かれている外部機器/ファイルを閉じるための予約手順です。		
<結果>:	パラメータ 1:結果変数		
	タイプ:	INT	
	値:	0	エラーなし
		16	無効な外部パスがプログラム指令されています。
		21	外部機器を閉じるときにエラーが発生しました。
<ExtDev>:	パラメータ 2: 閉じようとしている外部機器/ファイルのシンボル識別子。EXTOPEN を参照してください。 注: 識別子は EXTOPEN 呼び出しで指定した識別子と同一のものにしてください。		

例

プログラムコード	
N10	DEF INT RESULT
N20	DEF BOOL EXTDEVICE
N30	DEF STRING[80] OUTPUT

4.24 その他の機能

プログラムコード

```
N40      DEF INT PHASE
N50      EXTOPEN (RESULT, "LOCAL_DRIVE/my_file.txt", "SYN", "SHARED")
N60      IF RESULT > 0
N70          MSG("error for EXTOPEN:" << RESULT)
N80      ELSE
N90          EXTDEVICE=TRUE
N100     ENDIF
...
N200     PHASE=4
N210     IF EXTDEVICE
N220         OUTPUT=SPRINT("End phase: %D", PHASE)
N230         WRITE (RESULT, "LOCAL_DRIVE/my_file.txt", OUTPUT)
N240     ENDIF
...
```

下記も参照

文字列演算子 (ページ 503)

ファイルの書き込み(WRITE) (ページ 644)

4.24.5 アラーム(SETAL)

アラームは、NC プログラムで設定できます。アラームは、操作画面で個別のフィールドに表示されます。アラームは常に、アラームのカテゴリに従い、コントローラからの応答と連動します。

構文

SETAL (<番号> [, <文字列>])

意味

SETAL (...) :	あらかじめ設定されているアラーム設定手順 SETAL は、個別の NC ブロックでプログラム指令してください。		
<番号>:	アラーム番号		
	データタイプ:	INT	
	値の範囲:	60000 ... 64999 (予約済み)	SIEMENS サイクルのアラーム
		65000 ... 69999	ユーザーサイクルのアラーム
<文字列>:	文字列(オプション)		
	データタイプ:	STRING	
	ユーザーサイクルアラームのプログラム指令時には、4 個までのパラメータを含む文字列を指定できます。 これらのパラメータには可変ユーザーテキストを定義できます。 事前に定義された以下のパラメータを使用できます。		
	%1	チャンネル番号	
	%2	ブロック番号、ラベル	
	%3	サイクルアラームのテキストインデックス	
	%4	追加のアラームパラメータ	

注記

アラームテキストは、操作画面で設定してください。

注記

操作画面で有効になっている言語でメッセージを出力する場合、ユーザーは現在 HMI で設定されている言語についての情報が必要です。この情報は、システム変数 \$AN_LANGUAGE_ON_HMI (ページ 1582) を使用してパートプログラムとシンクロナイズドアクションで要求することができます。

例

プログラムコード	コメント
...	

4.24 その他の機能

プログラムコード	コメント
N100 SETAL (65000)	; アラーム番号 65000 設定
...	

詳細情報

アラーム応答と確認

ユーザーサイクルアラームは、アラーム応答と確認に関して異なった番号範囲に割り当てられます。

数値範囲	アラーム応答	アラーム確認
65000 - 65499	表示、NC スタート禁止	リセット
65500 - 65999	表示、NC スタート禁止(設定した MD20194 用の ASUB を除く)	リセット
66000 - 66999	表示、NC スタート禁止、事前にデコードされたブロックの実行後に移動が停止状態	リセット
67000 - 67999	ディスプレイ	キャンセル
68000 - 68999	表示、NC スタート禁止、補間器が即時停止	リセット
69000 - 69999	表示、NC スタート禁止、次のブロックの終点で停止	リセット

4.24.6 Define blank (WORKPIECE)

グラフィカルシミュレーションでブランクの形状とサイズを表示するには、コントローラがそれらを認識している必要があります。したがって、ユーザーはユーザーインターフェース経由で、または NC プログラム内で直接、ブランクを定義できます。ブランクの定義は(プログラム終了/チャンネル/モードグループ)のリセット後も維持されます。これらの定義は、コントローラの次の電源投入時に自動的に削除されます。

構文

```
WORKPIECE("<WP>", "<RefP>", "<ZeroOffset>", "<Type>", <Par5>,
<Par6>, ..., <Par12>)
```

意味

WORKPIECE (...):		ブランクを定義するための予約処理	
		先読み停止:	あり
		単独ブロック 指令:	あり
パラメータ:			
1	"<WP>":	ワークの名前(オプション)	
		データタイプ:	STRING
		指定する必要があるのは、1つのチャンネルに複数のワークが存在する可能性がある場合のみです。指定しない場合、「WORKP<n>」で自動的に確定します。ここで、<n>は宣言するチャンネルの数です。	
2	"<RefP>":	クランプ(オプション、フライス盤のみ)	
		データタイプ:	STRING
		値の範囲:	"Table" 固定テーブルのクランプ
			"A" 回転軸 A にクランプ
			"B" 回転軸 B にクランプ
			"C" 回転軸 C にクランプ
		必要条件: ブランクのクランプ用マシンデータを使用して、テーブルまたは回転軸を有効にする必要があります。(『SINUMERIK Operate 試運転マニュアル』を参照)。	
3	"<ZeroOffset>":	ブランクを位置決めするために設定可能なゼロオフセット(プログラム不可能) ブランクを位置決めするために設定可能なゼロオフセットの選択は、ユーザーインタフェースを使用したブランク入力でのみ可能です。パートプログラムでブランクを直接定義する場合、ブランクは常に現在有効なゼロオフセットに関連します。	

4.24 その他の機能

4	"<Type>":	素材形状		
		データタイプ:	STRING	
		値の範囲:	"CYLINDER":	円筒
			"PIPE":	風管
			"RECTANGLE":	中心のある直方体
			:	
			"BOX":	直方体
	"N_CORNER":	n 辺の多角形		
5 ... 12	<Par5> ... <Par12>:	ブランクの形状を説明するためのパラメータ		
		データタイプ:	REAL	
		必要なパラメータ数とその意味は、対応するブランクの形状とビットパラメータの値によって異なります。 参照: • 「ブランクの形状を説明するためのパラメータ」表 • 「ビットパラメータ」表		
WORPIECE ():		パラメータなしの WORKPIECE 呼び出しは、すべてのブランク定義を削除します。		
WORPIECE (<WP>):		ワーク名のための WORKPIECE 呼び出しは、そのブランク定義のみを削除します。		

表 4-6 ブランクの形状を説明するためのパラメータ

素材形状	パラメータ							
	<Par5>	<Par6>	<Par7>	<Par8>	<Par9>	<Par10>	<Par11>	<Par12>
円筒	ビットパラメータ ビットコード 整数値として解釈される実数値。ビットは次のパラメータの意味を定義します （「ビットパラメータ」表を参照）。	基準点 Z_0	長さ Z_1	加工寸法 Z_B	外径 d_0	-	回転軸を中心とする回転	-
風管		基準点 Z_0	長さ Z_1	加工寸法 Z_B	外径 d_0	壁厚 (inc)/内径 d_1 (abs)	回転軸を中心とする回転	-
中心のある直方体		基準点 Z_0	長さ Z_1	加工寸法 Z_B	幅 W	長さ L	回転軸を中心とする回転	-
直方体		基準点 Z_0	長さ Z_1	加工寸法 Z_B	X_0	Y_0	X_1	Y_1
n 辺の多角形		基準点 Z_0	長さ Z_1	加工寸法 Z_B	頂点の数	二面幅	回転軸を中心とする回転	-

表 4-7 ビットパラメータ

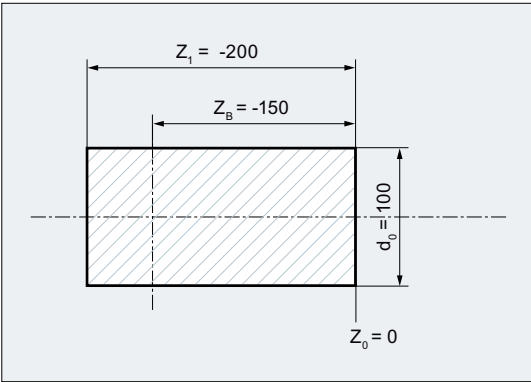
ビット	意味	
4 (0x0010)	直方体: X_1	
	= 0	inc
	= 1	abs
5 (0x0020)	直方体: Y_1	
	= 0	inc
	= 1	abs
6 (0x0040)	長さ Z_1 (最終寸法)	
	= 0	inc
	= 1	abs
ビット 7 (0x0080)	加工寸法 Z_B	
	= 0	inc
	= 1	abs

4.24 その他の機能

ビット	意味	
ビット 8 (0x0100)	風管:壁厚/内径	
	= 0	inc
	= 1	abs
9 (0x0200)	n 辺の多角形	
	= 0	二面幅
	= 1	辺の長さ
12 (0x1000)	旋盤のクランプ	
	= 0	主軸
	= 1	対向主軸
13 (0x2000)	対向主軸	
	= 0	ミラーリング付き
	= 1	ミラーリングなし

例

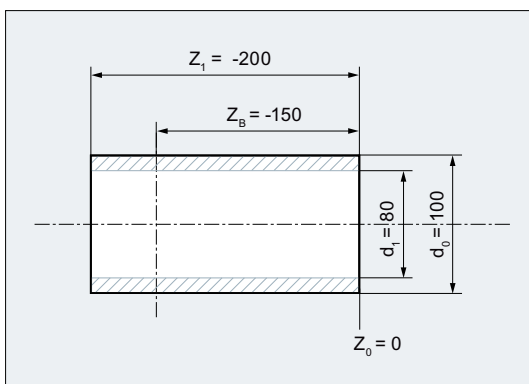
例 1:旋盤上の円筒状ブランク



プログラムコード	コメント
...	

プログラムコード	コメント
WORKPIECE(,,, "CYLINDER", 0, 0, -200, -150, 100)	; ブランク定義:
...	; ブランク形状:円筒
	; ビットパラメータ=0 (ビット設定なし) →
	長さ&加工寸法の値はインクリメンタル、主
	軸上のブランク
	; 基準点 (Z0)=0
	; 長さ (Z1)=-200
	; 加工寸法 (ZB)=-150
	; 外径 (d0)=100

例 2 :旋盤上の風管状ブランク



プログラムコード	コメント
...	
WORKPIECE(,,, "PIPE", 256, 0, -200, -150, 100, 80)	; ブランク定義:
	; ブランク形状:風管
	; ビットパラメータ=256 (Bit8=1) →内径
	は絶対値; 長さ&加工寸法はインクリメンタ
	ル、主軸上のブランク
	; 基準点 (Z0)=0
	; 長さ (Z1)=-200
	; 加工寸法 (ZB)=-150
	; 外径 (d0)=100
	; 内径 (d1)=80
...	

4.24 その他の機能

4.24.7 言語切り替えモード(G290, G291)

このコントローラを使用すると、外部 CNC システムからパートプログラムを読み込んで処理できます。対応する NC 言語モード(ISO 系言語)がセットアップ時に定義されていることが前提条件です。

ISO 系言語モードは、チャンネルごとに別々に有効化できます。たとえば、チャンネル 1 を ISO 系言語モードで実行し、チャンネル 2 を SINUMERIK モードで有効化することができます。

SINUMERIK モードと ISO 系言語モードの切り替えは、NC プログラムで G グループ 47 の命令を使用しておこなわれます。使用中の工具、工具補正、ワークオフセットはモード切り替えの影響を受けません。

構文

G291
...
G290

意味

G290:	SINUMERIK 言語モードの起動	
	単独ブロック指令:	あり
	効果:	モーダル
G291:	ISO 言語モードの起動	
	単独ブロック指令:	あり
	効果:	モーダル

条件

SINUMERIK モード

- G 命令の規定値はマシンデータを使用してチャンネルごとに定義できます。
- ISO 系言語の言語指令を SINUMERIK モードでプログラムすることはできません。

ISO 系言語モード

- コントロールシステムの初期設定として、マシンデータで ISO 系言語モードを設定できます。初期設定では、その後、ISO 系言語モードでコントロールシステムが再起動されます。
- ISO 系言語の G 命令だけがプログラム指令できます。SINUMERIK G 機能は ISO 系言語モードではプログラム指令できません。
- ISO 系言語と SINUMERIK 言語を同一 NC ブロックで混在させることはできません。
- G 指令で ISO 系言語 M(フライス削り)と ISO 系言語 T(旋盤)の切り替えはできません。
- SINUMERIK モードでプログラムされたサブプログラムを呼び出すことができます。
- SINUMERIK 機能を使用する場合は、まず SINUMERIK モードに切り替える必要があります(例を参照)。

例**ISO 系言語モードでの直線ブロックの圧縮**

プログラムコード	コメント
N5 G290	; SINUMERIK 言語モードの起動。
N10 COMPCAD	; COMPCAD は Siemens 言語のコマンドで、連続する直線ブロックを、可能な限り長い軌跡長の多項式ブロックで置き換えるコンプレッサ機能を有効化します。
N15 G291	; ISO 言語モードを起動します。
N20 G01 X100 Y100 F1000	; COMPCAD が SINUMERIK モードで有効化されているため、ISO 系言語モードの直線ブロックも、この機能を使用して圧縮できます。
...	

詳細情報

機能マニュアル G コード言語

4.25 ユーザー荒削りプログラム

4.25.1 荒削りの機能のサポート

事前にプログラム指令された切削プログラムを切削に使用できます。それ以外に、以下の機能を使用して、ユーザー専用の切削プログラムを作成できます。

- 輪郭テーブルの生成(CONTPRON)
- コード化された輪郭テーブルの生成(CONTDCON)
- 輪郭解析の解除(EXECUTE)
- 2 個の輪郭要素の交点の特定(INTERSEC)
(CONTPRON で生成したテーブルの場合のみ有効です)
- テーブルの輪郭要素をブロック毎に実行(EXECTAB)
(CONTPRON で生成したテーブルの場合のみ有効です)
- 円弧データの計算(CALCDAT)

注記

これらの機能は、切削だけでなく、汎用的に使用できます。

必要条件

CONTPRON または CONTDCON 機能を呼び出す前に、以下をおこなってください。

- 衝突が発生しないで加工が可能な起点に移動してください。
- G40 で、工具径補正を解除してください。

4.25.2 輪郭テーブルの生成(CONTPRON)

CONTPRON は輪郭解析を起動します。その後に呼び出される NC ブロックは実行されず、個々の移動に分割されて、輪郭テーブルに格納されます。各輪郭要素は、輪郭テーブルの 2 次元配列の 1 行に対応します。レリーフカットの数が返されます。

構文

輪郭解析の起動:

CONTPRON (<輪郭テーブル>, <加工タイプ>, <レリーフカット>, <加工方向>)

輪郭解析の解除と通常の実行モードへの復帰:

EXECUTE (<ERROR>)

「輪郭解析の解除(EXECUTE) (ページ 1194)」を参照してください。

意味

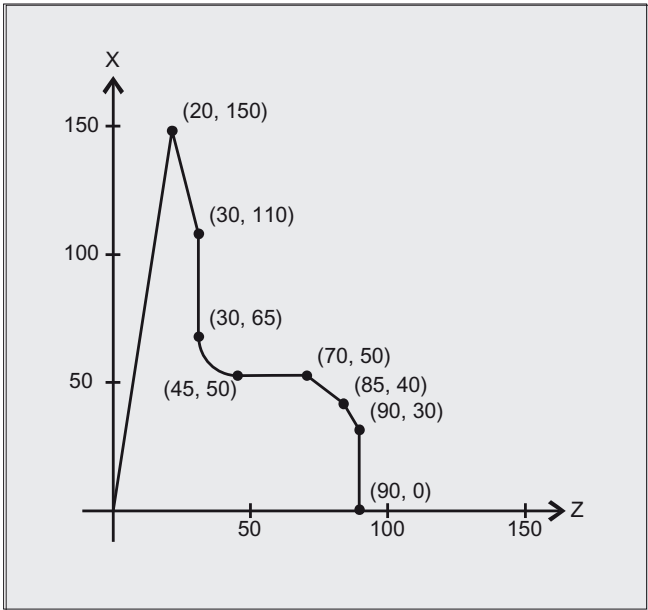
CONTPRON:	輪郭解析を起動して輪郭テーブルを生成する予約処理		
<輪郭テーブル>:	輪郭テーブルの名称		
<加工タイプ>:	加工タイプのパラメータ		
	タイプ:	CHAR	
	値:	"G":	長手方向の旋削:内径加工
		"L":	長手方向の旋削:外側加工
		"N":	正面旋削:内径加工
"P":		正面旋削:外側加工	
<レリーフカット>:	発生するレリーフカット要素数として得られる変数		
	タイプ:	INT	
<加工方向>:	加工方向のパラメータ		
	タイプ:	INT	
	値:	0	輪郭解析、前方(初期値)
		1	両方向の輪郭解析

例 1

以下で輪郭テーブルを生成します:

- 名称「KTAB」
- 最大 30 個の輪郭要素(円弧、直線)
- 発生するレリーフカット要素数を表わす 1 個の変数
- エラーメッセージを表わす 1 個の変数

4.25 ユーザー荒削りプログラム



NC プログラム:

プログラムコード	コメント
N10 DEF REAL KTAB[30,11]	; 最大 30 個の輪郭要素を持つ、KTAB という名称の輪郭テーブル、パラメータ値 11 (テーブルの列数) は固定数です。
N20 DEF INT ANZHINT	; ANZHINT という名称のレリーフカット要素の数を表わす変数。
N30 DEF INT ERROR	; エラーフィードバック信号を表わす変数 (0=エラーなし、1=エラーあり)。
N40 G18	
N50 CONTPRON(KTAB,"G",ANZHINT)	; 輪郭解析の起動
N60 G1 X150 Z20	; N60~N120:輪郭の記述
N70 X110 Z30	
N80 X50 RND=15	
N90 Z70	
N100 X40 Z85	
N110 X30 Z90	
N120 X0	
N130 EXECUTE (ERROR)	; 輪郭テーブルへの書き込みを終了し、通常のプログラム\モードに切り替えます。
N140 ...	; テーブルの処理を続行します。

輪郭テーブル KTAB

イン デッ クス 行	列									
(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
7	7	11	0	0	20	150	0	82.4053566 3	0	0
0	2	11	20	150	30	110	-1111	104.036243 5	0	0
1	3	11	30	110	30	65	0	90	0	0
2	4	13	30	65	45	50	0	180	45	65
3	5	11	45	50	70	50	0	0	0	0
4	6	11	70	50	85	40	0	146.309932 5	0	0
5	7	11	85	40	90	30	0	116.565051 2	0	0
6	0	11	90	30	90	0	0	90	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

列の内容の説明

(0) 次の輪郭要素(その列の行番号)へのポインタ

(1) 前の輪郭要素へのポインタ

(2) 輪郭モードの動作コード

X の指令値 = abc

a = 10² G90 = 0 G91 = 1b = 10¹ G70 = 0 G71 = 1c = 10⁰ G0 = 0 G1 = 1 G2 = 2 G3 = 3

(3), (4) 輪郭要素の始点

(3) =横座標、(4) =現在の平面上の縦座標

(5), (6) 輪郭要素の始点

(5) =横座標、(6) =現在の平面上の縦座標

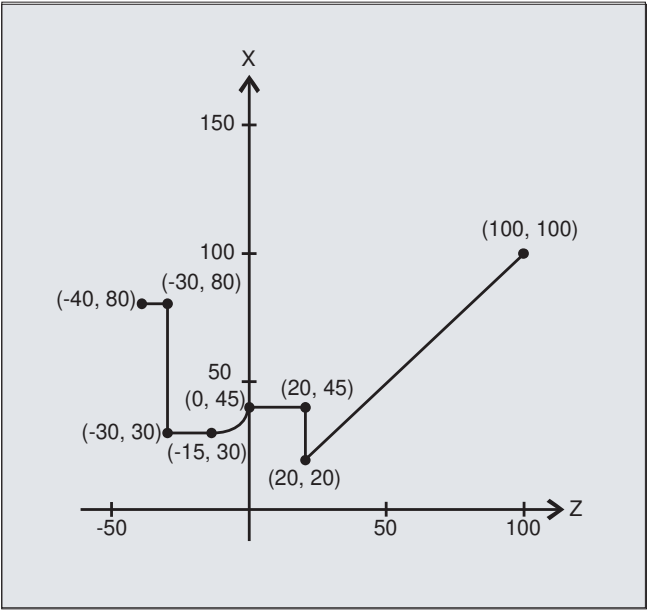
4.25 ユーザー荒削りプログラム

- (7) 最大/最小値:輪郭の部分の最大値と最小値を示します
- (8) 輪郭要素と、横座標(長手方向の加工の場合)、または縦座標(正面旋削の場合)の間の最大値。角度は、プログラム指令した加工のタイプによって異なります。
- (9), (10) 円弧ブロックの場合の、輪郭要素の中心点座標です。
(9) =横座標、(10) =縦座標

例 2

以下のような輪郭テーブルを生成します。

- 名称「KTAB」
- 最大 92 個の輪郭要素(円弧、直線)
- 運転モード:長手方向の旋削、外側加工
- 解析、前方と後方



NC プログラム

プログラムコード	コメント
N10 DEF REAL KTAB[92,11]	; 最大 92 個の輪郭要素を持つ、KTAB という名称の輪郭テーブルです、パラメータ値 11 は固定数です。
N20 DEF CHAR BT="L"	; CONTRON のモード:長手方向の旋削、外側加工
N30 DEF INT HE=0	; レリーフカット要素の数=0
N40 DEF INT MODE=1	; 解析、前方と後方
N50 DEF INT ERR=0	; エラーフィードバック信号

4.25 ユーザー荒削りプログラム

プログラムコード	コメント
...	
N100 G18 X100 Z100 F1000	
N105 CONTPRON (KTAB, BT, HE, MODE)	; 輪郭解析の起動
N110 G1 G90 Z20 X20	
N120 X45	
N130 Z0	
N140 G2 Z-15 X30 K=AC (-15) I=AC (45)	
N150 G1 Z-30	
N160 X80	
N170 Z-40	
N180 EXECUTE (ERR)	; 輪郭テーブルへの書き込みを終了し、通常のプログラム\モードに切り替えます。
...	

輪郭テーブル KTAB

輪郭解析が終了した後は、輪郭は両方向に有効です。

インデックス	列										
行	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
0	6 ¹⁾	7 ²⁾	11	100	100	20	20	0	45	0	0
1	0 ³⁾	2	11	20	20	20	45	-3	90	0	0
2	1	3	11	20	45	0	45	0	0	0	0
3	2	4	12	0	45	-15	30	5	90	-15	45
4	3	5	11	-15	30	-30	30	0	0	0	0
5	4	7	11	-30	30	-30	45	-1111	90	0	0
6	7	0 ⁴⁾	11	-30	80	-40	80	0	0	0	0
7	5	6	11	-30	45	-30	80	0	90	0	0
8	1 ⁵⁾	2 ⁶⁾	0	0	0	0	0	0	0	0	0
	...										
83	84	0 ⁷⁾	11	20	45	20	80	0	90	0	0
84	90	83	11	20	20	20	45	-1111	90	0	0
85	0 ⁸⁾	86	11	-40	80	-30	80	0	0	0	0
86	85	87	11	-30	80	-30	30	88	90	0	0
87	86	88	11	-30	30	-15	30	0	0	0	0
88	87	89	13	-15	30	0	45	-90	90	-15	45

4.25 ユーザー荒削りプログラム

89	88	90	11	0	45	20	45	0	0	0	0
90	89	84	11	20	45	20	20	84	90	0	0
91	83 ⁹⁾	85 ¹⁰⁾	11	20	20	100	100	0	45	0	0

各列の内容の説明と 0、1、6、8、83、85、および 91 行のコメント

例 1 の列の説明が適用されます。

テーブル 0 行では常に:

- 1) 先行ポインタ:n 行には輪郭終点が含まれます(前方へ)
- 2) 後続ポインタ:n 行は輪郭テーブルの終点です(前方へ)

輪郭要素内でそれぞれ 1 回前方へ:

- 3) 先行ポインタ:輪郭始点(前方へ)
- 4) 後続ポインタ:輪郭終点(前方へ)

常に輪郭テーブル終点(前方へ) +1 行:

- 5) 先行ポインタ:レリーフカットの数(前方へ)
- 6) 後続ポインタ:レリーフカットの数(後方へ)

輪郭要素内でそれぞれ 1 回後方へ:

- 7) 後続ポインタ:輪郭終点(後方へ)
- 8) 先行ポインタ:輪郭始点(後方へ)

常にテーブルの最終行:

- 9) 先行ポインタ:n 行は輪郭テーブルの始点です(後方へ)
- 10) 後続ポインタ:n 行には輪郭始点が含まれます(後方へ)

詳細情報

使用可能な移動指令、座標系

輪郭のプログラミングには、次の G 命令を使用できます。

- G グループ 1:G0、G1、G2、G3

さらに、以下を実行できます。

- 丸み付けと面取り
- CIP と CT による円弧プログラミング

スプライン、多項式、およびねじの各機能を実行すると、エラーが発生します。

CONTPRON と EXECUTE の間でフレームを起動して、座標系を変更することはできません。これは、G70 と G71、または G700 と G710 の間での変更でも同様です。

輪郭テーブルの解析中に、GEOAX でジオメトリ軸を置換すると、アラームが発生します。

レリーフカット要素

サブプログラムまたは個々のブロックで、個々のレリーフカット要素について輪郭を記述できます。

プログラム指令した輪郭方向に影響されない荒削り

CONTPRON による輪郭解析は、プログラム指令方向にかかわらず、呼び出し後に輪郭テーブルを使用できるように拡張されました。

4.25.3 コード化された輪郭テーブルの生成(CONTDCON)

CONTDCON で起動した輪郭解析の場合は、メモリの使用を最適化するために、呼び出した以下の NC ブロックを、6 列の輪郭テーブルにコード化形式で保存します。各輪郭要素は、輪郭テーブルの 1 行に対応します。以下に示すコード化の規則を熟知しているなどの場合は、テーブル行のサイクルに DIN コードプログラムを組み合わせることもできます。開始点のデータは、番号 0 のテーブル行に保存されます。

構文

輪郭解析の起動:

CONTDCON (<輪郭テーブル>, <加工方向>)

輪郭解析の解除と通常の実行モードへの復帰:

EXECUTE (<ERROR>)

「輪郭解析の解除(EXECUTE) (ページ 1194)」を参照してください。

意味

CONTDCON:	輪郭解析を起動して、コード化された輪郭テーブルを生成する予約処理
<輪郭テーブル>:	輪郭テーブルの名称

4.25 ユーザー荒削りプログラム

<加工方向>:	加工方向のパラメータ		
	タイプ:	INT	
	値:	0	輪郭ブロックの順序に従った輪郭解析(初期値)
		1	不可

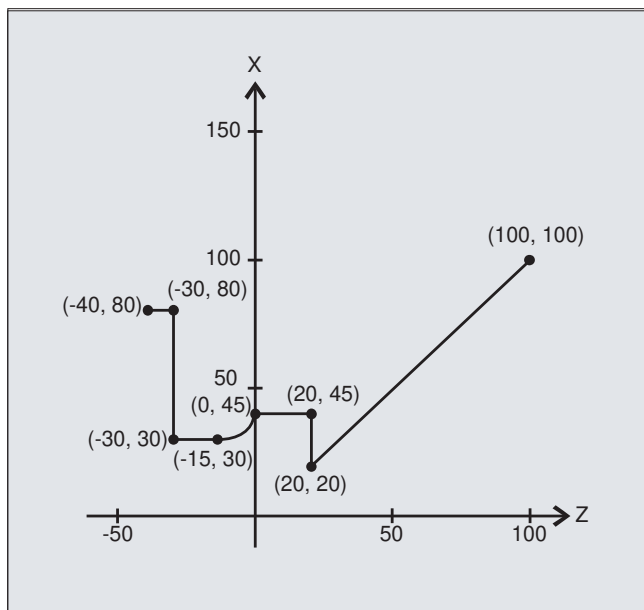
注記

テーブルに含まれるプログラム区間の CONTDCON に使用できる G 命令は、CONTPRON に使用できるものより広範囲です。さらに、送り速度と送り速度タイプを、輪郭区間毎に保存します。

例

以下で輪郭テーブルを生成します:

- 名称「KTAB」
- 輪郭要素(円弧、直線)
- 運転モード:旋削
- 加工方向:前方



4.25 ユーザー荒削りプログラム

NC プログラム:

プログラムコード	コメント
N10 DEF REAL KTAB[9,6]	; 名称が KTAB で、テーブルのセルが 9 個の輪郭テーブルです。これには、輪郭セットを 8 個格納できます。パラメータ値 6 (テーブルの列番号) は固定サイズです。
N20 DEF INT MODE = 0	; 加工方向の変数です。標準値 0: 輪郭のプログラム指令方向にのみ有効です。
N30 DEF INT ERROR = 0	; エラーフィードバック信号を示す変数です。
...	
N100 G18 G64 G90 G94 G710	
N101 G1 Z100 X100 F1000	
N105 CONTDCON (KTAB, MODE)	; 輪郭解析を呼び出します (MODE は省略可能です)。
N110 G1 Z20 X20 F200	; 輪郭を記述します。
N120 G9 X45 F300	
N130 Z0 F400	
N140 G2 Z-15 X30 K=AC(-15) I=AC(45) F100	
N150 G64 Z-30 F600	
N160 X80 F700	
N170 Z-40 F800	
N180 EXECUTE(ERROR)	; 輪郭テーブルへの書き込みを終了し、通常のプログラム\モードに切り替えます。
...	

輪郭テーブル KTAB:

	列インデックス					
	0	1	2	3	4	5
行インデックス	輪郭モード	終点の横座標	終点の縦座標	中心点の横座標	中心点の縦座標	送り速度
0	30	100	100	0	0	7
1	11031	20	20	0	0	200
2	111031	20	45	0	0	300
3	11031	0	45	0	0	400
4	11032	-15	30	-15	45	100
5	11031	-30	30	0	0	600
6	11031	-30	80	0	0	700
7	11031	-40	80	0	0	800
8	0	0	0	0	0	0

4.25 ユーザー荒削りプログラム

列の内容の説明:

始点の 0 行目のコード:

0 列目: 10^0 (1 の位):G0 = 0
 10^1 (10 の位):G70 = 0、G71 = 1、G700 = 2、G710 = 3
 1 列目: 起点の横座標
 2 列目: 起点の縦座標
 3～4 列目: 0
 5 列目: テーブルの最後の輪郭部分の行インデックス

1～n 行:輪郭部分の入力値

0 列目: 10^0 (1 の位):G0 = 0、G1 = 1、G2 = 2、G3 = 3
 10^1 (10 の位):G70 = 0、G71 = 1、G700 = 2、G710 = 3
 10^2 (100 の桁):G90 = 0、G91 = 1
 10^3 (1000 の桁):G93 = 0、G94 = 1、G95 = 2、G96 = 3
 10^4 (10000 の桁):G60 = 0、G44 = 1、G641 = 2、G642 = 3
 10^5 (100000 の桁):G9 = 1
 1 列目: 終点の横座標
 2 列目: 終点の縦座標
 3 列目: 円弧補間の中心点の横座標
 4 列目: 円弧補間の中心点の縦座標
 5 列目: 送り速度

詳細情報

使用可能な移動指令、座標系

輪郭のプログラミングには、次の G グループと G 命令を使用できます。

G グループ 1: G0、G1、G2、G3
 G グループ 10: G60、G64、G641、G642
 G グループ 11: G9
 G グループ 13: G70、G71、G700、G710
 G グループ 14: G90、G91
 G グループ 15: G93、G94、G95、G96G961

さらに、以下を実行できます。

- 丸み付けと面取り
- CIP と CT による円弧プログラミング

スプライン、多項式、およびねじの各機能を実行すると、エラーが発生します。

CONTDCON と EXECUTE の間でフレームを起動して、座標系を変更することはできません。これは、G70 と G71、または G700 と G710 の間での変更でも同様です。

輪郭テーブルの解析中に、GEOAX でジオメトリ軸を置換すると、アラームが発生します。

加工方向

CONTDCON で生成した輪郭テーブルを使用して、輪郭のプログラム指令方向に荒削りをおこないます。

4.25.4 2 個の輪郭要素の交点の特定(INTERSEC)

INTERSEC は、CONTPRON で生成した輪郭テーブルの標準化した 2 個の輪郭要素の交点を特定します。

構文

```
<状態>=INTERSEC (<輪郭テーブル_1>[<輪郭要素_1>],  
<輪郭テーブル_2>[<輪郭要素_2>],<交点>,<加工タイプ>)
```

意味

INTERSEC:	CONTPRON で生成した輪郭テーブルから取得した 2 個の輪郭要素の交点を特定する予約機能		
<状態>:	交点状態を表わす変数		
	タイプ:	BOOL	
	値:	TRUE	交点が見つかりました
		FALSE	交点が見つかりません
<輪郭テーブル_1>:	1 番目の輪郭テーブルの名称		
<輪郭要素_1>:	1 番目の輪郭テーブルの輪郭要素の番号		
<輪郭テーブル_2>:	2 番目の輪郭テーブルの名称		
<輪郭要素_2>:	2 番目の輪郭テーブルの輪郭要素の番号		

4.25 ユーザー荒削りプログラム

<交点>:	有効平面(G17 / G18 / G19)上の交点の座標		
	タイプ:	REAL	
<加工タイプ>:	加工タイプのパラメータ		
	タイプ:	INT	
	値:	0	有効平面上のパラメータ 2 による交点演算 (標準値)
		1	平面移動に影響されない交点演算

注記

変数は、使用する前に定義してください。

CONTPRON で定義した値を維持したまま、輪郭を移動してください。

パラメータ	意味
2	移動のための輪郭モードのコード
3	輪郭の始点の横座標
4	輪郭の始点の縦座標
5	輪郭の終点の横座標
6	輪郭の終点の縦座標
9	横座標の中心点座標(円弧輪郭の場合のみ)
10	縦座標の中心点座標(円弧輪郭の場合のみ)

例

テーブル TABNAME1 の輪郭要素 3 とテーブル TABNAME2 の輪郭要素 7 の交点演算です。有効平面上の交点座標は変数 ISCOORD に格納されます(1 番目の要素=横座標、2 番目の要素=縦座標)。交点がない場合は、プログラムが NOCUT にジャンプします(交点が見つかりません)。

プログラムコード	コメント
DEF REAL TABNAME1 [12,11]	; 輪郭テーブル 1
DEF REAL TABNAME2 [10,11]	; 輪郭テーブル 2
DEF REAL ISCOORD [2]	; 交点座標を表わす変数です。
DEF BOOL ISPOINT	; 交点状態を表わす変数です。
DEF INT MODE	; 加工タイプを表わす変数です。
...	
モード= 1	; 演算は有効平面とは無関係におこなわれます。

4.25 ユーザー荒削りプログラム

プログラムコード	コメント
N10 ISPOINT=INTERSEC(TABNAME1[3],TABNAME2[7], ISCOORD,MODE)	; 輪郭要素の交点を呼び出します。
N20 IF ISPOINT==FALSE GOTOF NOCUT	; NOCUT ヘジャンプします
...	

4.25.5 テーブルの輪郭要素をブロック毎に実行(EXECTAB)

EXECTAB を使用して、CONTPRON など生成したテーブルの輪郭要素をブロック毎に実行できます。

構文

EXECTAB (<輪郭テーブル> [<輪郭要素>])

意味

EXECTAB:	輪郭要素を実行する予約機能
<輪郭テーブル>:	輪郭テーブルの名称
<輪郭要素>:	輪郭要素の番号

例

テーブル KTAB の輪郭要素 0～2 をブロック毎に実行します。

プログラムコード	コメント
N10 EXECTAB(KTAB[0])	; テーブル KTAB の要素 0 を移動します。
N20 EXECTAB(KTAB[1])	; テーブル KTAB の要素 1 を移動します。
N30 EXECTAB(KTAB[2])	; テーブル KTAB の要素 2 を移動します。

4.25.6 円弧データの計算(CALCDAT)

CALCDAT を使用して、円弧上の既知の 3～4 個の点から、円弧の半径と中心点の座標を計算できます。それぞれ異なる点を指定してください。

4.25 ユーザー荒削りプログラム

円弧上に 4 点が直接に存在していない場合は、円弧の中心点と半径に対する平均値が計算されます。

注記

平均の計算規則

円弧の計算は、次の 4 回実行されます。

- 1. 円弧上の点 1、2、3 で
- 2. 円弧上の点 1、2、4 で
- 3. 円弧上の点 1、3、4 で
- 4. 円弧上の点 2、3、4 で

円弧の中心点の横座標と縦座標の値は、4 つの円弧計算の横座標と縦座標の値を合計して 4 で割って算出されます。

半径は、円弧計算で得られた 4 つの半径の合計の平方根を求め、その結果に 0.5 を乗算して算出されます。

構文

<状態>=CALCDAT (<円弧上の点> [<数>,<タイプ>] ,<数>,<結果>)

意味

CALCDAT:	円弧の半径と中心点座標を 3～4 点から計算する予約機能		
<状態>:	円弧の計算状態を示す変数		
	タイプ:	BOOL	
	値:	TRUE	指定した点が円弧上にあります。
		FALSE	指定した点が円弧上にありません。
<円弧上の点> []:	円弧上の点を パラメータで指定するための変数		
	<番号>:	円弧上の点の数(3 または 4)	
	<タイプ>:	座標データのタイプ 例: 2 点座標の場合は 2	
	<番号>:	計算に使用する点の数を示すパラメータ(3 または 4)	

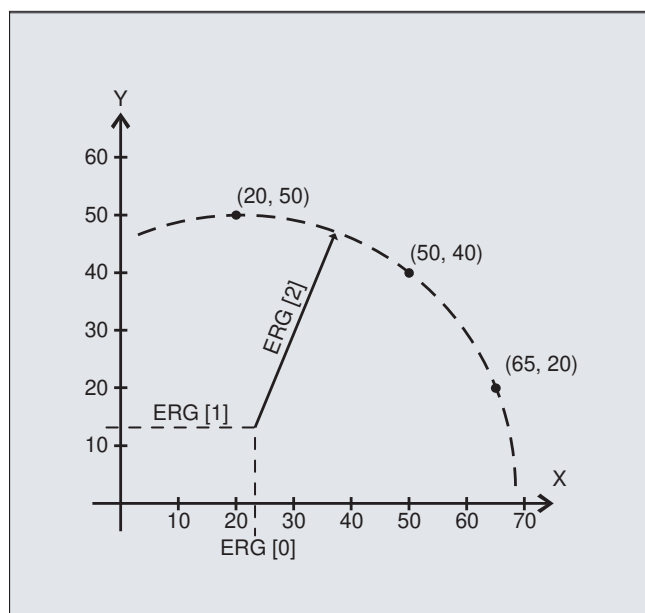
<結果>[3]:	結果を示す変数:	
	円弧の中心点座標と半径	
	0	円弧の中心点座標:横座標値
	1	円弧の中心点座標:縦座標値
	2	半径

注記

変数は、使用する前に定義してください。

例

3点を使用して、それらが、1つの円弧セグメント上にあるかどうかを特定します。



プログラムコード	コメント
N10 DEF REAL PT[3,2]=(20,50,50,40,65,20)	; 円弧の点を指定するための変数
N20 DEF REAL RES[3]	; 結果を示す変数
N30 DEF BOOL STATUS	; 状態を示す変数
N40 STATUS=CALCDAT(PKT,3,ERG)	; 特定した円弧データを呼び出します。
N50 IF STATUS == FALSE GOTO ERROR	; エラーへジャンプします。

4.25 ユーザー荒削りプログラム

4.25.7 輪郭解析の解除(EXECUTE)

EXECUTE は輪郭解析を解除します。これと同時に、システムは通常の実行モードに戻ります。

構文

EXECUTE (<ERROR>)

意味

EXECUTE:	輪郭解析を終了する予約機能	
<ERROR>:	; エラーフィードバック信号を示す変数	
	タイプ:	INT
	変数の値は、エラーが発生することなく、輪郭が解析可能であったかどうかを表わします。	
	0	異常
	1	エラーなし

例

```
プログラムコード
...
N30 CONTPRON(...)
N40 G1 X...Z...
...
N100 EXECUTE(...)
...
```

4.26 サイクルを外部的にプログラミング

4.26.1 テクノロジサイクル

4.26.1.1 はじめに

目次

本章では、旋削、フライス削り、および研削のサイクルについて説明します。

構成

サイクルの説明は、以下のように構成されています。

- **構文**
転送パラメータのサイクル名と呼び出し手順です。
- **パラメータ**
個々のパラメータを説明するためのテーブルです。

パラメータリスト

パラメータのテーブルには、名称、説明、値の範囲、および他のパラメータとの依存関係が含まれています。

この画面のパラメータの参照欄は、外部で生成されたサイクル呼び出しの再コンパイル時に、制御装置でプログラム指令された値をより簡単に特定するために提供されています。

「インタフェース専用」パラメータ

特定のパラメータはテーブル内で「インタフェース専用」の印が付いています。このようなパラメータは、サイクルの動作には関係ありません。サイクル呼び出しを完全に再コンパイルするためにだけです。このパラメータがプログラム指令されていなくても、サイクルは再コンパイルが可能です。このフィールドは色で識別され、画面内で完結します。

「予備」パラメータ

「予備」として記述されているパラメータは、以下の呼び出しパラメータが内部サイクルパラメータと一致するように値 0 またはカンマを使ってプログラム指令してください。例外:値""またはカンマを使用した文字列パラメータ

4.26 サイクルを外部的にプログラミング

位置決めパターンでのサイクルの繰り返し

穴あけサイクルとミリングサイクルは、位置決めパターン(モーダル呼び出し)で繰り返すことができます。このような場合、同一の行で MCALL をサイクルの前に書き込んでください(例:MCALL CYCLE83 (...))。

注記

特定の転送パラメータ(<_VARI>, <_GMODE>, <_DMODE>, <_AMODE>など)がパラメータとして間接的にプログラム指令されている場合は、再コンパイル時に入力画面が開かれますが、定義された選択欄への一義的な割り当てがないため、保存できません。

4.26.1.2 テクノロジ別の概要

以下の一覧表に、使用可能な外部でプログラム可能なすべてのテクノロジサイクルと、それぞれのサイクルに割り当てられたテクノロジをリストします。

テクノロジ	テクノロジサイクル
穴あけ	• CYCLE81 - 穴あけ、センタリング (ページ 1254) • CYCLE82 - 穴あけ、座ぐり (ページ 1255) • CYCLE85 - リーマ加工 (ページ 1267) • CYCLE86 - ボーリング (ページ 1268) • CYCLE83 - 深穴ドリル 1 (ページ 1258) • CYCLE830 - 深穴ドリル 2 (ページ 1305) • CYCLE84 - フローティングチャックなしのタッピング (ページ 1262) • CYCLE840 - フローティングチャックによるタッピング (ページ 1317) • CYCLE78 - ドリルねじフライス削り (ページ 1247) • CYCLE802 - 任意の位置 (ページ 1300) • HOLES1 - 行位置決めパターン (ページ 1199) • CYCLE801 - 格子またはフレーム位置決めパターン (ページ 1298) • HOLES2 - 円または円弧位置決めパターン (ページ 1200)
旋削	• CYCLE951 - 荒削り (ページ 1331) • CYCLE930 - 溝削り (ページ 1324) • CYCLE940 - アンダーカット形状 E と形状 F/ねじのアンダーカット (ページ 1327) • CYCLE99 - ねじの旋削 (ページ 1281) • CYCLE98 - 連続ねじ (ページ 1275) • CYCLE92 - 突切り (ページ 1270)
輪郭旋削	• CYCLE62 - 輪郭の呼び出し (ページ 1226) • CYCLE952 - 旋削/旋削削り残し/プランジ切削/プランジ切削削り残し/プランジ旋削/プランジ旋削削り残し (ページ 1335)

4.26 サイクルを外部的にプログラミング

テクノロジー	テクノロジーサイクル
フライス加工	• CYCLE61 - 正面削り (ページ 1223) • POCKET3 - 長方形ポケット (ページ 1202) • POCKET4 - 円形ポケット (ページ 1206) • CYCLE76 - 長方形スゴット (ページ 1241) • CYCLE77 - 円形スピゴット (ページ 1244) • CYCLE79 - 多角形 (ページ 1251) • SLOT1- 直線溝 (ページ 1210) • SLOT2 - 円周溝 (ページ 1214) • CYCLE899 - オープン溝 (ページ 1321) • LONGHOLE - 長穴 (ページ 1217) • CYCLE70 - ねじフライス削り (ページ 1234) • CYCLE60 - 彫刻 (ページ 1220)
輪郭切削	• CYCLE62 - 輪郭の呼び出し (ページ 1226) • CYCLE72 - 輪郭フライス削り (ページ 1236) • CYCLE63 - 輪郭ポケットのフライス削り/輪郭ポケットの削り残し/輪郭スピゴットのフライス削り/輪郭スピゴットの削り残し (ページ 1227) • CYCLE64 - 輪郭ポケットの前加工 (ページ 1231)
研削加工	• CYCLE495 - 成形研削 (ページ 1287) • CYCLE435 - ドレッサー座標系の設定 (ページ 1286) • CYCLE4071 - 反転点での切り込みによる長手方向の研削 (ページ 1342) • CYCLE4072 - 反転点での切り込みによる長手方向の研削とキャンセル信号 (ページ 1344) • CYCLE4073 - 連続切り込みによる長手方向の研削 (ページ 1348) • CYCLE4074 - 連続切り込みによる長手方向の研削とキャンセル信号 (ページ 1350) • CYCLE4075 - 反転点での切り込みによる平面研削 (ページ 1354) • CYCLE4077 - 反転点での切り込みによる平面研削とキャンセル信号 (ページ 1357) • CYCLE4078 - 連続切り込みによる平面研削 (ページ 1361) • CYCLE4079 - 間欠切り込みによる平面研削 (ページ 1363)

4.26 サイクルを外部的にプログラミング

テクノロジー	テクノロジーサイクル
他の	• CYCLE782 - 負荷に従った位置合わせ (ページ 1289) • CYCLE800 - 旋回平面/旋回工具/工具の位置合わせ (ページ 1292) • CYCLE806 - 補間旋削 (ページ 1304) • CYCLE832 - 高速設定 (ページ 1313)
すべて	• GROUP_BEGIN - プログラムブロック開始 (ページ 1366) • GROUP_END - プログラムブロック終了 (ページ 1367) • GROUP_ADDEND - 試し切り追加終了 (ページ 1367)

4.26.1.3 HOLES1 - 行位置決めパターン

構文

HOLES1(<SPCA>, <SPCO>, <STA1>, <FDIS>, <DBH>, <NUM>, <_VARI>, <_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	X0	<SPCA>	REAL	1 番目の軸の直線位置決めパターンの基準点(abs)
2	Y0	<SPCO>	REAL	2 番目の軸の直線位置決めパターンの基準点(abs)
3	α0	<STA1>	REAL	基本回転角度(1 番目の軸に対する角度)
4	L0	<FDIS>	REAL	1 番目の穴から基準点までの距離
5	L	<DBH>	REAL	穴間の間隔
6	N	<NUM>	INT	穴の数
7		<_VARI>	INT	予備
8		<_UMODE>	INT	予備
9		<_HIDE>	STRING [200]	非表示位置 • 最大 198 文字 • 連続する位置番号の指定。例:"1,3"(位置 1 と 3 は実行されません)
10		<_NSP>	INT	予備

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
11		<_DMODE>	INT	表示モード	
				UNITS(一の位):	加工平面 G17/18/19
					0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 = G17 (サイクルでのみ有効)
					2 = G18 (サイクルでのみ有効)
				3 =	G19 (サイクルでのみ有効)

4.26.1.4 HOLES2 – 円または円弧位置決めパターン

構文

HOLES2 (<CPA>, <CPO>, <RAD>, <STA1>, <INDA>, <NUM>, <_VARI>, <_UMODE>, <_HIDE>, <_NSP>, <_DMODE>)

パラメータ

No.	マスクパラメータ	パラメータ内部	データタイプ	意味	
1	X0	<CPA>	REAL	1 番目の軸の円弧位置決めパターンの中心点(abs) 1 番目の軸の基準点	(XY の場合) (XA、YB、ZC の場合)
2	Y0	<CPO>	REAL	2 番目の軸の円弧位置決めパターンの中心点(abs) 2 番目の軸の基準点	(XY の場合) (XA、YB、ZC の場合)
3	R	<RAD>	REAL	円弧位置決めパターンの半径	(XY の場合)
4	$\alpha 0$	<STA1>	REAL	開始角度 または 1 番目の回転軸の位置	(XY の場合) (XA、YB、ZC の場合)
5	$\alpha 1$	<INDA>	REAL	進み角度 - (円弧のみ)	
				<0 =	右回り
				>0 =	反時計回り

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	パラメータ内部	データタイプ	意味
6	N	<NUM>	INT	位置の数
7		<_VARI>	INT	加工タイプ
				UNITS(一の位): 予備
				TENS(十の位): 位置決めタイプ
				0 = アプローチ位置 - 直線
				1 = アプローチ位置 - 円弧軌跡
				HUNDREDS(百の位): 予備
				THOUSANDS(千の位): 円弧パターン
				0 = 互換性モード、INDA = の場合は一周円、INDA <> 0 の場合は円弧
				1 = 一周円
				2 = 円弧
				TEN THOUSANDS(万の位): 回転軸ありの位置決めパターン
				0 = XY (回転軸なし) (XY の場合)
				1 = XA (X 軸と X を中心とした回転軸) (XA の場合のみ)
				2 = YB (Y 軸と Y を中心とした回転軸) (YB の場合のみ)
				3 = ZC (Z 軸と C を中心とした回転軸) (ZC の場合のみ)
				TEN MILLIONS + ONE MILLION(百万+ 十万の位): オフセット(同じ軸を中心とした複数の回転軸用、インデックスが大きすぎる場合は、1 番目の軸)
				00 = 1 番目の A、B または C 軸
				01 = 2 番目の A、B または C 軸
				...
				10 = 20 番目の A、B または C 軸
8		<_UMODE>	INT	予備

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	パラメータ内部	データタイプ	意味
9		<_HIDE>	STRING [200]	予備
10		<_NSP>	INT	予備
11		<_DMODE>	INT	表示モード
		UNITS(一の位): 加工平面 G17/18/19		
		0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効		
		1 = G17 (サイクルでのみ有効)		
		2 = G18 (サイクルでのみ有効)		
		3 = G19 (サイクルでのみ有効)		

4.26.1.5 POCKET3 - 長方形ポケット

構文

```
POCKET3(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_CRAD>,
<_PA>, <_PO>, <_STA>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>,
<_CDIR>, <_VARI>, <_MIDA>, <_AP1>, <_AP2>, <_AD>, <_RAD1>,
<_DP1>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<_RTP>	REAL	イニシャル点(abs)
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<_DP>	REAL	ポケットの深さ(abs/inc)、<_AMODE>を参照
5	L	<_LENG>	REAL	ポケットの長さ(inc、符号なしで入力)
6	W	<_WID>	REAL	ポケットの幅(inc、符号なしで入力)
7	R	<_CRAD>	REAL	ポケットのコーナ半径
8	X0	<_PA>	REAL	基準点、1 番目の軸(abs)
9	Y0	<_PO>	REAL	基準点、2 番目の軸(abs)
10	α0	<_STA>	REAL	回転角度、長手軸(L)と 1 番目の軸との間の角度

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味			
11	DZ	<_MID>	REAL	最大切り込み深さ			
12	UXY	<_FAL>	REAL	平面の仕上げ代			
13	UZ	<_FALD>	REAL	深さの仕上げ代			
14	F	<_FFP1>	REAL	平面の送り速度			
15	FZ	<_FFD>	REAL	深さ切り込み速度			
16		<_CDIR>	INT	加工方向	0 =	ダウンカット	
					1 =	アップカット	
17		<_VARI>	INT	加工タイプ			
				UNITS(一の位):			
					1 =	荒削り	
					2 =	仕上げ	
					4 =	端面の仕上げ	
					5 =	面取り	
				TENS(十の位):			
					0 =	予備穴あけ、G0 による切り込み	
					1 =	縦方向、G1 による切り込み	
					2 =	ヘリカル	
					3 =	ポケットの長手軸の揺動	
				HUNDREDS(百の位):	予備		
18	DXY	<_MIDA>	REAL	最大平面切り込み、単位については、<_AMODE>を参照			
19	L1	<_AP1>	REAL	予備加工の長さ(inc)			
20	W1	<_AP2>	REAL	予備加工の幅(inc)			
21	AZ	<_AD>	REAL	予備加工の深さ(inc)			
22	ER	<_RAD1>	REAL	ヘリカル切込みのヘリカル軌跡の半径			
	EW			揺動時の最大切り込み角度			
23	EP	<_DP1>	REAL	ヘリカル切込み時のヘリカルピッチ			
24		<_UMODE>	INT	予備			
25	FS	<_FS>	REAL	面取り幅(inc)			
26	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs/inc)、<_AMODE>を参照			

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
27		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)	
				UNITS(一の位):	予備
				TENS(十の位):	予備
				HUNDREDS(百の位):	加工または開始点の計算のみのどちらかを選択します。
					0 = 互換性モード
					1 = 通常の加工
				THOUSANDS(千の位):	中心/コーナによる位置指令方法
					0 = 互換性モード
					1 = 中心による位置指令方法
					2 = コーナ点による位置指令方法、ポケット位置 +LENG/+WID
					3 = コーナ点による位置指令方法、ポケット位置 -LENG/+WID
					4 = コーナ点による位置指令方法、ポケット位置 +LENG/-WID
					5 = コーナ点による位置指令方法、ポケット位置 -LENG/-WID
				TEN THOUSANDS(万の位):	全て加工/再加工
					0 = 互換性モード(<_AP1>, <_AP2>と<_AD> を以前と同様に処理)
					1 = 全て加工
					2 = 後加工

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
28		<_DMODE>	INT	表示モード
				UNITS(一の位):
				加工平面 G17/G18/G19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
				TENS(十の位):
				送り速度タイプ:面と深さの送り速度の G グループ(G94/G95)
				0 = 互換性モード
				1 = サイクル呼び出し前と同じ G 命令。面と深さの送り速度と同じ G94/G95
				HUNDREDS(百の位): --- 予備
29		<_AMODE>	INT	代替モード
				UNITS(一の位):
				ポケットの深さ(Z1)
				0 = アブソリュート(互換性モード)
				1 = インクレメンタル
				TENS(十の位):
				平面切り込みの単位(DXY)
				0 = mm
				1 = 工具直径の%
				HUNDREDS(百の位):
				面取りの切り込み深さ(ZFS)
				0 = アブソリュート
				1 = インクレメンタル

4.26 サイクルを外部的にプログラミング

4.26.1.6 POCKET4 - 円形ポケット

構文

```
POCKET4 (<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_CDIAM>, <_PA>, <_PO>,
<_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>,
<_MIDA>, <_AP1>, <_AD>, <_RAD1>, <_DP1>, <_UMODE>, <_FS>, <_ZFS>,
<_GMODE>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	RP	<_RTP>	REAL	イニシャル点(abs)		
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)		
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)		
4	Z1	<_DP>	REAL	ポケットの深さ(abs/inc)、<_AMODE>を参照		
5	Ø	<_CDIAM>	REAL	ポケットの直径または半径、<_DMODE>を参照		
6	X0	<_PA>	REAL	基準点、1 番目の軸(abs)		
7	Y0	<_PO>	REAL	基準点、2 番目の軸(abs)		
8	DZ	<_MID>	REAL	最大深さ設定、<_VARI> = 平面毎を参照 最大ヘリカル設定、<_VARI> = ヘリカルを参照		
9	UXY	<_FAL>	REAL	平面の仕上げ代		
10	UZ	<_FALD>	REAL	深さの仕上げ代		
11	F	<_FFP1>	REAL	面加工時の送り速度		
12	FZ	<_FFD>	REAL	深さ切り込み速度		
13		<_CDIR>	INT	加工方向	0 =	ダウンカット
					1 =	アップカット

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
14		<_VARI>	INT	加工タイプ
				UNITS(一の位):
				加工
				1 = 荒削り
				2 = 仕上げ
				4 = 端面の仕上げ
				5 = 面取り
				TENS(十の位):
				切り込みタイプ(荒削りと仕上げ)
				0 = 予備穴あけ、G0 による切り込み(ポケットは前加工済み)
				1 = 縦方向、G1 による切り込み
				2 = ヘリカル
15	DXY	<_MIDA>	REAL	HUNDREDS(百の位):
				予備
				THOUSANDS(千の位):
16	Ø	<_AP1>	REAL	0 = 平面毎
				1 = ヘリカル
15				最大平面切り込み、<_AMODE>を参照、 0 = 0.8 x 工具直径
16	Ø	<_AP1>	REAL	予備加工の直径/半径(inc)
17	AZ	<_AD>	REAL	予備加工の深さ(inc)
18	ER	<_RAD1>	REAL	ヘリカル切込みのヘリカル軌跡の半径
19	EP	<_DP1>	REAL	ヘリカル軌跡上の切り込みのヘリカルピッチ
20		<_UMODE>	INT	予備
21	FS	<_FS>	REAL	面取り幅(inc)
22	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs/inc)、<_AMODE>を参照

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
23		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)	
				UNITS(一の位):	予備
				TENS(十の位):	予備
				HUNDREDS(百の位):	加工/開始点の計算
					0 = 互換性モード
					1 = 通常の加工
				THOUSANDS(千の位):	予備
				TEN THOUSANDS(万の位):	全て加工/再加工
					0 = 互換性モード(<_AP1>と<_AD>を以前と同様に処理)
					1 = 全て加工
					2 = 後加工

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
24		<_DMODE>	INT	表示モード	
				UNITS(一の位):	加工平面 G17/18/19
					0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 = G17 (サイクルでのみ有効)
					2 = G18 (サイクルでのみ有効)
					3 = G19 (サイクルでのみ有効)
				TENS(十の位):	送り速度タイプ:面と深さの送り速度の G グループ(G94/G95)
					0 = 互換性モード
					1 = サイクル呼び出し前と同じ G 命令。面と深さの送り速度と同じ G94/G95
				HUNDREDS(百の位):	
					0 = 互換性モード(半径として<_CDIAM>/<_AP1>を入力)
					1 = 直径として<_CDIAM>/<_AP1> を入力
				THOUSANDS(千の位):	--- 予備
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1368)
					0 = 入力:全て
					1 = 入力:簡易

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
25		<_AMODE>	INT	代替モード
				UNITS(一の位):
				ポケットの深さ(Z1)
				0 = アブソリュート(互換性モード)
				1 = インクレメンタル
				TENS(十の位):
				切り込み幅の単位(DXY)
				0 = mm
				1 = 工具直径の%
				HUNDREDS(百の位):
				面取りの切り込み深さ(ZFS)
				0 = アブソリュート
				1 = インクレメンタル

4.26.1.7 SLOT1- 直線溝

構文

SLOT1 (<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>, <WID>, <_CPA>, <_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <_MID>, <CDIR>, <_FAL>, <VARI>, <_MIDF>, <FFP2>, <SSF>, <_FALD>, <_STA2>, <_DP1>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<RFP>	REAL	工具軸の基準点(abs)
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<_DP>	REAL	溝の深さ(abs)
5		<_DPR>	REAL	Z0 を基準にした溝の深さ(inc)(符号なしで入力)
6		<NUM>	INT	溝の数 = 1
7	L	<LENG>	REAL	溝の長さ
8	W	<WID>	REAL	溝幅
9	X0	<_CPA>	REAL	平面の 1 番目の軸の基準点

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味			
10	Y0	<_CPO>	REAL	平面の 2 番目の軸の基準点			
11		<RAD>	REAL	予備			
12	α	<STA1>	REAL	回転角度			
13		<INDA>	REAL	予備			
14	FZ	<FFD>	REAL	深さ切り込み速度			
15	F	<FFP1>	REAL	送り速度			
16	DZ	<_MID>	REAL	最大切り込み深さ			
17		<CDIR>	INT	加工方向	0 =	ダウンカット	
					1 =	アップカット	
18	UXY	<_FAL>	REAL	平面または溝端面の仕上げ代			
19		<VARI>	INT	加工タイプ			
				UNITS(一の位):			
					0 =	予備	
					1 =	荒削り	
					2 =	仕上げ	
					4 =	端面の仕上げ(端面のみを加工)	
					5 =	面取り	
				TENS(十の位):	アプローチ		
					0 =	予備穴あけ、G0 による切り込み(溝は前加工済み)	
					1 =	縦方向、G1 による切り込み	
					2 =	ヘリカル	
					3 =	揺動	
				HUNDREDS(百の位):		予備	
20	DZF	<_MIDF>	REAL	予備			
21	FF	<FFP2>	REAL	予備			
22	SF	<SSF>	REAL	予備			
23	UZ	<_FALD>	REAL	深さの仕上げ代			
24	ER	<_STA2>	REAL	ヘリカル切り込みでのヘリカル軌跡の半径			
	EW			揺動時の最大切り込み角度			

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
25	EP	<_DP1>	REAL	ヘリカル時の 1 回転あたりの切り込み深さ
26		<_UMODE>	INT	予備
27	FS	<_FS>	REAL	面取りの面取り幅(inc)
28	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs または inc)、 <_AMODE>を参照
29		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)
				UNITS(一の位): 予備
				TENS(十の位): 予備
				HUNDREDS(百の位): 加工または開始点の計算のみのどちらかを選択します。
				1 = 通常の加工
				THOUSANDS(千の位): 溝の長さの基準点の位置指令方法
				0 = 中央
				1 = 内部左側 +L
				2 = 内部右側 -L
				3 = 左端 +L
				4 = 右端 -L

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
30		<_DMODE>	INT	表示モード
				UNITS(一の位):
				加工平面 G17/18/19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
				TENS(十の位):
				予備
				HUNDREDS(百の位):
				予備
				THOUSANDS(千の位):
				ソフトウェアバージョンの識別
31		<_AMODE>	INT	代替モード
				UNITS(一の位):
				最終深さ Z1 (abs/inc)
				0 = 互換性
				1 = Z1 (inc)
				2 = Z1 (abs)
				TENS(十の位):
				予備
				HUNDREDS(百の位):
				面取りの切り込み深さ(ZFS)
				0 = ZFS (abs)
				1 = ZFS (inc)

注記

このサイクルには、以前のソフトウェアバージョンにはなかった新しい機能が備わっています。その結果、入力画面(<NUM>, <RAD>, <INDA>)の特定のパラメータが表示されなくなりました。1つの位置決めパターン複数の溝を、「MCALL」を使用して希望する位置決めパターン、たとえば HOLES2 など呼び出すことでプログラム指令できます。

4.26 サイクルを外部的にプログラミング

4.26.1.8 SLOT2 - 円周溝

構文

```
SLOT2(<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <AFSL>, <WID>,
<_CPA>, <_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <_MID>,
<CDIR>, <_FAL>, <VARI>, <_MIDF>, <FFP2>, <SSF>, <_FFCP>,
<_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<RFP>	REAL	工具軸の基準点(abs)
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<_DP>	REAL	溝の深さ(abs)
5		<_DPR>	REAL	Z0 を基準にした溝の深さ(inc)(符号なしで入力)
6	N	<NUM>	INT	溝の数
7	$\alpha 1$	<AFSL>	REAL	溝の開口部角度
8	W	<WID>	REAL	溝幅
9	X0	<_CPA>	REAL	基準点 = 円弧の中心点、平面の 1 番目の軸
10	Y0	<_CPO>	REAL	基準点 = 円弧の中心点、平面の 2 番目の軸
11	R	<RAD>	REAL	円弧の半径
12	$\alpha 0$	<STA1>	REAL	開始角度
13	$\alpha 2$	<INDA>	REAL	増分角度
14	FZ	<FFD>	REAL	深さ切り込み速度
15	F	<FFP1>	REAL	送り速度
16	DZ	<_MID>	REAL	最大切り込み深さ
17		<CDIR>	INT	加工方向
				0 = ダウンカット 1 = アップカット
18	UXY	<_FAL>	REAL	平面または溝端面の仕上げ代

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
19		<VARI>	INT	加工タイプ
				UNITS(一の位):
				0 = 全て加工
				1 = 荒削り
				2 = 仕上げ
				3 = 端面の仕上げ
				5 = 面取り
				TENS(十の位):
				0 = G0 ラインによる中間位置決め
				1 = 円弧軌跡上での中間位置決め
				HUNDREDS(百の位): 予備
				THOUSANDS(千の位):
				0 = 互換性モード、 <INDA> = 0 の場合は一周円、 <INDA> <> 0 の場合は円弧
				1 = 一周円
				2 = 円弧
20	DZF	<_MIDF>	REAL	予備
21		<FFP2>	REAL	予備
22		<SSF>	REAL	予備
23	FF	<_FFCP>	REAL	予備
24		<_UMODE>	INT	予備
25	FS	<_FS>	REAL	面取り幅(inc)
26	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs または inc)、<_AMODE>を参照

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
27		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)
				UNITS(一の位): 予備
				TENS(十の位): 予備
				HUNDREDS(百の位): 加工または開始点の計算のみのどちらかを選択します。
				0 = 互換性モード
				1 = 通常の加工
28		<_DMODE>	INT	表示モード
				UNITS(一の位): 加工平面 G17/18/19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
				TENS(十の位): 予備
				HUNDREDS(百の位): 予備
				THOUSANDS(千の位): ソフトウェアバージョンの識別
				1 = ソフトウェアバージョン 2.5 以降の SLOT2 機能
				TEN THOUSANDS(万の位): サイクル画面の機能スケーリング (ページ 1368)
				0 = 入力:全て
				1 = 入力:簡易

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
29		<_AMODE>	INT	代替モード
				UNITS(一の位):
				最終深さ Z1 (abs/inc)
				0 = 互換性
				1 = Z1 (inc)
				2 = Z1 (abs)
				TENS(十の位):
				予備
				HUNDREDS(百の位):
				面取りの切り込み深さ(ZFS)
				0 = ZFS (abs)
				1 = ZFS (inc)

4.26.1.9 LONGHOLE - 長穴

構文

LONGHOLE(<RTP>, <RFP>, <SDIS>, <_DP>, <_DPR>, <NUM>, <LENG>, <_CPA>, <_CPO>, <RAD>, <STA1>, <INDA>, <FFD>, <FFP1>, <MID>, <_VARI>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<_DP>	REAL	長穴の深さ(abs)
5		<_DPR>	REAL	Z0 を基準にした長穴の深さ(inc)(符号なしで入力)
6		<NUM>	INT	長穴の数 = 1
7	L	<LENG>	REAL	長穴の長さ
8	X0	<_CPA>	REAL	平面における 1 番目の軸の基準点
9	Y0	<_CPO>	REAL	平面における 2 番目の軸の基準点
10		<RAD>	REAL	予備
11	α0	<STA1>	REAL	回転角度

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
12		<INDA>	REAL	予備
13	FZ	<FFD>	REAL	深さ切り込み速度
14	F	<FFP1>	REAL	送り速度
15	DZ	<MID>	REAL	最大切り込み深さ
16		<_VARI>	INT	加工タイプ
				UNITS(一の位):
				切り込みタイプ
				1 = G1 で縦方向
				3 = 揺動
				HUNDREDS(百の位):
				予備
17		<_UMODE>	INT	予備
18		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)
				UNITS(一の位):
				予備
				TENS(十の位):
				予備
				HUNDREDS(百の位):
				加工または開始点の計算のみのどちらかを選択します。
				0 = 互換性モード
				1 = 通常の加工
				THOUSANDS(千の位):
				溝の長さの基準点の位置指令方法
				0 = 中央
				1 = 内部左側 +L
				2 = 内部右側 -L
				3 = 左端 +L
				4 = 右端 -L

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
19		<_DMODE>	INT	表示モード
				UNITS(一の位):
				加工平面 G17/18/19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
				TENS(十の位):
				送り速度タイプ:面と深さの送り速度の G グループ(G94/G95)
				0 = 互換性モード
				1 = サイクル呼び出し前と同じ G 命令。面と深さの送り速度と同じ G94/G95
20		<_AMODE>	INT	HUNDREDS(百の位):
				予備
				THOUSANDS(千の位):
				ソフトウェアバージョンの識別
				1 = 機能の拡張 LONGHOLE (基準点の位置指令方法)
		<_AMODE>	INT	代替モード
				UNITS(一の位):
				最終深さ Z1 (abs/inc)
				0 = 互換性
				1 = Z1 (inc)
				2 = Z1 (abs)

注記

このサイクルには、以前のソフトウェアバージョンにはなかった新しい機能が備わっています。その結果、入力画面(<NUM>, <RAD>, <INDA>)の特定のパラメータが表示されなくなりました。1つの位置決めパターン複数の溝を、「MCALL」を使用して希望する位置決めパターン、たとえば HOLES2 など呼び出すことでプログラム指令できます。

4.26 サイクルを外部的にプログラミング

4.26.1.10 CYCLE60 - 彫刻

構文

```
CYCLE60 (<_TEXT>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_PA>,
<_PO>, <_STA>, <_CP1>, <_CP2>, <_WID>, <_DF>, <_FFD>, <_FFP1>,
<_VARI>, <_CODEP>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1		<_TEXT>	STRING [200]	彫刻するテキスト(最大 100 文字)
2	RP	<_RTP>	REAL	イニシャル点(abs)
3	Z0	<_RFP>	REAL	工具軸の基準点(abs)
4	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
5	Z1	<_DP>	REAL	深さ(abs)、<_AMODE>を参照
6	Z1	<_DPR>	REAL	深さ(inc)、<_AMODE>を参照
7	X0	<_PA>	REAL	平面の 1 番目の軸の基準点(abs) - 直交座標、<_VARI>を参照
	R			基準点、長さ(半径) - 極座標、<_VARI>を参照
8	Y0	<_PO>	REAL	平面の 2 番目の軸の基準点(abs) - 直交座標、<_VARI>を参照
	α0			基準点、1 番目の軸を基準にした角度 - 極座標、<_VARI>を参照
9	α1	<_STA>	REAL	テキストの方向、1 番目の軸を基準にしたテキスト行の角度、<_VARI>を参照
10	XM	<_CP1>	REAL	テキストの円弧の中心点、平面の 1 番目の軸(abs) - 直交座標、<_VARI> を参照
	LM			テキストの円弧の中心点、WNP を基準にした長さ(半径) - 極座標、<_VARI> を参照
11	YM	<_CP2>	REAL	テキストの円弧の中心点、平面の 2 番目の軸(abs) - 直交座標、<_VARI>を参照
	αM			テキストの円弧の中心点、1 番目の軸を基準にした角度 - 極座標、<_VARI>を参照
12	W	<_WID>	REAL	文字の高さ(符号なしで入力)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
13	DX1 DX2	<_DF>	REAL	文字間の距離/全体の幅、 <_VARI>を参照
	α2			開口角度、 <_VARI>を参照
14	FZ	<_FFD>	REAL	深さ切り込み速度、<_DMODE> を参照
15	F	<_FFP1>	REAL	面加工時の送り速度

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
16		<_VARI>	INT	加工(彫刻したテキストの配置と基準点)
				UNITS(一の位):
				基準点
				0 = 直交座標
				1 = 極座標
				TENS(十の位):
				テキストの配置
				0 = 1 つのライン上に配置
				1 = 上向きの円弧に配置
				2 = 下向きの円弧に配置
				HUNDREDS(百の位):
				予備
				THOUSANDS(千の位):
				テキストの基準点、横方向
				0 = 左側
				1 = 中央
				2 = 右側
				TEN THOUSANDS(万の位):
				テキストの基準点、縦方向
				0 = 底面
				1 = 中央
				2 = 上面
				HUNDRED THOUSANDS(十万の位):
				テキスト長
				0 = 文字の間隔
				1 = テキスト全体の幅(直線テキストのみ)
				2 = 開口角度(円形テキストのみ)
				ONE MILLION(百万の位):
				円弧の中心点
				0 = 直交(直交座標)
				1 = 極座標
				TEN MILLION(千万の位):
				ミラー書き込み
				0 = 互換性
				1 = ミラー書き込み ON
				2 = ミラー書き込み OFF
17		<_CODEP>	INT	書き込みのためのコードページ番号(現在は 1252 のみ)
18		<_UMODE>	INT	予備

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
19		_GMODE>	INT	形状モード(プログラム指令された形状データの評価)
				UNITS(一の位): 予備
				TENS(十の位): 予備
				HUNDREDS(百の位): 加工または開始点の計算のみのどちらかを選択します。
				0 = 互換性モード
				1 = 通常の加工
20		<_DMODE>	INT	表示モード
				UNITS(一の位): 加工平面 G17/18/19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
				TENS(十の位): 送り速度タイプ:面と深さの送り速度の G グループ(G94/G95)
				0 = 互換性モード
21		<_AMODE>	INT	代替モード
				UNITS(一の位): 最終深さ(<_DP>, <_DPR>)
				0 = 互換性
				1 = インクレメンタル(<_DPR>)
				2 = アブソリュート(<_DP>)

4.26.1.11 CYCLE61 - 正面削り

構文

```
CYCLE61 (<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_PA>, <_PO>, <_LENG>,
<_WID>, <_MID>, <_MIDA>, <_FALD>, <_FFP1>, <_VARI>, <_LIM>,
<_DMODE>, <_AMODE>)
```

4.26 サイクルを外部的にプログラミング

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<_RTP>	REAL	イニシャル点(abs)
2	Z0	<_RFP>	REAL	工具軸の基準点、素材の高さ(abs)
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<_DP>	REAL	仕上げ部の高さ(abs/inc)、<_AMODE>を参照
5	X0	<_PA>	REAL	1 番目の軸のコーナ点 1(abs)
6	Y0	<_PO>	REAL	2 番目の軸のコーナ点 1(abs)
7	X1	<_LENG>	REAL	1 番目の軸のコーナ点 2(abs または inc)、<_AMODE>を参照
8	Y1	<_WID>	REAL	2 番目の軸のコーナ点 2(abs または inc)、<_AMODE>を参照
9	DZ	<_MID>	REAL	最大切り込み深さ
10	DXY	<_MIDA>	REAL	最大平面切り込み(単位については、<_AMODE>を参照)
11	UZ	<_FALD>	REAL	深さの仕上げ代
12	F	<_FFP1>	REAL	加工送り速度
13		<_VARI>	INT	加工タイプ
				UNITS(一の位):
				加工タイプ
				1 = 荒削り
				2 = 仕上げ
				TENS(十の位):
				加工方向
				1 = 1 番目の軸に平行、一方向
				2 = 2 番目の軸に平行、一方向
				3 = 1 番目の軸に平行、可変方向
				4 = 2 番目の軸に平行、可変方向

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
14		<_LIM>	INT	制限	
				UNITS(一の位):	1 番目の軸の制限値、負
					0 = なし
					1 = あり
				TENS(十の位):	1 番目の軸の制限値、正
					0 = なし
					1 = あり
				HUNDREDS(百の位):	2 番目の軸の制限値、負
					0 = なし
					1 = あり
				THOUSANDS(千の位):	2 番目の軸の制限値、正
					0 = なし
					1 = あり
15		<_DMODE>	INT	表示モード	
				UNITS(一の位):	加工平面 G17/18/19
					0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 = G17 (サイクルでのみ有効)
					2 = G18 (サイクルでのみ有効)
					3 = G19 (サイクルでのみ有効)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
16		<_AMODE>	INT	代替モード
				UNITS(一の位):
				最終深さ(<_DP>)
				0 = アブソリュート
				1 = インクレメンタル
				TENS(十の位):
				平面切り込みの単位(<_MIDA>)
				0 = mm
				1 = 工具直径の%
				HUNDREDS(百の位):
				予備
				THOUSANDS(千の位):
				面の長さ
				0 = インクレメンタル
				1 = アブソリュート
				TEN THOUSANDS(万の位):
				面の幅
				0 = インクレメンタル
				1 = アブソリュート

4.26.1.12 CYCLE62 - 輪郭の呼び出し

構文

CYCLE62 (<_KNAME>, <_TYPE>, <_LAB1>, <_LAB2>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	PRG/CON	<_KNAME>	STRING [140]	輪郭名称またはサブプログラム名称は、_TYPE = 2 でプログラム指令しないでください。
2		<_TYPE>	INT	輪郭入力の特 定
				0 = サブプログラム
				1 = 輪郭名称
				2 = ラベル
				3 = サブプログラム内のラベル

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
3	LAB1	<_LAB1>	STRING[32]]	ラベル 1、輪郭の開始
4	LAB2	<_LAB2>	STRING[32]]	ラベル 2、輪郭の終了

4.26.1.13 CYCLE63 – 輪郭ポケットのフライス削り/輪郭ポケットの削り残し/輪郭スピゴットのフライス削り/輪郭スピゴットの削り残し**構文**

```
CYCLE63 (<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>, <_FZ>,  
<_DXY>, <_DZ>, <_UXY>, <_UZ>, <_CDIR>, <_XS>, <_YS>, <_ER>,  
<_EP>, <_EW>, <_FS>, <_ZFS>, <_TR>, <_DR>, <_UMODE>, <_GMODE>,  
<_DMODE>, <_AMODE>)
```

4.26 サイクルを外部的にプログラミング

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	PRG	<_PRG>	STRING [100]	切削プログラムの名称
2		<_VARI>	INT	加工タイプ
				UNITS(一の位):
				加工プロセス
				1 = 荒削り
				3 = 底面の仕上げ
				4 = 端面の仕上げ
				5 = 面取り
				TENS(十の位):
				切り込みタイプ
				0 = 中央切り込み
				1 = ヘリカル切り込み
				2 = 揺動切込み
				HUNDREDS(百の位):
				予備
				THOUSANDS(千の位):
				戻しモード
				0 = イニシャル点まで戻します
				1 = 基準点 + 安全距離まで戻します
				TEN THOUSANDS(万の位):
				底面の荒削りと仕上げの開始点
				0 = 自動
				1 = 手動
3	RP	<_RP>	REAL	イニシャル点(abs)
4	Z0	<_Z0>	REAL	工具軸の基準点(abs)
5	SC	<_SC>	REAL	安全距離(基準点に追加、符号なしで入力)
6	Z1	<_Z1>	REAL	最終深さ(<_AMODE> UNITS を参照)
7	F	<_F>	REAL	荒削り/仕上げ時の平面の送り速度
8	FZ	<_FZ>	REAL	深さ切り込み速度
9	DXY	<_DXY>	REAL	平面切り込み - 単位(<_AMODE> TENS を参照)
10	DZ	<_DZ>	REAL	深さ切り込み
11	UXY	<_UXY>	REAL	平面の仕上げ代

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
12	UZ	<_UZ>	REAL	深さの仕上げ代		
13		<_CDIR>	INT	加工方向	0 =	ダウンカット
					1 =	アップカット
14	XS	<_XS>	REAL	開始点 X、アブソリュート		
15	YS	<_YS>	REAL	開始点 Y、アブソリュート		
16	ER	<_ER>	REAL	ヘリカル切込み:半径		
17	EP	<_EP>	REAL	ヘリカル切込み:ピッチ		
18	EW	<_EW>	REAL	揺動切り込み:最大切り込み角度		
19	FS	<_FS>	REAL	面取りの面取り幅(inc)		
20	ZFS	<_ZFS>	REAL	面取り時の工具先端の切り込み深さ(<_AMODE> HUNDREDS を参照)		
21	TR	<_TR>	STRING[32]	削り残し除去加工時の基準工具の名称		
22	DR	<_DR>	INT	削り残し除去加工時の基準工具の D 番号		
23		<_UMODE>	INT	予備		
24		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	加工または開始点の計算のみのどちらかを選択します。	
					0 =	通常の加工(互換性モードは不要)
					1 =	通常の加工
					2 =	予備

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
25		<_DMODE>	INT	表示モード
				UNITS(一の位):
				加工平面 G17/G18/G19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
				TENS(十の位):
				予備
				HUNDREDS(百の位):
				テクノロジモード
				1 = ポケット
				2 = スピゴット
				THOUSANDS(千の位):
				削り残し除去加工
				0 = なし
				1 = あり
				TEN THOUSANDS(万の位):
				サイクル画面の機能スケーリング (ページ 1368)
				0 = 入力:全て
				1 = 入力:簡易
				HUNDRED THOUSANDS(十万の位):
				自動プログラム名
				0 = なし
				1 = あり

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
26		<_AMODE>	INT	代替モード	
				UNITS(一の位):	最終深さ(Z1)
					0 = アブソリュート(互換性モード)
					1 = インCREMENTAL
				TENS(十の位):	平面切り込みの単位(DXY)
					0 = mm
					1 = 工具直径の%
				HUNDREDS(百の位):	面取りの切り込み深さ(ZFS)
					0 = アブソリュート
					1 = インCREMENTAL
				THOUSANDS(千の位):	---
					予備

4.26.1.14 CYCLE64 - 輪郭ポケットの前加工

構文

```
CYCLE64 (<_PRG>, <_VARI>, <_RP>, <_Z0>, <_SC>, <_Z1>, <_F>,
<_DXY>, <_UXY>, <_UZ>, <_CDIR>, <_TR>, <_DR>, <_UMODE>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

4.26 サイクルを外部的にプログラミング

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	PRG	<_PRG>	STRING [100]	穴あけ/センタリングプログラムの名称		
2		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	予備	
				TENS(十の位):	予備	
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	戻しモード	
					0 =	イニシャル点まで戻します
					1 =	基準点 + 安全距離まで戻します
3	RP	<_RP>	REAL	イニシャル点(abs)		
4	Z0	<_Z0>	REAL	基準点(abs)		
5	SC	<_SC>	REAL	安全距離(基準点に追加、符号なしで入力)		
6	Z1	<_Z1>	REAL	穴あけ/センタリング深さ(<_AMODE> UNITS を参照)		
7	F	<_F>	REAL	穴あけ/センタリング送り速度		
8	DXY	<_DXY>	REAL	平面切り込み - 単位(<_AMODE> TENS を参照)		
9	UXY	<_UXY>	REAL	平面の仕上げ代		
10	UZ	<_UZ>	REAL	深さの仕上げ代		
11		<_CDIR>	INT	加工方向	0 =	ダウンカット
					1 =	アップカット
12	TR	<_TR>	STRING[20]]	基準工具名称		
13	DR	<_DR>	INT	基準工具 D 番号		
14		<_UMODE>	INT	予備		

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
15		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)
				UNITS(一の位): 予備
				TENS(十の位): 予備
				HUNDREDS(百の位): 加工または開始点の計算のみのどちらかを選択します。
				0 = 通常の加工(互換性モードは不要)
				1 = 通常の加工
				2 = 予備
25		<_DMODE>	INT	表示モード
				UNITS(一の位): 加工平面 G17/G18/G19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
				TENS(十の位): テクノロジモード
				1 = 予備穴あけ
				2 = センタリング
				HUNDREDS(百の位): --- 予備
				THOUSANDS(千の位): --- 予備
				TEN THOUSANDS(万の位): --- 予備
				HUNDRED THOUSANDS(十万の位): 自動プログラム名
				0 = なし
				1 = あり

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
26		<_AMODE>	INT	代替モード
				UNITS(一の位):
				穴あけ/センタリングの深さ Z1
				0 = アブソリュート(互換性モード)
				1 = インクレメンタル
				TENS(十の位):
				平面切り込みの単位(DXY)
				0 = mm
				1 = 工具直径の%

4.26.1.15 CYCLE70 - ねじフライス削り

構文

CYCLE70 (<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DIATH>, <_H1>, <_FAL>, <_PIT>, <_NT>, <_MID>, <_FFR>, <_TYPTH>, <_PA>, <_PO>, <_NSP>, <_VARI>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<_RTP>	REAL	イニシャル点(abs)
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<_DP>	REAL	ねじの長さ(abs、inc)、<_AMODE>を参照 穴の底面でのねじ切り上げを考慮します(最低でも半ピッチ)。
5	Ø	<_DIATH>	REAL	ねじの公称直径
6	H1	<_H1>	REAL	ねじ山深さ
7	U	<_FAL>	REAL	仕上げ代
8	P	<_PIT>	REAL	ピッチ(<_PITA>を選択:mm、inch、MODULE、ねじ山数/inch)
9	NT	<_NT>	INT	工具先端の刃数 工具長は常に一番下の刃を基準にしています。

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
10	DXY	<_MID>	REAL	一回切削当たりの最大切り込み <_MID>><_H1>:一回切削ですべて		
11	F	<_FFR>	REAL	フライス削り送り速度		
12		<_TYPH>	INT	ねじタイプ	0 =	めねじ
					1 =	おねじ
13	X0	<_PA>	REAL	1 番目の軸の円弧の中心点(abs)		
14	Y0	<_PO>	REAL	2 番目の軸の円弧の中心点(abs)		
15	αS	<_NSP>	REAL	開始角度(多条ねじ)		
16		<_VARI>	INT	加工タイプ		
				UNITS(一の位):		
					1 =	荒削り
					2 =	仕上げ
				TENS(十の位):		
					1 =	上から下へ
					2 =	下から上へ
				HUNDREDS(百の位):		
					0 =	右ねじ
					1 =	左ねじ
17		<_PITA>	INT	ねじピッチの評価		
					0 =	互換性モード
					1 =	ピッチ(mm)
					2 =	ピッチ(ねじ山数/inch)(TPI)
					3 =	ピッチ(inch)
					4 =	MODULE で表したピッチ
18		<_PITM>	STRING[15]	ピッチ入力のマーカとなる文字列(インタフェースのみ)		
19		<_PTAB>	STRING[20]	ねじテーブルの文字列(””、「ISO」、「BSW」、「BSP」、「UNC」) (インタフェースのみ)		
20		<_PTABA>	STRING[20]	ねじテーブルからの選択のための文字列(例:「M 10」、「M 12」、...) (インタフェースのみ)		

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
21		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)
				UNITS(一の位): 予備
				TENS(十の位): 予備
				HUNDREDS(百の位): 加工/開始点の計算
				0 = 互換性モード 1 = 通常の加工
22		<_DMODE>	INT	表示モード
				UNITS(一の位): 加工平面 G17/G18/G19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効) 3 = G19 (サイクルでのみ有効)
23		<_AMODE>	INT	代替モード
				UNITS(一の位): ねじ長さ(<_DP>)
				0 = アブソリュート 1 = インクレメンタル

4.26.1.16 CYCLE72 - 輪郭フライス削り

構文

```
CYCLE72 (<_KNAME>, <_RTP>, <_RFP>, <_SDIS>, <_DP>, <_MID>, <_FAL>,
<_FALD>, <_FFP1>, <_FFD>, <_VARI>, <_RL>, <_AS1>, <_LP1>, <_FF3>,
<_AS2>, <_LP2>, <_UMODE>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1		<_KNAME>	STRING [141]	輪郭サブプログラムの名称
2	RP	<_RTP>	REAL	イニシャル点(abs)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
3	Z0	<_RFP>	REAL	工具軸の基準点(abs)
4	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
5	Z1	<_DP>	REAL	終点、最終深さ(abs/inc)、<_AMODE>を参照
6	DZ	<_MID>	REAL	最大切り込み深さ(inc; 符号なしで入力)
7	UXY	<_FAL>	REAL	平面の仕上げ代(inc)、輪郭端面の削り代
8	UZ	<_FALD>	REAL	仕上げ代の深さ(inc)、底面の削り代(符号なしで入力)
9	FX	<_FFP1>	REAL	輪郭の送り速度
10	FZ	<_FFD>	REAL	深さ切り込み(または空間切り込み)の送り速度

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
11		<_VARI>	INT	加工タイプ	
				UNITS(一の位):	加工タイプ
					1 = 荒削り
					2 = 仕上げ
					5 = 面取り
				TENS(十の位):	
					0 = G0 による中間軌跡
					1 = G1 による中間軌跡
				HUNDREDS(百の位):	輪郭の最後で後退
					0 = 輪郭の最後で基準点まで後退
					1 = 輪郭の最後で基準点 + <_SDIS>まで後退
					2 = 輪郭の最後で<_SDIS>だけ後退
					3 = 輪郭の最後で後退せず、輪郭送り速度で次の開始点まで移動
				THOUSANDS(千の位):	予備
				TEN THOUSANDS(万の位):	輪郭を加工
					0 = 輪郭を前方に加工
					1 = 輪郭を後方に加工 後方に加工時の制約: - 最大で 170 個の輪郭要素 (面取りまたは丸み付けを含む)。 (X/Y) と F 平面の値だけが評価されます。

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
12		<_RL>	INT	加工方向
				40 = 輪郭の中心(G40、アプローチと後退:直線または垂直)
				41 = 輪郭の左側(G41、アプローチと後退:直線または円弧)
				42 = 輪郭の右側(G42、アプローチと後退:直線または円弧)
13		<_AS1>	INT	輪郭のアプローチ動作
				UNITS(一の位):
				1 = 直線
				2 = 四分円
				3 = 半円
				4 = 垂直方向にアプローチおよび後退
				TENS(十の位):
				0 = 平面での最後の動作
				1 = 空間での最後の動作
14	L1	<_LP1>	REAL	移動軌跡または移動半径(inc、符号なしで入力)
15	FZ	<_FF3>	REAL	中間軌跡の送り速度(輪郭の場合と同様に G94/G95)
16		<_AS2>	INT	輪郭の移動動作(垂直移動/後退以外)
				UNITS(一の位):
				1 = 直線
				2 = 四分円
				3 = 半円
				TENS(十の位):
				0 = 平面での最後の動作
				1 = 空間での最後の動作
17	L2	<_LP2>	REAL	後退軌跡または後退半径(inc、符号なしで入力)
18		<_UMODE>	INT	予備
19	FS	<_FS>	REAL	面取り幅(inc)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
20	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs または inc)、<_AMODE>を参照
21		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)
				UNITS(一の位): 予備
				TENS(十の位): 予備
				HUNDREDS(百の位): 加工または開始点の計算のみのどちらかを選択します。
				0 = 互換性モード
				1 = 通常の加工
22		<_DMODE>	INT	表示モード
				UNITS(一の位): 加工平面 G17/18/19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
				TENS(十の位): 送り速度タイプ:面と深さの送り速度の G グループ(G94/G95)
				0 = 互換性モード
				1 = サイクル呼び出し前と同じ G 命令。面と深さの送り速度と同じ G94/G95
				THOUSANDS(千の位):
				0 = 互換性モード:輪郭名称は<_KNAME>にあります。
				1 = 輪郭名称は CYCLE62 でプログラム指令され、_SC_CONT_NAME に転送されます。

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
23		<_AMODE>	INT	代替モード
				UNITS(一の位):
				終点 Z1 (<_DP>)
				0 = アブソリュート(互換性モード)
				1 = インCREMENTAL
				TENS(十の位):
				平面切り込みの単位
				0 = mm、inch
				1 = 予備
		<_AMODE>	HUNDREDS(百の位):	面取りの切り込み深さ(<_ZFS>)
				0 = アブソリュート
				1 = インCREMENTAL

注記

次の転送パラメータが間接的に(パラメータとして)プログラム指令されている場合、入力画面はリセットされません。

<_VARI>, <_RL>, <_AS1>, <_AS2>, <_UMODE>, <_GMODE>, <_DMODE>, <_AMODE>

4.26.1.17 CYCLE76 - 長方形スゴット

構文

CYCLE76(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_LENG>, <_WID>, <_CRAD>, <_PA>, <_PO>, <_STA>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>, <_VARI>, <_AP1>, <_AP2>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<_RTP>	REAL	イニシャル点(abs)
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
4	Z1	<_DP>	REAL	スピゴットの深さ(abs)
5		<_DPR>	REAL	Z0 を基準にしたスピゴットの深さ(inc)(符号なしで入力)
6	L	<_LENG>	REAL	スピゴットの長さ、<_GMODE>を参照(符号なしで入力)
7	W	<_WID>	REAL	スピゴットの幅、<_GMODE>を参照(符号なしで入力)
8	R	<_CRAD>	REAL	スピゴットのコーナ半径(符号なしで入力)
9	X0	<_PA>	REAL	平面の 1 番目の軸のスピゴットの基準点(abs)
10	Y0	<_PO>	REAL	平面の 2 番目の軸のスピゴットの基準点(abs)
11	α0	<_STA>	REAL	回転角度、長手軸(L)と平面の 1 番目の軸との間の角度
12	DZ	<_MID>	REAL	最大切り込み深さ(inc; 符号なしで入力)
13	UXY	<_FAL>	REAL	平面の仕上げ代(inc)、輪郭端面の削り代
14	UZ	<_FALD>	REAL	仕上げ代の深さ(inc)、底面の削り代(符号なしで入力)
15	FX	<_FFP1>	REAL	輪郭の送り速度
16	FZ	<_FFD>	REAL	深さ切り込み速度
17		<_CDIR>	INT	加工方向(符号なしで入力)
				UNITS(一の位):
				0 = ダウンカット
				1 = アップカット
18		<_VARI>	INT	加工タイプ
				UNITS(一の位):
				1 = 荒削り
				2 = 仕上げ
				5 = 面取り
19	L1	<_AP1>	REAL	スピゴット素材の長さ
20	W1	<_AP2>	REAL	スピゴット素材の幅
21	FS	<_FS>	REAL	面取り幅(inc)
22	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs または inc)、<_AMODE>を参照

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
23		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)
				UNITS(一の位): 予備
				TENS(十の位): 予備
				HUNDREDS(百の位): 加工または開始点の計算のみのどちらかを選択します。
				0 = 互換性モード
				1 = 通常の加工
				THOUSANDS(千の位): 中心またはコーナ基準のスピゴットの位置指令方法
				0 = 互換性モード
				1 = 中心による位置指令方法
				2 = コーナ点の位置指令方法、スピゴット +L +W
				3 = コーナ点の位置指令方法、スピゴット -L +W
				4 = コーナ点の位置指令方法、スピゴット +L -W
				5 = コーナ点の位置指令方法、スピゴット -L -W
				TEN THOUSANDS(万の位): 全て加工または再加工
				0 = 互換性モード
				1 = 全て加工
				2 = 後加工

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
24		<_DMODE>	INT	表示モード
				UNITS(一の位):
				加工平面 G17/18/19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
				TENS(十の位): --- 予備
				HUNDREDS(百の位): --- 予備
				THOUSANDS(千の位): --- 予備
25		<_AMODE>	INT	TEN THOUSANDS(万の位): サイクル画面の機能スケーリング (ページ 1368)
				0 = 入力:全て
				1 = 入力:簡易
				代替モード
				UNITS(一の位):
				最終深さ Z1 (DP)
				0 = 互換性
				1 = インクレメンタル
				2 = アブソリュート
				TENS(十の位): 予備
				HUNDREDS(百の位):
				面取りの切り込み深さ(ZFS)
				0 = アブソリュート
				1 = インクレメンタル

4.26.1.18 CYCLE77 - 円形スピゴット

構文

```
CYCLE77(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_DPR>, <_CDIAM>, <_PA>,
<_PO>, <_MID>, <_FAL>, <_FALD>, <_FFP1>, <_FFD>, <_CDIR>,
<_VARI>, <_AP1>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>, <_AMODE>)
```


4.26 サイクルを外部的にプログラミング

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<_RTP>	REAL	イニシャル点(abs)
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<_DP>	REAL	スピゴットの深さ(abs)
5		<_DPR>	REAL	Z0 を基準にしたスピゴットの深さ(inc)(符号なしで入力)
6	Ø	<_CDIAM>	REAL	スピゴットの直径(符号なしで入力)
7	X0	<_PA>	REAL	平面の 1 番目の軸のスピゴットの基準点(abs)
8	Y0	<_PO>	REAL	平面の 2 番目の軸のスピゴットの基準点(abs)
9	DZ	<_MID>	REAL	最大切り込み深さ(inc; 符号なしで入力)
10	UXY	<_FAL>	REAL	平面の仕上げ代(inc)、輪郭端面の削り代
11	UZ	<_FALD>	REAL	仕上げ代の深さ(inc)、底面の削り代(符号なしで入力)
12	FX	<_FFP1>	REAL	輪郭の送り速度
13	FZ	<_FFD>	REAL	深さ切り込み速度
14		<_CDIR>	INT	加工方向(符号なしで入力)
				UNITS(一の位):
				0 = ダウンカット 1 = アップカット
15		<_VARI>	INT	加工タイプ
				UNITS(一の位):
				加工タイプ
				1 = 最終加工代まで荒削り 2 = 仕上げ(仕上げ代 X/Y/Z=0) 5 = 面取り
16	Ø1	<_AP1>	REAL	スピゴット素材の直径
17	FS	<_FS>	REAL	面取り幅(inc)
18	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs または inc)、<_AMODE>を参照

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
19		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)
				UNITS(一の位): 予備
				TENS(十の位): 予備
				HUNDREDS(百の位): 加工または開始点の計算のみのどちらかを選択します。
				0 = 互換性モード
				1 = 通常の加工
				THOUSANDS(千の位): 予備
				TEN THOUSANDS(万の位): 全て加工/再加工
				0 = 互換性モード(以前と同様に<_AP1>を処理)
				1 = 全て加工
				2 = 後加工
20		<_DMODE>	INT	表示モード
				UNITS(一の位): 加工平面 G17/18/19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
				TENS(十の位): --- 予備
				HUNDREDS(百の位): --- 予備
				THOUSANDS(千の位): --- 予備
				TEN THOUSANDS(万の位): サイクル画面の機能スケーリング (ページ 1368)
				0 = 入力:全て
				1 = 入力:簡易

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
21		<_AMODE>	INT	代替モード
				UNITS(一の位):
				最終深さ Z1 (DP)
				0 = アブソリュート(互換性モード)
				1 = インCREMENTAL
				2 = アブソリュート
				TENS(十の位):
				予備
				HUNDREDS(百の位):
				面取りの切り込み深さ(ZFS)
				0 = アブソリュート
				1 = インCREMENTAL

4.26.1.19 CYCLE78 - ドリルねじフライス削り

構文

```
CYCLE78(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_ADPR>, <_FDPR>,
<_LDPR>, <_DIAM>, <_PIT>, <_PITA>, <_DAM>, <_MDEP>, <_VARI>,
<_CDIR>, <_GE>, <_FFD>, <_FRDP>, <_FFR>, <_FFP2>, <_FFA>,
<_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<_RTP>	REAL	イニシャル点(abs)
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<_DP>	REAL	最終穴あけ深さ(abs/inc)、<_AMODE>を参照
5		<_ADPR>	REAL	減速された穴あけ送り速度の予備穴あけ深さ (inc)、<_VARI> TEN THOUSANDS(万の位)で有効
6	D	<_FDPR>	REAL	最大切り込み深さ(inc) D ≥ Z1 ⇒ 最終穴あけ深さまで一回で切り込み D < Z1 ⇒ 複数回の切り込みによる深穴サイクルと切り屑除去
7	ZR	<_LDPR>	REAL	FR 送り速度での貫通穴あけ時の残りの穴あけ深さ(inc)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
8	Ø	<_DIAM>	REAL	ねじの公称直径
9	P	<_PIT>	REAL	数値で表したピッチ
10		<_PITA>	INT	ねじピッチ P の評価
				1 = ピッチ(mm/rev)
				2 = ピッチ(ねじ山数/inch)
				3 = ピッチ(inch/rev)
				4 = MODULE で表したピッチ
11	DF	<_DAM>	REAL	各々の追加切り込みの絶対値/パーセント値(通減量)、<_AMODE>を参照
12	V1	<_MDEP>	REAL	最小切り込み(inc)、通減パーセンテージ値の場合のみ有効

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
13		<_VARI>	INT	加工タイプ	
				UNITS(一の位):	予備
				TENS(十の位):	ねじ切り加工前の切り屑除去
					0 = ねじフライス削り前の切り屑除去なし(最終穴あけ深さでのみ有効)
					1 = ねじフライス削り前の切り屑除去(最終穴あけ深さでのみ有効)
				HUNDREDS(百の位):	右/左ねじ
					0 = 右ねじ
					1 = 左ねじ
				THOUSANDS(千の位):	穴あけ送り速度での残りの穴あけ深さ
					0 = 穴あけ送り速度 FR での残りの穴あけ深さなし
					1 = 穴あけ送り速度 FR での残りの穴あけ深さ
				TEN THOUSANDS(万の位):	減速された送り速度での予備穴あけ
					0 = 減速された送り速度での予備穴あけなし
					1 = 減速された送り速度での予備穴あけ 予備穴あけ送り速度 = 0.3 F1(F1 < 0.15 mm/rev の場合) 予備穴あけ送り速度 = 0.1 mm/rev(F1 ≥ 0.15 mm/rev の場合)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味			
14		<_CDIR>	INT	加工方向	0 =	ダウンカット	
					1 =	アップカット	
					4 =	アップカット+ダウンカット (荒削りと仕上げの組み合わせ)	
15	Z2	<_GE>	REAL	ねじフライス削り前の後退距離(inc)			
16	F1	<_FFD>	REAL	穴あけ送り速度(mm/min、in/min、または mm/rev)			
17	FR	<_FRDP>	REAL	残りの穴あけ深さに対応した穴あけ送り速度(mm/min または mm/rev)			
18	F2	<_FFR>	REAL	ねじフライス削り時の送り速度(mm/min または mm/tooth)			
19	FS	<_FFP2>	REAL	<_CDIR> =4 の仕上げ送り速度(mm/min または mm/刃)			
20		<_FFA>	INT	送り速度の評価			
				UNITS(一の位):		穴あけ送り速度 F1	
				TENS(十の位):		残りの穴あけ深さに対応した穴あけ送り速度 FR	
				HUNDREDS(百の位):		ねじフライス削り時の送り速度 F2	
				THOUSANDS(千の位):		仕上げ送り速度 FS	
21		<_PITM>	STRING[15]	ピッチ入力のマーカとなる文字列(インタフェースのみ) ¹⁾			
22		<_PTAB>	STRING[20]	ねじテーブルの文字列(“”、「ISO」、「BSW」、「BSP」、「UNC」) (インタフェースのみ) ¹⁾			
23		<_PTABA>	STRING[20]	ねじテーブルからの選択のための文字列(例: 「M 10」、「M 12」、...) (インタフェースのみ) ¹⁾			
24		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)、予約済み			
25		<_DMODE>	INT	表示モード			
				UNITS(一の位):	加工平面 G17/18/19		
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効	
					1 =	G17 (サイクルでのみ有効)	
					2 =	G18 (サイクルでのみ有効)	
					3 =	G19 (サイクルでのみ有効)	

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
26		<_AMODE>	INT	代替モード
				UNITS(一の位):
				穴あけ深さ = 最終穴あけ深さ Z1 (abs/inc)
				0 = アブソリュート
				1 = インクレメンタル
				TENS(十の位):
				各々の追加切り込みの絶対値/パーセント値(逓減量)
				0 = 絶対値
				1 = パーセント値(0.001~100%)

注記

¹⁾パラメータ 21、22、23 は、入力画面ねじテーブルでのねじの選択にのみ使用されます。ねじテーブルは、サイクル運転時にサイクル定義を介してアクセスすることはできません。

4.26.1.20 CYCLE79 - 多角形

構文

```
CYCLE79(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_NUM>, <_SWL>, <_PA>,
<_PO>, <_STA>, <_RC>, <_AP1>, <_MIDA>, <_MID>, <_FAL>, <_FALD>,
<_FFP1>, <_CDIR>, <_VARI>, <_FS>, <_ZFS>, <_GMODE>, <_DMODE>,
<_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<_RTP>	REAL	イニシャル点(abs)
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<_DP>	REAL	多角形の深さ(abs/inc)、<_AMODE>を参照
5	N	<_NUM>	INT	端面の数(1...n)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
6	SW/L	<_SWL>	REAL	二面幅または辺の長さ(<_VARI>による) (「SW」は二面幅、「L」は辺の長さ) 偶数の辺と単数の辺の場合は、二面幅のみ		
7	X0	<_PA>	REAL	スピゴットの基準点、1 番目の軸(abs)		
8	Y0	<_PO>	REAL	スピゴットの基準点、2 番目の軸(abs)		
9	α0	<_STA>	REAL	回転角度、1 番目の軸(X 軸)に対する端面の中心		
10	R1/FS1	<_RC>	REAL	<_NUM>によるコーナ丸み付け>2 (半径/面取り、<_AMODE>を参照) (inc、符号なしで入力) (「R1」は丸み付け、「FS1」は面取り)		
11	Ø	<_AP1>	REAL	スピゴット素材の直径		
12	DXY	<_MIDA>	REAL	最大切り込み幅(単位については、<_AMODE>を参照)		
13	DZ	<_MID>	REAL	最大切り込み深さ		
14	UXY	<_FAL>	REAL	平面の仕上げ代		
15	UZ	<_FALD>	REAL	深さの仕上げ代		
16	F	<_FFP1>	REAL	加工送り速度		
17		<_CDIR>	INT	加工方向	0 =	ダウンカット
					1 =	アップカット
18		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	加工タイプ	
					1 =	荒削り
					2 =	仕上げ
					3 =	端面の仕上げ
					5 =	面取り
				TENS(十の位):	二面幅または辺の長さ	
					0 =	二面幅
					1 =	辺の長さ
19	FS	<_FS>	REAL	面取り幅(inc)		
20	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs または inc)、<_AMODE>を参照		

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
21		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)	
				UNITS(一の位):	予備
				TENS(十の位):	予備
				HUNDREDS(百の位):	加工または開始点の計算のみのどちらかを選択します。
				1 =	通常の加工
22		<_DMODE>	INT	表示モード	
				UNITS(一の位):	加工平面 G17/18/19
					0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 = G17 (サイクルでのみ有効)
					2 = G18 (サイクルでのみ有効)
					3 = G19 (サイクルでのみ有効)
				TENS(十の位):	--- 予備
				HUNDREDS(百の位):	--- 予備
				THOUSANDS(千の位):	--- 予備
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1368)
					0 = 入力:全て
					1 = 入力:簡易

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
23		<_AMODE>	INT	代替モード
				UNITS(一の位):
				最終深さ(<_DP>)
				0 = アブソリュート
				1 = インCREMENTAL
				TENS(十の位):
				平面切り込みの単位(<_MIDA>)
				0 = mm
				1 = 工具直径の%
				HUNDREDS(百の位):
				面取りの切り込み深さ(<_ZFS>)
				0 = アブソリュート
				1 = インCREMENTAL
				THOUSANDS(千の位):
				コーナ丸み付け(<_RC>)
				0 = 半径
				1 = 面取り

4.26.1.21 CYCLE81 - 穴あけ、センタリング

構文

CYCLE81 (<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>, <_DMODE>, <_AMODE>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<RFP>	REAL	基準点(abs)
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1/Ø	<DP>	REAL	穴あけ深さ(abs)/センタリング径(abs)、<_GMODE>を参照
5	Z1	<DPR>	REAL	穴あけ深さ(inc)
6	DT	<DTB>	REAL	最終穴あけ深さでのドウェル時間、<_AMODE>を参照

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
7		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)
				UNITS(一の位): 予備
				TENS(十の位): 深さ/直径を基準にしたセンタリング
				0 = 互換性、深さ
				1 = Diameter
8		<_DMODE>	INT	表示モード
				UNITS(一の位): 加工平面 G17/G18/G19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
9		<_AMODE>	INT	代替モード
				UNITS(一の位): 穴あけ深さ Z1 (abs/inc)
				0 = 互換性、DP/DPR のプログラム指令から
				1 = インクレメンタル
				2 = アブソリュート
				TENS(十の位): 最終穴あけ深さ DT でのドウェル時間(秒(s))/回転数(rev)の単位
				0 = 互換性、DTB の符号から(> 0 は秒(s)、< 0 は回転数(rev))
				1 = 秒(s)
				2 = 回転数(rev)

4.26.1.22 CYCLE82 - 穴あけ、座ぐり

構文

```
CYCLE82 (<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <_GMODE>,
<_DMODE>, <_AMODE>, <_VARI>, <S_ZA>, <S_FA>, <S_ZD>, <S_FD>)
```

4.26 サイクルを外部的にプログラミング

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<RFP>	REAL	基準点(abs)
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<DP>	REAL	穴あけ深さ(abs)、<_AMODE>を参照
5	Z1	<DPR>	REAL	穴あけ深さ(inc)、<_AMODE>を参照
6	DT	<DTB>	REAL	最終穴あけ深さでのドウェル時間、<_AMODE>を参照
7		<_GMODE>	INT	ジオメトリックモード(プログラム指令されたジオメトリックデータの評価)
				UNITS(一の位): 予備
				TENS(十の位): 先端/シャンクを基準にした穴あけ深さ
				0 = 互換性、先端 1 = シャンク
8		<_DMODE>	INT	表示モード
				UNITS(一の位): 加工平面 G17/G18/G19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
				TENS(十の位): 予備
				HUNDREDS(百の位): 予備
				THOUSANDS(千の位): 予備
				TEN THOUSANDS(万の位): サイクル画面の機能スケーリング (ページ 1368)
				0 = 入力:全て
				1 = 入力:ベーシック

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
9		<_AMODE>	INT	代替モード
				UNITS(一の位):
				0 = 互換性、DP/DPR のプログラム指令から
				1 = インCREMENTAL
				2 = アブソリュート
				TENS(十の位):
				最終穴あけ深さでのドウェル時間 DT(秒(s)/回転数(rev)の単位)
				0 = 互換性、DT の符号から(> 0 は秒(s)、< 0 は回転数(rev))
				1 = 秒(s)
				2 = 回転数(rev)
				HUNDREDS(百の位):
				穴あけ深さ DZA abs/inc
				0 = インCREMENTAL
				1 = アブソリュート
				THOUSANDS(千の位):
				予備穴あけ送り速度の評価
				0 = 穴あけ送り速度の%として
				1 = F/min
				2 = F/rev
				TEN THOUSANDS(万の位):
				残りの穴あけ深さ ZD abs/inc
				0 = インCREMENTAL
				1 = アブソリュート
				HUNDRED THOUSANDS(十万の位):
				残りの穴あけ送り速度の評価
				0 = 穴あけ送り速度の%として
				1 = F/min
				2 = F/rev

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
10		<_VARI>	INT	予備穴あけ/貫通穴あけ加工タイプ
				UNITS(一の位): 予備
				TENS(十の位): 予備
				HUNDREDS(百の位): 予備
				THOUSANDS(千の位): 貫通穴あけ
				0 = 貫通穴あけ「なし」
				1 = 貫通穴あけ「あり」
				TEN THOUSANDS(万の位): 前加工
				0 = 予備穴あけ「なし」
				1 = 予備穴あけ「あり」
11	ZA	<S_ZA>	REAL	レファレンス点またはアブソリュートを基準にしたインクレメンタル予備穴あけ深さ(<_AMODE> HUNDREDS を参照)
12	FA	<S_FA>	REAL	予備穴あけ送り速度(値としてまたは%単位で)(<_AMODE> THOUSANDS と組み合わせて)
13	ZD	<S_ZD>	REAL	最終穴あけ深さまたはアブソリュートを基準にしたインクレメンタルの残りの穴あけ深さ(<_AMODE> TEN THOUSANDS)を参照)
14	FD	<S_FD>	REAL	残りの穴あけ送り速度(値としてまたは%単位で)(<_AMODE> HUNDRED THOUSANDS と組み合わせて)

4.26.1.23 CYCLE83 - 深穴ドリル 1

構文

```
CYCLE83(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <FDEP>, <FDPR>,
<_DAM>, <DTB>, <DTS>, <FRF>, <VARI>, <_AXN>, <_MDEP>, <_VRT>,
<_DTD>, <_DIS1>, <_GMODE>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<RFP>	REAL	基準点(abs)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<DP>	REAL	最終穴あけ深さ(abs)、<_AMODE>を参照
5	Z1	<DPR>	REAL	最終穴あけ深さ(inc)、<_AMODE>を参照
6	D	<FDEP>	REAL	1 番目の穴あけ深さ(abs)、<_AMODE>を参照
7	D	<FDPR>	REAL	1 番目の穴あけ深さ(inc)、<_AMODE>を参照
8	DF	<_DAM>	REAL	各々の追加切り込みの通減値/パーセント値。<_AMODE>を参照。
9	DTB	<DTB>	REAL	穴あけ深さでのドウェル時間、<_AMODE>を参照
10	DTS	<DTS>	REAL	開始点でのドウェル時間(切り屑除去のみ)、<_AMODE>を参照
11	FD1	<FRF>	REAL	最初の切り込み送り速度のパーセント値、<_AMODE>を参照
12		<VARI>	INT	加工タイプ UNITS(一の位): 切り屑破断/切り屑除去 0 = 切り屑切断 1 = 切り屑除去
13		<_AXN>	INT	工具軸 0 = 3 番目のジオメトリ軸 1 = 1 番目のジオメトリ軸 2 = 2 番目のジオメトリ軸 > 2 3 番目のジオメトリ軸
14	V1	<_MDEP>	REAL	最小切り込み(通減パーセント値のみ)
15	V2	<_VRT>	REAL	各加工ステップ終了後の後退量(切り屑破断の場合のみ) > 0 可変 後退距離 0 = 初期値 1 mm
16	DT	<_DTD>	REAL	最終穴あけ深さでのドウェル時間、<_AMODE>を参照
17	V3	<_DIS1>	REAL	リミット距離(切り屑除去の場合のみ)、<_AMODE>を参照
18		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価) UNITS(一の位): 予備 TENS(十の位): 先端/シャンクを基準にした穴あけ深さ 0 = 先端 1 = シャンク

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
19		<_DMODE>	INT	表示モード	
				UNITS(一の位):	加工平面 G17/G18/G19
					0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 = G17 (サイクルでのみ有効)
					2 = G18 (サイクルでのみ有効)
					3 = G19 (サイクルでのみ有効)
				TENS(十の位):	--- 予備
				HUNDREDS(百の位):	--- 予備
				THOUSANDS(千の位):	--- 予備
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1368)
					0 = 入力:全て
					1 = 入力:簡易

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
20		<_AMODE>	INT	代替モード
				UNITS(一の位):
				穴あけ深さ = 最終穴あけ深さ Z1 (abs/inc)
				0 = 互換性、<DE>/<DPR>のプログラム指令から
				1 = インクレメンタル
				2 = アブソリュート
				TENS(十の位):
				穴あけ深さでのドウェル時間 DTB(秒(s)/回転数(rev))
				0 = 互換性、DTB の符号から(> 0 は秒(s)、< 0 は回転数(rev))
				1 = 秒(s)
				2 = 回転数(rev)
				HUNDREDS(百の位):
				DTS の開始点でのドウェル時間(秒(s)/回転(rev))
				0 = 互換性、DTS の符号から(> 0 は秒(s)、< 0 は回転数(rev))
				1 = 秒(s)
				2 = 回転数(rev)
				THOUSANDS(千の位):
				最終穴あけ深さでのドウェル時間 DTD(秒(s)/回転数(rev)の単位)
				0 = 互換性、DTD の符号から(> 0 は秒(s)、< 0 は回転数(rev))
				1 = 秒(s)
				2 = 回転数(rev)
				TEN THOUSANDS(万の位):
				1 番目の穴あけ深さ D (abs/inc)
				0 = 互換性、<FDEPF>/<DPR>のプログラム指令から
				1 = インクレメンタル
				2 = アブソリュート
				HUNDRED THOUSANDS(十万の位):
				各々の追加切り込みの通減値/パーセント値<_DAM>

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
				0 = 互換性、<_DAM>の符号から(> 0 は通減値、< 0 は係数 0.001~1.0) 1 = 通減値 2 = パーセント値(0.001~100%)
				ONE MILLION(百万の位): リミット距離 V3 自動/手動 0 = 互換性、<_DIS1>の符号から(= 0 は自動、> 0 は手動) 1 = 自動(サイクルで計算) 2 = 手動(プログラム指令値)
				TEN MILLION(千万の位): 係数/パーセント値としての最初の切り込み送り速度係数 <FRF> 0 = 互換性、係数として(0.001~1.0、FRF = 0 は 100%) 1 = パーセント値 (0.001~999.999%)

4.26.1.24 CYCLE84 - フローティングチェックなしのタッピング

構文

```
CYCLE84 (<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDAC>, <MPIT>,
<PIT>, <POSS>, <SST>, <SST1>, <AXN>, <_PITA>, <_TECHNO>,
<_VARI>, <_DAM>, <_VRT>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>,
<_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<RFP>	REAL	基準点(abs)
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<DP>	REAL	穴あけ深さ = 最終穴あけ深さ(abs)、<_AMODE>を参照

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味			
5	Z1	<DPR>	REAL	穴あけ深さ = 最終穴あけ深さ(inc)、<_AMODE>を参照			
6	DT	<DTB>	REAL	穴あけ深さでのドウェル時間(s)			
7	SDE	<SDAC>	INT	サイクル終了後の回転方向			
8		<MPIT>	REAL	「ISO メトリック」のねじサイズのみ(ピッチは運転時に内部で計算されます)			
9	P	<PIT>	REAL	数値で表したピッチ、単位については <_PITA>を参照			
10	αS ¹⁾	<POSS>	REAL	主軸オリエンテーションの主軸位置			
11	S	<SST>	REAL	タッピングの主軸速度			
12	SR	<SST1>	REAL	後退時の主軸速度			
13		<_AXN>	INT	穴あけ軸	0 =	3 番目のジオメトリ軸	
					1 =	1 番目のジオメトリ軸	
					2 =	2 番目のジオメトリ軸	
					≥ 3 =	3 番目のジオメトリ軸	
14		<_PITA>	INT	ピッチ単位(<PIT>と<MPIT>の評価)			
					0 =	ピッチ(mm)	- 評価<MPIT>/<PIT>
					1 =	ピッチ(mm)	- 評価<PIT>
					2 =	ピッチ(TPI)	- <PIT>の評価(ねじ山数/inch)
					3 =	ピッチ(inch)	- 評価<PIT>
					4 =	MODULUS	- 評価<PIT>

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
15		<_TECHNO>	INT	用途 ¹⁾
				UNITS(一の位):
				イグザクトストップ動作
				0 = サイクル呼び出し前と同じイグザクトストップ動作が有効
				1 = イグザクトストップ G601
				2 = イグザクトストップ G602
				3 = イグザクトストップ G603
				TENS(十の位):
				フィードフォワード制御
				0 = サイクル呼び出し前と同じフィードフォワード制御あり/なしが有効
				1 = フィードフォワード制御あり FFWON
				2 = フィードフォワード制御なし FFWOF
				HUNDREDS(百の位):
				加減速
				0 = サイクル呼び出し前と同じ SOFT/BRISK/DRIVE が有効
				1 = 加々速度制限あり SOFT
				2 = 加々速度制限なし BRISK
				3 = 加速度で減速 DRIVE
16		<_VARI>	INT	加工タイプ
				UNITS(一の位):
				0 = 1 回切削
				1 = 切り屑破断(深穴タッピング)
				2 = 切り屑除去(深穴タッピング)
				THOUSANDS(千の位):
17	d	<_DAM>	REAL	ISO/シーメンスモードは入力画面と無関係
				0 = ISO 互換からの呼び出し
				1 = シーメンスコンテキストからの呼び出し
17	d	<_DAM>	REAL	最大切り込み深さ(切り屑除去/切り屑破断のみ)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
18	V2	<_VRT>	REAL	各加工ステップ終了後の後退量(切り屑破断の場合のみ)、<_AMODE>を参照
19		<_PITM>	STRING[15]	ピッチ入力のマーカとなる文字列 ²⁾
20		<_PTAB>	STRING[5]	ねじテーブルの文字列(″、″ISO″、″BSW″、″BSP″、″UNC″) ²⁾
21		<_PTABA>	STRING[20]	ねじテーブルからの選択のための文字列(例:″M 10″、″M 12″、...) ²⁾
22		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)
				UNITS(一の位): 予備
				TENS(十の位): 予備
23		<_DMODE>	INT	表示モード
				UNITS(一の位): 加工平面 G17/G18/G19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
				TENS(十の位): 予備
				HUNDREDS(百の位): 予備
				THOUSANDS(千の位): 互換性モード(再コンパイル入力画面のみ)、MD 52216 ビット 0 = 1 の場合 ¹⁾
				0 = テクノロジパラメータが表示されます(互換性):TECHNO パラメータが有効
				1 = テクノロジパラメータは表示されません:″サイクル呼び出し前と同じ″プログラム値が有効
				TEN THOUSANDS(万の位): サイクル画面の機能スケーリング (ページ 1368)
				0 = 入力:全て
				1 = 入力:簡易

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味			
24		<_AMODE>	INT	代替モード			
				UNITS(一の位):	穴あけ深さ = 最終穴あけ深さ Z1 (abs/inc)		
					0 =	互換性、 <DP>/<DPR>のプログラム指令から	
					1 =	インクレメンタル	
					2 =	アブソリュート	
				TENS(十の位):	予備		
				HUNDREDS(百の位):	予備		
				THOUSANDS(千の位):	ねじの回転方向、右/左		
					0 =	互換性、PIT/MPTI の符号から	
					1 =	右側	
					2 =	左側	
				TEN THOUSANDS(万の位):	予備		
				HUNDRED THOUSANDS(十万の位):	予備		
				ONE MILLION(百万の位):	各加工ステップ終了後の後退量 V2、手動/自動		
					0 =	互換性、<_VRT> のプログラム指令から(> 0 は可変値、≤ 0 は標準値 1 mm/0.0394 inch)	
					1 =	自動(標準値 1 mm / 0.0394 inch)	
					2 =	手動(V2 以下となるようプログラム指令)	

¹⁾テクノロジーフィールドは、SD52216 \$MCS_FUNCTION_MASK_DRILL の設定に応じて非表示になる場合があります。

²⁾パラメータ 19、20、21 は、入力画面ねじテーブルでのねじの選択にのみ使用されます。ねじテーブルは、サイクル運転時にサイクル定義を介してアクセスすることはできません。

4.26 サイクルを外部的にプログラミング

4.26.1.25 CYCLE85 - リーマ加工

構文

CYCLE85 (<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <FFR>, <RFF>, <_GMODE>, <_DMODE>, <_AMODE>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<RFP>	REAL	基準点(abs)
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<DP>	REAL	穴あけ深さ(abs)、<_AMODE>を参照
5	Z1	<DPR>	REAL	穴あけ深さ(inc)、<_AMODE>を参照
6	DT	<DTB>	REAL	最終穴あけ深さでのドウェル時間、<_AMODE>を参照
7	F	<FFR>	REAL	送り速度
8	FR	<RFF>	REAL	後退時の送り速度
9		<_GMODE>	INT	予備
10		<_DMODE>	INT	表示モード
				UNITS(一の位):
				加工平面 G17/G18/G19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
11		<_AMODE>	INT	代替モード(穴あけ)
				UNITS(一の位):
				穴あけ深さ Z1 (abs/inc)
				0 = 互換性、DP/DPR のプログラム指令から
				1 = インクレメンタル
				2 = アブソリュート
				TENS(十の位):
				最終穴あけ深さでのドウェル時間 DT(秒(s))/回転数(rev))
				0 = 互換性、DT の符号から(> 0 は秒(s)、< 0 は回転数(rev))
				1 = 秒(s)
				2 = 回転数(rev)

4.26.1.26 CYCLE86 - ボーリング

構文

CYCLE86(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDIR>, <RPA>, <RPO>, <RPAP>, <POSS>, <_GMODE>, <_DMODE>, <_AMODE>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<RFP>	REAL	基準点(abs)
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<DP>	REAL	穴あけ深さ(abs)、<_AMODE>を参照
5	Z1	<DPR>	REAL	穴あけ深さ(inc)、<_AMODE>を参照
6	DT	<DTB>	REAL	最終穴あけ深さでのドウェル時間、<_AMODE>を参照
7	DIR	<SDIR>	INT	主軸回転方向
				3 = M3 4 = M4
8	Dx	<RPA>	REAL	X 方向の戻し距離

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
9	DY	<RPO>	REAL	Y 方向の戻し距離
10	DZ	<RPAP>	REAL	Z 方向の戻し距離
11	SPOS	<POSS>	REAL	戻し時の主軸位置(主軸オリエンテーションの場合、°単位)
12		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)
				UNITS(一の位): 戻しモード
				0 = 戻し、互換性
				1 = 輪郭から戻しをおこなわない
13		<_DMODE>	INT	表示モード
				UNITS(一の位): 加工平面 G17/G18/G19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
14		<_AMODE>	INT	代替モード
				UNITS(一の位): 穴あけ深さ Z1 (abs/inc)
				0 = 互換性、<DP>/<DPR> のプログラム指令から
				1 = インクレメンタル
				2 = アブソリュート
				TENS(十の位): 最終穴あけ深さ DT でのドウェル時間(秒(s)/回転数(rev))
				0 = 互換性、DT の符号から(> 0 は秒(s)、< 0 は回転数(rev))
				1 = 秒(s)
				2 = 回転数(rev)

4.26 サイクルを外部的にプログラミング

4.26.1.27 CYCLE92 - 突切り

構文

```
CYCLE92 (<_SPD>, <_SPL>, <_DIAG1>, <_DIAG2>, <_RC>, <_SDIS>,
<_SV1>, <_SV2>, <_SDAC>, <_FF1>, <_FF2>, <_SS2>, <_DIAGM>,
<_VARI>, <_DN>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	X0	<_SPD>	REAL	基準点(abs、常に直径)		
2	Y0	<_SPL>	REAL	基準点(abs)		
3	X1	<_DIAG1>	REAL	減速時の深さ、<_AMODE> (UNITS)を参照		
4	X2	<_DIAG2>	REAL	最終深さ、<_AMODE> (TENS)を参照		
5	R/FS	<_RC>	REAL	丸み付け状態または面取り幅、<_AMODE> (THOUSANDS)を参照		
6	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)		
7	S	<_SV1>	REAL	一定主軸速度、<_AMODE> (TEN THOUSANDS)を参照		
	V			定切削速度		
8	SV	<_SV2>	REAL	定切削速度時の最大速度		
9	DIR	<_SDAC>	INT	主軸回転方向	3 =	M3 の場合
					4 =	M4 の場合
10	F	<_FF1>	REAL	減速時の深さまで切り込み		
11	FR	<_FF2>	REAL	最終深さまでの減速切り込み		
12	SR	<_SS2>	REAL	最終深さまでの減速速度		
13	XM	<_DIAGM>	REAL	部品固定具引き抜き深さ(abs、常に直径)		

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
14		<_VARI>	INT	加工タイプ
				UNITS(一の位):
				後退
				0 = <_SPD> + <_SDIS>へ後退
				1 = 最後での後退なし
				TENS(十の位):
15		<_DN>	INT	部品固定具
				0 = なし、M 命令を実行しない
				1 = あり、CUST_TECHCYC(101)から呼び出し- ドロアを開く、CUST_TECHCYC(102)- ドロアを閉じる
				工具の 2 番目の刃先の D 番号。プログラム指令されていない場合 ⇒ D+1
				表示モード
				UNITS(一の位):
20		<_DMODE>	INT	加工平面 G17/G18/G19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
21		<_AMODE>	INT	代替モード
				UNITS(一の位):
				減速時の深さ(<_DIAG1>)
				0 = アブソリュート、径方向軸の直径の値
				1 = インクレメンタル、径方向軸の半径の値
				TENS(十の位):
				最終深さ(<_DIAG2>)
				0 = アブソリュート、径方向軸の直径の値
				1 = インクレメンタル、径方向軸の半径の値
				HUNDREDS(百の位):
				予備
				THOUSANDS(千の位):
				丸み付け/面取り(<_RC>)
				0 = 半径
				1 = 面取り
				TEN THOUSANDS(万の位):
				主軸速度/切削速度(<_SV1>)
				0 = 一定主軸速度
				1 = 定切削速度

4.26.1.28 CYCLE95 - 輪郭切削

構文

CYCLE95 (<NPP>, <MID>, <FALZ>, <FALX>, <FAL>, <FF1>, <FF2>, <FF3>, <_VARI>, <DT>, <DAM>, <_VRT>, <_GMODE>, <_DMODE>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	CON	<NPP>	STRING [140]	輪郭名称
2	d	<MID>	REAL	荒削り時の最大切り込み深さ、 <_GMODE>を参照

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
3	UZ	<FALZ>	REAL	Z の仕上げ代		
4	UX	<FALX>	REAL	X の仕上げ代		
5	U	<FAL>	REAL	輪郭に平行の仕上げ代(両方の軸で有効)		
6	F	<FF1>	REAL	荒削りの送り速度		
7	FY	<FF2>	REAL	レリーフカット切り込み送り速度		
8	FS	<FF3>	REAL	仕上げの送り速度		
9		<_VARI>	INT	加工タイプ		
				UNITS(一の位)と TENS(十の位):		
				1 =	荒削り、長手方向、外部	
				2 =	荒削り、径方向、外部	
				3 =	荒削り、長手方向、内部	
				4 =	荒削り、径方向、内部	
				5 =	仕上げ、長手方向、外部	
				6 =	仕上げ、径方向、外部	
				7 =	仕上げ、長手方向、内部	
				8 =	仕上げ、径方向、内部	
				9 =	全て加工、長手方向、外部	
				10 =	全て加工、径方向、外部	
				11 =	全て加工、長手方向、内部	
				12 =	全て加工、径方向、内部	
				HUNDREDS(百の位):		
0 =	輪郭での丸み付けあり、コーナの削り残しなし					
1 =	輪郭での丸み付けなし					
2 =	前の交点に対してのみ丸み付け、コーナの削り残しが生じるおそれあり					
10	DT	<DT>	REAL	送り中断時のドウェル時間		
11	Di	<DAM>	REAL	送り中断の距離		

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
12	VRT	<_VRT>	REAL	輪郭からの戻し距離		
					0 =	有効な単位(インチまたはメートル)とは無関係に、戻し距離 1 mm が内部的に使用されます。
					> 0 =	戻し距離
13		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)		
				UNITS(一の位):	切り込み深さの評価	
					0 =	切り込み深さは、G グループ DIAMON/ DIAMOF に対応して計算されます
14		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)
				THOUSANDS(千の位):		
					0 =	互換性モード:NPP 内の輪郭名称
					1 =	輪郭名称は CYCLE62 でプログラム指令され、_SC_CONT_NAME に転送されます。

4.26.1.29 CYCLE98 - 連続ねじ

構文

```
CYCLE98 (<_PO1>, <_DM1>, <_PO2>, <_DM2>, <_PO3>, <_DM3>, <_PO4>,
<_DM4>, <APP>, <ROP>, <TDEP>, <FAL>, <_IANG>, <NSP>, <NRC>,
<NID>, <_PP1>, <_PP2>, <_PP3>, <_VARI>, <_NUMTH>, <_VRT>, <_MID>,
<_GDEP>, <_IFLANK>, <_PITA>, <_PITM1>, <_PITM2>, <_PITM3>,
<_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	Z0	<_PO1>	REAL	Z の基準点(abs)
2	X0	<_DM1>	REAL	X の基準点(abs)、直径
3	Z1	<_PO2>	REAL	Z の中間点 1 (abs/inc)、<_AMODE> (UNITS)を参照
4	X1 X1α	<_DM2>	REAL	X の中間点 1 (abs/inc)、<_AMODE> (TENS)を参照、または ねじ傾斜角 1 (-90°~90°) abs では常に直径、inc では常に半径
5	Z2	<_PO3>	REAL	Z の中間点 2 (abs/inc)、<_AMODE> (HUNDREDS)を参照
6	X2 X2α	<_DM3>	REAL	X の中間点 2 (abs/inc)、<_AMODE> (THOUSANDS)を参照、または ねじ傾斜角 2 (-90°~90°) abs では常に直径、inc では常に半径
7	Z3	<_PO4>	REAL	Z の終点、(abs/inc)、 <_AMODE>(TEN THOUSANDS)を参照
8	X3 X3α	<_DM4>	REAL	X の終点、(abs/inc)、<_AMODE> (HUNDRED THOUSANDS)を参照、または ねじ傾斜角 3 (-90°~90°) abs では常に直径、inc では常に半径
9	LW	<APP>	REAL	ねじ切り始め(inc、符号なしで入力)
10	LR	<ROP>	REAL	ねじ切り上げ(inc、符号なしで入力)
11	H1	<TDEP>	REAL	ねじ山深さ(inc、符号なしで入力)
12	U	<FAL>	REAL	X と Z の仕上げ代

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
13	DP	<_IANG>	REAL	距離または角度で表した切り込み斜面、<_AMODE> (ONE MILLION)を参照		
	αP			切り込み斜面は、パラメータ<_VARI>(HUNDREDS)の設定に従って適用されます。		
					<_VARI_HUNDREDS = 0 の定義 - 互換性モード:	
					> 0 =	片面だけを切り込み
					0 =	ねじを垂直切り込み
					< 0 =	面を交互に切り込み
					<_VARI_HUNDREDS<>0 の定義:	
					> 0 =	プラス側の切り込み
					0 =	中央に配置した電源装置
< 0 =	マイナス側の切り込み					
14	α0	<NSP>	REAL	1 番目のねじの開始角度オフセット		
15		<NRC>	INT	荒削り切削の回数、<_VARI> (TEN THOUSANDS)を参照		
16	NN	<NID>	INT	エアーカットの回数		
17	P0	<_PP1>	REAL	ねじの 1 番目の区間のピッチ、<_PITA>を参照		
18	P1	<_PP2>	REAL	ねじの 2 番目の区間のピッチ、<_PITA>を参照		
19	P2	<_PP3>	REAL	ねじの 3 番目の区間のピッチ、<_PITA>を参照		

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
20		<_VARI>	INT	加工タイプ
				UNITS(一の位):
				用途
				1 = 直線切り込みのおねじ
				2 = 直線切り込みのめねじ
				3 = 通減切り込みのおねじ、切削の断面は一定
				4 = 通減切り込みのめねじ、切削の断面は一定
				TENS(十の位):
				予備
				HUNDREDS(百の位):
				切り込みタイプ
				0 = <_IANG>の互換性モード
				1 = 片面だけを切り込み
				2 = 両面を交互に切り込み
				THOUSANDS(千の位):
				予備
21	n	<_NUMTH>	INT	TEN THOUSANDS(万の位):
				交互深さ切り込み
				0 = 互換性、荒削り切削の回数を設定(<_NRC>)
				1 = 1 回目の切り込みの初期設定(<_MID>)
				HUNDRED THOUSANDS(十万の位):
				加工タイプ
				0 = 互換性(荒削りと仕上げ)
				1 = 荒削り
				2 = 仕上げ
				3 = 荒削りと仕上げ
				ONE MILLION(百万の位):
				多条ねじの加工処理
				0 = ねじを昇順で加工
				1 = ねじを降順で加工
				ねじの数

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
22		<_VRT>	REAL	戻し距離(inc)
				0 = 有効な単位(インチまたはメトリック)とは無関係に、戻し距離 1 mm が内部的に使用されます。
				> 0 = 戻し距離
23	D1	<_MID>	REAL	1 番目の切り込み<_VARI> (TEN THOUSANDS)を参照
24	DA	<_GDEP>	REAL	ねじ切り替え深さ(「多条ねじ」でのみ有効)
				0 = ねじ切り替え深さに従わない
				> 0 = ねじ切り替え深さに従う
25		<_IFLANK>	REAL	幅で表された切込み斜面(インタフェースのみ)
26		<_PITA>	INT	ねじピッチの評価
				0 = ピッチの互換性モード、有効な単位(メトリック/インチ)に従って、以前と同じように<_PP1>～<_PP3>を評価
				1 = ピッチ(mm)
				2 = ピッチ(TPI)(ねじ山数/inch)
				3 = ピッチ(inch)
				4 = MODULUS
27		<_PITM1>	STRING[15]	ピッチ入力のマーカとなる文字列(インタフェースのみ)
28		<_PITM2>	STRING[15]	ピッチ入力のマーカとなる文字列(インタフェースのみ)
29		<_PITM3>	STRING[15]	ピッチ入力のマーカとなる文字列(インタフェースのみ)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
30		<_DMODE>	INT	表示モード	
				UNITS(一の位):	加工平面 G17/G18/G19
					0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 = G17 (サイクルでのみ有効)
					2 = G18 (サイクルでのみ有効)
					3 = G19 (サイクルでのみ有効)
				TENS(十の位):	--- 予備
				HUNDREDS(百の位):	--- 予備
				THOUSANDS(千の位):	--- 予備
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1368)
					0 = 入力:全て
					1 = 入力:簡易

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
31		<_AMODE>	INT	代替モード
				UNITS(一の位):
				TENS(十の位):
				HUNDREDS(百の位):
				THOUSANDS(千の位):
				TEN THOUSANDS(万の位):
				HUNDRED THOUSANDS(十万の位):
				ONE MILLION(百万の位):
				TEN MILLION(千万の位):

4.26.1.30 CYCLE99 - ねじの旋削

構文

```
CYCLE99 (<_SPL>, <_SPD>, <_FPL>, <_FPD>, <_APP>, <_ROP>, <_TDEP>,
<_FAL>, <_IANG>, <_NSP>, <_NRC>, <_NID>, <_PIT>, <_VARI>,
<_NUMTH>, <_SDIS>, <_MID>, <_GDEP>, <_PIT1>, <_FDEP>, <_GST>,
<_GUD>, <_IFLANK>, <_PITA>, <_PITM>, <_PTAB>, <_PTABA>, <_DMODE>,
<_AMODE>, <_S_XRS>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
1	Z0	<_SPL>	REAL	基準点(abs)	
2	X0	<_SPD>	REAL	基準点(abs、常に直径)	
3	Z1	<_FPL>	REAL	<_AMODE> (UNITS)と組み合わせた終点	
4	X1	<_FPD>	REAL	<_AMODE>(TENS)と組み合わせた終点	
5	LW/LW2	<_APP>	REAL	<_AMODE> (HUNDREDS)と組み合わせたねじ切り始め、または <_AMODE> (HUNDREDS)と組み合わせたねじ切り始めとねじ切り上げ	
6	LR	<_ROP>	REAL	ねじ切り上げ	
7	H1	<_TDEP>	REAL	ねじ山深さ	
8	U	<_FAL>	REAL	X と Z の仕上げ代	
9	DP	<_IANG>	REAL	<_AMODE> (THOUSANDS)と組み合わせた、距離または角度で表した切り込み傾斜	
				> 0 =	プラス側の切り込み
				< 0 =	マイナス側の切り込み
				0 =	中央に配置した電源装置
10	α0	<_NSP>	REAL	開始角度オフセット(「一条ねじ」でのみ有効)	
11	ND	<_NRC>	INT	荒削り切削の回数、<_VARI> (TEN THOUSANDS)と組み合わせ	
12	NN	<_NID>	INT	エアーカットの回数	
13	P	<_PIT>	REAL	<_PITA>と組み合わせた値としてのピッチ	

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
14		<_VARI>	INT	加工タイプ
				UNITS(一の位):
				用途
				1 = 直線切り込みのおねじ
				2 = 直線切り込みのめねじ
				3 = 通減切り込みのおねじ、切削の断面は一定
				4 = 通減切り込みのめねじ、切削の断面は一定
				TENS(十の位):
				予備
				HUNDREDS(百の位):
				切り込みタイプ
				1 = 片面だけを切り込み
				2 = 両面を交互に切り込み
				THOUSANDS(千の位):
				予備
15	n	<_NUMTH>	INT	TEN THOUSANDS(万の位):
				交互深さ切り込み
				0 = 荒削り切削の回数を設定(<_NRC>)
				1 = 1 回目の切り込みの初期設定(<_MID>)
				HUNDRED THOUSANDS(十万の位):
				加工タイプ
				1 = 荒削り
				2 = 仕上げ
				3 = 荒削りと仕上げ
				ONE MILLION(百万の位):
				多条ねじの加工処理
				0 = ねじを昇順で加工
				1 = ねじを降順で加工
16	VR	<_SDIS>	REAL	戻り距離、inc
17	D1	<_MID>	REAL	<_VARI> (TEN THOUSANDS)と組み合わせた 1 番目の切り込み深さ

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
18	DA	<_GDEP>	REAL	ねじ切り替え深さ(「多条ねじ」でのみ有効)
				0 = ねじ切り替え深さに従わない
				>0 = ねじ切り替え深さに従う
19	g	<_PIT1>	REAL	一回転当たりのピッチの変更
				0 = ピッチは一定(G33)
				>0 = ピッチを増やす(G34)
				<0 = ピッチを減らす(G35)
20		<_FDEP>	REAL	切り込み深さ(符号なしで入力)
21	N1	<_GST>	INT	<_AMODE> (HUNDRED THOUSANDS)と組み合わせた開始ねじ N1 = 1...N
22		<_GUD>	INT	予備
23		<_IFLANK>	REAL	幅で表された切込み斜面(インタフェースのみ)
24		<_PITA>	INT	ピッチの単位(PIT および/または MPIT の評価)
				0 = ピッチ(mm) - MPIT/PIT 評価
				1 = ピッチ(mm) - PIT 評価
				2 = ピッチ(TPI) - PIT 評価(ねじ山数/inch)
				3 = ピッチ(inch) - PIT 評価
				4 = MODULE - PIT 評価
25		<_PITM>	STRING[15]	ピッチ入力のマーカとなる文字列(インタフェースのみ) ¹⁾
26		<_PTAB>	STRING[20]	ねじテーブルの文字列(インタフェースのみ) ¹⁾
27		<_PTABA>	STRING[20]	ねじテーブルの選択のための文字列(インタフェースのみ) ¹⁾

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
28		<_DMODE>	INT	表示モード
				UNITS(一の位):
				加工平面 G17/G18/G19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
				TENS(十の位):
				ねじのタイプ
				0 = 長手方向ねじ
				1 = スクロールねじ
				2 = テーパねじ
				HUNDREDS(百の位): --- 予備
				THOUSANDS(千の位): --- 予備
				TEN THOUSANDS(万の位):
				サイクル画面の機能スケーリング (ページ 1368)
				0 = 入力:全て
				1 = 入力:ベーシック

NC プログラミング
プログラミングマニュアル, 07/2021, 6FC5398-2EP40-0TA2

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
30	XS/RS	<_S_XRS>	REAL	<_AMODE>と組み合わせた長手方向ねじの真直度補正:ONE MILLION(百万の位)

注記

¹⁾パラメータ <_PITM>, <_PTAB>と<_PTABA>は、入力画面のねじテーブルでのねじの選択にのみ使用されます。

ねじテーブルは、サイクル運転時にサイクル定義を介してアクセスすることはできません。

4.26.1.31 CYCLE435 - ドレッサー座標系の設定

構文

CYCLE435 (<_T>, <_DD>, <S_TA>, <S_DA>, <S_AD>, <S_AL>, <S_PVD>, <S_PVL>, <S_PD>, <S_PL>, <_AMODE>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1		<_T>	STRING[32]	砥石の工具名
2		<_DD>	INT	砥石の刃先番号
3		<S_TA>	STRING[32]	目立て工具のレファレンス点 - 目立て工具の名称
4		<S_DA>	INT	目立て工具の刃先番号
5		<S_AD>	REAL	目立て値、直径
6		<S_AL>	REAL	目立て値、面
7		<S_PVD>	REAL	成形研削オフセット、直径
8		<S_PVL>	REAL	成形研削オフセット、面
9		<S_PD>	REAL	成形研削代、直径
10		<S_PL>	REAL	成形研削代、面

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
11		<_AMODE>	INT	代替モード	
				UNITS(一の位):	サイクルの終了時に有効な工具
					0 = 目立て工具が有効
					1 = といしが有効

4.26.1.32 CYCLE495 - 成形研削

構文

CYCLE495 (<_T>, <_DD>, <_SC>, <_F>, <_VARI>, <_D>, <_DX>, <_DZ>, <_S_PA>, <_S_N>, <_DMODE>, <_AMODE>, <_S_FW>, <_S_HW>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1		<_T>	STRING[20]	砥石の工具名
2		<_DD>	INT	砥石の刃先番号
3		<_SC>	REAL	障害物を回避するための戻り距離、インクレメンタル
4		<_F>	REAL	成形研削の送り速度

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
5		<_VARI>	INT	加工タイプ
				UNITS(一の位):
				1 = 軸と平行
				2 = 輪郭に平行
				TENS(十の位):
				加工方向
				0 = 引き 刃先位置 1～4 で可能
				1 = 押し 刃先位置 1～4 で可能
				2 = 交互 刃先位置 1～8 で可能
				3 = 始点 → 終点 刃先位置 1～8 で可能
				4 = 終点 → 始点 刃先位置 1～8 で可能
				HUNDREDS(百の位):
				切り込み方向
6		<_D>	REAL	軸に平行な成形研削タイプの目立て値
7		<_DX>	REAL	輪郭に平行な成形研削タイプの G18 の目立て値 X と G19 の目立て値 Y
8		<_DZ>	REAL	軸に平行な成形研削タイプの G18 および G19 の目立て値 Z
9		<S_PA>	REAL	成形研削代
10		<S_N>	INT	成形研削プログラムでのストローク数

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
11		<_DMODE>	INT	表示モード
				UNITS(一の位):
				加工平面 G17/G18/G19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
12		<_AMODE>	INT	代替モード
				UNITS(一の位):
				成形研削の選択、新規/継続
				1 = 新規
				2 = 続行
				TENS(十の位):
13		<S_FW>	REAL	目立て工具のクリア角度
				成形研削代の選択
				0 = 荒削り輪郭から輪郭の最低点まで
14		<S_HW>	REAL	目立て工具のホルダー角度
				1 = 荒削り輪郭から輪郭の最高点まで

注記

CYCLE495 を使用したプロファイリングが予約処理 CONTDCON を使用して実行されるときに輪郭解釈。CONTDCON は工具径補正(G41/G42)が有効な場合に許可されていないので、CYCLE495 を呼び出す前に工具径補正は G40 で無効にされていなければなりません。

4.26.1.33 CYCLE782 - 負荷に従った位置合わせ

注記

CYCLE782 を使用するには、「インテリジェント負荷調整」オプションのライセンスが必要です。

構文

CYCLE782 (<S_MODE>, <S_TESTAXIS>, <S_VALUE>)

4.26 サイクルを外部的にプログラミング

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
1		<S_MODE>	INT	処理モード	
				UNITS(一の位):	選択/解除
					0 = 調整を無効にします
					1 = 調整を有効にします
					2 = 軸の基本慣性の特定と有効化(試運転機能)
				TENS(十の位):	計測バージョン(<S_MODE> HUNDREDS POSITION = 0 の場合のみ)
					0 = 標準モード
					1 = 精密モード(摩擦を特定する場合など)
				百の位	慣性モーメントを計測または指定します
					0 = 測定 このモードでは、負荷を特定するための動動作が実行されます。
					1 = 慣性モーメントが指定されます このモードでは、負荷を特定するための移動動作が実行されません。代わりに、自動サーボ調整機能(AST)などで特定した慣性モーメントが転送されます。
				1000 の位	計測結果表示
					0 = 測定結果画面がオフ
					1 = 測定結果画面がオン
				万の位	計測結果の表示時間
					1 = 表示が 8 秒後に自動的に消えます
					3 = NC スタートでの確認
				十万の位	選択の読み込み ¹⁾
					0 = 読み込みプロセス全体の特定:空の軸 + ワーク

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
				1 = 個別読み込みの特定:ワーク
2		<S_TESTAXIS>	AXIS	チャンネル軸: - 調整に対して有効で - 慣性モーメントを特定する必要があります(<S_MODE> HUNDREDS POSITION = 0 の場合のみ)
3		<S_VALUE>	REAL	慣性モーメント値(<S_MODE> HUNDREDS POSITION = 1 の場合のみ)

1) 値は機械データ 52212 \$MCS_FUNCTION_MASK_Tech ビット 18 から自動的に取られます。

注記

試運転機能「軸の基本慣性の特定と有効化」(<S_MODE> UNIT PLACE = 2)は、ユーザーインタフェースではサポートされていません。このためにサイクル呼び出しをプログラム指令してください。測定結果は機械データ 53350 \$MAS_ILC_BASE_VALUE に入力されます。「読み込みの選択」パラメータ(<S_MODE> ONE HUNDRED THOUSANDS DIGIT)の値が、ユーザーインタフェースによって機械データ 52212 \$MCS_FUNCTION_MASK_Tech ビット 18 から読み取られ、サイクル呼び出しに入力されます。

読み込みは直線軸と回転軸に対して決定または指定することができます。直線軸の場合は、質量、回転軸の場合は、慣性モーメントが測定結果画面で常に表示されるか、ユーザーインタフェースで指定されます。

軸タイプにかかわらず、慣性モーメントは機械データ 53350 \$MAS_ILC_BASE_VALUE と補正に常書き込まれます。

例

例 1:

NC プログラムの開始時に、軸 MX1 の負荷レベルが特定されます – この値は調整を更新するためのベースとして使用されます。結果は表示され、NC スタートで確認されます。

プログラムコード

```
...
CYCLE782 (31011,MX1)
...
```

4.26 サイクルを外部的にプログラミング

例 2:

プログラムの途中で、たとえば AST を使用して特定された慣性モーメント値が変数 `_MY_VALUE` に転送され、この値で負荷サイクルが呼び出され、MX2 に対して調整が有効になります。結果は表示されません。

プログラムコード

```
DEF REAL _MY_VALUE
_MY_VALUE=...
...
CYCLE782 (101,MX2,_MY_VALUE)
...
```

例 3:

軸 MX2 の調整が無効化されます。

プログラムコード

```
...
CYCLE782 (0,MX2)
...
```

4.26.1.34 CYCLE800 – 旋回平面/旋回工具/工具の位置合わせ

構文

CYCLE800 (<_FR>, <_TC>, <_ST>, <_MODE>, <_X0>, <_Y0>, <_Z0>, <_A>, <_B>, <_C>, <_X1>, <_Y1>, <_Z1>, <_DIR>, <_FR_I>, <_DMODE>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1		<_FR>	INT	後退モード	0 =	後退なし
					1 =	機械軸 Z の後退
					2 =	機械軸 Z、次に XY の後退
					3 =	予備
					4 =	工具方向の最大後退
					5 =	工具方向のインクレメンタル後退

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
2		<_TC>	STRING[32]]	旋回データブロックの名称:	""	""(名称なし)、存在する旋回データブロックが1つのみの場合
					"0"	旋回データブロックの選択を解除(旋回フレームを削除)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
3		<_ST>	INT	座標変換状態
				UNITS(一の位):
				0 = 新規。旋回レベルが削除され、現在のパラメータを使用して再計算されます。
				1 = 追加。旋回レベルが現在の旋回レベルに追加されます。
				TENS(十の位):
				工具先端追従の有無(旋回機能がセットアップで設定されている場合のみ有効)
				0 = 工具先端を追従しない
				1 = 工具先端を追従する(TRAORI)
				HUNDREDS(百の位):
				工具の移動/位置合わせ(機能は工具旋回入力画面で表示されます)
				0 = 工具はアプローチしない
				1 = 工具はアプローチする(できれば、ラジアルミルで)
				2 = 旋削工具を位置合わせする(B軸キネマティックが旋回セットアップでフライス削り用に設定されている場合)
				3 = フライス工具を位置合わせする(B軸キネマティックが旋回セットアップでフライス削り用に設定されている場合)
				THOUSANDS(千の位):
				内部「JOG での旋回」パラメータ
				TEN THOUSANDS(万の位):
				方向パラメータ<_DIR>を参照してください
				0 = 旋回「あり」
				1 = 旋回「なし」、「負」方向 ³⁾
				2 = 旋回「なし」、「正」方向 ³⁾
				HUNDRED THOUSANDS(十万の位):
				方向パラメータ<_DIR>を参照してください

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
					0 =	互換性
					1 =	方向の選択「負」が最適化されています(ユーザーインタフェースのみ) ⁴⁾
					2 =	方向の選択「正」が最適化されています(ユーザーインタフェースのみ) ⁴⁾

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
4		<_MODE> 5)	INT	旋回モード旋回角度と旋回処理(ビット指定)の評価
				ビット:7 6
				0 0: 軸毎の旋回角度 -> パラメータ<_A>, <_B>, <_C> を参照してください
				0 1: 立体角 -> パラメータ<_A>, <_B> ¹⁾ を参照してください
				1 0: 投影角 -> パラメータ<_A>, <_B>, <_C> ¹⁾ を参照してください
				1 1: ダイレクト回転軸旋回モード -> パラメータ<_A>, <_B> ¹⁾ を参照してください
				ビット:5 4 3 2 1 0 (立体角には適用されません)
				xxxx01 Xを中心とした1番目の回転_A
				xxxx10 Yを中心とした1番目の回転_A
				xxxx11 Zを中心とした1番目の回転_A
				xx01xx Xを中心とした2番目の回転_B
				xx10xx Yを中心とした2番目の回転_B
				xx11xx Zを中心とした2番目の回転_B
				01xxxx Xを中心とした3番目の回転_C
				10xxxx Yを中心とした3番目の回転_C
				11xxxx Zを中心とした3番目の回転_C
5	X0	<_X0>	REAL	回転前の基準点 X
6	Y0	<_Y0>	REAL	回転前の基準点 Y
7	Z0	<_Z0>	REAL	回転前の基準点 Z

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
8	X(A)	<_A>	REAL	パラメータ<_MODE>の設定に基づいた 1 番目の回転
9	Y(B)	<_B>	REAL	パラメータ<_MODE>の設定に基づいた 2 番目の回転
10	Z(C)	<_C>	REAL	パラメータ<_MODE>の設定に基づいた 3 番目の回転
11	X1	<_X1>	REAL	回転後の基準点 X
12	Y1	<_Y1>	REAL	回転後の基準点 Y
13	Z1	<_Z1>	REAL	回転後の基準点 Z
14	-または+	<_DIR>	INT	回転軸の移動を開始(初期設定 = -1)
				-1 = 回転軸 1 と 2 のうち、どちらか小さい値のほうに位置決めする ²⁾
				+1 = 回転軸 1 と 2 のうち、どちらか大きい値のほうに位置決めする ²⁾
				0 = 旋回しない(旋回フレームの計算のみ) ^{1) 3)}
15	FR	<_FR_I>	REAL	工具方向の後退の値(inc)、インクレメンタル
16		<_DMODE>	INT	表示モード
				UNITS(一の位):
				加工平面 G17/G18/G19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
				TENS(十の位):
				工具位置合わせ時のベータ値の表示
				0 = 値
				1 = 矢印

4.26 サイクルを外部的にプログラミング

注記

次の転送パラメータが間接的に(パラメータとして)プログラム指令されている場合、入力画面はリセットされません: <_FR>, <_ST>, <_TC>, <_MODE>, <_DIR>

- 1) 旋回機能がセットアップで設定されている場合のみ選択できます。
- 2) 回転軸 1 または 2 を基準にした方向が旋回セットアップで設定されている場合に選択可能です。

方向基準が「なし」の場合、選択欄はありません。

- 3) 旋回の選択「なし」は、SD 55221 ビット 0 でグレーの透過色で表示できます。

旋回「なし」、「負」方向は<_DIR> = 0 と_ST TEN THOUSANDS = 1 に対応しています。

旋回「なし」、「正」方向は<_DIR> = 0 と_ST TEN THOUSANDS = 2 に対応しています。

- 4) 回転軸 1 または 2 の方向の選択は、方向基準のある回転軸が極位置にある場合(位置データが 0 に等しい)にも発生します。

- 5) 割り当て例:軸毎の回転、回転順序 ZYX

2 進法:00011011 10 進法:27

軸識別子 XYZ は、NC チャネルのジオメトリ軸に対応しています。XYZ 軸を中心にした個別の回転がおこなえます。たとえば、ZXZ を中心にした回転順序は、1 回の CYCLE800 呼び出しでは行えません。

4.26.1.35 CYCLE801 – 格子またはフレーム位置決めパターン

構文

```
CYCLE801 (<_SPCA>, <_SPCO>, <_STA>, <_DIS1>, <_DIS2>, <_NUM1>,  
<_NUM2>, <_VARI>, <_UMODE>, <_ANG1>, <_ANG2>, <_HIDE>, <_NSP>,  
<_DMODE>)
```

パラメータ

No.	マスクパラメータ	パラメータ内部	データタイプ	意味	
1	X0	<_SPCA>	REAL	1 番目の軸の位置決めパターン(格子/フレーム)の基準点(abs)	
2	Y0	<_SPCO>	REAL	2 番目の軸の位置決めパターン(格子/フレーム)の基準点(abs)	
3	α0	<_STA>	REAL	基本回転角度 (1 番目の軸に対する角度)	< 0 = 右回り
					> 0 = 左回り

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	パラメータ内部	データタイプ	意味
4	L1	<_DIS1>	REAL	列間の距離(1 番目の軸からの位置、符号なしで入力)
5	L2	<_DIS2>	REAL	行間の距離(2 番目の軸からの距離、符号なしで入力)
6	N1	<_NUM1>	INT	列数
7	N2	<_NUM2>	INT	行数
8		<_VARI>	INT	加工タイプ
				UNITS(一の位):
				位置決めパターン
				0 = 格子
				1 = フレーム
				TENS(十の位):
				予備
				HUNDREDS(百の位):
				予備
9		<_UMODE>	INT	予備
10	αX	<_ANG1>	REAL	1 番目の軸によるせん断角(線が 1 番目の軸に対して斜めに配置されています)
				<0 = 右回りの角度 (0~90°)
				>0 = 左回りの角度 (0~90°)
11	αY	<_ANG2>	REAL	2 番目の軸に対するせん断角(線が 2 番目の軸に対して斜めに配置されています)
				<0 = 右回りの角度 (0~90°)
				>0 = 左回りの角度 (0~90°)
12		<_HIDE>	STRING [200]	非表示位置 - 最大 198 文字 - 連続する位置番号の指定。例:"1,3" (位置 1 と 3 は実行されません)
13		<_NSP>	INT	予備

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	パラメータ内部	データタイプ	意味		
14		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/G18/G19	
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 =	G17 (サイクルでのみ有効)
					2 =	G18 (サイクルでのみ有効)
					3 =	G19 (サイクルでのみ有効)

4.26.1.36 CYCLE802 - 任意の位置

構文

```
CYCLE802 (<_XA>, <_YA>, <_X0>, <_Y0>, <_X1>, <_Y1>, <_X2>, <_Y2>,
<_X3>, <_Y3>, <_X4>, <_Y4>, <_X5>, <_Y5>, <_X6>, <_Y6>, <_X7>,
<_Y7>, <_X8>, <_Y8>, <_VARI>, <_UMODE>, <_DMODE>, <S_ABA>,
<S_AB0>, <S_AB1>, <S_AB2>, <S_AB3>, <S_AB4>, <S_AB5>, <S_AB6>,
<S_AB7>, <S_AB8>)
```

パラメータ

No.	マスクパラメータ	パラメータ内部	データタイプ	意味		
1		<_XA>	INT	すべての X 位置の代替値(9 桁の 10 進数値) 10 進数値の桁数は次のとおりです。876543210 (桁位置は穴あけ位置 Xn に対応しています)		
				位置データ:	1 =	アブソリュート(1 番目のプログラム指令位置は常にアブソリュートです)
					2 =	インクレメンタル

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	パラメータ内部	データタイプ	意味
2		<_YA>	INT	すべての Y 位置の代替値(9 桁の 10 進数値) 10 進数値の桁数は次のとおりです。876543210 (桁位置は穴あけ位置 Yn に対応しています)
				位置データ:
				1 = アブソリュート(1 番目のプログラム指令位置は常にアブソリュートです) 2 = インクレメンタル
3	X0	<_X0>	REAL	1 番目の位置 X
4	Y0	<_Y0>	REAL	1 番目の位置 Y
5	X1	<_X1>	REAL	2 番目の位置 X
6	Y1	<_Y1>	REAL	2 番目の位置 Y
7	X2	<_X2>	REAL	3 番目の位置 X
8	Y2	<_Y2>	REAL	3 番目の位置 Y
9	X3	<_X3>	REAL	4 番目の位置 X
10	Y3	<_Y3>	REAL	4 番目の位置 Y
11	X4	<_X4>	REAL	5 番目の位置 X
12	Y4	<_Y4>	REAL	5 番目の位置 Y
13	X5	<_X5>	REAL	6 番目の位置 X
14	Y5	<_Y5>	REAL	6 番目の位置 Y
15	X6	<_X6>	REAL	7 番目の位置 X
16	Y6	<_Y6>	REAL	7 番目の位置 Y
17	X7	<_X7>	REAL	8 番目の位置 X
18	Y7	<_Y7>	REAL	8 番目の位置 Y
19	X8	<_X8>	REAL	9 番目の位置 X
20	Y8	<_Y8>	REAL	9 番目の位置 Y

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	パラメータ内部	データタイプ	意味	
21		<_VARI>	INT	加工	
				HUNDREDS(百の位):	(注文生産現場からの呼び出しの場合のみ)(現時点では 0 と 2 のみが評価されます)
					0 = 主軸をクランプしません
					1 = G00 または G01 による垂直切り込みの場合のみ主軸をクランプします
					2 = 加工中は常に主軸をクランプします
				THOUSANDS(千の位):	予備
				TEN THOUSANDS(万の位):	回転軸あり/なしの位置決めパターン - 軸の組み合わせ (<_VARI> HUNDRED THOUSANDS と)
					0 = XY (XY のみで回転軸なし、互換性)
					1 = X、Y または Z と回転軸: XA、YB、ZC (1 つの回転軸と回転軸の回転の中心となるジオメトリ軸)
					2 = XY と回転軸: XYA、XYB、XYC (1 つの回転軸と 1 番目および 2 番目のジオメトリ軸、TRACYL なし)
				HUNDRED THOUSANDS(十万の位):	回転軸
					0 = 回転軸なし (XY のみ、互換性)
					1 = A 軸(X を中心とした回転軸)
					2 = B 軸(Y を中心とした回転軸)
					3 = C 軸(Z を中心とした回転軸)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	パラメータ内部	データタイプ	意味	
				TEN MILLIONS + ONE MILLION(千万 + 百万の位):	回転軸ありの位置決めパターン – オフセット (同じ軸を中心とした複数の回転軸用、インデックスが大きすぎる場合は、1 番目の軸) 00 = 1 番目の A、B または C 軸または互換性のため 01 = 2 番目の A、B または C 軸 ... 19 = 20 番目の A、B または C 軸
22		<_UMODE>	INT	クランプする主軸の選択:(注文生産現場からの呼び出しの場合のみ) (ユーザーサイクル CUST_TECHCYC の呼び出し)	3 = 主軸のクランプ/解放 23 = 対向主軸のクランプ/解放
23		<_DMODE>	INT	表示モード	UNITS(一の位): 加工平面 G17/G18/G19 0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効 1 = G17 (サイクルでのみ有効) 2 = G18 (サイクルでのみ有効) 3 = G19 (サイクルでのみ有効)
24		<S_ABA>	INT	すべての AB 位置の代替値(9 桁の 10 進数値) 10 進数値の桁数は次のとおりです。876543210 (桁位置は位置 ABn に対応しています)	位置データ: 1 = アブソリュート(1 番目のプログラム指令位置は常にアブソリュートです) 2 = インクレメンタル
25	A0	<S_AB0>	REAL	回転軸ありの位置決めパターンの 1 番目の回転軸位置 (<_VARI>と組み合わせて))	
26	A1	<S_AB1>	REAL	回転軸ありの位置決めパターンの 2 番目の回転軸位置	
27	A2	<S_AB2>	REAL	回転軸ありの位置決めパターンの 3 番目の回転軸位置	

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	パラメータ内部	データタイプ	意味
28	A3	<S_AB3>	REAL	回転軸ありの位置決めパターンの 4 番目の回転軸位置
29	A4	<S_AB4>	REAL	回転軸ありの位置決めパターンの 5 番目の回転軸位置
30	A5	<S_AB5>	REAL	回転軸ありの位置決めパターンの 6 番目の回転軸位置
31	A6	<S_AB6>	REAL	回転軸ありの位置決めパターンの 7 番目の回転軸位置
32	A7	<S_AB7>	REAL	回転軸ありの位置決めパターンの 8 番目の回転軸位置
33	A8	<S_AB8>	REAL	回転軸ありの位置決めパターンの 9 番目の回転軸位置

注記

パラメータ X1/Y1/A1~X8/Y8/A8 に必要でない位置は、無視することができます。ただし、<_XA>,<_YA>と<S_ABA>の代替値を 9 つのすべての位置に対して全て与えてください。

位置決めパターン XA、YB または ZC (ジオメトリ軸と回転軸)の場合、位置決めパターンによって移動しない加工平面の軸(G17 の Y と XA)は、サイクル呼び出しの前に位置決めしてください。

4.26.1.37 CYCLE806 - 補間旋削

注記

CYCLE806 による補間旋削はライセンスが必要なオプションです。

構文

CYCLE806 (<_MODE>, <S_XC>, <S_YC>, <_STA>, <_SV1>, <_SV2>,
<_SDIR>, <_AMODE>, <_DMODE>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1		<_MODE>	INT	モード		
				UNITS(一の位):	選択 / 選択解除	
					0 =	補間旋削の選択解除
					1 =	補間旋削の選択

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
2	XC	<S_XC>	REAL	回転 X の中心		
3	YC	<S_YC>	REAL	回転 Y の中心		
4	ガンマ	<_STA>	REAL	角度ガンマ(工具軸を中心とする回転); 予約済み		
5	S	<_SV1>	REAL	定主軸速度または定切削速度(<_AMODE>ユニットを参照)		
6	PS	<_SV2>	REAL	定切削速度の最大速度(予約済み)		
7		<_SDIR>	INT	予約		
					0 =	工具からの回転方向
					3 =	回転方向 M3
					4 =	回転方向 M4
8		<_AMODE>	INT	代替モード		
				UNITS(一の位):	主軸移動(<_SV1>)	
					0 =	一定主軸速度
				1 =	定切削速度	
9		<_DMODE>	INT	表示モード		
				UNITS(一の位):	加工平面 G17/18/19	
					1 =	G17 (サイクルでのみ有効)
10	TC	<S_TRA>	STRING	- 座標変換の名前 - 空:1 番目のキネマティック座標変換 TRAJNT を使用		

4.26.1.38 CYCLE830 - 深穴ドリル 2

構文

```

CYCLE830 (<RTP>, <RFP>, <SDIS>, <_DP>, <FDEP>, <_DAM>, <DTB>,
<DTS>, <FRF>, <VARI>, <MDEP>, <_VRT>, <DTD>, <_DIS1>, <S_FP>,
<S_SDAC2>, <S_SV2>, <S_FB>, <_SDAC>, <_SV1>, <S_SPOS>, <S_ZA>,
<S_FA>, <S_ZP>, <S_FS>, <S_ZS1>, <S_ZS2>, <S_N>, <S_ZD>, <S_FD>,
<S_FR>, <S_SDAC3>, <S_SV3>, <S_CON>, <S_COFF>, <_GMODE>,
<_DMODE>, <_AMODE>, <S_AMODE2>, <S_AMODE3>, <S_ZPV>)

```

4.26 サイクルを外部的にプログラミング

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<RFP>	REAL	基準点(abs)
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なし)
4	Z1	<_DP>	REAL	最終穴あけ深さ abs/inc (<_AMODE>UNITS を参照)
5	d	<FDEP>	REAL	アブソリュートまたはインクレメンタルの切り屑破断/除去のための 1 番目の穴あけ深さ、予備穴あけあり/なしで基準点を基準、またはパイロット穴深さを基準(<_AMODE> TEN THOUSANDS を参照)
6	DF	<_DAM>	REAL	各々の追加切り込みの絶対値/パーセント値、通減アブソリュート値/パーセント値(<_AMODE> HUNDRED THOUSANDS を参照)
7	DTB	<DTB>	REAL	各穴あけ深さでのドウェル時間(<_AMODE> TENS を参照)
8	DTS	<DTS>	REAL	始点での切り屑除去中のドウェル時間(<_AMODE> HUNDREDS を参照)
9	FD1	<FRF>	REAL	最初の切り込み送り速度のパーセント値(<_AMODE> TEN MILLION を参照)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
10		<VARI>	INT	加工
				UNITS(一の位):
				0 = 一回の切削
				1 = 切り屑切断
				2 = 切り屑除去
				3 = 切り屑切断と切り屑除去
				TENS(十の位):
				0 = 切り屑除去中の後退
				0 = パイロット穴深さまで
				1 = 安全距離まで
				HUNDREDS(百の位):
				ソフトな最初の切削
				0 = なし
				1 = あり
				THOUSANDS(千の位):
				貫通穴あけ
				0 = なし
				1 = あり
				TEN THOUSANDS(万の位):
				予備穴あけ/パイロット穴
				0 = 予備穴あけなし
				1 = 予備穴あけあり
				2 = パイロット穴あり
				HUNDRED THOUSANDS(十万の位):
				後退
				0 = パイロット穴深さまで
				1 = イニシャル点
11	V1	<_MDEP>	REAL	最小インクレメンタル切り込み(通減パーセント値のみ)
12	V2	<_VRT>	REAL	各インクレメンタル加工ステップ終了後の後退量(切り屑破断の場合のみ)
				0 = 初期値 1 mm
				>0 = 可変 後退距離
13	DT	<_DTD>	REAL	最終穴あけ深さでのドウェル時間(<_AMODE> THOUSANDS を参照)
14	V3	<_DIS1>	REAL	切り屑除去の場合のみのインクレメンタルのリミット距離 (<_AMODE> ONE MILLION を参照)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
15	FP	<S_FP>	REAL	パイロット穴への移動の送り速度(値としてまたは%単位で) (<S_AMODE2> HUNDREDS と組み合わせて)		
16		<S_SDAC2>	INT	アプローチ時の主軸の回転方向		
					3 =	M3
					4 =	M4
				5 =	M5 (初期設定)	
17	SP	<S_SV2>	REAL	一定主軸速度でのアプローチ	(<S_AMODE2> TEN MILLION を参照)	
	定切削速度					
	V4			穴あけ速度の%での主軸速度		
18	F	<S_FB>	REAL	穴あけ送り速度(<S_AMODE2> UNITS を参照)		
19		<_SDAC>	REAL	穴あけ時の主軸の回転方向		
					3 =	M3
					4 =	M4
20	S	<_SV1>	REAL	一定主軸速度での	穴あけ(<S_AMODE2> ONE MILLION を参照)	
	定切削速度					
	V5					
21	SPOS	<S_SPOS>	REAL	主軸位置決め、M5 でのアプローチの場合のみ		
22	ZA	<S_ZA>	REAL	基準点を基準にしたインクリメンタルの予備穴あけ深さまたはアブソリュート(<S_AMODE3> UNITS を参照)		
23	FA	<S_FA>	REAL	予備穴あけ送り速度(値としてまたは%単位で)(<S_AMODE2> TENS と組み合わせて)		
24	ZP	<S_ZP>	REAL	基準点を基準にしたインクリメンタルのパイロット穴またはアブソリュートまたは穴直径の係数(<S_AMODE3> TENS を参照)		
25	FS	<S_FS>	REAL	最初の切削の送り速度(値としてまたは%単位で) (<S_AMODE2> THOUSANDS と組み合わせて)		
26	ZS1	<S_ZS1>	REAL	一定送り速度での各 1 番目切削の深さ(inc)		
27	ZS2	<S_ZS2>	REAL	送り速度増速での各 1 番目切削の深さ(inc)		
28	N	<S_N>	INT	各切り屑除去の前の切り屑切断の回数		
29	ZD	<S_ZD>	REAL	最終穴あけ深さを基準にしたインクリメンタルの残りの穴あけ深さまたはアブソリュート(<S_AMODE3> HUNDREDS を参照)		

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
30	FD	<S_FD>	REAL	残りの穴あけ送り速度(値としてまたは%単位で) (<S_AMODE2> TEN THOUSANDS と組み合わせて)	
31	FR	<S_FR>	REAL	後退送り速度(<S_AMODE2> HUNDRED THOUSANDS と組み合わせて)	
32		<S_SDAC3>	INT	後退のときの主軸の回転方向	
				3 =	M3
				4 =	M4
				5 =	M5
33	SR	<S_SV3>	REAL	後退速度	一定主軸速度(<S_AMODE2> HUNDRED MILLION を参照)
	V6				定切削速度
				穴あけ速度の%での主軸速度	
34	クーラント ON	<S_CON>	STRING[10]	クーラントオン、M 命令またはサブプログラム呼び出し	
35	クーラント オフ	<S_COFF>	STRING[10]	クーラントオフ、M 命令またはサブプログラム呼び出し	
36		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)	
				UNITS(一の位):	予備
				TENS(十の位):	先端/シャンクを基準にした穴あけ深さ
				0 =	先端
				1 =	シャンク

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
37		<_DMODE>	INT	表示モード	
				UNITS(一の位):	加工平面 G17/G18/G19
					0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 = G17 (サイクルでのみ有効)
					2 = G18 (サイクルでのみ有効)
					3 = G19 (サイクルでのみ有効)
				TENS(十の位):	予備
				HUNDREDS(百の位):	予備
				THOUSANDS(千の位):	予備
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1368)
					0 = 入力:全て
					1 = 入力:ベーシック

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
38		<_AMODE>	INT	代替モード 1
				UNITS(一の位):
				穴あけ深さ = 最終穴あけ深さ Z1 abs/inc
				0 = インクレメンタル
				1 = アブソリュート
				TENS(十の位):
				各穴あけ深さでのドウェル時間 DTB(秒(s)/回転数(rev))
				0 = 秒(s)
				1 = 回転数(rev)
				HUNDREDS(百の位):
				切り屑除去のドウェル時間 DTS(秒(s)/回転(rev))
				0 = 秒(s)
				1 = 回転数(rev)
				THOUSANDS(千の位):
				最終穴あけ深さ DT でのドウェル時間(秒(s)/回転数(rev))
				0 = 秒(s)
				1 = 回転数(rev)
				TEN THOUSANDS(万の位):
				1 番目の穴あけ深さ D (abs/inc)
				0 = インクレメンタル
				1 = アブソリュート
				HUNDRED THOUSANDS(十万の位):
				各々の追加切り込みの絶対値/パーセント値(通減量)
				0 = 絶対値
				1 = パーセント値(0.001~100%)
				ONE MILLION(百万の位):
				リミット距離 V3 自動/手動
				0 = 自動(サイクルで計算)
				1 = 手動(プログラム指令値)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
39		<S_AMODE 2>	INT	代替モード 2
				UNITS(一の位):
				UNITS(一の位):穴あけ送り速度 F
				0 = F/min
				1 = F/rev
				TENS(十の位):
				予備穴あけ送り速度の評価 FA
				0 = 穴あけ送り速度の%として
				1 = F/min
				2 = F/rev
				HUNDREDS(百の位):
				パイロット穴への移動のための送り速度 FP の評価
				0 = 穴あけ送り速度の%として
				1 = F/min
				2 = F/rev
				THOUSANDS(千の位):
				最初の切削の送り速度 FS の評価
				0 = 穴あけ送り速度の%として
				1 = F/min
				2 = F/rev
				TEN THOUSANDS(万の位):
				貫通穴あけ送り速度 FD の評価
				0 = 穴あけ送り速度の%として
				1 = F/min
				2 = F/rev
				HUNDRED THOUSANDS(十万の位):
				後退送り速度 FR
				0 = F/min
				1 = 早送り
				ONE MILLION(百万の位):
				穴あけ - 主軸速度/切削速度 (S/V5)
				0 = 一定主軸速度
				1 = 定切削速度
				TEN MILLION(千万の位):
				主軸速度/切削速度(SP/V4)でのアプローチ
				0 = 一定主軸速度
				1 = 定切削速度

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
					2 =	穴あけ速度の%での主軸速度
				HUNDRED MILLIONS(一億の位):	後退 - 主軸速度/切削速度(SR/V6)	
					0 =	一定主軸速度
					1 =	定切削速度
					2 =	穴あけ速度の%での主軸速度
40		<S_AMODE 3>	INT	代替モード 3		
				UNITS(一の位):	穴あけ深さ ZA abs/inc	
					0 =	インクリメンタル
					1 =	アブソリュート
				TENS(十の位):	パイロット穴 ZP の深さ	
					0 =	インクリメンタル
					1 =	アブソリュート
					2 =	穴直径の係数
				HUNDREDS(百の位):	残りの穴あけ深さ ZD abs/inc	
0 =	インクリメンタル					
1 =	アブソリュート					
41	ZPV	<S_ZPV>	REAL	パイロット穴深さからのインクリメンタルのリミット距離		

4.26.1.39 CYCLE832 - 高速設定

構文

CYCLE832 (<S_TOL>, <S_TOLM>, <S_OTOL>)

注記

CYCLE832 により、機械のセットアップ時に必要な工作機械メーカーによる調整の仕事が軽減されることはありません。このサイクルは、加工プロセスと NCU 設定(プリコントロール、加々速度一定など)に関連する軸の調整をおこないます。

4.26 サイクルを外部的にプログラミング

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
1	許容範囲	<S_TOL>	REAL	輪郭の許容範囲 輪郭の許容範囲は、ジオメトリ軸の軸の許容範囲に対応しています。	
2		<S_TOLM>	INT	加工タイプ(用途)	
				UNITS(一の位):	
					0 = 解除
					1 = 仕上げ
					2 = 中仕上げ(荒仕上げ)
					3 = 荒削り
					4 = 滑らかな仕上げ(精密)
				TENS(十の位):	
					0 = 互換性 ¹⁾ または旋回の許容範囲なし
					1 = パラメータ<S_OTOL>の旋回許容範囲
				百の位 ... HUNDRED THOUSANDS(十万の位)	割り付けられた互換性が理由で
				ONE MILLION(百万の位):	
					0 = 互換性。最上の金型形成機能が自動的に使用されます。 • オプション[最上面]が無効です。 ▮ Advanced Surface • オプション[最上面]が有効です。 ⇒ スムージングを行う最上面
					1 = スムージングを行わない最上面
					2 = スムージングを行う最上面

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
3	ORI の許容範囲	<S_OTOL>	REAL	旋回の許容範囲またはタイプの識別子 CYCLE832 ワークの旋回の許容範囲パラメータです。 これは、ダイナミックな方向座標変換(5 軸同時加工など)を備えた機械での高速加工プログラムの実行時に必要です。 パラメータ<S_OTOL>をプログラム指令してください。これは 3 軸機械での工具のオリエンテーションを使用しないプログラムの用途にも適用されます(<S_OTOL> = 1)。

- 1) 係数 SD55441 ... SD55444 で乗算された、サイクルのセッティングデータ SD55451 ... SD55454 (ダイナミック応答モードのオリエンテーション許容範囲...)または SD55445 ... SD55449 (ダイナミック応答モードの輪郭許容範囲...)から取得されたオリエンテーション許容範囲。

詳しくは、以下を参照してください。SINUMERIK Operate 試運転マニュアル

プレーンテキスト入力

サイクル呼び出しを読みやすくするために、パラメータ<S_TOLM>(加工タイプ)をプレーンテキストで入力することもできます。プレーンテキストは言語に依存しません。次の入力が可能です。

_OFF	FO R	0	解除
_FINISH	FO R	1	仕上げ
_SEMIFIN	FO R	2	中仕上げ
_ROUGH	FO R	3	荒削り
_PRECISION	FO R	4	滑らかな仕上げ
_ORI_FINISH	FO R	11	旋回の許容範囲の入力による仕上げ
_ORI_SEMIFIN	FO R	12	旋回の許容範囲の入力による荒仕上げ
_ORI_ROUGH	FO R	13	旋回の許容範囲の入力による荒削り
_ORI_PRECISION	FO R	14	旋回の許容範囲の入力による滑らかな仕上げ

4.26 サイクルを外部的にプログラミング

<code>_TOP_SURFACE_SMOOTH_OFF</code>	<code>FO</code>	<code>1000000</code>	スムージングを行わない最上面
	<code>R</code>		
<code>_TOP_SURFACE_SMOOTH_ON</code>	<code>FO</code>	<code>2000000</code>	スムージングを行う最上面
	<code>R</code>		

最上面のプレーンテキストの入力では、プレーンテキストは以下の例に示すように組み合わせられます。

```
CYCLE832 (0.1, _TOP_SURFACE_SMOOTH_OFF+_ORI_FINISH, 1)
```

注記

プレーンテキストは G グループ 59 の機能名称(軌跡補間のダイナミック応答モード)に基づいています。これらのプレーンテキストにより、3 軸機械と多軸方向座標変換(TRAORI)を備えた機械は用途で明確に区別されます。

CYCLE832 の選択解除

CYCLE832 を選択解除する場合は、パラメータ<S_TOL>を 0 で転送してください。

例:CYCLE832 (0,0,1)

CYCLE832 の選択解除には、構文 CYCLE832 () も使用できます。

例

例 1:3 軸機械での方向座標変換を使用しない CYCLE832

a) プレーンテキスト入力によるサイクル呼び出し

プログラムコード	コメント
G710	; 単位系はメトリックです。
CYCLE832 (0.004,_FINISH,1)	; 次の設定による CYCLE832 呼び出し: 輪郭の許容範囲 = 0.004 mm、加工タイプ:仕上げ
...	; 高速加工プログラムの実行

b) プレーンテキスト入力を使用しないサイクル呼び出し

プログラムコード	コメント
G710	; 上記を参照してください
CYCLE832 (0.004,1,1)	; 上記を参照してください
...	; 上記を参照してください

4.26 サイクルを外部的にプログラミング

例 2:5 軸機械での方向座標変換を使用した CYCLE832

a) プレーンテキスト入力によるサイクル呼び出しと選択解除

プログラムコード	コメント
G710	; 単位系はメトリックです。
TRAORI	; 方向座標変換を起動します。
CYCLE832(0.3,_ORI_ROUGH,0.8)	; 次の設定による CYCLE832 呼び出し: 輪郭の許容範囲 = 0.3 mm、加工タイプ:旋回の許容範囲の入力 による荒削り; 旋回の許容範囲 = 0.8°
...	; 高速加工プログラムの実行
CYCLE832(0,_OFF,1)	; 輪郭の許容範囲 = 0、 加工タイプ:CYCLE832 の解除、 旋回の許容範囲 = 0°

b) プレーンテキスト入力を使用しないサイクル呼び出しと選択解除

プログラムコード	コメント
G710	; 上記を参照してください
TRAORI	; 上記を参照してください
CYCLE832(0.3,13,0.8)	; 上記を参照してください
...	; 上記を参照してください
CYCLE832(0,0,1)	; 上記を参照してください

4.26.1.40 CYCLE840 - フローティングチャックによるタッピング

構文

CYCLE840(<RTP>, <RFP>, <SDIS>, <DP>, <DPR>, <DTB>, <SDR>, <SDAC>, <ENC>, <MPIT>, <PIT>, <_AXN>, <_PITA>, <_TECHNO>, <_PITM>, <_PTAB>, <_PTABA>, <_GMODE>, <_DMODE>, <_AMODE>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<RTP>	REAL	イニシャル点(abs)
2	Z0	<RFP>	REAL	基準点(abs)
3	SC	<SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<DP>	REAL	穴あけ深さ(abs)、<_AMODE>を参照

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味				
5	Z1	<DPR>	REAL	穴あけ深さ(inc)、<_AMODE>を参照				
6	DT	<DTB>	REAL	穴あけ深さでのドウェル時間(s)/後退後の安全距離、<ENC>を参照				
7		<SDR>	INT	後退時の回転方向				
8	SDE	<SDAC>	INT	サイクル終了後の回転方向				
9		<ENC>	INT	主軸エンコーダを搭載したタッピング(G33)/主軸エンコーダを搭載しないタッピング(G63)				
					0 =	主軸エンコーダを搭載	- <MPIT>/<PIT> からのピッチ - DT なし	
					20 =	主軸エンコーダを搭載	- <MPIT>/<PIT> からのピッチ - 安全距離への後退後の DT あり	
					11 =	主軸エンコーダを搭載しない	- <MPIT>/<PIT> からのピッチ - 穴あけ深さでの DT あり	
					1 =	主軸エンコーダを搭載しない	- プログラム指令送り速度からのピッチ - 穴あけ深さ DT による (送り速度 = 回転速度・ピッチ)	
10		<MPIT>	REAL	「ISO メトリック」のねじサイズのみ(ピッチは運転時に内部で計算されます) 値の範囲:3~48 (M3~M48 の場合)、<PIT>の代わり				
11		<PIT>	REAL	数値で表したピッチ、単位については<_PITA>を参照 値の範囲:> 0、MPIT の代わり				
12		<_AXN>	INT	穴あけ軸	0 =	3 番目のジオメトリ軸		
					1 =	1 番目のジオメトリ軸		
					2 =	2 番目のジオメトリ軸		
					≥ 3 =	3 番目のジオメトリ軸		

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
13		<_PITA>	INT	ピッチ単位(<PIT>と<MPIT>の評価)
				0 = ピッチ(mm) - 評価<MPIT>/<PIT>
				1 = ピッチ(mm) - 評価<PIT>
				2 = ピッチ(TPI) - <PIT>の評価(ねじ山数/inch)
				3 = ピッチ(inch) - 評価<PIT>
				4 = MODULUS - 評価<PIT>
14		<_TECHNO>	INT	用途 ¹⁾
				UNITS(一の位): イグザクトストップ動作
				0 = サイクル呼び出し前と同じイグザクトストップ動作が有効
				1 = イグザクトストップ G601
				2 = イグザクトストップ G602
				3 = イグザクトストップ G603
				TENS(十の位): フィードフォワード制御
				0 = サイクル呼び出し前と同じフィードフォワード制御あり/なしが有効
				1 = フィードフォワード制御あり FFWON
				2 = フィードフォワード制御なし FFWOF
15		<_PITM>	STRING[15]	ピッチ入力のマーカーとなる文字列 ²⁾
16		<_PTAB>	STRING[5]	ねじテーブルの文字列(、"、"ISO"、"BSW"、"BSP"、"UNC") ²⁾
17		<_PTABA>	STRING[20]	ねじテーブルからの選択のための文字列(例: "M 10"、"M 12"、...) ²⁾
18		<_GMODE>	INT	予備

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味			
19		<_DMODE>	INT	表示モード			
				UNITS(一の位):	加工平面 G17/G18/G19		
					0 =	互換性、サイクル呼び出し前に有効な平面がそのまま有効	
					1 =	G17 (サイクルでのみ有効)	
					2 =	G18 (サイクルでのみ有効)	
					3 =	G19 (サイクルでのみ有効)	
				TENS(十の位):	予備		
				HUNDREDS(百の位):	予備		
				THOUSANDS(千の位):	互換性モード(または再コンパイル入力画面のみ)、MD 52216 ビット 0=1 の場合 ¹⁾		
					0 =	テクノロジーパラメータが表示されず(互換性):TECHNO パラメータが有効	
					1 =	テクノロジーパラメータは表示されません:「サイクル呼び出し前と同じ」プログラム値が有効	
TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1368)						
	0 =	入力:全て					
	1 =	入力:簡易					
20		<_AMODE>	INT	代替モード			
				UNITS(一の位):	穴あけ深さ Z1 (abs/inc)		
					0 =	互換性、 <DP>/<DPR>のプログラム指令から	
					1 =	インクレメンタル	
					2 =	アブソリュート	

¹⁾テクノロジーフィールドは、SD52216 \$MCS_FUNCTION_MASK_DRILL の設定データに応じて非表示になる場合があります。

²⁾パラメータ 15、16、17 は、入力画面ねじテーブルでのねじの選択にのみ使用されます。ねじテーブルは、サイクル運転時にサイクル定義を介してアクセスすることはできません。

4.26.1.41 CYCLE899 – オープン溝

構文

```
CYCLE899(<_RTP>, <_RFP>, <_SDIS>, <_DP>, <_LENG>, <_WID>, <_PA>,
<_PO>, <_STA>, <_MID>, <_MIDA>, <_FAL>, <_FALD>, <_FFP1>,
<_CDIR>, <_VARI>, <_GMODE>, <_DMODE>, <_AMODE>, <_UMODE>, <_FS>,
<_ZFS>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	RP	<_RTP>	REAL	イニシャル点(abs)
2	Z0	<_RFP>	REAL	工具軸の基準点(abs)
3	SC	<_SDIS>	REAL	安全距離(基準点に追加、符号なしで入力)
4	Z1	<_DP>	REAL	溝の深さ(abs/inc)、<_AMODE>を参照
5	L	<_LENG>	REAL	溝の長さ(inc)
6	W	<_WID>	REAL	溝の幅(inc)
7	X0	<_PA>	REAL	基準点/開始位置 1 番目の軸(abs)
8	Y0	<_PO>	REAL	基準点/開始位置 2 番目の軸(abs)
9	α0	<_STA>	REAL	1 番目の軸を基準にした回転角度
10	DZ	<_MID>	REAL	最大切り込み深さ(inc)、渦巻きフライス削りの場合のみ
11	DXY	<_MIDA>	REAL	最大平面切り込み、<_AMODE>を参照
12	UXY	<_FAL>	REAL	平面の仕上げ代
13	UZ	<_FALD>	REAL	深さの仕上げ代
14	F	<_FFP1>	REAL	送り速度
15		<_CDIR>	INT	加工方向
				UNITS(一の位):
				0 = ダウンカット
				1 = アップカット
				4 = 点減

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
16		<_VARI>	INT	加工タイプ
				UNITS(一の位):
				1 = 荒削り
				2 = 仕上げ
				3 = 底面の仕上げ
				4 = 端面の仕上げ
				5 = 荒仕上げ
				6 = 面取り
				TENS(十の位): 予備
				HUNDREDS(百の位): 予備
17		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)
				UNITS(一の位): 予備
				TENS(十の位): 予備
				HUNDREDS(百の位): 加工または開始点の計算のみのどちらかを選択します。
				1 = 通常の加工
				THOUSANDS(千の位): 中心/端面による位置指令方法
				0 = 中心による位置指令方法
				1 = 端面を使用した「左側」の位置指令方法(1 番目の軸の「-」方向)
				2 = 端面を使用した「右側」の位置指令方法(1 番目の軸の「+」方向)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
18		<_DMODE>	INT	表示モード
				UNITS(一の位):
				加工平面 G17/G18/G19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
				TENS(十の位): --- 予備
				HUNDREDS(百の位): --- 予備
				THOUSANDS(千の位): --- 予備
19		<_AMODE>	INT	代替モード
				UNITS(一の位):
				溝の深さ Z1
				0 = アブソリュート
				1 = インクレメンタル
				TENS(十の位):
				平面切り込み(<_MIDA>) DXY の単位
				0 = mm
				1 = 工具直径の%
				HUNDREDS(百の位):
20	FS	<_FS>	REAL	面取り幅(inc)
				面取りの切り込み深さ ZFS
				0 = アブソリュート
21	ZFS	<_ZFS>	REAL	面取りの切り込み深さ(工具先端) (abs/inc)、<_AMODE>を参照
				0 = アブソリュート
				1 = インクレメンタル
22		<_UMODE>	INT	予備

4.26 サイクルを外部的にプログラミング

4.26.1.42 CYCLE930 - 溝削り

構文

```
CYCLE930 (<_SPD>, <_SPL>, <_WIDG>, <_WIDG2>, <_DIAG>, <_DIAG2>,
<_STA>, <_ANG1>, <_ANG2>, <_RCO1>, <_RCI1>, <_RCI2>, <_RCO2>,
<_FAL>, <_IDEP1>, <_SDIS>, <_VARI>, <_DN>, <_NUM>, <_DBH>,
<_FF1>, <_NR>, <_FALX>, <_FALZ>, <_DMODE>, <_AMODE>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
1	X0	<_SPD>	REAL	平面軸の基準点(常に直径)		
2	Z0	<_SPL>	REAL	長手軸の基準点		
3	B1	<_WIDG>	REAL	溝の底面の幅		
4	B2	<_WIDG2>	REAL	溝の上部の幅(インタフェースのみ)		
5	T1	<_DIAG>	REAL	基準点の溝の深さ abs で長手方向の加工の場合 = 直径、その他の場合は inc		
6	T2	<_DIAG2>	REAL	基準点の反対側の溝の深さ(インタフェースのみ)、 abs で長手方向の加工の場合 = 直径、その他の場合は inc		
7	α0	<_STA>	REAL	傾斜角(-180 ≤ <_STA> ≤ 180)		
8	α1	<_ANG1>	REAL	基準点で特定された溝の側面の側面角度 1 (0 ≤ <_ANG1> < 90)		
9	α2	<_ANG2>	REAL	基準点の反対側の側面角度 2 (0 ≤ <_ANG2> < 90)		
10	R1/FS1	<_RCO1>	REAL	丸み付け半径または面取り幅 1、基準点の外部		
11	R2/FS2	<_RCI1>	REAL	丸み付け半径または面取り幅 2、基準点の内部		
12	R3/FS3	<_RCI2>	REAL	丸み付け半径または面取り幅 3、基準点の反対側の内部		
13	R4/FS4	<_RCO2>	REAL	丸み付け半径または面取り幅 4、基準点の反対側の外部		
14	U	<_FAL>	REAL	X と Z の仕上げ代、<_VARI> (TEN THOUSANDS)を参照(符号なしで入力)		
15	D	<_IDEP1>	REAL	切り込み時の最大切り込み深さ(符号なしで入力)		
					0 =	1 回目の切削は直接深さいっぱいまで
					> 0 =	1 回目の切削<_IDEP1>、2 回目の切削 2 <_IDEP1>など

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
16	SC	<_SDIS>	REAL	安全距離(符号なしで入力)
17		<_VARI>	INT	加工タイプ
				UNITS(一の位): 予備
				TENS(十の位): 加工プロセス
				1 = 荒削り
				2 = 仕上げ
				3 = 荒削りと仕上げ
				HUNDREDS(百の位): 長手方向/径方向の外部/内部 +Z/+Z と+X/-X の位置決め
				1 = 長手方向/外部 +Z
				2 = 径方向/内部 -X
				3 = 長手方向/内部 +Z
				4 = 径方向/内部 +X
				5 = 長手方向/外部 -Z
				6 = 径方向/外部 -X
				7 = 長手方向/内部 -Z
				8 = 径方向/外部 +X
				THOUSANDS(千の位): 基準点の位置
				0 = 上側の基準点
				1 = 下側の基準点
				TEN THOUSANDS(万の位): 仕上げ代の仕様を定義します
				0 = 輪郭に平行の仕上げ代 U
				1 = UX と UZ 仕上げ代を分離
18		<_DN>	INT	工具の 2 番目の刃先の D 番号
				>0 = 溝削り工具の 2 番目の刃先の工具オフセットの D 番号
				0 = 2 番目の刃先はプログラム指令なし
19	N	<_NUM>	INT	溝の数(0 = 1 つの溝)
20	DP	<_DBH>	REAL	溝間の距離(<_NUM> > 1 の場合にのみ必要)
21	F	<_FF1>	REAL	送り速度

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
22		<_NR>	INT	溝の形状の識別は、形状選択用の垂直ソフトキーに対応しています。
				0 = 面取り/丸み付けなしの 90°の側面
				1 = 面取り/丸み付けあり(α0 なし)の傾斜側面
				2 = 1 と同じ、ただしテーパ上(α0 あり)
23	UX	<_FALX>	REAL	X 軸の仕上げ代、<_VARI> (TEN THOUSANDS)を参照(符号なしで入力)
24	UZ	<_FALZ>	REAL	Z 軸の仕上げ代、<_VARI> (TEN THOUSANDS)を参照(符号なしで入力)
25		<_DMODE>	INT	表示モード
				UNITS(一の位): 加工平面 G17/G18/G19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
26		<_AMODE>	INT	代替モード
				UNITS(一の位):
				溝の上部の位置指令方法(インタフェースのみ)
				0 = 基準点
				1 = 基準点の反対側
				TENS(十の位):
				深さ
				0 = アブソリュート
				1 = インクレメンタル
				HUNDREDS(百の位):
				幅の位置指令方法(インタフェースのみ)
				0 = 外径(上部)
				1 = 内径(底面)
				THOUSANDS(千の位):
				丸み付け/面取り 1 (<_RCO1>)
				0 = 半径
				1 = 面取り
				TEN THOUSANDS(万の位):
				丸み付け/面取り 2 (<_RCI1>)
				0 = 半径
				1 = 面取り
				HUNDRED THOUSANDS(十万の位):
				丸み付け/面取り 3 (<_RCI2>)
				0 = 半径
				1 = 面取り
				ONE MILLION(百万の位):
				丸み付け/面取り 4 (<_RCO2>)
				0 = 半径
				1 = 面取り

4.26.1.43 CYCLE940 – アンダーカット形状 E と形状 F/ねじのアンダーカット

CYCLE940 サイクルを使用して、さまざまなアンダーカットをプログラム指令できます。場合によっては、パラメータ設定が大きく異なることがあります。

4.26 サイクルを外部的にプログラミング

表の追加列で、どのアンダーカットタイプにどのパラメータが必要かを示します。これはサイクル画面の垂直選択ソフトキーに対応します。

- E:アンダーカット形状 E
- F:アンダーカット形状 F
- A-D:DIN 規格ねじのアンダーカット(形状 A~D)
- T:ねじのアンダーカット(形状は自由に定義)

構文

```
CYCLE940 (<_SPD>, <_SPL>, <_FORM>, <_LAGE>, <_SDIS>, <_FFP>,
<_VARI>, <_EPD>, <_EPL>, <_R1>, <_R2>, <_STA>, <_VRT>, <_MID>,
<_FAL>, <_FALX>, <_FALZ>, <_PITI>, <_PTAB>, <_PTABA>, <_DMODE>,
<_AMODE>)
```

パラメータ

N o.	マスク パラメ ータ	内部パラ メータ	データタ イプ	形状のプログ ラミング				意味
				E	F	A- D	T	
1	X0	<_SPD>	REAL	X	X	X	X	平面軸の基準点(常に直径)
2	Z0	<_SPL>	REAL	X	X	X	X	長手軸の基準点(abs)
3	FORM	<_FORM>	CHAR	X	X	X	X	アンダーカットの形状(大文字、たとえば「T」) アンダーカット値を選択する選択テーブル
								A = 外径、DIN76 基準、 A = 通常
								B = 外径、DIN76 基準、 B = ショート
								C = 内径、DIN76 基準、 C = 通常
								D = 内径、DIN76 基準、 D = ショート
								E = DIN509 基準
								F = DIN509 基準
								T = 自由形状

4.26 サイクルを外部的にプログラミング

N o.	マスク パラメ ータ	内部パラ メータ	データタ イプ	形状のプログ ラミング				意味			
4	POSITI ON	<_LAGE>	INT	X	X	X	X	アンダーカット の位置(Z に平行)	0 =	外径 +Z: ____	
									1 =	外径 -Z: ____/	
									2 =	内径 +Z: /-----	
									3 =	内径 -Z: -----\	
5	SC	<_SDIS>		X	X	X	X	安全距離(inc)			
6	F	<_FFP>		X	X	X	X	加工送り速度(mm/rev)			
7		<_VARI>	INT	-	-	X	X	加工タイプ			
								UNITS(一の位):	加工タイプ		
									1 =	荒削り	
									2 =	仕上げ	
									3 =	荒削り + 仕上げ	
								TENS(十の位):	加工方法		
									0 =	輪郭に平行	
									1 =	長手方向	
アンダーカット形状 E と F は常に、仕上げと同様に 1 回のパスで加工されます。											
8	X1	<_EPD>		X	X	-	-	仕上げ代 X (abs/inc)、<_AMODE>を参照			
				-	-	-	X	アンダーカットの深さ(abs/inc)、<_AMODE>を参照			
9	Z1	<_EPL>		-	X	-	-	仕上げ代 Z			
				-	-	-	X	アンダーカットの幅(abs/inc)、<_AMODE>を参照			
10	R1	<_R1>		-	-	-	X	斜面の丸み付け半径			
11	R2	<_R2>		-	-	-	X	コーナの丸み付け半径			
12	α	<_STA>		-	-	X	X	切り込み角度			
13	VX	<_VRT>		X	X	-	-	横送り X (abs/inc)、<_AMODE>を参照			
				-	-	X	X	仕上げ時の横送り X (abs/inc)、<_AMODE>を参照			
14	D	<_MID>		-	-	X	X	深さ切り込み			
15	U	<_FAL>		-	-	X	X	輪郭に平行の仕上げ代、<_AMODE>を参照			
16	UX	<_FALX>		-	-	X	X	仕上げ代 X			
17	UZ	<_FALZ>		-	-	X	X	仕上げ代 Z			

4.26 サイクルを外部的にプログラミング

N o.	マスク パラメ ータ	内部パラ メータ	データタ イプ	形状のプログ ラミング				意味
18	P	<_PITI>	INT	-	-	X	-	ピッチ、形状 A~D を選択。M1 ... M68 に対応。
								0 = 0.20 6 = 0.50 12 = 1.25 18 = 3.50 1 = 0.25 7 = 0.60 13 = 1.50 19 = 4.00 2 = 0.30 8 = 0.70 14 = 1.75 20 = 4.50 3 = 0.35 9 = 0.75 15 = 2.00 21 = 5.00 4 = 0.40 10 = 0.80 16 = 2.50 22 = 5.50 5 = 0.45 11 = 1.00 17 = 3.00 23 = 6.00
				X	X	-	-	半径/深さ、形状 E、F を選択。
								0 = 0.6 x 0.3 4 = 2.5 x 0.4 8 = 0.1 x 0.1 1 = 1.0 x 0.4 5 = 4.0 x 0.5 9 = 0.2 x 0.1 2 = 1.0 x 0.2 6 = 0.4 x 0.2 3 = 1.6 x 0.3 7 = 0.6 x 0.2
19		<_PTAB>	STRING [5]					ねじテーブルの文字列(、"「ISO」、「BSW」、「BSP」、 「UNC」) (インタフェースのみ)
20		<_PTABA >	STRING [20]					ねじテーブルからの選択のための文字列 (例: 「M10」、「M12」、...) (インタフェースのみ)
21		<_DMODE >	INT					表示モード
				X	X	X	X	UNITS(一の位): 加工平面 G17/G18/G19
								0 = 互換性、サイクル呼び出し 前に有効な平面がそのまま 有効
								1 = G17 (サイクルでのみ有効)
								2 = G18 (サイクルでのみ有効)
								3 = G19 (サイクルでのみ有効)

4.26 サイクルを外部的にプログラミング

N o.	マスク パラメ ータ	内部パラ メータ	データタ イプ	形状のプログ ラミング	意味	
22		<_AMODE >	INT		代替モード	
				X X - X	UNITS(一の位):	パラメータ<_EPD>仕上げ代 X またはアンダーカットの深さ
						0 = アブソリュート(常に直径)
						1 = インCREMENTAL
				X X - X	TENS(十の位):	パラメータ<_EPL> 仕上げ代 Z またはアンダーカットの幅
						0 = アブソリュート
						1 = インCREMENTAL
				X X X X	HUNDREDS(百の位):	パラメータ<_VRT>横送り X
						0 = アブソリュート(常に直径)
						1 = インCREMENTAL
				- - X X	THOUSANDS(千の位):	仕上げ代
						0 = 輪郭に平行の仕上げ代 (<_FAL>)
						1 = 個別の加工仕上げ代 (<_FALX>/<_FALZ>)

4.26.1.44 CYCLE951 - 荒削り

構文

CYCLE951(<_SPD>, <_SPL>, <_EPD>, <_EPL>, <_ZPD>, <_ZPL>, <_LAGE>, <_MID>, <_FALX>, <_FALZ>, <_VARI>, <_RF1>, <_RF2>, <_RF3>, <_SDIS>, <_FF1>, <_NR>, <_DMODE>, <_AMODE>)

パラメータ

No.	マスクパラ メータ	内部パラメ ータ	データタイ プ	意味
1	X0	<_SPD>	REAL	基準点(abs、常に直径)
2	Z0	<_SPL>	REAL	基準点(abs)
3	X1	<_EPD>	REAL	終点

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
4	Z1	<_EPL>	REAL	終点		
5	XM α1 α2	<_ZPD>	REAL	中間点、<_DMODE> (TENS)を参照		
6	ZM α1 α2	<_ZPL>	REAL	中間点、<_DMODE> (TENS)を参照		
7	位置	<_LAGE>	INT	切削コーナの位置	0 =	外部/背面
					1 =	外部/前面
					2 =	内部/背面
					3 =	内部/前面
8	d	<_MID>	REAL	切り込み時の最大切り込み深さ		
9	UX	<_FALX>	REAL	X の仕上げ代		
10	UZ	<_FALZ>	REAL	Z の仕上げ代		
11		<_VARI>	INT	加工タイプ		
				UNITS(一の位):	座標系の切削方向(長手方向または径方向)	
					1 =	長手方向
					2 =	正面
				TENS(十の位):		
					1 =	最終加工代まで荒削り
					2 =	仕上げ
				HUNDREDS(百の位):	予備	
				THOUSANDS(千の位):	予備	
				TEN THOUSANDS(万の位):	予備	
12	R1/FS1	<_RF1>	REAL	丸み付け半径または面取り幅 1、<_AMODE> (TEN THOUSANDS)を参照		
13	R2/FS2	<_RF2>	REAL	丸み付け半径または面取り幅 2、<_AMODE> (HUNDRED THOUSANDS)を参照		

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
14	R3/FS3	<_RF3>	REAL	丸み付け半径または面取り幅 3、<_AMODE> (ONE MILLION) を参照
15	SC	<_SDIS>	REAL	安全距離
16	F	<_FF1>	REAL	荒削り/仕上げの送り速度
17		<_NR>	INT	切削タイプの識別(形状選択用の垂直ソフトキーに対応)
				0 = 切削 1、面取り/丸み付けなしの 90°コーナ
				1 = 切削 2、面取り/丸み付けありの 90°コーナ
				2 = 切削 3、面取り/丸み付けありの任意のコーナ
18		<_DMODE>	INT	表示モード
				UNITS(一の位):
				加工平面 G17/G18/G19
				0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
				1 = G17 (サイクルでのみ有効)
				2 = G18 (サイクルでのみ有効)
				3 = G19 (サイクルでのみ有効)
				TENS(十の位):
				入力<_ZPD>/<_ZPL>の形式
				0 = Xm/Zm
				1 = Xm/a1
				2 = Xm/a2
				3 = a1/Zm
				4 = a2/Zm
				5 = a1/a2

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
21		<_AMODE>	INT	代替モード	
				UNITS(一の位):	X の中間点
					0 = アブソリュート、径方向軸の直径の値
					1 = インクレメンタル、径方向軸の半径の値
				TENS(十の位):	Z の中間点
					0 = アブソリュート
					1 = インクレメンタル
				HUNDREDS(百の位):	X の終点
					0 = アブソリュート、径方向軸の直径の値
					1 = インクレメンタル、径方向軸の半径の値
				THOUSANDS(千の位):	Z の終点
					0 = アブソリュート
					1 = インクレメンタル
				TEN THOUSANDS(万の位):	丸み付け/面取り 1
					0 = 半径
					1 = 面取り
				HUNDRED THOUSANDS(十万の位):	丸み付け/面取り 2
					0 = 半径
					1 = 面取り
				ONE MILLION(百万の位):	丸み付け/面取り 3
					0 = 半径
					1 = 面取り

4.26.1.45 CYCLE952 – 旋削/旋削削り残し/プランジ切削/プランジ切削削り残し/プランジ旋削/プランジ旋削削り残し

構文

```
CYCLE952 (<_PRG>, <_CON>, <_CONR>, <_VARI>, <_F>, <_FR>, <_RP>,
<_D>, <_DX>, <_DZ>, <_UX>, <_UZ>, <_U>, <_U1>, <_BL>, <_XD>,
<_ZD>, <_XA>, <_ZA>, <_XB>, <_ZB>, <_XDA>, <_XDB>, <_N>, <_DP>,
<_DI>, <_SC>, <_DN>, <_GMODE>, <_DMODE>, <_AMODE>, <_PK>, <_DCH>,
<_FS>)
```

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1	PRG	<_PRG>	STRING[100]	旋削プログラムの名称
2	CON	<_CON>	STRING[100]	素材の更新済みの輪郭が呼び出されるプログラムの名称(削り残し加工時)
3	CONR	<_CONR>	STRING[100]	素材の更新済みの輪郭(<_AMODE> TEN THOUSANDS を参照)が書き込まれるプログラムの名称

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
4		<_VARI>	INT	加工タイプ	
				UNITS(一の位):	切削のタイプ
					1 = 長手方向
					2 = 正面
					3 = 輪郭に平行
				TENS(十の位):	製造工程(<_GMODE> HUNDREDS を参照)
					1 = 荒削り
					2 = 仕上げ
					3 = 予備
					4 = 荒削り、2 個のチャンネル
					5 = 仕上げ、2 個のチャンネル
				HUNDREDS(百の位):	加工方向
					1 = 加工方向 X -
					2 = 加工方向 X +
					3 = 加工方向 Z -
					4 = 加工方向 Z +
				THOUSANDS(千の位):	切り込み方向
					1 = 外部 X -
					2 = 内部 X +
					3 = 前面 Z -
					4 = 背面 Z +
				TEN THOUSANDS(万の位):	仕上げ代の仕様を定義します
					0 = UX と UZ 仕上げ代を分離
					1 = 輪郭に平行の仕上げ代 U
				HUNDRED THOUSANDS(十万の位):	丸み付け
					0 = 互換性、自動丸み付け
					1 = 輪郭での丸み付けあり
					2 = 丸み付けなし
					3 = 自動丸み付け

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
				ONE MILLION(百万の位):	レリーフカット	
					0 =	溝加工時に位置は評価されません - 削り残しと溝旋削 - 削り残し
					1 =	レリーフカットの加工
					2 =	レリーフカットの加工なし
				TEN MILLION(千万の位):	回転中心の背面/前面	
					0 =	回転中心の前面で加工
					1 =	予備
5	F	<_F>	REAL	荒削り/仕上げの送り速度		
	FZ			横軸の溝旋削の切り込み		
6	FR	<_FR>	REAL	レリーフカット切り込み時の送り速度、荒削り		
	FX			縦軸の溝旋削の切り込み		
7	RP	<_RP>	REAL	内部加工用のイニシャル点(abs、常に直径)		
8	D	<_D>	REAL	荒削り切り込み(<_AMODE> UNITS を参照)		
9	Dx	<_DX>	REAL	X 切り込み(<_AMODE> UNITS を参照)		
10	DZ	<_DZ>	REAL	Z 切り込み(<_AMODE> UNITS を参照)		
11	UX	<_UX>	REAL	仕上げ代 X、(<_VARI> TEN THOUSANDS を参照)		
12	UZ	<_UZ>	REAL	仕上げ代 Z、(<_VARI> TEN THOUSANDS を参照)		
13	U	<_U>	REAL	輪郭に平行の仕上げ代、(<_VARI> TEN THOUSANDS を参照)		
14	U1	<_U1>	REAL	仕上げ時の追加仕上げ代(<_AMODE> THOUSANDS を参照)		
15	BL	<_BL>	INT	素材の定義	1 =	仕上げ代付きの円筒
					2 =	仕上げ部の輪郭の仕上げ代
					3 =	素材の輪郭が指定されています
16	XD	<_XD>	REAL	素材 X の定義(<_AMODE> HUNDRED THOUSANDS を参照)		
17	ZD	<_ZD>	REAL	素材 Z の定義(<_AMODE> ONE MILLION を参照)		
18	XA	<_XA>	REAL	限界値 1 X (abs、常に直径)		
19	ZA	<_ZA>	REAL	限界値 1 Z (abs)		
20	XB	<_XB>	REAL	限界値 2 X (<_AMODE> TEN MILLIONS を参照)		
21	ZB	<_ZB>	REAL	限界値 2 Z (<_AMODE> HUNDRED MILLIONS を参照)		

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味		
22	XDA	<_XDA>	REAL	端面の 1 番目の溝位置の溝加工限界値 1(abs、常に直径)		
23	XDB	<_XDB>	REAL	端面の 2 番目の溝位置の溝加工限界値 1(abs、常に直径)		
24	N	<_N>	INT	溝の数		
25	DP	<_DP>	REAL	溝の間の距離	直線溝:Z 軸に平行	
					径方向の溝:X 軸に平行	
26	Di	<_DI>	REAL	切り込みの中断のための距離	0 =	中断なし
					> 0 =	中断あり
27	SC	<_SC>	REAL	障害物を回避するための安全距離、インクレメンタル		
28	D2	<_DN>	INT	工具の 2 番目の刃先の D 番号。プログラムされていない場合 ⇒ D+1		

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
29		<_GMODE>	INT	形状モード(プログラム指令された形状データの評価)
				UNITS(一の位):
				TENS(十の位):
				HUNDREDS(百の位):
				加工または開始点の計算のみのどちらかを選択します。
				0 = 通常の加工(互換性モードは不要)
				1 = 通常の加工
				2 = 開始点を計算 - 加工なし (ShopMill/ShopTurn からの呼び出しのみ)
				THOUSANDS(千の位):
				限界値
				0 = なし
				1 = あり
				TEN THOUSANDS(万の位):
				限界値 1 X の入力
				0 = なし
				1 = あり
				HUNDRED THOUSANDS(十万の位):
				限界値 2 X の入力
				0 = なし
				1 = あり
				ONE MILLION(百万の位):
				限界値 1 Z の入力
				0 = なし
				1 = あり
				TEN MILLION(千万の位):
				限界値 2 Z の入力
				0 = なし
				1 = あり

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
30		<_DMODE>	INT	表示モード	
				UNITS(一の位):	加工平面 G17/G18/G19
					0 = 互換性、サイクル呼び出し前に有効な平面がそのまま有効
					1 = G17 (サイクルでのみ有効)
					2 = G18 (サイクルでのみ有効)
					3 = G19 (サイクルでのみ有効)
				TENS(十の位):	テクノロジーモード
					1 = 輪郭に沿った切削
					2 = 輪郭の溝削り
					3 = 溝旋削
				HUNDREDS(百の位):	削り残し除去加工
					0 = なし
					1 = あり
				THOUSANDS(千の位):	--- 予備
				TEN THOUSANDS(万の位):	サイクル画面の機能スケーリング (ページ 1368)
					0 = 入力:全て
					1 = 入力:簡易
				HUNDRED THOUSANDS(十万の位):	自動プログラム名
					0 = なし
					1 = あり

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
31		<_AMODE>	INT	代替モード	
				UNITS(一の位):	切り込み方法
					0 = 輪郭に平行した切削時の DX と DZ 切り込み
					1 = D 切り込み
				TENS(十の位):	切り込み方法
					0 = 可変の切削深さ (90 ... 100%)
					1 = 一定の切削深さ
				HUNDREDS(百の位):	分割切削
					0 = 均一
					1 = 端面に割り当て
				THOUSANDS(千の位):	輪郭仕上げ代 U1 の選択、二重仕上げ
					0 = なし
					1 = あり
				TEN THOUSANDS(万の位):	素材の選択の更新
					0 = なし
					1 = あり
				HUNDRED THOUSANDS(十万の位):	素材 XD の仕上げ代の選択
					0 = アブソリュート、径方向軸の直径の値
					1 = インクレメンタル、径方向軸の半径の値
				ONE MILLION(百万の位):	素材 ZD の仕上げ代の選択
					0 = アブソリュート
					1 = インクレメンタル
				TEN MILLION(千万の位):	限界値 2 XB の選択
					0 = アブソリュート、径方向軸の直径の値
					1 = インクレメンタル、径方向軸の半径の値

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
				HUNDRED MILLION(一億の位):
				限界値 2 ZB の選択
				0 = アブソリュート
				1 = インクレメンタル
				ONE BILLION(一兆の位):
				0 = マスタチャネル
				1 = スレーブチャネル
32		<_PK>	INT	機械で利用できるチャネルが 2 個より多い場合は、協調チャネルの番号
33	DCH	<_DCH>	REAL	チャネルオフセット
34	FS	<_FS>	REAL	全て加工時の仕上げ送り速度

4.26.1.46 CYCLE4071 - 反転点での切り込みによる長手方向の研削

構文

CYCLE4071 (<S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>, <S_H>, <S_A1>, <S_A2>)

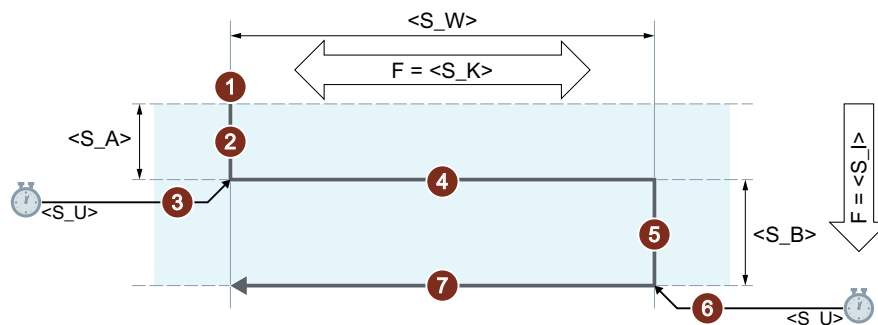
パラメータ

No.	パラメータ	データタイプ	意味
1	<S_A>	REAL	開始位置での切り込み深さ
2	<S_B>	REAL	終了位置での切り込み深さ
3	<S_W>	REAL	研削幅
4	<S_U>	REAL	スパークアウト時間
5	<S_I>	REAL	切り込みの送り速度
6	<S_K>	REAL	径方向の切り込みの送り速度
7	<S_H>	INT	繰り返し回数
8	<S_A1>	AXIS	切り込み軸(任意選択)または 1 番目のジオメトリ軸
9	<S_A2>	AXIS	揺動軸(任意選択)または 2 番目のジオメトリ軸

機能

このサイクルは繰り返し切り込みの実行に使用されます。切り込み深さは開始位置と終了位置で変更することができます。切り込み間で接線方向の移動があります。

手順



- ① 揺動軸の現在の位置でサイクルを開始します。
- ② 切り込み軸を開始点<S_A>の切込み深さまで、切り込み<S_I>用の送り速度で移動します。
- ③ スパークアウト時間<S_U>のスパークアウト。
- ④ 研削幅<S_W>の揺動軸を移動軌跡として、径方向の切り込み<S_K>用の送り速度で移動します。
- ⑤ 切り込み軸を終了点の切込み深さ<S_B>まで、切り込み<S_I>用の送り速度で移動します。
- ⑥ スパークアウト時間<S_U>のスパークアウト。
- ⑦ 研削幅<S_W>の揺動軸を移動軌跡として、径方向の切り込み<S_K>用の送り速度で開始点まで移動します。

繰り返しの手順を示しています。

この手順は、プログラムされた繰り返し回数<S_H>に達するまで繰り返されます。

注記

この手順はシングルブロックによって中断することはできません。

4.26 サイクルを外部的にプログラミング

例

以下のサイクルパラメータによる 2 つの揺動移動の実行

- 開始位置での切り込み深さ:0.02 mm
- 終了位置での切り込み深さ:0.01 mm
- ストローク : 100 mm
- スパークアウト時間:1 秒
- 切り込みの送り速度:1 mm/min
- 径方向の送り速度:1000 mm/min
- 繰り返し回数:2
- 揺動軸と切り込み軸:標準ジオメトリ軸

プログラムコード

```
N10 T1 D1
N20 CYCLE4071(0.02,0.01,100,1,1,1000,2)
N30 M30
```

4.26.1.47 CYCLE4072 - 反転点での切り込みによる長手方向の研削とキャンセル信号

構文

```
CYCLE4072(<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_I>, <S_K>,
<S_H>, <S_A1>, <S_A2>)
```

パラメータ

No.	パラメータ	データタイプ	意味
1	<S_GAUGE>	STRING	切り込みのキャンセル条件: 1. 高速入力回数 2. 論理式
2	<S_A>	REAL	開始位置での切り込み深さ
3	<S_B>	REAL	終了位置での切り込み深さ
4	<S_W>	REAL	研削幅
5	<S_U>	REAL	スパークアウト時間
6	<S_I>	REAL	切り込みの送り速度

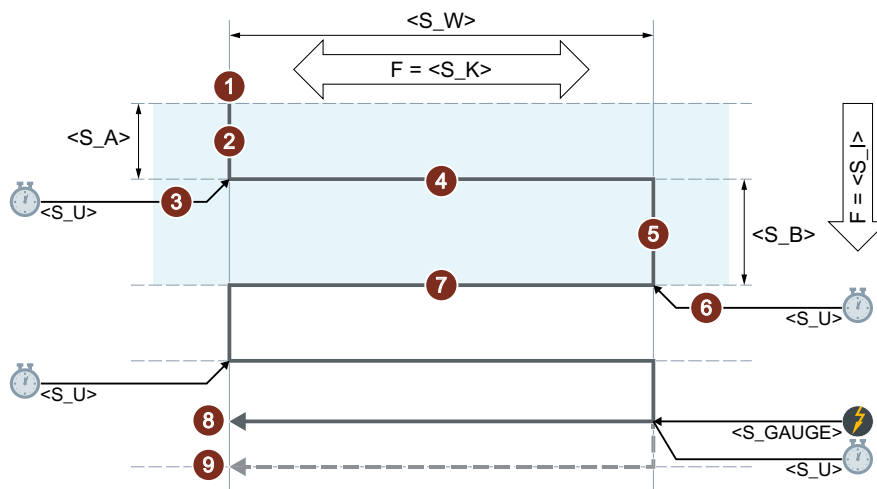
No.	パラメータ	データタイプ	意味
7	<S_K>	REAL	径方向の切り込みの送り速度
8	<S_H>	INT	繰り返し回数
9	<S_A1>	AXIS	切り込み軸(任意選択)または 1 番目のジオメトリ軸
10	<S_A2>	AXIS	揺動軸(任意選択)または 2 番目のジオメトリ軸

機能

このサイクルは、外部キャンセル信号を考慮した繰り返し切り込みの実行に使用されます。切り込み深さは開始位置と終了位置で変えることができます。切り込み間で接線方向の移動があります。キャンセル条件が満たされたときに、深さ切り込みがキャンセルされます。深さ切り込みのキャンセル後、常に完全なストロークが 1 回実行されます。

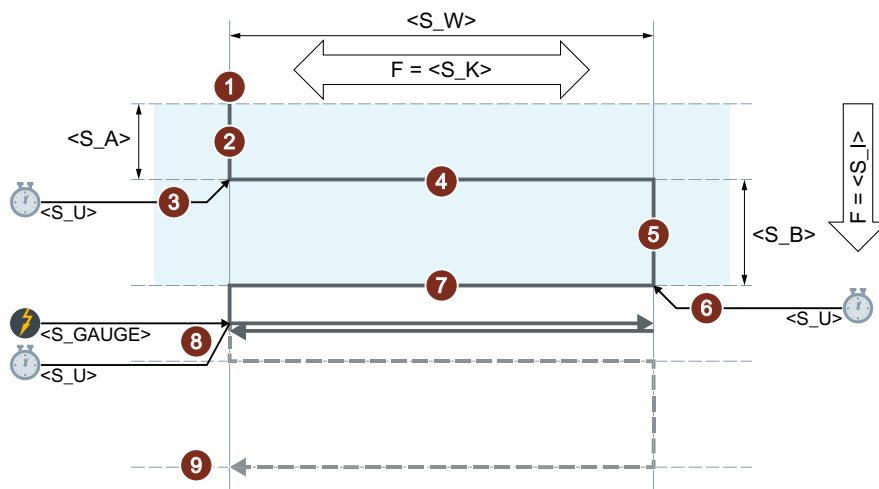
手順

終了位置での切り込みのキャンセル



4.26 サイクルを外部的にプログラミング

開始位置での切り込みのキャンセル



- ① 揺動軸の現在の位置でサイクルを開始します。
 - ② 切り込み軸を開始点<S_A>の切込み深さまで、切り込み<S_I>用の送り速度で移動します。
 - ③ スパークアウト時間<S_U>のスパークアウト。
 - ④ 研削幅<S_W>の揺動軸を移動軌跡として、径方向の切り込み<S_K>用の送り速度で移動します。
 - ⑤ 切り込み軸を開始点<S_B>の切込み深さまで、切り込み<S_I>用の送り速度で移動します。
 - ⑥ スパークアウト時間<S_U>のスパークアウト。
 - ⑦ 切り込み軸を開始点<S_W>の切込み深さまで、切り込み<S_K>用の送り速度で移動します。
 - ⑧ キャンセル信号:次の開始点に達すると加工が停止します。
 - ⑨ キャンセル信号なし:この手順は、プログラムされた繰り返し回数<S_H>に達するまで繰り返されます。
- 繰り返しの手順を示しています。

注記

この手順はシングルブロックによって中断することはできません。

リソース

サイクルでは、リソースとして、ブロック全体でのシンクロナイズドアクションとシンクロナイズドアクション変数を使用します。シンクロナイズドアクションは、シンクロナイズドアクション範囲(CUS.DIR-1...、CMA.DIR-1000...、CST.DIR-1199...)の空き領域から自動的に決定されます。シンクロナイズドアクション変数として SYG_IS[1] が使用されます。

例

例 1:2 ストロークによる揺動:

サイクルパラメータ

- 開始位置での切り込み深さ:0.02 mm
- 終了位置での切り込み深さ:0.01 mm
- ストローク : 100 mm
- スパークアウト時間:1 秒
- 切り込みの送り速度:1 mm/min
- 径方向の送り速度:1000 mm/min
- 繰り返し回数:2
- 揺動軸と切り込み軸:標準ジオメトリ軸

キャンセル信号:高速入力 1 (\$A_IN[1])

プログラムコード

```
N10 T1 D1
N20 CYCLE4072 ("1",0.02,0.01,100,1,1,1000,2)
N30 M30
```

例 2:2 ストロークによる揺動:

サイクルパラメータ

- 開始位置での切り込み深さ:0.02 mm
- 終了位置での切り込み深さ:0.01 mm
- ストローク : 100 mm
- スパークアウト時間:1 秒
- 切り込みの送り速度:1 mm/min

4.26 サイクルを外部的にプログラミング

- 径方向の送り速度:1000 mm/min
- 繰り返し回数:2
- 揺動軸と切り込み軸:標準ジオメトリ軸

キャンセル信号:変数\$A_DBR[20] < 0.01

プログラムコード

```
N10 T1 D1
N20 CYCLE4072 ("($A_DBR[20]<0.01)",0.02,0.01,100,1,1,1000,2)
N30 M30
```

4.26.1.48 CYCLE4073 - 連続切り込みによる長手方向の研削

構文

```
CYCLE4073(<S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>, <S_A1>,
<S_A2>)
```

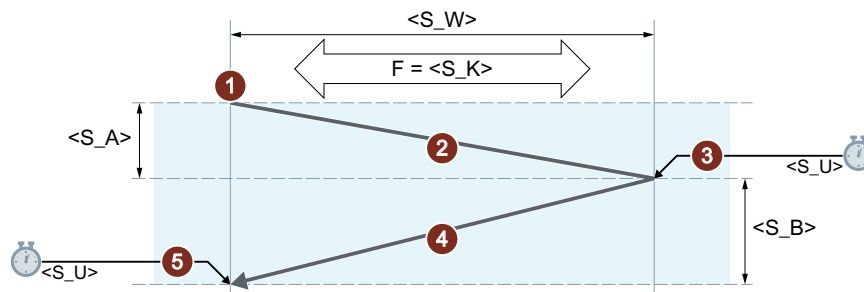
パラメータ

No.	パラメータ	データタイプ	意味
1	<S_A>	REAL	開始位置での切り込み深さ
2	<S_B>	REAL	終了位置での切り込み深さ
3	<S_W>	REAL	研削幅
4	<S_U>	REAL	スパークアウト時間
5	<S_K>	REAL	径方向の切り込みの送り速度
6	<S_H>	INT	繰り返し回数
7	<S_A1>	AXIS	切り込み軸(任意選択)または 1 番目のジオメトリ軸
8	<S_A2>	AXIS	揺動軸(任意選択)または 2 番目のジオメトリ軸

機能

このサイクルは繰り返し切り込みの実行に使用されます。開始位置から終了位置までの切り込みと、終了位置から開始位置までの切り込みは変えることができます。

手順



- ① 切り込み深さ 0 の揺動軸の現在の位置でサイクルを開始します。
 - ② 研削幅<S_W>の揺動軸を移動軌跡として、径方向の切り込み<S_K>用の送り速度で、切り込み深さを開始位置の切り込み深さ<S_A>まで連続して深くしながら移動します。
 - ③ スパークアウト時間<S_U>のスパークアウト。
 - ④ 研削幅<S_W>の揺動軸を移動軌跡として、径方向の切り込み<S_K>用の送り速度で、切り込み深さを終了位置の切り込み深さ<S_B>まで連続して深くしながら開始点まで移動します。
 - ⑤ スパークアウト時間<S_U>のスパークアウト。
- 繰り返しの手順を示しています。
- この手順は、プログラムされた繰り返し回数<<S_H>>に達するまで繰り返されます。

注記

この手順はシングルブロックによって中断することはできません。

例

2 ストロークによる揺動:

サイクルパラメータ

- 開始位置での切り込み深さ:0.02 mm
- 終了位置での切り込み深さ:0.01 mm
- ストローク : 100 mm
- スパークアウト時間:1 秒
- 径方向の送り速度:1000 mm/min

4.26 サイクルを外部的にプログラミング

- 繰り返し回数:2
- 揺動軸と切り込み軸:標準ジオメトリ軸

プログラムコード

```

N10 T1 D1
N20 CYCLE4073(0.02,0.01,100,1,1000,2)
N30 M30

```

4.26.1.49 CYCLE4074 - 連続切り込みによる長手方向の研削とキャンセル信号

構文

```

CYCLE4074 (<S_GAUGE>, <S_A>, <S_B>, <S_W>, <S_U>, <S_K>, <S_H>,
<S_A1>, <S_A2>)

```

パラメータ

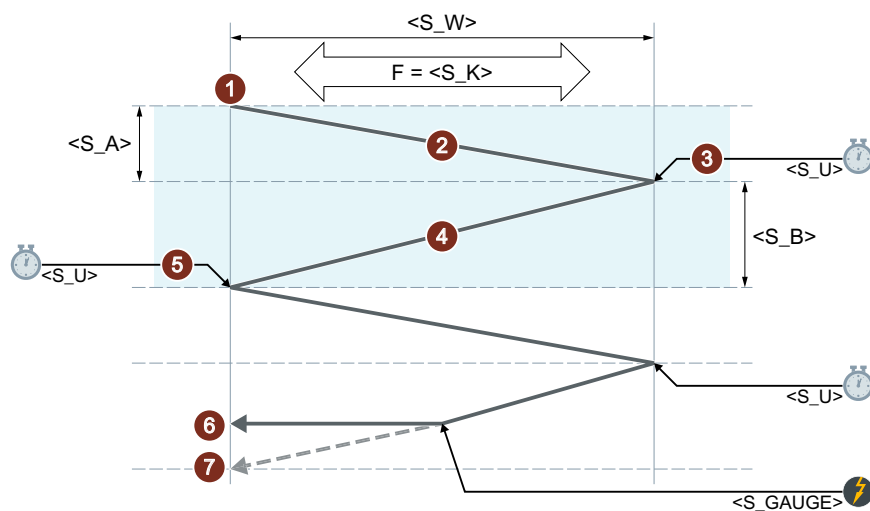
No.	パラメータ	データタイプ	意味
1	<S_GAUGE>	STRING	切り込みのキャンセル条件: 1. 高速入力回数 2. 論理式
2	<S_A>	REAL	開始位置での切り込み深さ
3	<S_B>	REAL	終了位置での切り込み深さ
4	<S_W>	REAL	研削幅
5	<S_U>	REAL	スパークアウト時間
6	<S_K>	REAL	径方向の切り込みの送り速度
7	<S_H>	INT	繰り返し回数
8	<S_A1>	AXIS	切り込み軸(任意選択)または 1 番目のジオメトリ軸
9	<S_A2>	AXIS	揺動軸(任意選択)または 2 番目のジオメトリ軸

機能

このサイクルは、外部キャンセル信号などを考慮した切り込みの繰り返しの実行に使用されます。切り込み深さは開始位置と終了位置で変えることができます。キャンセル条件が満たされたときに、深さ切り込みがキャンセルされます。深さ切り込みのキャンセル後、常に完全なストロークが 1 回実行されます。

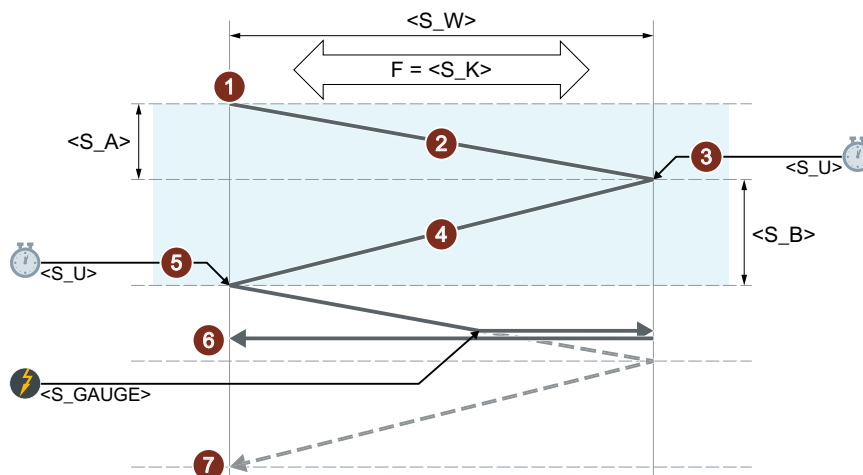
手順

終了位置から開始位置までの切り込みのキャンセル



4.26 サイクルを外部的にプログラミング

開始位置から終了位置までの切り込みのキャンセル



- ① 切り込み深さ 0 の揺動軸の現在の位置でサイクルを開始します。
 - ② 研削幅<<S_W>>の揺動軸を移動軌跡として、径方向の切り込み<<S_K>>用の送り速度で、切り込み深さを開始位置の切り込み深さ<<S_A>>まで連続して深くしながら移動します。
 - ③ スパークアウト時間<S_U>のスパークアウト。
 - ④ 研削幅<<S_W>>の揺動軸を移動軌跡として、径方向の切り込み<<S_K>>用の送り速度で、切り込み深さを終了位置の切り込み深さ<<S_B>>まで連続して深くしながら移動します。
 - ⑤ スパークアウト時間<S_U>のスパークアウト。
 - ⑥ キャンセル信号:深さ切り込みがキャンセルされます。次の開始点に達すると加工が停止します。
 - ⑦ キャンセル信号なし:この手順は、プログラムされた繰り返し回数<S_H>に達するまで繰り返されます。
- 繰り返しの手順を示しています。

注記

この手順はシングルブロックによって中断することはできません。

リソース

サイクルでは、リソースとして、ブロック全体でのシンクロナイズドアクションとシンクロナイズドアクション変数を使用します。シンクロナイズドアクションは、シンクロナイズドアクション範囲(CUS.DIR-1 ...、CMA.DIR-1000 ...、CST.DIR-1199 ...)の空き領域から自動的に決定されます。シンクロナイズドアクション変数として SYG_IS[1] が使用されます。

例

例 1:2 ストロークによる揺動:

サイクルパラメータ

- 開始位置での切り込み深さ:0.02 mm
- 終了位置での切り込み深さ:0.01 mm
- ストローク : 100 mm
- スパークアウト時間:1 秒
- 径方向の送り速度:1000 mm/min
- 繰り返し回数:2
- 揺動軸と切り込み軸:標準ジオメトリ軸

キャンセル信号:高速入力 1 (\$A_IN[1])

プログラムコード

```
N10 T1 D1
N20 CYCLE4074("1",0.02,0.01,100,1,1000,2)
N30 M30
```

例 2:2 ストロークによる揺動:

サイクルパラメータ

- 開始位置での切り込み深さ:0.02 mm
- 終了位置での切り込み深さ:0.01 mm
- ストローク : 100 mm
- スパークアウト時間:1 秒
- 径方向の送り速度:1000 mm/min

4.26 サイクルを外部的にプログラミング

- 繰り返し回数:2
- 揺動軸と切り込み軸:標準ジオメトリ軸

キャンセル信号:変数\$A_DBR[20] < 0.01

プログラムコード

```
N10 T1 D1
N20 CYCLE4074 ("($A_DBR[20]<0.01)",0.02,0.01,100,1,1000,2)
N30 M30
```

4.26.1.50 CYCLE4075 - 反転点での切り込みによる平面研削

構文

```
CYCLE4075(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>,
<S_A1>, <S_A2>)
```

パラメータ

No.	パラメータ	データタイプ	意味
1	<S_I>	REAL	開始位置での切り込み深さ
2	<S_J>	REAL	終了位置での切り込み深さ
3	<S_K>	REAL	合計切り込み深さ
4	<S_A>	REAL	研削幅
5	<S_R>	REAL	切り込みの送り速度
6	<S_F>	REAL	径方向の切り込みの送り速度
7	<S_P>	REAL	スパークアウト時間
8	<S_A1>	AXIS	切り込み軸(任意選択)
9	<S_A2>	AXIS	揺動軸(任意選択)

機能

このサイクルは、切り込みステップでの切り込み全体の深さでの加工に使用されます。切り込み深さは開始位置と終了位置で変えることができます。切り込み間で接線方向の移動があります。

位置データ P1~P4 は負または正とすることができます。

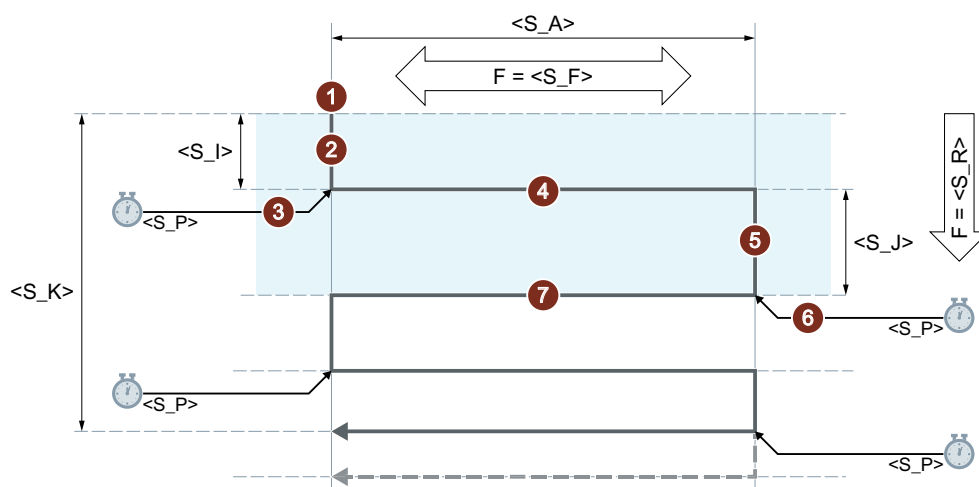
4.26 サイクルを外部的にプログラミング

切り込み軸または揺動軸の指定は任意です。一方または両方のパラメータを指定していない場合、サイクルはチャンネルの最初の 2 つのジオメトリ軸を使用します。

開始点と終了点の切り込み深さの合計が 0、または合計切り込み深さが 0 の場合、スパークアウトストロークのみが実行されます。

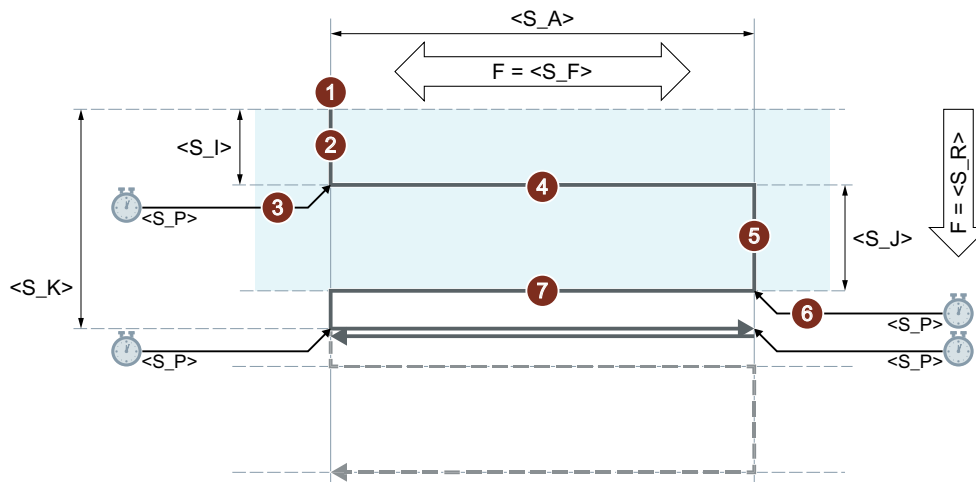
手順

2 番目の反転点での切り込みで合計切り込み深さに到達



4.26 サイクルを外部的にプログラミング

1 番目の反転点での切り込みで合計切り込み深さに到達



- ① 揺動軸の現在の位置でサイクルを開始します。
- ② 切り込み軸を開始点<S_I>の切込み深さまで、切り込み<S_R>用の送り速度で移動します。
- ③ スパークアウト時間<S_P>のスパークアウト。
- ④ 研削幅<S_A>の揺動軸を移動軌跡として、径方向の切り込み<S_F>用の送り速度で移動します。
- ⑤ 切り込み軸を開始点<S_J>の切込み深さまで、切り込み<S_R>用の送り速度で移動します。
- ⑥ スパークアウト時間<S_P>のスパークアウト。
- ⑦ 切り込み軸を開始点<S_A>の切込み深さまで、切り込み<S_F>用の送り速度で移動します。

繰り返しの手順を示しています。

この手順は、合計切り込み深さ<S_K>に達するまで繰り返されます。最後のストロークは、不均等に配分されます。

注記

この手順はシングルブロックによって中断することはできません。

例

以下の仕様での揺動

- 開始位置での切り込み深さ 0.02 mm
- 終了位置での切り込み深さ 0.01 mm

4.26 サイクルを外部的にプログラミング

- 合計切り込み深さ 1 mm
- ストローク 100 mm
- 切り込みの送り速度 1 mm/min
- 径方向の送り速度 1000 mm/min
- スパークアウト時間 1 秒
- 標準ジオメトリ軸

プログラムコード

```

N10 T1 D1
N20 CYCLE4075(0.02,0.01,1,100,1,1000,1)
N30 M30

```

4.26.1.51 CYCLE4077 - 反転点での切り込みによる平面研削とキャンセル信号

構文

CYCLE4077(<S_GAUGE>, <S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>, <S_A2>)

パラメータ

No.	パラメータ	データタイプ	意味
1	<S_GAUGE>	STRING	切り込みのキャンセル条件 • 高速入力回数 • 論理式
2	<S_I>	REAL	開始位置での切り込み深さ
3	<S_J>	REAL	終了位置での切り込み深さ
4	<S_K>	REAL	合計切り込み深さ
5	<S_A>	REAL	研削幅
6	<S_R>	REAL	切り込みの送り速度
7	<S_F>	REAL	径方向の切り込みの送り速度
8	<S_P>	REAL	スパークアウト時間
9	<S_A1>	AXIS	切り込み軸(任意選択)
10	<S_A2>	AXIS	揺動軸(任意選択)

4.26 サイクルを外部的にプログラミング

機能

このサイクルは、切り込みステップでの切り込み全体の深さでの加工に使用されます。切り込み深さは開始位置と終了位置で変えることができます。切り込み間で接線方向の移動があります。高速入力のカンセル信号が 1 であるか、キャンセル条件が満たされた場合、深さ切り込みがキャンセルされます。キャンセル後、常に完全なストロークが 1 回実行されます。

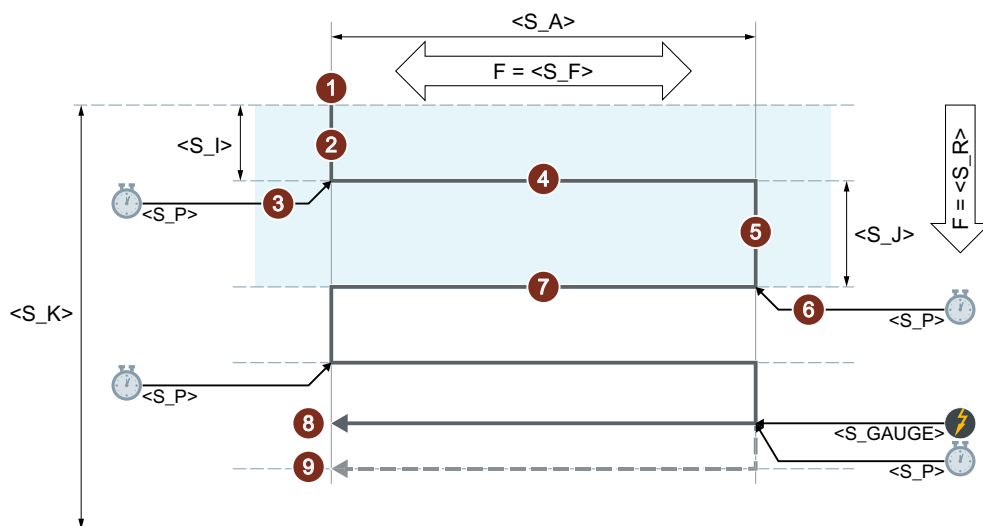
位置データ P2~P5 は負または正とすることができます。

切り込み軸または揺動軸の指定は任意です。一方または両方のパラメータを指定していない場合、サイクルはチャネルの最初の 2 つのジオメトリ軸を使用します。

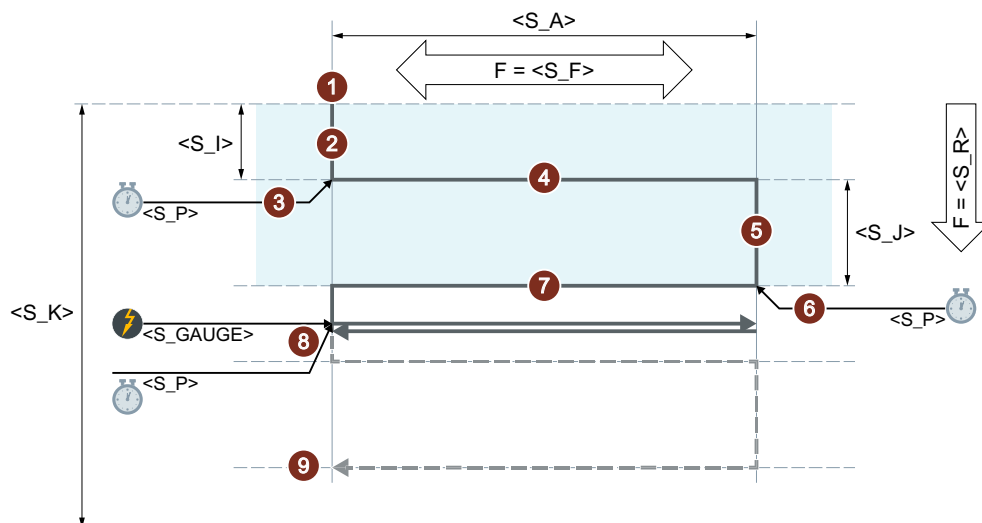
開始点と終了点の切り込み深さの合計が 0、または合計切り込み深さが 0 の場合、スパークアウトストロークのみが実行されます。

手順

終了位置での切り込みのキャンセル



開始位置での切り込みのキャンセル



- ① 揺動軸の現在の位置でサイクルを開始します。
 - ② 切り込み軸を開始点の切り込み深さ<S_I>まで、切り込み<S_R>用の送り速度で移動します。
 - ③ スパークアウト時間<S_P>のスパークアウト。
 - ④ 研削幅<S_A>の揺動軸を移動軌跡として、径方向の切り込み<S_F>用の送り速度で移動します。
 - ⑤ 切り込み軸を開始点<S_J>の切り込み深さまで、切り込み<S_R>用の送り速度で移動します。
 - ⑥ スパークアウト時間<S_P>のスパークアウト。
 - ⑦ 切り込み軸を開始点<S_A>の切り込み深さまで、切り込み<S_F>用の送り速度で移動します。
 - ⑧ キャンセル信号:次の開始点に達すると加工が停止します。
 - ⑨ キャンセル信号なし:この手順は、合計切り込み深さ<S_K>に達するまで繰り返されます。最後のストロークは、不均等に配分されます。
- 繰り返しの手順を示しています。

注記

この手順はシングルブロックによって中断することはできません。

4.26 サイクルを外部的にプログラミング

リソース

サイクルでは、リソースとして、ブロック全体でのシンクロナイズドアクションとシンクロナイズドアクション変数を使用します。シンクロナイズドアクションは、シンクロナイズドアクション範囲(CUS.DIR-1 ...、CMA.DIR-1000 ...、CST.DIR-1199 ...)の空き領域から自動的に決定されます。シンクロナイズドアクション変数として SYG_IS[1] が使用されます。

例

例 1

以下の仕様での揺動

- 開始位置での切り込み深さ 0.02 mm
- 終了位置での切り込み深さ 0.01 mm
- 合計切り込み深さ 1 mm
- ストローク 100 mm
- 切り込みの送り速度 1 mm/min
- 径方向の送り速度 1000 mm/min
- スパークアウト時間 1 秒
- 標準ジオメトリ軸

キャンセル信号:高速入力 1 (\$A_IN[1])

プログラムコード

```
N10 T1 D1
N20 CYCLE4077("1",0.02,0.01,1,100,1,1000,1)
N30 M30
```

例 2

以下の仕様での揺動

- 開始位置での切り込み深さ 0.02 mm
- 終了位置での切り込み深さ 0.01 mm
- 合計切り込み深さ 1 mm
- ストローク 100 mm
- 切り込みの送り速度 1 mm/min
- 径方向の送り速度 1000 mm/min

- スパークアウト時間 1 秒
- 標準ジオメトリ軸

キャンセル信号:デュアルポート RAM 変数 20 が 0.01 未満(\$A_DBR[20] < 0.01)

プログラムコード
N10 T1 D1
N20 CYCLE4077("(\$A_DBR[20]<0.01)",0.02,0.01,1,100,1,1000,1)
N30 M30

4.26.1.52 CYCLE4078 - 連続切り込みによる平面研削

構文

CYCLE4078(<S_I>, <S_J>, <S_K>, <S_A>, <S_F>, <S_P>, <S_A1>, <S_A2>)

パラメータ

No.	パラメータ	データタイプ	意味
1	<S_I>	REAL	開始位置から終了位置までの切り込み深さ
2	<S_J>	REAL	終了位置から開始位置までの切り込み深さ
3	<S_K>	REAL	合計切り込み深さ
4	<S_A>	REAL	研削幅
5	<S_F>	REAL	送り速度
6	<S_P>	REAL	スパークアウト時間
7	<S_A1>	AXIS	切り込み軸(任意選択)
8	<S_A2>	AXIS	揺動軸(任意選択)

機能

このサイクルは、連続切り込みを使用した切り込み全体の深さでの加工に使用されます。開始位置から終了位置までの切り込み深さと、終了位置から開始位置までの切り込み深さは変えることができます。

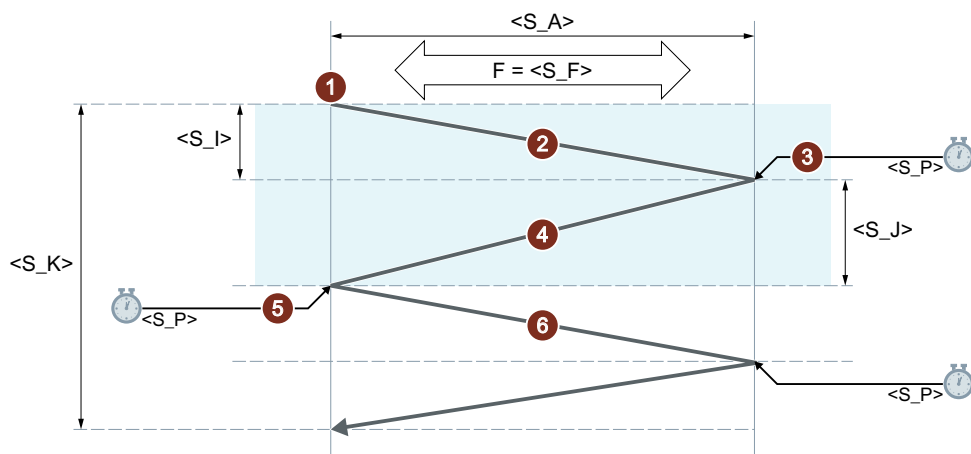
位置データ P1~P4 は負または正とすることができます。

切り込み軸または揺動軸の指定は任意です。一方または両方のパラメータを指定していない場合、サイクルはチャネルの最初の 2 つのジオメトリ軸を使用します。

4.26 サイクルを外部的にプログラミング

P1 と P2 の切り込み深さの合計が 0、または合計切り込み深さが 0 の場合、スパークアウトストロークのみが実行されます。

手順



- ① 切り込み深さ 0 の揺動軸の現在の位置でサイクルを開始します。
- ② 研削幅<S_A>の揺動軸を移動軌跡として、送り速度<S_F>で、切り込み深さを開始位置の切り込み深さ<S_I>まで連続して深くしながら移動します。
- ③ スパークアウト時間<S_P>のスパークアウト。
- ④ 研削幅<S_A>の揺動軸を移動軌跡として、送り速度<S_F>で、切り込み深さを終了点の切り込み深さ<S_J>まで連続して深くしながら、開始点まで移動します。
- ⑤ スパークアウト時間<S_P>のスパークアウト。
- ⑥ 研削幅<S_A>の揺動軸を移動軌跡として、送り速度<S_F>で開始点まで移動します。

繰り返しの手順を示しています。

この手順は、合計切り込み深さ<S_K>に達するまで繰り返されます。最後のストロークは、不均等に配分されます。

注記

この手順はシングルブロックによって中断することはできません。

例

以下の仕様での揺動

- 開始位置での切り込み深さ 20 mm
- 終了位置での切り込み深さ 10 mm
- 合計切り込み深さ 100 mm
- ストローク 100 mm
- 送り速度 1000 mm/min
- スパークアウト時間 1 秒
- 標準ジオメトリ軸

プログラムコード

```
N10 T1 D1
N20 CYCLE4078(20,10,100,100,1000,1)
N30 M30
```

4.26.1.53 CYCLE4079 - 間欠切り込みによる平面研削

構文

CYCLE4079(<S_I>, <S_J>, <S_K>, <S_A>, <S_R>, <S_F>, <S_P>, <S_A1>, <S_A2>)

パラメータ

No.	パラメータ	データタイプ	意味
1	<S_I>	REAL	開始位置での切り込み深さ
2	<S_J>	REAL	終了位置での切り込み深さ
3	<S_K>	REAL	合計切り込み深さ
4	<S_A>	REAL	研削幅
5	<S_R>	REAL	切り込みの送り速度
6	<S_F>	REAL	径方向の切り込みの送り速度
7	<S_P>	REAL	スパークアウト時間
8	<S_A1>	AXIS	切り込み軸(任意選択)
9	<S_A2>	AXIS	揺動軸(任意選択)

4.26 サイクルを外部的にプログラミング

機能

このサイクルは、切り込みステップでの切り込み全体の深さでの加工に使用されます。切り込み深さは開始位置と終了位置で変えることができます。切り込み間で接線方向の移動があります。

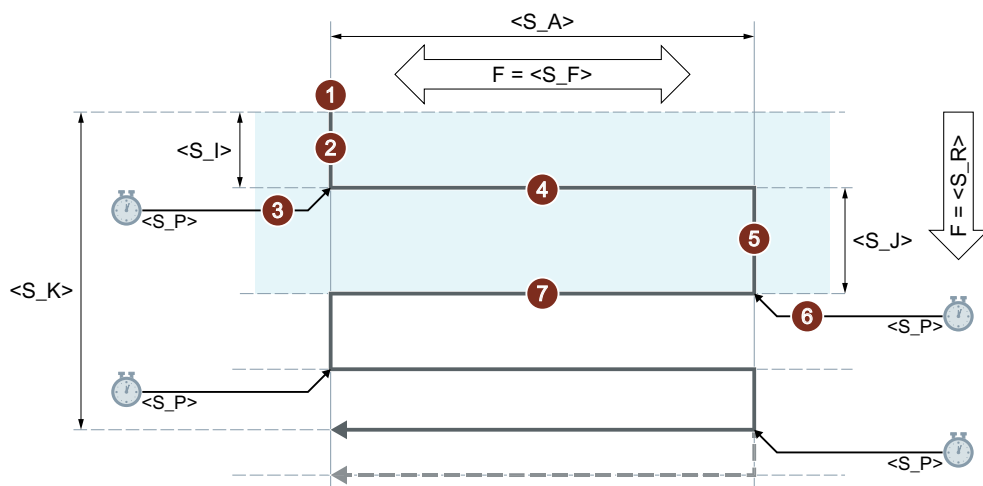
位置データ P1~P4 は負または正とすることができます。

切り込み軸または揺動軸の指定は任意です。一方または両方のパラメータを指定していない場合、サイクルはチャンネルの最初の 2 つのジオメトリ軸を使用します。

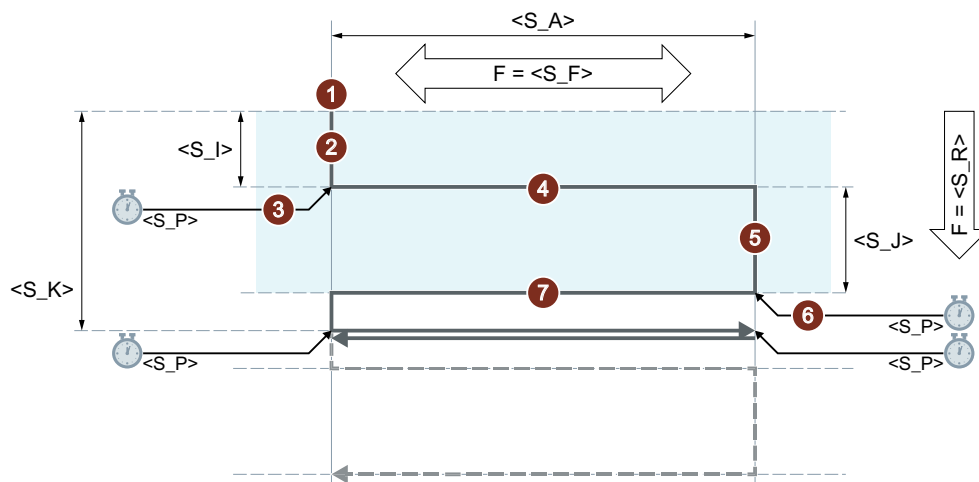
開始点と終了点の切り込み深さの合計が 0、または合計切り込み深さが 0 の場合、スパークアウトストロークのみが実行されます。

手順

2 番目の反転点での切り込みで合計切り込み深さに到達



1 番目の反転点での切り込みで合計切り込み深さに到達



- ① 揺動軸の現在の位置でサイクルを開始します。
- ② 切り込み軸を開始点の切り込み深さ<S_I>まで、切り込み<S_R>用の送り速度で移動します。
- ③ スパークアウト時間<S_P>のスパークアウト。
- ④ 研削幅<S_A>の揺動軸を移動軌跡として、径方向の切り込み<S_F>用の送り速度で移動します。
- ⑤ 切り込み軸を開始点<S_J>の切り込み深さまで、切り込み<S_R>用の送り速度で移動します。
- ⑥ スパークアウト時間<S_P>のスパークアウト。
- ⑦ 切り込み軸を開始点<S_A>の切り込み深さまで、切り込み<S_F>用の送り速度で移動します。

繰り返しの手順を示しています。

この手順は、合計切り込み深さ<S_K>に達するまで繰り返されます。最後のストロークは、不均等に配分されます。

注記

この手順はシングルブロックによって中断することはできません。

例

以下の仕様での揺動

- 開始位置での切り込み深さ 0.02 mm
- 終了位置での切り込み深さ 0.01 mm

4.26 サイクルを外部的にプログラミング

- 合計切り込み深さ 1 mm
- ストローク 100 mm
- 切り込みの送り速度 1 mm/min
- 径方向の送り速度 1000 mm/min
- スパークアウト時間 1 秒
- 標準ジオメトリ軸

プログラムコード
N10 T1 D1
N20 CYCLE4079(0.02,0.01,1,100,1,1000,1)
N30 M30

4.26.1.54 GROUP_BEGIN - プログラムブロック開始

構文

GROUP_BEGIN (<_LEVEL>, <_NAME>, <_SP>, <_MODE>, <S_ICON>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味	
1		<_LEVEL>	INT	レベル	
				0 =	メインレベル
				1 =	1 番目のサブレベル
2		<_NAME>	STRING[128]	ブロック名称	
3		<_SP>	INT	主軸	
				0 =	主軸なし
				1 =	主軸
				2 =	対向主軸

4.26 サイクルを外部的にプログラミング

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
4		<_MODE>	INT	モード
				ビット 0 = 1 GROUP_ADDEND が存在します
				ビット 1 = 1 ShopTurn:自動後退(工具交換位置に移動)
				ビット 12 予備
				ビット 13 予備
5		<S_ICON>	STRING[32]	アイコン名(操作画面専用)

4.26.1.55 GROUP_END - プログラムブロック終了

構文

GROUP_END (<_LEVEL>, <_SP>)

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1		<_LEVEL>	INT	レベル
				0 = メインレベル
				1 = 1 番目のサブレベル
2		<_SP>	INT	主軸
				0 = 主軸なし
				1 = 主軸
				2 = 対向主軸

4.26.1.56 GROUP_ADDEND - 試し切り追加終了

構文

GROUP_ADDEND (<_LEVEL>, <_SP>)

4.26 サイクルを外部的にプログラミング

パラメータ

No.	マスクパラメータ	内部パラメータ	データタイプ	意味
1		<_LEVEL>	INT	レベル
				0 = メインレベル
				1 = 1 番目のサブレベル
2		<_SP>	INT	主軸
				0 = 主軸なし
				1 = 主軸
				2 = 対向主軸

4.26.1.57 必要条件

サイクル画面の機能スケーリング

機能スケーリングを有効にすると、さまざまなサイクル画面で簡易入力を選択し、最も重要なサイクルパラメータのみを表示することができます。

たとえば、簡易入力は、次のサイクル画面で選択できます。

機能	サイクル画面
穴あけ	深穴ドリル
	タッピング
フライス加工	長方形ポケット
	輪郭フライス加工:ポケット
旋削	ねじの旋削:長手方向
	輪郭旋削:切削
	輪郭旋削:溝切り
	輪郭旋削:溝旋削

当該のサイクル画面の操作画面で、オプション[入力:簡易]と[入力:完全]を選択できます。

表示されないサイクルパラメータ

簡易入力画面で表示されないサイクルパラメータは、プリセットされた固定のパラメータで、技術的には有用ですが、可変の値がないものです。あるいは、チャンネル別サイクルセッティングデータによって、パラメータ設定された値がサイクルパラメータに割り当てら

れます。「セットアップ」の「チャンネル別サイクルセッティングデータ」の下の節を参照してください。

[入力:完全]から[入力:簡易]への切り替え

[入力:完全]の設定でサイクル画面に入力し、[入力:簡易]に切り替えた場合、サイクル呼び出しを生成したときに、表示されなくなったパラメータには初期設定またはセッティングデータの値が使用されます。

セットアップ

チャンネル別構成マシンデータ

サイクル画面の機能スケーリングは、マシンデータで有効にできます。

MD52210 \$MCS_FUNCTION_MASK_DISP, bit 9 = 1 (select display "Input simple")

チャンネル別サイクルセッティングデータ

サイクル画面で簡易入力を有効にしている場合、次のセッティングデータで特定のサイクルパラメータの値を指定できます。

番号	識別子	意味
SD5530 0	\$SCS_EASY_SAFETY_CLEARANCE	安全距離
SD5530 1	\$SCS_EASY_DWELL_TIME	ドウェル時間
SD5530 5	\$SCS_EASY_DRILL_DEEP_FD1	深穴ドリル:パーセンテージ:1 番目の送り速度
SD5530 6	\$SCS_EASY_DRILL_DEEP_DF	深穴ドリル:パーセンテージ:切り込み
SD5530 7	\$SCS_EASY_DRILL_DEEP_V1	深穴ドリル:最小切り込み深さ
SD5530 8	\$SCS_EASY_DRILL_DEEP_V2	深穴ドリル:後退距離
SD5530 9	\$SCS_EASY_THREAD_RETURN_DIST	ねじの旋削:戻り距離

4.26 サイクルを外部的にプログラミング

4.26.2 計測サイクル

計測サイクルは、当社が提供する特別なサブプログラムで、特定の計測作業を実行します。一般的なサイクルと同様に、計測サイクルは、パラメータ設定によって特定の問題に適応させることもできます。

計測サイクルは、次の領域および用途で使用できます。

- 工具計測、旋盤/フライス盤
- ワーク計測、旋盤/フライス盤

詳細情報

プログラミングマニュアル、計測サイクル

4.26.2.1 計測サイクルパラメーター一覧

CYCLE973 計測サイクルパラメータ

```
PROC CYCLE973 (INT S_MVAR, INT S_PRNUM, INT S_CALNUM, REAL S_SETV, INT S_MA, INT S_MD, REAL  
S_FA, REAL S_TSA, REAL S_VMS, INT S_NMSP, INT S_MCBIT, INT _DMODE, INT _AMODE)
```

表 4-8 CYCLE973 呼び出しパラメータ ¹⁾

番号	画面形式 パラメータ	サイクル パラメータ	意味
1		S_MVAR	計測タイプ(初期設定 = 0012103)
			規格値: UNITS(一の位):面、端面または溝での校正 0 = 既知の指令値による(WCS の)面/端面での長さ 1 = 既知の指令値による(WCS の)面での半径 2 = (WCS の)溝での長さ、「S_CALNUM」を参照してください 3 = (WCS の)溝での半径、「S_CALNUM」を参照してください
			TENS(十の位):予約済み 0 = 0
			HUNDREDS(百の位):予約済み 0 = 0
			THOUSANDS(千の位):校正の計測軸 と計測方向の選択 ²⁾ 0 = 指定なし(溝の底部での面の校正に対して、計測軸と計測方向の選択なし) ⁴⁾ 1 = 計測軸と計測方向の選択を指定、S_MA、S_MD を参照してください(計測軸の 1 つの計測方向) 2 = 計測軸の選択を指定、S_MA を参照してください(計測軸の 2 つの計測方向)
			TEN THOUSANDS(万の位):位置の偏りの特定(プローブ傾斜) ^{2), 3)} 0 = 位置の偏りの特定 1 = 位置の偏りを特定しない
			HUNDRED THOUSANDS(十万の位):予約済み 0 = 0
			ONE MILLION:工具長の補正 ⁷⁾ 0 = 工具長さを補正しない(トリガポイントのみ) 1 = 工具長さを補正する
2	アイコン+ 数値	S_PRNUM	プローブパラメータのフィールドの番号(プローブ番号ではありません) (初期設定 = 1)

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
3		S_CALNUM	溝での校正の校正溝の数値(初期設定 = 1) ⁵⁾	
4		S_SETV	面での校正の指令値	
5	X0	S_MA	計測軸(軸の数値) ⁶⁾ (初期設定 = 1)	
			規格 値:	1 = 平面の 1 番目の軸(G18 の場合、Z) 2 = 平面の 2 番目の軸(G18 の場合、X) 3 = 平面の 3 番目の軸(G18 の場合、Y) ⁶⁾
6	+-	S_MD	計測方向(初期設定 = 1)	
			規格 値:	0 = 正の計測方向 1 = 負の計測方向
7	DFA	S_FA	計測距離	
8	TSA	S_TSA	安全領域	
9	VMS	S_VMS	校正の可変の計測速度 ²⁾	
10	計測	S_NMSP	同じ位置での計測回数 ²⁾ (初期設定 = 1)	
11		S_MCBIT	予約済み	
12		_DMODE	表示モード	
			規格 値:	UNITS(一の位):加工平面 G17/G18/G19 0 = 互換性、サイクル呼び出しの前に有効な平面が有効 1 = G17 (サイクルのみで有効) 2 = G18 (サイクルのみで有効) 3 = G19 (サイクルのみで有効)
13		_AMODE	代替モード	

¹⁾ すべての初期値 = 0 または初期設定=x として指定

²⁾ 表示は一般 SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE で決定

³⁾ 軸の 2 方向での校正の場合のみ該当

⁴⁾ プローブの刃先位置(SL)から計測軸と計測方向のみが自動的に特定されます。SL=8 → -X、SL=7 → -Z

⁵⁾ 校正溝の番号(n)は次の一般セッティングデータを示します(MCS のすべての位置):

刃先位置 SL=7 の場合:

SD54615 \$SNS_MEA_CAL_EDGE_BASE_AX1[n] 平面の 1 番目の軸の溝底部の位置(G18 の場合、Z)

SD54621 \$SNS_MEA_CAL_EDGE_PLUS_DIR_AX2[n] 平面の 2 番目の軸の正方向の溝壁面の位置(G18 の場合、X)

SD54622 \$SNS_MEA_CAL_EDGE_MINUS_DIR_AX2[n] 平面の 2 番目の軸の負方向の溝壁面の位置:

刃先位置 SL=8 の場合:

SD54619 \$SNS_MEA_CAL_EDGE_BASE_AX2[n] 平面の 2 番目の軸の溝底部の位置

SD54620 \$SNS_MEA_CAL_EDGE_UPPER_AX2[n] 平面の 2 番目の軸の溝端面の上端の位置(プローブの事前位置
決め用のみ)

SD54617 \$SNS_MEA_CAL_EDGE_PLUS_DIR_AX1[n] 平面の 1 番目の軸の正方向の溝壁面の位置

SD54618 \$SNS_MEA_CAL_EDGE_MINUS_DIR_AX1[n] 平面の 1 番目の軸の負方向の溝壁面の位置

注:

溝壁面の位置の値+-は、ラフに特定できます。

溝壁面の位置の値の差から求められる溝幅は、正確に特定できることが必要です(精密ダイヤルゲージ)。

溝で校正する場合、校正する軸のプローブの工具長さ = 0 とします。

溝底部の位置の値も、機械で(外形寸法図ではなく)正確に特定してください。

6) 計測軸 S_MA=3 は、面の校正をおこなうための平面の 3 番目の実軸を備えた旋盤の軸です(G18 の場合、Y)。

7) 溝の長さ、または面の長さを校正する場合に、工具長を補正します。

旋盤のワークプローブは、2 つの長さ(X Z)を使用して定義できます。

旋削プローブ、タイプ 580

刃先位置 7:長さの校正の場合は、オプションで長さ Z が補正されます。

旋削プローブ、タイプ 580

刃先位置 8:長さの校正の場合は、オプションで長さ X が補正されます

工具長は、計測バージョン、溝の半径、または面の半径に合わせて補正されません。

対応するトリガ位置のみが保存されます。

CYCLE974 計測サイクルパラメータ

```
PROC CYCLE974 (INT S_MVAR, INT S_KNUM, INT S_KNUM1, INT S_PRNUM, REAL S_SETV, INT S_MA, REAL
S_FA, REAL S_TSA, REAL S_STA1, INT S_NMSP, STRING[32] S_TNAME, INT S_DLNUM, REAL S_TZL, REAL
S_TDIF, REAL S_TUL, REAL S_TLL, REAL S_TMV, INT S_K, INT S_EVNUM, INT S_MCBIT, INT _DMODE, INT
_AMODE, INT _DP)
```

4.26 サイクルを外部的にプログラミング

表 4-9 CYCLE974 呼び出しパラメータ ¹⁾

番号	画面形式 パラメータ	サイクル パラメータ	意味	
1		S_MVAR	計測タイプ	
			規格 値:	UNITS(一の位): 0 = 正面の計測 1 = 内径の計測 2 = 外径の計測
				TENS(十の位):予約済み
				HUNDREDS(百の位):補正対象 0 = 計測のみ(WO の補正なし、または工具オフセットなし) 1 = 計測、WO の特定と補正(S_KNUM を参照してください) ³⁾ 2 = 計測と工具オフセット(S_KNUM1 を参照してください)
				THOUSANDS(千の位):予約済み
				TEN THOUSANDS(万の位):主軸(旋削主軸)の反転あり/なしの計測 0 = 反転なしの計測 1 = 反転ありの計測

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
2	選択	S_KNUM	ゼロオフセット(WO)、基本 WO、または基本レファレンスでの補正 ²⁾	
			規格 値:	UNITS(一の位):
				TENS(十の位): 0 = 補正なし ゼロオフセットの 1～最大 99 の数値または 基本オフセットの 1～最大 16 の数値
				HUNDREDS(百の位):予約済み
				THOUSANDS(千の位):WO または基本 WO または基本レファレンスの補正 0 = 設定可能 WO の補正 1 = チャネル別基本 WO の補正 2 = 基本レファレンスの補正 3 = グローバル基本 WO の補正 9 = 有効な WO、または G500 の場合は最後に有効だったチャネル別基本 WO の補正
				TEN THOUSANDS(万の位):WO、基本 WO または基本レファレンスの汎用/精密補正 0 = 精密補正 ⁶⁾ 1 = 汎用補正

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
3	選択	S_KNUM1	工具オフセットの補正 ²⁾ 、 ⁴⁾	
			規格 値:	UNITS(一の位):
				TENS(十の位):
				HUNDREDS(百の位): 0 = 補正なし 工具オフセットの 1～最大 999 個の D 番号(刃先番号)。 サムオフセットとセットアップオフセットについては、S_DLNUM も参照してください
				THOUSANDS(千の位):0 または一義的な D 番号
				TEN THOUSANDS(万の位):0 または一義的な D 番号
				MD の一義的な D 番号がセットアップされている場合、1～最大 32000
				HUNDRED THOUSANDS(十万の位):工具オフセット ²⁾ 0 = 指定なし(工具形状のオフセット) 1 = 長さ L1 のオフセット 2 = 長さ L2 のオフセット 3 = 長さ L3 のオフセット 4 = 半径のオフセット
				ONE MILLION(百万の位):工具オフセット ²⁾ 0 = 指定なし(工具長さ摩耗のオフセット) 1 = 工具オフセット、サムオフセット(AO) ⁵⁾ 工具オフセット値を既存の AO に加算 2 = 工具オフセット、セットアップオフセット(SO) ⁵⁾ SO (新) = SO (旧) + AO (旧)オフセット値、AO (新) = 0 3 = 工具オフセット、セットアップオフセット(SO) ⁵⁾ 工具オフセット値を既存の SO に加算 4 = 工具オフセット、形状
				TEN MILLION(千万の位):工具オフセット ²⁾ 0 = 指定なし(通常工具形状のオフセット、(反転なし)) 1 = 反転オフセット
				HUNDRED MILLIONS(一億の位):工具オフセット 0 = 予備工具なしの工具オフセット 1 = 予備工具での工具オフセット(_DP)

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
4	アイコン + 数値	S_PRNUM	プローブパラメータのフィールドの番号(プローブ番号ではありません) (初期設定 = 1)	
5	X0	S_SETV	指令値	
6	X	S_MA	計測軸(軸の数値) (初期設定 = 1)	
			規格 値:	1 = 平面の 1 番目の軸(G18 の場合、Z) 2 = 平面の 2 番目の軸(G18 の場合、X) 3 = 平面の 3 番目の軸(G18 の場合、Y) ⁵⁾
7	DFA	S_FA	計測距離	
8	TSA	S_TSA	安全領域	
9	α	S_STA1	反転ありの計測の開始角度	
10	計測	S_NMSP	同じ位置での計測回数 ²⁾ (初期設定 = 1)	
11	T	S_TNAME	工具名称 ²⁾	
12	DL	S_DLNUM	セットアップ サムオフセット DL 番号 ⁵⁾	
13	ST	_DP	修正する予備工具の番号(予備工具番号)	
14	TZL	S_TZL	ゼロオフセット ^{2)、4)}	
15	DIF	S_TDIF	寸法差チェック ^{2)、4)}	
16	TUL	S_TUL	許容上限値(指令値へのインクレメンタル方式) ⁴⁾	
17	TLL	S_TLL	許容下限値(指令値へのインクレメンタル方式) ⁴⁾	
18	TMV	S_TMV	平均化のオフセット範囲 ²⁾	
19	FW	S_K	平均化の加重係数 ²⁾	
20	EVN	S_EVNUM	経験平均値メモリ番号 ^{2)、7)}	
21		S_MCBIT	予約済み	
22		_DMODE	表示モード	
			規格 値:	UNITS(一の位):加工平面 G17/G18/G19 0 = 互換性、サイクル呼び出しの前に有効な平面が有効 1 = G17 (サイクルのみで有効) 2 = G18 (サイクルのみで有効) 3 = G19 (サイクルのみで有効)

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
23		_AMODE	代替モード	
			規格 値:	UNITS(一の位):寸法許容誤差 あり/なし 0 = なし 1 = あり

- 1) すべての初期値 = 0 または初期設定=x として指定
- 2) 表示は一般 SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE で決定
- 3) WO の補正は反転なしの計測のみで可能
- 4) チャンネル別 MD 20360 TOOL_PARAMETER_DEF_MASK の工具オフセットでは、ビット 0 とビット 1 を参照
- 5) 「セットアップサムオフセット」機能が一般 MD 18108 \$MN_MM_NUM_SUMCORR で設定されている場合のみ
また、一般 MD 18080 \$MN_MM_TOOL_MANAGEMENT_MASK で、ビット 8 を 1 にセットしてください。
- 6) MD で WO 「精密」が設定されていない場合、補正は WO 「汎用」に基づいておこなわれます。
- 7) 経験平均化は、工具オフセットのみで可能
経験平均値メモリの数値の範囲::
経験値メモリの 1~20 の数値(n)。チャンネル別の SD55623 \$SCS_MEA_EMPIRIC_VALUE[n-1]を参照してください
平均値メモリの 10000~200000 の数値(n)。チャンネル別の SD55625 \$SCS_MEA_AVERAGE_VALUE[n-1]を参照してください

CYCLE994 計測サイクルパラメータ

```
PROC CYCLE994 (INT S_MVAR, INT S_KNUM, INT S_KNUM1, INT S_PRNUM, REAL S_SETV, INT S_MA, REAL
S_SZA, REAL S_SZO, REAL S_FA, REAL S_TSA, INT S_NMSP, STRING[32] S_TNAME, INT S_DLNUM, REAL
S_TZL, REAL S_TDIF, REAL S_TUL, REAL S_TLL, REAL S_TMV, INT S_K, INT S_EVNUM, INT S_MCBIT, INT
_DMODE, INT _AMODE, INT _DP)
```

4.26 サイクルを外部的にプログラミング

表 4-10 CYCLE994 呼び出しパラメータ ¹⁾

番号	画面形式 パラメータ	サイクル パラメータ	意味	
1		S_MVAR	計測タイプ	
			規格 値:	UNITS(一の位):内径/外径の計測(初期設定 = 1) 1 = 内径の計測 2 = 外径の計測
				TENS(十の位):予約済み
				HUNDREDS(百の位):補正対象 0 = 計測のみ(WO の補正なし、または工具オフセットなし) 1 = WO の 計測、特定と補正(S_KNUM を参照してください) ³⁾ 2 = 計測と工具オフセット(S_KNUM1 を参照してください)
				THOUSANDS(千の位):バイパス領域 0 = バイパス領域なし 1 = バイパス軸、平面の 1 番目の軸(G18 の場合、Z)。計測軸、S_MA を参照してください。 2 = バイパス軸、平面の 2 番目の軸(G18 の場合、X)。計測軸、S_MA を参照してください。 3 = バイパス軸、平面の 3 番目の軸(G18 の場合、Y)。計測軸、S_MA を参照してください。 ⁸⁾

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味
2	選択	S_KNUM	ゼロオフセット(WO)、基本 WO、または基本レファレンスの補正 ²⁾
			規格
			値:
			UNITS(一の位):
			TENS(十の位): 0 = 補正なし ゼロオフセットの 1～最大 99 の数値または 基本オフセットの 1～最大 16 の数値
			HUNDREDS(百の位):予約済み
			THOUSANDS(千の位):WO または基本 WO または基本レファレンスの補正 0 = 設定可能 ZO の補正 1 = チャネル別基本 ZO の補正 2 = 基本レファレンスの補正 3 = グローバル基本 ZO の補正 9 = 有効な ZO、または G500 の場合は最後に有効だったチャネル別基本 ZO の補正
			TEN THOUSANDS(万の位):WO、基本 WO または基本レファレンスの汎用/精密補正 0 = 精密補正 ⁶⁾ 1 = 汎用補正

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
3	選択	S_KNUM1	工具オフセットの補正 ²⁾ 、 ⁴⁾	
			規格 値:	UNITS(一の位):
				TENS(十の位):
				HUNDREDS(百の位): 0 = 補正なし 工具オフセットの 1～最大 999 個の D 番号(刃先番号)。 サムオフセットとセットアップオフセットについては、S_DLNUM も参照してください。
				THOUSANDS(千の位):0 または一義的な D 番号
				TEN THOUSANDS(万の位):0 または一義的な D 番号 MD の一義的な D 番号がセットアップされている場合、1～最大 32000
				HUNDRED THOUSANDS(十万の位):工具オフセット ²⁾ 0 = 指定なし(工具形状のオフセット) 1 = 長さ L1 のオフセット 2 = 長さ L2 のオフセット 3 = 長さ L3 のオフセット 4 = 半径のオフセット
				ONE MILLION(百万の位):工具オフセット ²⁾ 0 = 指定なし(工具長さ摩耗のオフセット) 1 = 工具オフセット、サムオフセット(AO) ⁵⁾ 工具オフセット値を既存の AO に加算 2 = 工具オフセット、セットアップオフセット(SO) ⁵⁾ SO (新) = SO (旧) + AO (旧)オフセット値、AO (新) = 0 3 = 工具オフセット、セットアップオフセット(SO) ⁵⁾ 工具オフセット値を既存の SO に加算 4 = 工具オフセット、形状
				TEN MILLION(千万の位):工具オフセット ²⁾ 0 = 指定なし(通常工具形状のオフセット、反転なし) 1 = 反転オフセット
				HUNDRED MILLIONS(一億の位):工具オフセット 0 = 予備工具なしの工具オフセット 1 = 予備工具での工具オフセット(_DP)

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
4	アイコン+ 数値	S_PRNUM	プローブパラメータのフィールドの番号(プローブ番号ではありません) (初期設定 = 1)	
5	X0	S_SETV	指令値	
6	X	S_MA	計測軸の数値(初期設定 = 1))	
			規格 値:	1 = 平面の 1 番目の軸(G18 の場合、Z) 2 = 平面の 2 番目の軸(G18 の場合、X) 3 = 平面の 3 番目の軸(G18 の場合、Y) ⁸⁾
7	X1	S_SZA	計測された軸でのバイパス距離	
8	Y1	S_SZO	バイパス軸でのバイパス距離	
9	DFA	S_FA	計測距離	
10	TSA	S_TSA	安全領域	
11	計測	S_NMSP	同じ位置での計測回数 ²⁾ (初期設定 = 1)	
12	T	S_TNAME	工具名称 ²⁾	
13	DL	S_DLNUM	セットアップ サムオフセット DL 番号 ⁵⁾	
14	ST	_DP	修正する予備工具の番号(予備工具番号)	
15	TZL	S_TZL	ゼロオフセット ²⁾ , ⁴⁾	
16	DIF	S_TDIF	寸法差チェック ²⁾ , ⁴⁾	
17	TUL	S_TUL	許容上限値(指令値へのインクレメンタル方式) ⁴⁾	
18	TLL	S_TLL	許容下限値(指令値へのインクレメンタル方式) ⁴⁾	
19	TMV	S_TMV	平均化のオフセット範囲 ²⁾	
20	FW	S_K	平均化の加重係数 ²⁾	
21	EVN	S_EVNUM	経験値メモリ番号 ²⁾ , ⁷⁾	
22		S_MCBIT	予約済み	
23		_DMODE	表示モード	
			規格 値:	UNITS(一の位):加工平面 G17/G18/G19 0 = 互換性、サイクル呼び出しの前に有効な平面が有効 1 = G17 (サイクルのみで有効) 2 = G18 (サイクルのみで有効) 3 = G19 (サイクルのみで有効)

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
24		_AMODE	代替モード	
			規格 値:	UNITS(一の位):寸法許容誤差 あり/なし 0 = なし 1 = あり

- 1) すべての初期値 = 0 または初期設定=x として指定
- 2) 表示は一般 SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE で決定
- 3) WO の補正は反転なしの計測のみで可能
- 4) 工具オフセットの場合、チャネル MD20360 TOOL_PARAMETER_DEF_MASK を参照
- 5) 「セットアップ サムオフセット」機能が一般 MD 18108 \$MN_MM_NUM_SUMCORR で設定されている場合のみ
また、一般 MD 18080 \$MN_MM_TOOL_MANAGEMENT_MASK で、ビット 8 を 1 にセットしてください。
- 6) MD で WO 「精密」が設定されていない場合、補正は WO 「汎用」に基づいておこなわれます。
- 7) 経験平均化は、工具オフセットのみで可能
経験平均値メモリの数値の範囲::
経験値メモリの 1~20 の数値(n)。チャネル別の SD55623 \$SCS_MEA_EMPIRIC_VALUE[n-1]を参照してください
平均値メモリの 10000~200000 の数値(n)。チャネル別の SD55625 \$SCS_MEA_AVERAGE_VALUE[n-1]を参照
してください
- 8) 機械に Y 軸がある場合

CYCLE976 計測サイクルパラメータ

```
PROC CYCLE976(INT S_MVAR,INT S_PRNUM,REAL S_SETV,REAL S_SETV0,INT S_MA,INT S_MD,REAL
S_FA,REAL S_TSA,REAL S_VMS,REAL S_STAL,INT S_NMSP,INT S_SETV1,INT _DMODE,INT _AMODE)
```

4.26 サイクルを外部的にプログラミング

表 4-11 CYCLE976 呼び出しパラメータ ¹⁾

番号	画面形式 パラメータ	サイクル パラメータ	意味
1		S_MVAR	計測タイプ(初期設定 = 1000)
			規格 値:
			UNITS(一の位):面、キャリブレーションボールまたはキャリブレーションリングでの校正 ²⁾ 0 = 既知の指令値の面上の長さ 1 = 既知の直径(指令値)と既知の中心点をもつキャリブレーションリングの半径。 2 = 直径が既知(指令値)で、中心点が不明なキャリブレーションリングの半径 3 = キャリブレーションボールの半径と長さ 4 = 既知の指令値の端面の半径。計測軸と計測方向の選択には注意してください。 ³⁾ 5 = 既知の指令値と端面のクリアランスをもつ2つの端面間の半径。計測軸を選択してください。
			TENS(十の位):予約済み 0 = 0
			HUNDREDS(百の位):予約済み 0 = 0
			THOUSANDS(千の位):校正時の計測軸と計測方向の選択 0 = 指定なし(計測軸と計測方向の選択は不要です) ⁸⁾ 1 = 計測軸と計測方向の選択を指定、「S_MA」を参照してください、S_MD (計測軸の1つの計測方向) 2 = 計測軸の選択を指定、「S_MA」を参照してください(計測軸での2つの計測方向)
			TEN THOUSANDS(万の位):位置の偏り(プローブ傾斜)の特定 ²⁾ 0 = プローブの位置の偏りを特定する ⁶⁾ 1 = 位置の偏りを特定しない
			HUNDRED THOUSANDS(十万の位):軸と平行の校正または角度での校正 0 = 有効な WCS での軸と平行な校正 1 = 角度での校正 ⁷⁾
			ONE MILLION(百万の位):面またはボールでの校正中の工具長さの特定

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
				0 = 工具の長さが特定されていません 1 = 工具の長さが特定されています ⁴⁾ 2 = 切り込み軸がボールで校正されています、工具の長さが特定されています、工具の長さの計測された差異が校正データに入力されています
2	アイコン + 数値	S_PRNUM	プローブパラメータのフィールドの番号(プローブ番号ではありません) (初期設定 = 1)	
3		S_SETV	指令値	
4	Z0	S_SETV0	ボールキャリブレーションの縦方向の基準の指令値	
5	X/Y/Z	S_MA	計測軸(軸の数値) ^{2), 6)} (初期設定 = 1)	
			規格値:	1 = 平面の 1 番目の軸(G17 の場合、X) 2 = 平面の 2 番目の軸(G17 の場合、Y) 3 = 平面の 3 番目の軸(G17 の場合、Z)
6	+-	S_MD	計測方向 ^{2), 6)}	
			規格値:	0 = 正 1 = 負
7	DFA	S_FA	計測距離	
8	TSA	S_TSA	安全領域	
9	VMS	S_VMS	校正の可変の計測速度 ²⁾	
10	α	S_STA1	開始角度 ^{2), 5)}	
11	計測	S_NMSP	同じ位置での計測回数 ²⁾ (初期設定 = 1)	
12	X0	S_SETV1	2 つの端面間の校正時の端面の基準点 ³⁾	
13		_DMODE	表示モード	
			規格値:	UNITS(一の位):加工平面 G17/G18/G19 0 = 互換性、サイクル呼び出しの前に有効な平面が有効 1 = G17 (サイクルのみで有効) 2 = G18 (サイクルのみで有効) 3 = G19 (サイクルのみで有効)
14		_AMODE	代替モード	

1) すべての初期値 = 0 または初期設定=x として指定

2) 表示は一般 SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE で決定

4.26 サイクルを外部的にプログラミング

- 3) "キャリブレーションリングでの半径"校正では、リングの直径と中心点が既知であることが必要です(4つの計測方向)。
 "2つの端面での半径"校正では、計測軸方向の端面への距離が既知であることが必要です(2つの計測方向)。
 "1つの端面での半径"校正では、面の指令値が既知であることが必要です。
- 4) 面での校正(面での長さ)のみの計測タイプでは、補正された工具長さは S_MD と S_MA から得られます。
- 5) 「キャリブレーションリング、...および中心点が既知」(S_MVAR=1xxx02)の場合のみ。
- 6) 下記の計測タイプのみの計測軸 S_MVAR=0 または=xx1x01 または=xx2x01 または=20000
 計測タイプ:「面での校正」→計測軸と計測方向の選択
 または「キャリブレーションリング、...および既知の中心点」→軸方向の選択と計測軸と計測方向の選択
 または「キャリブレーションリング、...および既知の中心点」→2つの軸方向の選択と計測軸の選択
 または「プローブ長さの特定」→S_MA=3 →平面の3番目の軸(G17の場合、Z)
- 7) 計測タイプ、キャリブレーションリングまたはキャリブレーションボールによる校正のみ
 「キャリブレーションボールでの校正」の場合、角度での計測の場合、軸はボールの赤道のまわりを回転します。
- 8) 未知の中心点での「キャリブレーションリングでの半径」の校正の場合、平面での4つの計測方向(G17の場合、+X、+Y)です。
 「面での長さ」校正の場合、工具軸の負方向(G17の場合、-Z)です。

CYCLE978 計測サイクルパラメータ

```
PROC CYCLE978 (INT S_MVAR, INT S_KNUM, INT S_KNUM1, INT S_PRNUM, REAL S_SETV, REAL S_FA, REAL
S_TSA, INT S_MA, INT S_MD, INT S_NMSP, STRING[32] S_TNAME, INT S_DLNUM, REAL S_TZL, REAL
S_TDIF, REAL S_TUL, REAL S_TLL, REAL S_TMV, INT S_K, INT S_EVNUM, INT S_MCBIT, INT _DMODE, INT
_AMODE, INT _DP)
```

4.26 サイクルを外部的にプログラミング

表 4-12 CYCLE978 呼び出しパラメータ ¹⁾

番号	画面形式パラメータ	サイクルパラメータ	意味	
1		S_MVAR	計測タイプ	
			規格 値:	UNITS(一の位):輪郭要素 0 = 面を計測
				TENS(十の位):予約済み
				HUNDREDS(百の位):補正対象 0 = 計測のみ(ZO の補正なし、または工具オフセットなし) 1 = 計測、ZO の特定と補正(S_KNUM を参照してください) 2 = 計測と工具オフセット(S_KNUM1 を参照してください)
				THOUSANDS(千の位):予約済み
				TEN THOUSANDS(万の位):主軸反転あり/なし、または、スイッチング方向へのプローブの割り出しあり/なしの計測 ⁹⁾ 0 = 主軸反転およびプローブ割り出しなしの計測 1 = 主軸反転ありの計測 2 = スwitchング方向へのプローブの割り出し

4.26 サイクルを外部的にプログラミング

番号	画面形式パラメータ	サイクルパラメータ	意味
2	選択	S_KNUM	ゼロオフセット(WO)、基本 WO、または基本レファレンスの補正 ²⁾
			規格
			値:
			UNITS(一の位):
			TENS(十の位):
			0 = 補正なしゼロオフセットの 1～最大 99 の数値または基本オフセットの 1～最大 16 の数値
			HUNDREDS(百の位):予約済み
			THOUSANDS(千の位):WO または基本 WO または基本レファレンスの補正
			0 = 設定可能 ZO の補正
			1 = チャネル別基本 ZO の補正
			2 = 基本レファレンスの補正
			3 = グローバル基本 ZO の補正
			9 = 有効な ZO、または G500 の場合は最後に有効だったチャネル別基本 ZO の補正
			TEN THOUSANDS(万の位):WO、基本 WO または基本レファレンスの汎用/精密補正
			0 = 精密補正 ⁶⁾
			1 = 汎用補正

4.26 サイクルを外部的にプログラミング

番号	画面形式パラメータ	サイクルパラメータ	意味
3	選択	S_KNUM1	工具オフセットの補正 ²⁾
			規格値:
			UNITS(一の位):
			TENS(十の位):
			HUNDREDS(百の位): 0 = 補正なし 工具オフセットの 1～最大 999 個の D 番号(刃先番号)。 サムオフセットとセットアップオフセットについては、S_DLNUM も参照してください
			THOUSANDS(千の位):0 または一義的な D 番号
			TEN THOUSANDS(万の位):0 または一義的な D 番号 MD の一義的な D 番号がセットアップされている場合、1～最大 32000
			HUNDRED THOUSANDS(十万の位):工具オフセット ²⁾ 0 = 指定なし(工具形状のオフセット) 1 = 長さ L1 のオフセット 2 = 長さ L2 のオフセット 3 = 長さ L3 のオフセット 4 = 半径のオフセット
			ONE MILLION(百万の位):工具オフセット ²⁾ 0 = 指定なし(工具半径摩耗のオフセット) 1 = 工具オフセット、サムオフセット(AO) ⁵⁾ 工具オフセット値を既存の AO に加算 2 = 工具オフセット、セットアップオフセット(SO) ⁵⁾ SO (新) = SO (旧) + AO (旧)オフセット値、AO (新) = 0 3 = 工具オフセット、セットアップオフセット(SO) ⁵⁾ 工具オフセット値を既存の SO に加算 4 = 工具オフセット、形状
			TEN MILLION(千万の位):工具オフセット ²⁾ 0 = 指定なし(通常工具形状のオフセット、反転なし) 1 = 反転オフセット
			HUNDRED MILLIONS(一億の位):工具オフセット 0 = 予備工具なしの工具オフセット 1 = 予備工具での工具オフセット(_DP)

4.26 サイクルを外部的にプログラミング

番号	画面形式パラメータ	サイクルパラメータ	意味	
4	アイコン +数値	S_PRNUM	プローブパラメータのフィールドの番号(プローブ番号ではありません) (値の範囲 1~40)	
5	X0	S_SETV	指令値	
6	DFA	S_FA	計測距離	
7	TSA	S_TSA	安全領域	
8	X	S_MA	計測軸の数値 ⁷⁾ (数値の範囲 1~3)	
			規格 値:	1 = 平面の 1 番目の軸(G17 の場合、X) 2 = 平面の 2 番目の軸(G17 の場合、Y) 3 = 平面の 3 番目の軸(G17 の場合、Z) 工具方向の計測
9		S_MD	計測軸の計測方向	
			規格 値:	1 = 正の計測方向 2 = 負の計測方向
10	計測	S_NMSP	同じ位置での計測回数 ²⁾ (数値の範囲 1~9)	
11	TR	S_TNAME	工具名称 ³⁾	
12	DL	S_DLNUM	セットアップ サムオフセット DL 番号 ⁵⁾	
13	ST	_DP	修正する予備工具の番号(予備工具番号)	
14	TZL	S_TZL	ゼロオフセット ^{2)、3)}	
15	DIF	S_TDIF	寸法差チェック ^{2)、3)}	
16	TUL	S_TUL	許容上限値(指令値へのインクレメンタル方式) ³⁾	
17	TLL	S_TLL	許容下限値(指令値へのインクレメンタル方式) ³⁾	
18	TMV	S_TMV	平均化のオフセット範囲 ²⁾	
19	FW	S_K	平均化の加重係数 ²⁾	
20	EVN	S_EVNUM	データセット、経験値メモリ ^{2)、8)}	
21		S_MCBIT	予約済み	
22		_DMODE	表示モード	
			規格 値:	UNITS(一の位):加工平面 G17/G18/G19 0 = 互換性、サイクル呼び出しの前に有効な平面が有効 1 = G17 (サイクルのみで有効) 2 = G18 (サイクルのみで有効) 3 = G19 (サイクルのみで有効)

4.26 サイクルを外部的にプログラミング

番号	画面形式パラメータ	サイクルパラメータ	意味	
23		_AMODE	代替モード	
			規格値:	UNITS(一の位):寸法許容誤差 あり/なし 0 = なし 1 = あり

- 1) すべての初期値 = 0 または a～b の値のデータ範囲として指定
- 2) 表示は一般 SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE で決定
- 3) 工具のオフセットのみ、それ以外はパラメータ = ""
- 4) 工具のオフセットと寸法許容誤差「あり」の場合のみ、それ以外はパラメータ = 0
- 5) 「セットアップサムオフセット」機能が一般 MD 18108 \$MN_MM_NUM_SUMCORR で設定されている場合のみ
また、一般 MD 18080 \$MN_MM_TOOL_MANAGEMENT_MASK で、ビット 8 を 1 にセットしてください。
- 6) MD で WO 「精密」が設定されていない場合、補正は WO 「汎用」に基づいておこなわれます。
- 7) 工具形状のオフセット:
平面での計測の場合(S_MA=1 または S_MA=2) 工具半径のオフセット
工具方向での計測の場合(S_MA=3) 工具長さ L1 のオフセット
- 8) 工具オフセットと ZO の補正で経験平均化が可能
経験平均値メモリの数値の範囲:
経験値メモリの 1～20 の数値(n)。チャネル別 SD55623 \$SCS_MEA_EMPIRIC_VALUE[n-1]
を参照してください。平均値メモリの 10000～200000 の数値(n)。チャネル別 SD55625
\$SCS_MEA_AVERAGE_VALUE[n-1]を参照してください
- 9) 主軸反転ありで計測する場合、プローブの半径/直径を精密に特定してください。これは、リング、端面、またはボールでの CYCLE976 の半径の校正タイプで行います。それ以外は、計測結果が不正になります。

CYCLE998 計測サイクルパラメータ

```
PROC CYCLE998(INT S_MVAR,INT S_KNUM,INT S_RA,INT S_PRNUM,REAL S_SETV,REAL S_STA1,REAL
S_INCA,REAL S_FA,REAL S_TSA,INT S_MA,INT S_MD,REAL S_ID,REAL S_SETV0,REAL S_SETV1,REAL
S_SETV2,REAL S_SETV3,INT S_NMSP,INT S_EVNUM,INT _DMODE,INT _AMODE)
```

4.26 サイクルを外部的にプログラミング

表 4-13 CYCLE998 呼び出しパラメータ ¹⁾

番号	画面形式パラメータ	サイクルパラメータ	意味	
1		S_MVAR	計測タイプ(初期設定 = 5)	
			規格値:	UNITS(一の位):輪郭要素 5 = 端面を計測(1 つの角度) 6 = 平面を計測(2 つの角度)
				TENS(十の位):予約済み
				HUNDREDS(百の位):補正対象 0 = 計測のみで WO の補正なし 1 = 計測、WO の特定と補正(S_KNUM を参照してください)
				THOUSANDS(千の位):プロテクションゾーン 0 = プロテクションゾーンを考慮しない 1 = プロテクションゾーンを考慮する
				TEN THOUSANDS(万の位):主軸反転ありの計測(差分計測) 0 = 主軸反転なしの計測 1 = 反転ありの計測
				HUNDRED THOUSANDS(十万の位):角度でのまたは軸と平行の計測 0 = 角度での計測 1 = 軸と平行の計測

4.26 サイクルを外部的にプログラミング

番号	画面形式パラメータ	サイクルパラメータ	意味	
2	選択	S_KNUM	ゼロオフセット(WO)、基本 WO、または基本レファレンスの補正 ²⁾	
			規格値:	UNITS(一の位):
				TENS(十の位): 0 = 補正なし ゼロオフセットの 1～最大 99 の数値または 基本オフセットの 1～最大 16 の数値
				HUNDREDS(百の位):予約済み
				THOUSANDS(千の位):WO または基本 WO または基本レファレンスの補正 0 = 設定可能 WO の補正 1 = チャンネル別基本 WO の補正 2 = 基本レファレンスの補正 9 = 有効な WO または G500 の場合は最後に有効だったチャンネル別基本 WO の補正
				TEN THOUSANDS(万の位):ZO、基本 ZO または基本レファレンスの汎用/精密補正 ³⁾ 0 = 精密補正 1 = 汎用補正
3	A、B、C	S_RA	座標回転または回転軸のオフセット対象 ¹⁴⁾	
			規格値:	0 = パラメータ S_MA ⁴⁾ から得られる軸まわりの座標回転の補正対象>0 = 回転軸の補正対象回転軸(できれば回転テーブル)のチャンネル軸番号の数値。角度オフセットは、回転軸の WO の平行移動部分で行います。
4	アイコン + 数値	S_PRNUM	プローブパラメータのフィールドの数値 (初期設定 = 1)	
5	DX/DY/DZ	S_SETV	計測軸の開始位置から計測点 P1 までの距離(インクレメンタル値)(S_MA) ⁵⁾	
6	α	S_STA1	平面の 1 番目の軸(G17 の場合、X)まわりの「端面の割り出し」または「平面の割り出し」の角度指令値 ⁹⁾	
7	β	S_INCA	平面の 2 番目の軸(G17 の場合、Y)まわりの「平面の割り出し」の角度指令値 ⁹⁾	
8	DFA	S_FA	計測距離	
9	TSA	S_TSA	安全領域 角度指令値に対する角度差の監視[°] ⁶⁾	

4.26 サイクルを外部的にプログラミング

番号	画面形式パラメータ	サイクルパラメータ	意味	
10	X/Y/Z	S_MA	計測軸、オフセット軸 ⁷⁾ (初期設定 = 201)	
			規格値:	UNITS(一の位):計測軸の数値 1 = 平面の 1 番目の軸(G17 の場合、X) 2 = 平面の 2 番目の軸(G17 の場合、Y) 3 = 平面の 3 番目の軸(G17 の場合、Z)
				TENS(十の位):予約済み
				HUNDREDS(百の位):オフセット軸の数値 1 = 平面の 1 番目の軸(G17 の場合、X) 2 = 平面の 2 番目の軸(G17 の場合、Y) 3 = 平面の 3 番目の軸(G17 の場合、Z)
11	+-	S_MD	計測軸の計測方向 ⁸⁾	
			規格値:	0 = 計測方向を計測軸の指令値と実位置から特定(互換性) 1 = 正の計測方向 2 = 負の計測方向
12	L2	S_ID	計測タイプ「端面の割り出し」の場合: オフセット軸の計測点 P1 と P2 の距離(インクレメンタル値)(値>0) 計測タイプ「平面の割り出し」の場合、次のパラメータが適用されます。	
13	L2	S_SETV 0	平面の 1 番目の軸の計測点 P1 と P2 の距離 ¹⁰⁾	
14		S_SETV 1	平面の 2 番目の軸の計測点 P1 と P2 の距離 ^{11)、12)}	
15	L3x	S_SETV 2	平面の 1 番目の軸の計測点 P1 と P3 の距離 ¹¹⁾	
16	L3y	S_SETV 3	平面の 2 番目の軸の計測点 P1 と P3 の距離 ¹⁰⁾	
17	計測	S_NMSP	同じ位置での計測回数 ²⁾ (初期設定 = 1)	
18		S_EVNUM	データセット、経験値メモリ ^{2)、13)}	

4.26 サイクルを外部的にプログラミング

番号	画面形式パラメータ	サイクルパラメータ	意味
19		_DMODE	表示モード
			規格値: UNITS(一の位):加工平面 G17/G18/G19 0 = 互換性、サイクル呼び出しの前に有効な平面が有効 1 = G17 (サイクルのみで有効) 2 = G18 (サイクルのみで有効) 3 = G19 (サイクルのみで有効)
20		_AMODE	予約済み(切替モード)

- 1) すべての初期値 = 0 または初期設定=x として指定
- 2) 表示は一般 SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE で決定
- 3) 補正対象が回転軸で MD 52207 \$MCS_AXIS_USAGE_ATTRIB[n] ビット 6=1 の場合のみ ZO 「精密」。
MD で ZO がセットアップされていない場合、補正は ZO 「汎用」に基づいて行われます。
- 4) 座標回転のオフセットの例:S_MA=102 計測軸 Y、オフセット軸 X で(G17 の場合)Z まわりの座標回転が得られます。
- 5) プロテクションゾーン「あり」の場合のみ有効な値です(S_MVAR THOUSANDS の位 = 1)
- 6) オフセット軸で計測点 P1 から計測点 P2 に位置決めする場合、パラメータ S_STA1 と S_TSA の角度が加算されます。
- 7) 計測軸の数値がオフセット軸の数値と一致しないようにしてください(例: 101 は許可されません)
- 8) 「端面の割り出し」と「軸と平行の計測」のみの計測方向です(S_MVAR=10x105)
- 9) 角度範囲 S_STA1 「端面の割り出し」で±45°
角度範囲 S_STA1 0~+60°および S_INCA 「平面の割り出し」で±30°
- 10) 計測タイプ「平面の割り出し」と「端面の割り出し」の場合です
- 11) 計測タイプ「平面の計測」と「軸と平行に計測」の場合です
- 12) 計測サイクルタイプ SW04.04 では不可です
- 13) WO の補正のための経験値生成; 経験平均値メモリの値の範囲:
経験値メモリの 1~20 の数値(n)、以下のチャンネル別を参照してください。SD55623
\$SCS_MEA_EMPIRIC_VALUE[n-1]
- 14) パラメータ S_RA は、計測タイプ「端面の割り出し」(S_MVAR xxxxx5)にのみ関係します。

CYCLE977 計測サイクルパラメータ

```
PROC CYCLE977 (INT S_MVAR, INT S_KNUM, INT S_KNUM1, INT S_PRNUM, REAL S_SETV, REAL S_SETV0, REAL
S_SETV1, REAL S_FA, REAL S_TSA, REAL S_STA1, REAL S_ID, REAL S_SZA, REAL S_SZO, INT S_MA, INT
S_NMSP, STRING[32] S_TNAME, INT S_DLNUM, REAL S_TZL, REAL S_TDIF, REAL S_TUL, REAL S_TLL, REAL
S_TMV, INT S_K, INT S_EVNUM, INT S_MCBIT, INT _DMODE, INT _AMODE, REAL S_XM, REAL S_YM, INT _DP)
```

表 4-14 CYCLE977 呼び出しパラメータ ¹⁾

番号	画面形式 パラメータ	サイクル パラメータ	意味
1		S_MVAR	計測タイプ
			規格 値:
			UNITS(一の位):加工形状(数値の範囲 1~6) 1 = 穴の計測 2 = スピゴットの計測(シャフト) 3 = 溝の計測 4 = リブの計測 5 = 長方形の計測、内側 6 = 長方形の計測、外側
			TENS(十の位):予約済み
			HUNDREDS(百の位):補正対象 0 = 計測のみ(WO の補正なし、または工具オフセットなし) 1 = 計測と、WO の特定と補正(S_KNUM を参照してください) 2 = 計測と工具オフセット(S_KNUM1 を参照してください)
			THOUSANDS(千の位):プロテクションゾーン 0 = プロテクションゾーンを考慮しない 1 = プロテクションゾーンを考慮する
			TEN THOUSANDS(万の位):主軸反転(差分計測)あり/なし、または、スイッチング方向へのプローブの割り出しあり/なしの計測 0 = 主軸反転なしの計測、プローブの割り出しをおこなってはいけません 1 = 主軸反転ありの計測 2 = スwitching方向へのプローブの割り出し

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
2	選択	S_KNUM	ゼロオフセット(WO)、基本 WO、または基本レファレンスの補正 ²⁾	
			規格 値:	UNITS(一の位):
				TENS(十の位): 0 = 補正なし ゼロオフセットの 1～最大 99 の数値または 基本オフセットの 1～最大 16 の数値
				HUNDREDS(百の位):予約済み
				THOUSANDS(千の位):WO または基本 WO または基本レファレンスの補正 0 = 設定可能 ZO の補正 1 = チャネル別基本 ZO の補正 2 = 基本レファレンスの補正 3 = グローバル基本 ZO の補正 9 = 有効な ZO、または G500 の場合は最後に有効だったチャネル別基本 ZO の補正
				TEN THOUSANDS(万の位):WO、基本 WO または基本レファレンスの汎用/精密補正 0 = 精密補正 ⁶⁾ 1 = 汎用補正

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
3	選択	S_KNUM1	工具オフセットの補正 ²⁾	
			規格 値:	UNITS(一の位):
				TENS(十の位):
				HUNDREDS(百の位): 0 = 補正なし 工具オフセットの 1～最大 999 個の D 番号(刃先番号)。サムオフセットとセットアップオフセットについては、S_DLNUM も参照してください
				THOUSANDS(千の位):0 または一義的な D 番号
				TEN THOUSANDS(万の位):0 または一義的な D 番号 MD の一義的な D 番号がセットアップされている場合、1～最大 32000
				HUNDRED THOUSANDS(十万の位):工具オフセット ²⁾ 0 = 指定なし(工具半径のオフセット) 1 = 長さ L1 のオフセット 2 = 長さ L2 のオフセット 3 = 長さ L3 のオフセット 4 = 半径のオフセット
				ONE MILLION(百万の位):工具オフセット ²⁾ 0 = 指定なし(工具半径摩耗のオフセット) 1 = 工具オフセット、サムオフセット(AO) ⁵⁾ 工具オフセット値を既存の AO に加算 2 = 工具オフセット、セットアップオフセット(SO) ⁵⁾ SO (新) = SO (旧) + AO (旧)オフセット値、AO (新) = 0 3 = 工具オフセット、セットアップオフセット(SO) ⁵⁾ 工具オフセット値を既存の SO に加算 4 = 工具オフセット、形状
				TEN MILLION(千万の位):工具オフセット ²⁾ 0 = 指定なし(通常の工具形状のオフセット、反転なし) 1 = 反転オフセット
				HUNDRED MILLIONS(一億の位):工具オフセット 0 = 予備工具なしの工具オフセット 1 = 予備工具での工具オフセット(_DP)

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味
4	アイコン+ 数値	S_PRNUM	プローブパラメータのフィールドの番号(プローブ番号ではありません) (値の範囲 1~40)
5	X0	S_SETV	指令値
6	X0	S_SETV0	平面の 1 番目の軸(G17 の場合、X)の長方形の指令値
7	Y0	S_SETV1	平面の 2 番目の軸(G17 の場合、Y)の長方形の指令値
8	XM	S_XM	セットポイントを中心点としてジオメトリ軸 X を入力
9	YM	S_YM	セットポイントを中心点としてジオメトリ軸 Y を入力
10	DFA	S_FA	計測距離
11	TSA	S_TSA	安全領域
12	α 0	S_STA1	開始角度
13	DZ	S_ID	インクレメンタル量 1. 平面の 3 番目の軸(G17 の場合、Z)のインクレメンタル送り S_ID の符号による送り方向。スピゴット、リブおよび長方形の外側の計測 の場合、計測高さへの下降の設定に S_ID を使用します。 2. プロテクションゾーンの考慮 穴、溝および長方形の内側の計測とプロテクションゾーンの場合、S_ID を 使用してオーバートラベルの高さを設定します。
14	X1	S_SZA	プロテクションゾーンの直径または長さ(幅) ⁷⁾
15	Y1	S_SZO	「長方形の計測」の場合:平面の 2 番目の軸のプロテクションゾーンの幅
16	X	S_MA	計測軸の数値 ⁷⁾ (溝またはリブの計測の場合のみ) 規格 値: 1 = 平面の 1 番目の軸(G17 の場合、X) 2 = 平面の 2 番目の軸(G17 の場合、Y)
17	ST	_DP	修正する予備工具の番号(予備工具番号)
18	計測	S_NMSP	同じ位置での計測回数 ²⁾ (数値の範囲 1~9)
19	TR	S_TNAME	工具名称 ²⁾
20	DL	S_DLNUM	セットアップ サムオフセット DL 番号 ⁵⁾
21	TZL	S_TZL	ゼロオフセット ^{2)、4)}
22	DIF	S_TDIF	寸法差チェック ^{2)、4)}
23	TUL	S_TUL	許容上限値(指令値へのインクレメンタル方式) ⁴⁾
24	TLL	S_TLL	許容下限値(指令値へのインクレメンタル方式) ⁴⁾
25	TMV	S_TMV	平均化のオフセット範囲 ²⁾
26	FW	S_K	平均化の加重係数 ²⁾

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
27		S_EVNUM	データセット、経験平均値メモリ ²⁾ 、 ⁸⁾	
28		S_MCBIT	予約済み	
29		_DMODE	表示モード	
			規格 値:	UNITS(一の位):加工平面 G17/G18/G19 0 = 互換性、サイクル呼び出しの前に有効な平面が有効 1 = G17 (サイクルのみで有効) 2 = G18 (サイクルのみで有効) 3 = G19 (サイクルのみで有効)
30		_AMODE	代替モード	
			規格 値:	UNITS(一の位):寸法許容誤差 あり/なし 0 = なし 1 = あり

- 1) すべての初期値 = 0 または a~b の値のデータ範囲として指定
- 2) 表示は一般 SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE で決定
- 3) 工具のオフセットのみ、それ以外はパラメータ = ""
- 4) 工具のオフセットと寸法許容誤差「あり」の場合のみ、それ以外はパラメータ = 0
- 5) 「セットアップサムオフセット」機能が一般 MD 18108 \$MN_MM_NUM_SUMCORR で設定されている場合のみ
また、一般 MD 18080 \$MN_MM_TOOL_MANAGEMENT_MASK で、ビット 8 を 1 にセットしてください。
- 6) MD で WO 「精密」が設定されていない場合、補正は WO 「汎用」に基づいておこなわれます。
- 7) 穴または溝の内側のプロテクションゾーンの直径または幅
スピゴットまたはリブの外側のプロテクションゾーンの直径または幅
- 8) 経験平均化は、工具オフセットで可能
経験平均値メモリの数値の範囲:
経験値メモリの 1~20 の数値(n)。チャンネル別 SD55623 \$SCS_MEA_EMPIRIC_VALUE[n-1]を参照してください
平均値メモリの 10000~200000 の数値(n)。チャンネル別 SD55625 \$SCS_MEA_AVERAGE_VALUE[n-1]を参照してください。

CYCLE961 計測サイクルパラメータ

```
PROC CYCLE961 (INT S_MVAR, INT S_KNUM, INT S_PRNUM, REAL S_SETV0, REAL S_SETV1, REAL
S_SETV2, REAL S_SETV3, REAL S_SETV4, REAL S_SETV5, REAL S_SETV6, REAL S_SETV7, REAL S_SETV8, REAL
S_SETV9, REAL S_STA1, REAL S_INCA, REAL S_ID, REAL S_FA, REAL S_TSA, INT S_NMSP, INT S_MCBIT, INT
_DMODE, INT _AMODE)
```

4.26 サイクルを外部的にプログラミング

表 4-15 CYCLE961 呼び出しパラメータ¹⁾

番号	画面形式 パラメータ	サイクル パラメータ	意味
1		S_MVAR	計測タイプ(初期設定 ≥ 6)
			規格 値: UNITS(一の位):輪郭要素 5 = 内側のコーナを直角に設定、角度と距離 A1～A3 の指令値指定 6 = 外側のコーナを直角に設定、角度と距離 A1～A3 の指令値指定 7 = 内側のコーナの設定、角度と距離 A1～A4 の指定 8 = 外側のコーナの設定、角度と距離 A1～A4 の指定
			TENS(十の位):距離として、または 4 点を使った指令値指定 0 = 距離としての指令値指定(極座標) 1 = 4 点を使った指令値指定(計測点 P1～P4)
			HUNDREDS(百の位):補正対象 0 = 計測のみ(WO の補正なしまたは工具オフセットなし) 1 = 計測と、WO の特定と補正、S_KNUM を参照してください
			THOUSANDS(千の位):プロテクションゾーン 0 = プロテクションゾーン(障害物)を考慮しない 1 = プロテクションゾーン(障害物)を考慮する。S_ID を参照してください
			TEN THOUSANDS(万の位):WCS でのコーナの位置 0 = コーナの位置をパラメータ S_STA1 で特定(互換性) 1 = 位置決めされた計測開始点のコーナの位置 ¹⁾ 2 = コーナの位置 2、平面の 1 番目の軸(G17 の場合、X)の距離は負 (S_SETV0、S_SETV1 を参照してください) 3 = コーナの位置 3、平面の 1 番目と 2 番目の軸(G17 の場合、XY)の距離は負 (S_SETV0 ～S_SETV3 を参照してください) 4 = コーナの位置 4、平面の 2 番目の軸(G17 の場合、Y)の距離は負 (S_SETV2、S_SETV3 を参照してください)

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味
2	選択	S_KNUM	ゼロオフセット(WO)、基本 WO、または基本レファレンスの補正 ²⁾
			規格 値:
			UNITS(一の位):
			TENS(十の位): 0 = 補正なし ゼロオフセットの 1～最大 99 の数値または 基本オフセットの 1～最大 16 の数値
			HUNDREDS(百の位):予約済み
3	アイコン+ 数値	S_PRNUM	THOUSANDS(千の位):WO または基本 WO または基本レファレンスの補正 0 = 設定可能 WO の補正 1 = チャンネル別基本 WO の補正 2 = 基本レファレンスの補正 9 = 有効な WO または G500 の場合は最後に有効だったチャンネル別基本 WO の補正
			TEN THOUSANDS(万の位):WO、基本 WO または基本レファレンスの汎用/精密補正 0 = 精密補正 ⁵⁾ 1 = 汎用補正
4	L1/X1	S_SETV0	平面の 1 番目の軸(G17 の場合、X)の方向での極と計測点 P1 の距離 L1 ³⁾ (実際の距離 L1=0 の場合、 $L1 = M_SETV1 / 2$ を自動的に計算)または 平面の 1 番目の軸(G17 の場合、X)での起点 P1x ⁴⁾
5	L2/Y1	S_SETV1	平面の 1 番目の軸の方向での極と計測点 P2 の距離 L2 ³⁾ または平面の 2 番目の軸(G17 の場合、Y)の起点 P1y ⁴⁾
6	L3/X2	S_SETV2	平面の 2 番目の軸の方向での極と計測点 P3 の距離 L3 ³⁾ (距離 L3=0 の場合、直角でないコーナに対して $L3 = M_SETV3 / 2$ を自動的に計算) または平面の 1 番目の軸の起点 P2x ⁴⁾
7	L4/Y2	S_SETV3	直角でないコーナのある平面の 2 番目の軸の方向での極と計測点 P3 の距離 L4 ³⁾ または平面の 2 番目の軸の起点 P2y ⁴⁾
8	XP/X3	S_SETV4	平面の 1 番目の軸の極の位置 ³⁾ または平面の 1 番目の軸の起点 P3x ⁴⁾

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味
9	XP/Y3	S_SETV5	平面の 2 番目の軸の極の位置 ³⁾ または平面の 2 番目の軸の起点 P3y ⁴⁾
10	X4	S_SETV6	平面の 1 番目の軸の起点 P4x ⁴⁾
11	Y4	S_SETV7	平面の 2 番目の軸の起点 P4y ⁴⁾
12	X0	S_SETV8	WO の補正のための平面の 1 番目の軸の計測コーナの指令値
13	Y0	S_SETV9	WO の補正のための平面の 2 番目の軸の計測コーナの指令値
14	α0	S_STA1	平面の 1 番目の軸の正方向からワークの基準端面に対する MCS での開始角度 (+270°)
15	α1	S_INCA	直角でないコーナを計測する場合のワークの基準端面間の角度 ⁷⁾
16	DZ	S_ID	有効なプロテクションゾーンの各計測点の計測高さでの送り量(S_MVAR を参照 してください)。
17	DFA	S_FA	計測距離
18	TSA	S_TSA	安全領域 角度指令値に対する角度差の監視[°]
19	計測	S_NMSP	同じ位置での計測回数 ²⁾ (数値の範囲 1~9)
20		S_MCBIT	予約済み
21		_DMODE	表示モード 規格 値: UNITS(一の位):加工平面 G17/G18/G19 0 = 互換性、サイクル呼び出しの前に有効な平面が有効 1 = G17 (サイクルのみで有効) 2 = G18 (サイクルのみで有効) 3 = G19 (サイクルのみで有効)
22		_AMODE	代替モード

1) すべての初期値 = 0 または a~b の値のデータ範囲として指定

2) 表示は一般 SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE で決定

3) 極座標での計測点の入力、分割角度 S_INCA の計測点 3 または 4 の開始角度 S_STA1 を考慮。

4) 直角座標系での計測点の入力(4 点を使った入力)

5) MD で WO 「精密」が設定されていない場合、補正は WO 「汎用」に基づいておこなわれます。

6) 角度の範囲 S_INCA: -180°~+180°

7) 開始角度 S_STA1、数値の範囲:直角のコーナ: +- 90°、すべてのコーナ: +- 45°

CYCLE979 計測サイクルパラメータ

```
PROC CYCLE979 (INT S_MVAR, INT S_KNUM, INT S_KNUM1, INT S_PRNUM, REAL S_SETV, REAL S_FA, REAL
S_TSA, REAL S_CPA, REAL S_CPO, REAL S_STA1, REAL S_INCA, INT S_NMSP, STRING[32] S_TNAME, REAL
S_DLNUM, REAL S_TZL, REAL S_TDIF, REAL S_TUL, REAL S_TLL, REAL S_TMV, INT S_K, INT S_EVNUM, INT
S_MCBIT, INT _DMODE, INT _AMODE, REAL S_XM, REAL S_YM, INT _DP)
```

表 4-16 CYCLE979 呼び出しパラメータ ⁰⁾

番号	画面形式 パラメータ	サイクル パラメータ	意味
1		S_MVAR	計測タイプ
			規格 値:
			UNITS(一の位):輪郭要素 1 = 穴を計測 2 = スピゴットを計測(シャフト)
			TENS(十の位):予約済み
			HUNDREDS(百の位):補正対象 0 = 計測のみ(WO の補正なし、または工具オフセットなし) 1 = 計測と、WO の特定と補正(S_KNUM を参照してください) 2 = 計測と工具オフセット(S_KNUM1 を参照してください)
			THOUSANDS(千の位):計測点数 0 = 3 つの計測点 1 = 4 つの計測点
			TEN THOUSANDS(万の位):主軸反転(差分計測)あり/なし、または、スイッチング方向へのプローブの割り出しあり/なしの計測 0 = 主軸反転およびプローブ割り出しなしの計測 1 = 主軸反転ありの計測 2 = スwitchング方向へのプローブの割り出し

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
2	選択	S_KNUM	ゼロオフセット(WO)、基本 WO、または基本レファレンスの補正 ²⁾	
			規格 値:	UNITS(一の位):
				TENS(十の位):0 = 補正なし ゼロオフセットの 1～最大 99 の数値または 基本オフセットの 1～最大 16 の数値
				HUNDREDS(百の位):予約済み
				THOUSANDS(千の位):WO または基本 WO または基本レファレンスの補正 0 = 設定可能 ZO の補正 1 = チャネル別基本 ZO の補正 2 = 基本レファレンスの補正 3 = グローバル基本 ZO の補正 9 = 有効な ZO、または G500 の場合は最後に有効だったチャネル別基本 ZO の補正
				TEN THOUSANDS(万の位):WO、基本 WO または基本レファレンスの汎用/精密補正 0 = 精密補正 ⁶⁾ 1 = 汎用補正

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
3	選択	S_KNUM1	工具オフセットの補正 ²⁾	
			規格 値:	UNITS(一の位):
				TENS(十の位):
				HUNDREDS(百の位): 0 = 補正なし 工具オフセットの 1～最大 999 個の D 番号(刃先番号)。サムオフセットとセットアップオフセットについては、S_DLNUM も参照してください
				THOUSANDS(千の位):0 または一義的な D 番号
				TEN THOUSANDS(万の位):0 または一義的な D 番号 MD の一義的な D 番号がセットアップされている場合、1～最大 32000
				HUNDRED THOUSANDS(十万の位):工具オフセット ²⁾ 0 = 指定なし(工具半径のオフセット) 1 = 長さ L1 のオフセット 2 = 長さ L2 のオフセット 3 = 長さ L3 のオフセット 4 = 半径のオフセット
				ONE MILLION(百万の位):工具オフセット ²⁾ 0 = 指定なし(工具半径摩耗のオフセット) 1 = 工具オフセット、サムオフセット(AO) ⁵⁾ 工具オフセット値を既存の AO に加算 2 = 工具オフセット、セットアップオフセット(SO) ⁵⁾ SO (新) = SO (旧) + AO (旧)オフセット値、AO (新) = 0 3 = 工具オフセット、セットアップオフセット(SO) ⁵⁾ 工具オフセット値を既存の SO に加算 4 = 工具オフセット、形状
				TEN MILLION(千万の位):工具オフセット ²⁾ 0 = 指定なし(通常の工具形状のオフセット、反転なし) 1 = 反転オフセット
				HUNDRED MILLIONS(一億の位):工具オフセット 0 = 予備工具なしの工具オフセット 1 = 予備工具での工具オフセット(_DP)

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
4	アイコン+ 数値	S_PRNUM	プローブパラメータのフィールドの番号(プローブ番号ではありません) (値の範囲 1~40)	
5	X0	S_SETV	指令値	
6	DFA	S_FA	計測距離	
7	TSA	S_TSA	安全領域	
8	X0	S_CPA	平面の 1 番目の軸(G17 の場合、X)の中心点	
9	Y0	S_CPO	平面の 2 番目の軸(G17 の場合、Y)の中心点	
10	XM	S_XM	セットポイントを中心点としてジオメトリ軸 X を入力	
11	YM	S_YM	セットポイントを中心点としてジオメトリ軸 Y を入力	
12	α0	S_STA1	開始角度 ⁷⁾	
13	α1	S_INCA	分割角度 ⁸⁾	
14	計測	S_NMSP	同じ位置での計測回数 ¹⁾ (数値の範囲 1~9)	
15	ST	_DP	修正する予備工具の番号(予備工具番号)	
16	T	S_TNAME	工具名称 ²⁾	
17	DL	S_DLNUM	セットアップ サムオフセット DL 番号 ^{1)、4)}	
18	TZL	S_TZL	ゼロオフセット ^{1)、2)}	
19	DIF	S_TDIF	寸法差チェック ^{1)、2)}	
20	TUL	S_TUL	許容上限値(指令値へのインクレメンタル方式) ²⁾	
21	TLL	S_TLL	許容下限値(指令値へのインクレメンタル方式) ²⁾	
22	TMV	S_TMV	平均化のオフセット範囲 ¹⁾	
23	FW	S_K	平均化の加重係数 ¹⁾	
24		S_EVNUM	データセット、経験値メモリ ^{1)、6)}	
25		S_MCBIT	予約済み	
26		_DMODE	表示モード	
			規格 値:	UNITS(一の位):加工平面 G17/G18/G19 0 = 互換性、サイクル呼び出しの前に有効な平面が有効 1 = G17 (サイクルのみで有効) 2 = G18 (サイクルのみで有効) 3 = G19 (サイクルのみで有効)

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
27		_AMODE	代替モード	
			規格 値:	UNITS(一の位):寸法許容誤差 あり/なし 0 = なし 1 = あり

- 0) すべての初期値 = 0 または a~b の値のデータ範囲として指定。
- 1) 表示は一般 SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE で決定
- 2) 工具のオフセットのみ、それ以外はパラメータ = ""
- 3) 工具のオフセットと寸法許容誤差「あり」の場合のみ、それ以外はパラメータ = 0
- 4) 「セットアップ サムオフセット」機能が一般 MD 18108 \$MN_MM_NUM_SUMCORR で設定されている場合のみ
- 5) MD で WO 「精密」が設定されていない場合、補正は WO 「汎用」に基づいておこなわれます。
- 6) 経験平均化は、工具オフセットのみで可能
経験平均値メモリの数値の範囲:
経験値メモリの 1~20 の数値(n)。チャンネル別 SD55623 \$SCS_MEA_EMPIRIC_VALUE[n-1]
を参照してください平均値メモリの 10000~200000 の数値(n)。チャンネル別 SD55625
\$SCS_MEA_AVERAGE_VALUE[n-1]を参照してください
- 7) 開始角度の数値の範囲-360°~+360°
- 8) 4 つの計測点の場合、0°分割角度 ≤90°の数値の範囲、または 3 点計測の場合、0°分割角度 ≤120°。

CYCLE997 計測サイクルパラメータ

```
PROC CYCLE997 (INT S_MVAR,INT S_KNUM,INT S_PRNUM,REAL S_SETV,REAL S_FA,REAL S_TSA,REAL
S_STA1,REAL S_INCA,REAL S_SETV0,REAL S_SETV1,REAL S_SETV2,REAL S_SETV3,REAL S_SETV4,REAL
S_SETV5,REAL S_SETV6,REAL S_SETV7,REAL S_SETV8,REAL S_TNVL,INT S_NMSP,INT S_MCBIT,INT
_DMODE,INT _AMODE)
```

4.26 サイクルを外部的にプログラミング

表 4-17 CYCLE997 呼び出しパラメータ 1)、2)

番号	画面形式 パラメータ	サイクル パラメータ	意味	
1		S_MVAR	計測タイプ(初期設定 = 9)	
			規格 値:	UNITS(一の位):輪郭要素 9 = ボールの計測
				TENS(十の位):計測の繰り返し 0 = 計測の反復なし 1 = 計測の反復あり
				HUNDREDS(百の位):補正対象 0 = 計測のみ(WO の補正なし) 1 = 計測と、WO の特定と補正(S_KNUM を参照してください)
				THOUSANDS(千の位):計測方法 0 = 軸と平行に計測、開始角度なし、SD55740、ビット 1 に従ったプローブ調整 1 = 円計測、開始角度あり、SD55740、ビット 1 に従ったプローブ調整 2 = 円計測、開始角度あり、切り替え方向でプローブを調整 3 = 軸と平行に計測、開始角度あり、SD55740、ビット 1 に従ったプローブ調整 4 = 軸と平行に計測、開始角度あり、切り替え方向でプローブを調整
				TEN THOUSANDS(万の位):計測するボールの数 0 = 1 つのボールを計測 1 = 3 つのボールを計測
				HUNDRED THOUSANDS(十万の位):計測点の数、角度での計測の場合のみ(注: 計測方法:THOUSANDS の位 > 0) 0 = 角度での計測で 3 つの計測点(ボールのまわりを移動) 1 = 角度での計測で 4 つの計測点(ボールのまわりを移動)
				ONE MILLION(百万の位):ボールの直径指令値の特定 0 = ボールの直径指令値を特定しない 1 = ボールの直径指令値を特定する

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味
2	選択	S_KNUM	ゼロオフセット(WO)または基本 WO または基本レファレンスで補正 ³⁾
			規格 値:
			UNITS(一の位):
			TENS(十の位): 0 = 補正なし ゼロオフセットの 1～最大 99 の数値または 基本オフセットの 1～最大 16 の数値
			HUNDREDS(百の位):予約済み
			THOUSANDS(千の位):WO または基本 WO または基本レファレンスの補正 0 = 設定可能 WO の補正 1 = チャネル別基本 WO の補正 2 = 基本レファレンスの補正 3 = グローバル基本 WO の補正 ⁷⁾ 9 = 有効な WO、または G500 の場合は最後に有効だったチャネル別基本 WO の補正
3	アイコン+ 数値	S_PRNUM	TEN THOUSANDS(万の位):WO、基本 WO、または基本レファレンスの汎用/精密補正 0 = 精密補正 ⁶⁾ 1 = 汎用補正
3	アイコン+ 数値	S_PRNUM	プローブパラメータのフィールドの番号(プローブ番号ではありません) (値の範囲 1～40)
4		S_SETV	ボールの直径 ⁴⁾
5	DFA	S_FA	計測距離
6	TSA	S_TSA	安全領域
7	α0	S_STA1	角度での計測の開始角度
8	α1	S_INCA	角度での計測の分割角度
9	X1	S_SETV0	3 つのボールの計測での平面の 1 番目の軸(G17 の場合、X)の 1 番目のボールの位置指令
10	Y1	S_SETV1	3 つのボールの計測での平面の 2 番目の軸(G17 の場合、Y)の 1 番目のボールの位置指令
11	Z1	S_SETV2	3 つのボールの計測での平面の 3 番目の軸(G17 の場合、Z)の 1 番目のボールの位置指令

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味
12	X2	S_SETV3	3つのボールの計測での平面の1番目の軸の2番目のボールの位置指令
13	Y2	S_SETV4	3つのボールの計測での平面の2番目の軸の2番目のボールの位置指令
14	Z2	S_SETV5	3つのボールの計測での平面の3番目の軸の2番目のボールの位置指令
15	X3	S_SETV6	3つのボールの計測での平面の1番目の軸の3番目のボールの位置指令
16	Y3	S_SETV7	3つのボールの計測での平面の2番目の軸の3番目のボールの位置指令
17	Z3	S_SETV8	3つのボールの計測での平面の3番目の軸の3番目のボールの位置指令
18	TVL	S_TNVL	3つのボールの計測での三角形の歪みの制限値(偏りの合計) ⁵⁾
19	計測	S_NMSP	同じ位置での計測回数 ²⁾ (数値の範囲 1~9)
20		S_MCBIT	予約済み
21		_DMODE	表示モード
			規格 値: UNITS(一の位):加工平面 G17/G18/G19 0 = 互換性、サイクル呼び出しの前に有効な平面が有効 1 = G17 (サイクルのみで有効) 2 = G18 (サイクルのみで有効) 3 = G19 (サイクルのみで有効)
22		_AMODE	代替モード

1) すべての初期値 = 0 または a~b の値のデータ範囲として指定

2) 表示は一般 SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE で決定

3) 中間位置決め、球体の赤道のまわりを回転

4) 3つのボールの計測:3つすべてのボールに同じ直径の指令を適用(_SETV)

5) S_TNVL=1.2

の初期値 WO の補正:特定された歪みが S_TNVL 制限値を下回った場合のみ WO の補正がおこなわれます。

6) MD で WO 「精密」が設定されていない場合、補正は WO 「汎用」に基づいておこなわれます。

7) 計測タイプ「3つのボールの計測」の場合、グローバル基本フレームでの補正はできません(S_KNUM = 3001~3016)。これは、フレームに回転成分がないからです。

CYCLE995 計測サイクルパラメータ

```
PROC CYCLE995 (INT S_MVAR,INT S_KNUM,INT S_PRNUM,REAL S_SETV,REAL S_FA,REAL S_TSA,REAL
S_STA1,REAL S_INCA,REAL S_DZ,REAL S_SETV0,REAL S_SETV1,REAL S_SETV2,REAL S_TUL,REAL
S_TZL,INT S_NMSP,INT S_MCBIT,INT _DMODE,INT _AMODE)
```


4.26 サイクルを外部的にプログラミング

表 4-18 CYCLE995 呼び出しパラメータ ¹⁾

番号	画面パラメータ	サイクルパラメータ	意味
1		S_MVAR	計測タイプ(初期設定 = 5) 規格値: UNITS(一の位):輪郭要素 5 = 主軸形状(工具軸と平行) TENS(十の位):繰り返し計測 1 = 繰り返し計測あり HUNDREDS(百の位):オフセット対象なし 0 = 計測のみ THOUSANDS(千の位):計測方法 2 = 角度での計測、スイッチング方向へのプローブの割り出し TEN THOUSANDS(万の位):計測するボールの数 0 = ボールの計測 HUNDRED THOUSANDS(十万の位):計測点数 1 = 角度での計測時の 4 つの計測点(ボールのまわりを回る) ONE MILLION(百万の位):ボールの直径指令値の特定 0 = ボールの直径指令値を特定しない 1 = ボールの直径指令値を特定する
2	選択	S_KNUM	補正対象 0 = 0
3	アイコン + 数値	S_PRNUM	プローブパラメータのフィールドの番号(プローブ番号ではありません) (値の範囲 1~40)
4	DM	S_SETV	キャリブレーション球体の直径 ⁴⁾
5	DFA	S_FA	計測距離
6	TSA	S_TSA	安全領域 ⁵⁾
7	α0	S_STA1	角度での計測の開始角度 ³⁾
8		S_INCA	角度での計測の分割角度 ²⁾
9	DZ	S_DZ	プローブのシャフトでの 1 番目の計測 P1 から 2 番目の計測 P2 までの距離
10		S_SETV0	平面の 1 番目の軸(G17 の場合、X)のボールの指令位置 ²⁾

4.26 サイクルを外部的にプログラミング

番号	画面パラメータ	サイクルパラメータ	意味	
11		S_SETV1	平面の 2 番目の軸(G17 の場合、Y)のボールの指令位置 ²⁾	
12		S_SETV2	平面の 3 番目の軸(G17 の場合、Z)のボールの指令位置 ²⁾	
13	TUL	S_TUL	角度偏りの許容上限値	
14	TZL	S_TZL	ゼロオフセット範囲 ^{1), 4)}	
15	番号	S_NMSP	同じ位置での計測回数 ²⁾ (数値の範囲 1～9)	
16		S_MCBIT	予約済み ²⁾	
17		_DMODE	表示モード	
			規格値:	UNITS(一の位):加工平面 G17/G18/G19 0 = 互換性、サイクル呼び出しが有効になる前の平面が有効 1 = G17 (サイクルのみで有効) 2 = G18 (サイクルのみで有効) 3 = G19 (サイクルのみで有効)
18		_AMODE	代替モード	
			規格値:	UNITS(一の位):寸法許容誤差あり/なし 0 = なし 1 = あり

すべての初期値 = 0 または a～b の値の範囲として指定

¹⁾ 表示は一般 SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE で決定

²⁾ パラメータが現在未使用で、入力画面でも表示されません。
 パラメータ「分割角度」S_INCA は固定値として 90°が設定されます。

³⁾ 開始角度の数値の範囲-360°～+360°

⁴⁾ 寸法許容誤差ありの場合:
 計測角度がゼロオフセット範囲 TZL の値未満の場合、角度(_OVR[2], _OVR[3])および偏り(_OVR[7], _OVR[8])の結果パラメータはゼロに設定されます。
 TZL 表示は一般 SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE のビット 25=1 を使用しておこなわれます。
 (角度、主軸の計測時に選択したゼロオフセットを有効化してください)

⁵⁾ パラメータ TSA は、キャリブレーションボールの 1 番目の計測を示します。

CYCLE996 計測サイクルパラメータ

```
PROC CYCLE996 (INT S_MVAR, INT S_TC, INT S_PRNUM, REAL S_SETV, REAL S_STA1, REAL S_SETV0, REAL  
S_SETV1, REAL S_SETV2, REAL S_SETV3, REAL S_SETV4, REAL S_SETV5, REAL S_TNVL, REAL S_FA, REAL  
S_TSA, INT S_NMSP, INT S_MCBIT, INT _DMODE, INT _AMODE)
```

4.26 サイクルを外部的にプログラミング

表 4-19 CYCLE996 呼び出しパラメータ ¹⁾

番号	画面形式 パラメータ	サイクル パラメータ	意味
1		S_MVAR	計測タイプ(初期設定 = 1)
			規格値:
			UNITS(一の位):計測順序 0 = キネマティックを計算(選択:オペレータの確認が必要な結果表示、記録、旋回データセットの変更)、_AMODE を参照してください。1 = 1 番目の計測 2 = 2 番目の計測 3 = 3 番目の計測
			TENS(十の位):予約済み 0 = 0
			HUNDREDS(百の位):1 番目～3 番目の計測の計測タイプ 0 = キャリブレーションボールの軸と平行の計測 1 = キャリブレーションボールの角度での計測、主軸補正なし ³⁾ 2 = キャリブレーションボールの計測とプローブの切り替え方向での主軸の補正 ³⁾ 3 = 軸と平行の計測、開始角度あり ⁸⁾ 4 = 軸と平行の計測、開始角度あり、プローブの切り替え方向での主軸の実速度追従 ⁸⁾
			THOUSANDS(千の位):キネマティックの補正対象を計算 ⁴⁾ 0 = 計測のみ。旋回データセットを計算、変更はおこなわない 1 = 旋回データセットを計算。必要に応じて、オペレータの確認後に旋回データセットを変更 ⁴⁾
			TEN THOUSANDS(万の位):キネマティック計算のための計測軸(回転軸 1 または 2)または開ベクトル結合または閉ベクトル結合 0 = 閉ベクトル結合(キネマティック計算のみ) 1 = 回転軸 1 (1 番目～3 番目の計測のみ) 2 = 回転軸 2 (1 番目～3 番目の計測のみ) ⁵⁾ 3 = 開ベクトル結合(キネマティック計算のみ)
			HUNDRED THOUSANDS(十万の位):キネマティック計算のための回転軸 1 の正規化 0 = 回転軸 1 のスケーリングを行わない 1 = 平面の 1 番目の軸(G17 の場合、X)の方向のスケーリング

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味
			2 = 平面の 2 番目の軸(G17 の場合、Y)の方向のスケーリング 3 = 平面の 3 番目の軸(G17 の場合、Z)の方向のスケーリング ONE MILLION(百万の位):キネマティック計算のための回転軸 2 の正規化⁵⁾ 0 = 回転軸 2 のスケーリングを行わない 1 = 平面の 1 番目の軸(G17 の場合、X)の方向のスケーリング 2 = 平面の 2 番目の軸(G17 の場合、Y)の方向のスケーリング 3 = 平面の 3 番目の軸(G17 の場合、Z)の方向のスケーリング TEN MILLION(千万の位):ログファイル 0 = 記録ファイルなし 1 = 計算されたベクトル(工具ホルダ)および、MD で設定されている場合は、1 番目のダイナミックな 5 軸座標変換(TRAORI(1))を含む記録ファイル
2		S_TC	旋回データセットの番号(工具ホルダ)
3	アイコン + 数値	S_PRNUM	プローブパラメータのフィールドの番号(プローブ番号ではありません) (初期設定 = 1)
4		S_SETV	キャリブレーションボールの直径
5	α0	S_STA1	角度での計測の開始角度
6	α0	S_SETV0	回転軸 1 の位置の値(回転軸が手動か半自動の場合)
7	α1	S_SETV1	回転軸 2 の位置の値(回転軸が手動か半自動の場合) ⁶⁾
8	XN	S_SETV2	回転軸 1 を正規化するための位置の値
9	XN	S_SETV3	回転軸 2 を正規化するための位置の値 ⁶⁾
10	Δ	S_SETV4	オフセットベクトル I1~I4 の許容値
11	Δ	S_SETV5	回転軸ベクトル V1 と V2 の許容値
12	TVL	S_TNVL	回転軸の角度セグメントの制限値(値の範囲 1~60°) (初期設定=20) ⁷⁾
13	DFA	S_FA	計測距離
14	TSA	S_TSA	安全領域
15	計測	S_NMSP	同じ位置での計測回数 ²⁾ (初期設定 = 1)
16		S_MCBIT	予約済み

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
17		_DMODE	表示モード	
			規格 値:	UNITS(一の位):加工平面 G17/G18/G19 0 = 互換性、サイクル呼び出しの前に有効な平面が有効 1 = G17 (サイクルのみで有効) 2 = G18 (サイクルのみで有効) 3 = G19 (サイクルのみで有効)
18		_AMODE	代替モード	
			規格 値:	UNITS(一の位):許容誤差チェックあり/なし 0 = なし 1 = あり:ベクトル S_SETV4、S_SETV5 の許容値の評価
				TENS(十の位):計算されたベクトルを旋回データセットに入力する場合 のオペレータの確認 ⁴⁾ 0 = あり:オペレータが変更を確認してください 1 = なし:計算されたベクトルをすぐに入力(HUNDREDS と THOUSANDS の位= 0 の場合のみ有効)
				HUNDREDS(百の位):計測結果表示 ⁵⁾ 0 = なし 1 = あり
				THOUSANDS(千の位):計測結果表示は編集可能です。 0 = なし 1 = あり、編集可(HUNDREDS の位 = 1 の場合のみ有効です)

1) すべての初期値 = 0 または初期設定=x として指定

2) 表示は一般 SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE で決定。

3) このタイプを使って、たとえば 90°の位置で、キャリブレーションボールの保持シャフトとの干渉なしでキネマティックをキャリブレーションボールで計測することができます。開始角度 S_STA1 (0°~360°)を入力できます。ボールのまわりを回る場合の分割角度は 90°になります。

円弧軌跡の送り速度として、チャンネル別 SD55634 \$SCS_MEA_FEED_PLANE_VALUE が使用されます。

4) 入力前に M0 のオペレータ確認があります。ベクトルは NC スタートのみで入力されます。

計測プログラムが RESET で中止された場合、計算されたベクトルは入力されません。

ベクトルは、計算中にオフセットベクトルの許容誤差を超えなかった場合のみに入力されます。

5) 計算されたキネマティック計測タイプ専用の計測結果表示。

1 番目から 3 番目の計測後にも計測結果を表示させたい場合、チャンネル別 SD55613

\$SCS_MEA_RESULT_DISPLAY を設定します。

4.26 サイクルを外部的にプログラミング

- 6) 回転軸 2 は 2 つの回転軸を持つキネマティックの場合のみです
- 7) 回転軸の角度セグメントの制限値 20° ~ 60° の S_TNVL の値の範囲 S_TNVL の値 $< 20^{\circ}$ の場合、プローブのミクロンの範囲の計測の不正確さにより、精度低下の可能性があります。この制限値に違反すると、エラーメッセージ 61430 が出力されます。このとき、最小制限値も表示されます。
- 8) SD54760 ビット 17 = 1 の場合、主軸はプローブの切り替え方向に追跡されます。

CYCLE9960 計測サイクルパラメータ

```
PROC CYCLE9960 (INT S_MVAR, STRING[40] S_TNAME, INT S_PRNUM, REAL S_SETV, REAL S_SETV1, REAL  
S_START_RA1, REAL S_END_RA1, INT S_CMEA_RA1, REAL S_POS_RA2, REAL S_SETV2, REAL  
S_START_RA2, REAL S_END_RA2, INT S_CMEA_RA2, REAL S_POS_RA1, REAL S_SETV4, REAL S_FA, REAL  
S_TSA, INT S_NMSP, INT S_DMODE, INT S_AMODE, INT S_KNUM)
```


4.26 サイクルを外部的にプログラミング

表 4-20 CYCLE9960 呼び出しパラメータ¹⁾

番号	画面パラメータ	サイクルパラメータ	意味
1	選択	S_MVAR	計測タイプ(初期設定 = 1)
			規格値: UNITS(一の位):計測タイプ 0 = キネマティックの計測と計算(次により選択:結果の表示、プロトコル、旋回データセット(千の位 S_MVAR を参照) 1 = 基準ヘッドの測定 2 = ヘッドを基準ヘッドに合わせて最適化 3 = 補間点の計測と計算(E996)
			TENS(十の位):予約済み
			HUNDREDS (百の位):ボール計測タイプ 2 = キャリブレーションボールの計測とプローブの切り替え方向への軸の追跡 4 = 開始角度での軸と平行の計測、プローブの切り替え方向への軸の追跡 ²⁾
			THOUSANDS(千の位):キネマティックの補正対象を計算 ³⁾ 0 = 計測と計算。データセットが計算され、未変更のままです 1 = 計算されたデータセットがオペレータの確認後に変更されます 2 = 以前に計測したキネマティックが計算され、該当する場合は、オペレータの確認後に変更されます
			TENTHOUSANDS(万の位):軸の計測(回転軸 1 または 2) 1 = すべての既存の回転軸の計測と計算 4 = 回転軸 1 のみを計測および計算 5 = 回転軸 2 のみを計測および計算
2		S_TNAME	座標変換の名称(旋回データセットまたはキネマティックチェーンに基づいた座標変換)
3	アイコン + 数値	S_PRNUM	プローブパラメータのフィールドの数値(プローブ番号ではありません)(初期設定 = 1)
4		S_SETV	キャリブレーションボールの直径

4.26 サイクルを外部的にプログラミング

番号	画面パラメータ	サイクルパラメータ	意味
5	α1	S_SETV1	1 番目の回転軸の角度での計測のための開始角度
6			S_START_RA1:1 番目の回転軸の開始角度
7			S_END_RA1:1 番目の回転軸の最終角度
8			S_CMEA_RA1:1 番目の回転軸の計測回数、キネマティックの計測と計算の場合は 3 個。補間点の場合は、12 個まで設定できます。
9			S_POS_RA2:1 番目の回転軸を計測している間の 2 番目の回転軸の位置 ⁴⁾
10	α2	S_SETV2	2 番目の回転軸の角度での計測のための開始角度 ⁴⁾
11			S_START_RA2:2 番目の回転軸の開始角度 ⁴⁾
12			S_END_RA2:2 番目の回転軸の最終角度 ⁴⁾
13			S_CMEA_RA2:2 番目の回転軸の計測回数、キネマティックの計測と計算の場合は 3 個。補間点の場合は、12 個まで設定できます。 ⁴⁾
14			S_POS_RA1:2 番目の回転軸を計測している間の 1 番目の回転軸の位置
15	デルタ結線	S_SETV4	オフセットベクトルの許容値
16	DFA	S_FA	計測距離
17	TSA	S_TSA	安全領域
18	数値	S_NMSP	同じ位置での計測回数 ⁵⁾ (初期設定 = 1)
19		_DMODE	表示モード
			規格値: UNITS(一の位):加工平面 G17/G18/G19 0 = 互換性、サイクル呼び出しの前に有効な平面が有効 1 = G17 (サイクルのみで有効) 2 = G18 (サイクルのみで有効) 3 = G19 (サイクルのみで有効)

番号	画面パラメータ	サイクルパラメータ	意味	
20		_AMODE	代替モード	
			規格値:	ONES(一の位):許容範囲のチェックあり/なし 0 = なし 1 = あり:ベクトルの許容値の評価 -> S_SETV4、S_SETV5
				TENS(十の位):自動校正 0 = 自動校正なし 1 = 自動校正あり
				HUNDREDS (百の位):自動開始角度 0 = 設定された開始角度を使用 1 = 測定点ごとに開始角度を自動計算
21		S_KNUM	基準ヘッドの計測のために補正する WO の数	
			値	UNITS(一の位):補正は常に荒削りオフセットに適用され、仕上げオフセットは削除されます
				TENS(十の位):1 ... 99 の調整可能な WO の番号(1=G54)
				THOUSANDS(千の位):9 回の補正が有効、調整可能な WO

- 1) すべての初期値 = 0 または初期設定=xx として指定
- 2) SD54760 ビット 17 = 1 の場合、主軸はプローブの切り替え方向に追跡されます。
- 3) 入力前に M0 のオペレータ確認があります。NC が開始されるまでベクトルを入力できません。計測プログラムが RESET で中断された場合、計算されたベクトルは入力されません。計算でオフセットベクトルの許容値を超えていない場合のみ、ベクトルが入力されます。
- 4) 回転軸 2 は 2 つの回転軸を持つキネマティックの場合のみです
- 5) 表示は一般マシンデータ SD54760 \$SNS_MEA_FUNCTION_MASK_PIECE によって異なります。

CYCLE982 計測サイクルパラメータ

```
PROC CYCLE982 (INT S_MVAR, INT S_KNUM, INT S_PRNUM, INT S_MA, INT S_MD, REAL S_ID, REAL S_FA, REAL
S_TSA, REAL S_VMS, REAL S_STA1, REAL S_CORA, REAL S_TZL, REAL S_TDIF, INT S_NMSP, INT S_EVNUM, INT
S_MCBIT, INT _DMODE, INT _AMODE)
```

4.26 サイクルを外部的にプログラミング

表 4-21 CYCLE982 呼び出しパラメータ ¹⁾

番号	画面形式 パラメータ	サイクル パラメータ	意味
1		S_MVAR	計測タイプ
			規格 値:
			UNITS(一の位):校正/計測 0 = 工具プローブを校正する 1 = 単一工具計測 ³⁾ 2 = マルチ工具計測、長さで工具半径を特定(フライス工具の場合)
			TENS(十の位):MCS または WCS での校正または計測 0 = 機械基準 ⁴⁾ 1 = ワーク基準
			HUNDREDS(百の位):フライス工具の反転あり/なしの計測 0 = 反転なしの計測 1 = 反転ありの計測
			THOUSANDS(千の位):フライス工具の補正対象 0 = 長さまたは長さで半径を特定します(S_MVAR の 1 番目の位置を参照してください) 1 = S_MVAR の 1 番目の位置 = 1 の場合、半径を特定します 2 = S_MVAR の 1 番目の位置 = 1 または 2 の場合、長さで半径を特定します(正面)、 3 = サイドカッタ工具、上側の刃先(背面)、長さで半径を特定します ⁵⁾
			TEN THOUSANDS(万の位):フライス工具またはドリルの位置 0 = フライス工具またはドリルの軸方向位置、平面の 2 番目の軸(G18 の場合、X)での半径 ⁷⁾ 1 = フライス工具またはドリルの半径方向位置、平面の 1 番目の軸(G18 の場合、Z)での半径 ⁷⁾
			HUNDRED THOUSANDS(十万の位):インクレメンタル校正または計測 0 = 指定しない 1 = インクレメンタル校正または計測
			ONE MILLION(百万の位):開始角度の主軸位置決め S_STA1 (フライス工具の計測のみ) 0 = 主軸を位置決めしない 1 = 主軸を開始角度で位置決め S_STA1

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
2	選択	S_KNUM	オフセットタイプ ²⁾	
			規格 値:	UNITS(一の位):工具オフセット 0 = 指定なし(形状の工具オフセット) 1 = 摩耗の工具オフセット
3	アイコン+ 数値	S_PRNUM	プローブパラメータのフィールドの番号(プローブ番号ではありません) (初期設定 = 1)	
4	X0	S_MA	計測軸	
			規格 値:	1 = 平面の 1 番目の軸(G18 の場合、Z) 2 = 平面の 2 番目の軸(G18 の場合、X)
5	+-	S_MD	計測方向	
			規格 値:	0 = 選択なし(計測方向は現在値から特定) 1 = 正 2 = 負
6	Z2	S_ID	オフセット	
7	DFA	S_FA	計測距離	
8	TSA	S_TSA	安全領域	
9	VMS	S_VMS	校正の可変の計測速度 ²⁾	
10	α1	S_STA1	フライス工具計測時の開始角度	
11	α2	S_CORA	反転ありでフライス工具を計測する場合のオフセット角度 ⁸⁾	
12	TZL	S_TZL	フライス工具の計測時のゼロオフセット校正時 S_TZL = 0	
13	DIF	S_TDIF	寸法差チェック	
14	計測	S_NMSP	同じ位置での計測回数 ²⁾ (初期設定 = 1)	
15	EVN	S_EVNUM	経験平均値メモリ番号 ^{2), 9)}	
16		S_MCBIT	予約済み	

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
17		_DMODE	表示モード	
			規格値:	UNITS(一の位):加工平面 G17/G18/G19 0 = 互換性、サイクル呼び出しの前に有効な平面が有効 1 = G17 (サイクルのみで有効) 2 = G18 (サイクルのみで有効) 3 = G19 (サイクルのみで有効)
				TENS(十の位):旋盤工具とフライス工具の刃先位置 (入力画面での表示のみ 1~9)
				HUNDREDS(百の位):工具タイプ 0 = 旋盤工具 1 = フライス工具 2 = ドリル
				THOUSANDS(千の位):工具プローブを基準としたアプローチ方法 0 = PLUS [X/Z]; 工具位置が軸方向の場合 X、工具位置が半径方向の場合 Z 1 = MINUS [X/Z]; 工具位置が軸方向の場合 X、工具位置が半径方向の場合 Z
18		_AMODE	代替モード	
			規格値:	UNITS(一の位):予約済み
				TENS(十の位):予約済み
				HUNDREDS(百の位):予約済み
				THOUSANDS(千の位):計測後に校正および単一の計測のための開始位置にアプローチします(S_MVAR - UNITS を参照) 0 = 工具が配置され、プローブエッジを基準にした DFA によるオフセット 1 = 開始位置にアプローチ

- すべての初期値 = 0 または初期設定=x として指定
- 表示は一般 SD54762 _MEA_FUNCTION_MASK_TOOL で決定
- 旋盤またはフライス工具またはドリルを計測。パラメータ S_MA での計測軸。刃先位置 1~8 で旋盤工具を指定、フライス工具の場合はパラメータ S_MVAR の HUNDREDS から THOUSANDS の位を使用。

4.26 サイクルを外部的にプログラミング

- 4) 計測と校正は基本座標系(キネマティックトランスフォーメーションがオフの場合は MCS)でおこなわれます。
- 5) インクレメンタル計測の場合は不可
- 6) 複数計測 S_MVAR=x2x02 または x3x02 のみ(例: ディスクタイプまたは溝フライス工具)
- 7) チャンネル別 SD42950 \$SC_TOOL_LENGTH_TYPE = 2 の場合、工具長さの成分は旋盤工具と同様に割り当てられます。
- 8) 反転ありの計測の場合のみ S_MVAR=xx1x1
- 9) 経験値生成
経験値メモリの数値の範囲:経験値メモリの 1~20 の数値(n)、チャンネル別 SD55623
\$SCS_MEA_EMPIRIC_VALUE[n-1]を参照してください。

CYCLE971 計測サイクルパラメータ

```
PROC CYCLE971 (INT S_MVAR, INT S_KNUM, INT S_PRNUM, INT S_MA, INT S_MD, REAL S_ID, REAL S_FA, REAL  
S_TSA, REAL S_VMS, REAL S_TZL, REAL S_TDIF, INT S_NMSP, REAL S_F1, REAL S_S1, REAL S_F2, REAL  
S_S2, REAL S_F3, REAL S_S3, INT S_EVNUM, INT S_MCBIT, INT _DMODE, INT _AMODE)
```

4.26 サイクルを外部的にプログラミング

表 4-22 CYCLE971 呼び出しパラメータ¹⁾

番号	画面形式 パラメータ	サイクル パラメータ	意味	
1		S_MVAR	計測タイプ	
			規格 値:	UNITS(一の位): 0 = 工具プローブを校正 1 = 主軸停止で工具を計測 (長さまたは半径) 2 = 主軸回転で工具を計測(長さまたは半径)、パラメータ S_F1 ~S_S4 を参照してください
				TENS(十の位):機械座標系またはワーク座標系での計測 0 = MCS(機械基準)で計測、工具計測または工具プローブの校正 1 = WCS (ワーク基準)で計測、工具計測または工具プローブの校正
				HUNDREDS(百の位):各刃の点検 0 = なし 1 = あり
				THOUSANDS(千の位): 0 = 0
				TEN THOUSANDS(万の位):インクレメンタル校正または計測 0 = 指定しない 1 = インクレメンタル校正または計測
				HUNDRED THOUSANDS(十万の位):工具プローブの自動校正 0 = 工具プローブを自動的に校正しない 1 = 工具プローブを自動的に校正する
				ONE MILLION(百万の位):主軸反転ありの平面での校正 0 = 主軸反転なしの平面での校正 1 = 主軸反転ありの平面での校正
2	選択	S_KNUM	オフセットタイプ ²⁾	
			規格 値:	UNITS(一の位):工具オフセット 0 = 指定なし(形状の工具オフセット) 1 = 摩耗の工具オフセット
3	アイコン+ 数値	S_PRNUM	プローブパラメータのフィールドの数値(プローブの番号ではありません)	

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味	
4	X0	S_MA	計測軸、オフセット軸 ⁴⁾	
			規格値:	UNITS(一の位):計測軸の数値 1 = 平面の 1 番目の軸(G17 の場合、X) 2 = 平面の 2 番目の軸(G17 の場合、Y) 3 = 平面の 3 番目の軸(G17 の場合、Z)
				TENS(十の位): 0 = 0
				HUNDREDS(百の位):オフセット軸の数値 0 = オフセット軸ではありません 1 = 平面の 1 番目の軸(G17 の場合、X) 2 = 平面の 2 番目の軸(G17 の場合、Y)
5	+-	S_MD	計測方向	
			規格値:	0 = 選択なし(計測方向は現在値から特定) 1 = 正 2 = 負
6	V	S_ID	オフセット	
			規格値:	0 = オフセットなしの工具用 >0 = - 校正:校正工具の直径がプローブの直径の上限より大きい場合、オフセットは平面の 3 番目の軸(G17 の場合、Z)に適用されます。プローブの中心からの工具半径から S_ID の値を引いただけ、工具がオフセットされます。オフセット軸は、S_MA でも指定されます。 - 対策:複数の刃先をもつ場合、半径の計測では工具長さのオフセットおよび刃先の最も高い位置を指定してください。また、長さを計測する際には刃先の一番高い位置に対する工具半径のオフセットを指定してください。
7	DFA	S_FA	計測距離	
8	TSA	S_TSA	安全領域	
9	VMS	S_VMS	校正の可変の計測速度 ²⁾	
10	TZL	S_TZL	ゼロオフセット(工具計測の場合のみ)	
11	DIF	S_TDIF	工具計測の寸法差チェック(S_MVAR=xx1 または S_MVAR=xx2)	
12	計測	S_NMSP	同じ位置での計測回数 ²⁾	
13	F1	S_F1	回転主軸によるプロービングのための 1 番目の送り速度 ²⁾	

4.26 サイクルを外部的にプログラミング

番号	画面形式 パラメータ	サイクル パラメータ	意味
14	S1	S_S1	回転主軸によるプロービングのための 1 番目の回転数 ²⁾
15	F2	S_F2	回転主軸によるプロービングのための 2 番目の送り速度 ²⁾
16	S2	S_S2	回転主軸によるプロービングのための 2 番目の回転数 ²⁾
17	F3	S_F3	回転主軸によるプロービングのための 3 番目の送り速度 ³⁾
18	S3	S_S3	回転主軸によるプロービングのための 3 番目の回転数 ³⁾
19	EVN	S_EVNUM	経験値メモリ番号 ²⁾
20		S_MCBIT	_CBIT または _CHBIT のマスク
21		_DMODE	表示モード
		規格 値:	UNITS(一の位):加工平面 G17/G18/G19 0 = 互換性、サイクル呼び出しの前に有効な平面が有効 1 = G17 (サイクルのみで有効) 2 = G18 (サイクルのみで有効) 3 = G19 (サイクルのみで有効)
22		_AMODE	代替モード
		規格 値:	UNITS(一の位):半径の工具オフセットの計測 1 = なし 2 = あり
			TENS(十の位):平面の 3 番目の軸(G17 の場合、Z)の半径の計測時の工具 オフセットの方向 1 = 正 2 = 負
			HUNDREDS(百の位):長さの計測または 3 番目の軸のプローブの校正時 の工具オフセット 0 = 互換性、自動 1 = なし 2 = あり
			THOUSANDS(千の位):オフセット軸の長さの計測時の工具オフセットの 方向(S_MA HUNDREDS を参照) 1 = 正 2 = 負

¹⁾ すべての初期値 = 0 または初期設定=x として指定

²⁾ 表示は一般 SD54762 MEA_FUNCTION_MASK_TOOL で決定

4.26 サイクルを外部的にプログラミング

- 3) 工具のオフセットと寸法許容誤差「あり」の場合のみ、それ以外はパラメータ = 0
- 4) 自動計測の場合(S_MVAR=1x00xx)、計測軸の表示なし、オフセット軸 ⇒ S_MA=0。

CYCLE150 計測サイクルパラメータ

```
PROC CYCLE150 (INT S_PICT, INT S_PROT, STRING[160] S_PATH) SAVE
ACTBLOCNO DISPLOF
```

表 4-23 CYCLE150 呼び出しパラメータ

番号	画面パラメータ	サイクルパラメータ	意味	
1	計測結果画面	S_PICT	結果表示の選択(初期設定 = 0)	
			規格値:	UNITS(一の位): 0 = 計測結果画面 OFF 1 = 計測結果画面 ON
				TENS(十の位):表示モードの選択(値は SD55613 と同じ) 1 = 計測結果画面の表示 - 8 秒後に自動的に選択解除 3 = 計測結果画面の表示 - NC スタートを使用して確認 4 = 計測結果画面の表示 - アラーム時のみ(61303 ~ 61306)
2		S_PROT	ログの選択(初期設定 = 0)	
	ログ		規格値:	UNITS(一の位):記録オフ / オン / 最後の計測を選択 0 = ログ OFF 1 = ログ ON 2 = 最後の計測をログ
	ログタイプ			TENS(十の位):ログタイプの選択 0 = 標準ログ 1 = ユーザーログ(自由に定義可能)
	ログ形式			HUNDREDS(百の位):ログ形式の選択 0 = テキスト形式 1 = 表形式(Excel へのインポート用)

4.26 サイクルを外部的にプログラミング

番号	画面パラメータ	サイクルパラメータ	意味	
	ログデータ			THOUSANDS(千の位):選択の書き直しまたは追加 0 = 新規 1 = 追加
	ログアーカイブ			TEN THOUSANDS(万の位):ログアーカイブの選択 0 = パートプログラムとして 1 = ディレクトリ
3		S_PATH	ログアーカイブの選択に対応したログファイルのパス (完全なパス名またはファイル名のみ、例: "//NC:/WKS.DIR/NAME.WPD"または"MESSPROTOKOLL.TXT"	

テーブル

5.1 命令

注記

サイクル

操作リストには、NC プログラム(G コード)で発生するすべてのサイクルが含まれます。サイクルには、マスクを使用してプログラムエディタでプログラムできるものと、プログラムサポートなしでループ用にプログラムしなければならないものがあります。互換性維持のために制御装置で引き続き利用できるものの、SINUMERIK Operate プログラムエディタを使って編集できないサイクル(「互換性サイクル」)は考慮されません。

命令 A ... C

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
:	O	NC メインブロック番号、ジャンプラベル終了、結合演算子		+		PM-NC
*	O	乗算演算子		+		PM-NC
+	O	加算演算子		+		PM-NC
-	O	減算演算子		+		PM-NC
<	O	比較演算子、より小さい		+		PM-NC
<<	O	文字列用結合演算子		+		PM-NC
<=	O	比較演算子、以下		+		PM-NC
=	O	代入演算子		+		PM-NC
>=	O	比較演算子、以上		+		PM-NC
/	O	除算演算子		+		PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
I0 I7		ブロックをスキップします(スキップレベル 1) ブロックをスキップします(スキップレベル 8)		+		PM-NC
A	A	軸名称	m/s	+		PM-NC
A2	A	工具オリエンテーション:RPY 角またはオイラー角	s	+		PM-NC
A3	A	工具オリエンテーション:方向ベクトルの第 1 成分	s	+		PM-NC
A4	A	工具オリエンテーション:ブロック開始時の面法線ベクトルの第 1 成分	s	+		PM-NC
A5	A	工具オリエンテーション:ブロック終了時の面法線ベクトルの第 1 成分	s	+		PM-NC
A6	A	工具オリエンテーション:テーパの回転軸の方向ベクトルの第 1 成分	s	+		PM-NC
A7	A	工具オリエンテーション:テーパの円筒面の中間位置決めの第 1 ベクトル要素	s	+		PM-NC
ABS	F	アブソリュート値(数量)		+	+	PM-NC
AC	K	座標/位置のアブソリュート指令	s	+		PM-NC
ACC	K	現在の軸加減速度の働き	M	+	+	PM-NC
ACCLIMA	K	現在の最大軸加減速度の働き	M	+	+	PM-NC
ACN	K	回転軸のアブソリュート指令、負方向位置へのアプローチ	s	+		PM-NC
ACOS	F	アークコサイン (三角関数)		+	+	PM-NC
ACP	K	回転軸のアブソリュート指令、正方向位置へのアプローチ	s	+		PM-NC
ACTBLOCNO	P	「実行中のブロックの表示をマスク」(DISPLOF)が有効の場合でも、アラームブロックの実行中のブロック番号を出力。		+		PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
ADDFRAME	F	計測されたフレームの結合と起動		+	-	PM-NC、FM-B
ADIS	A	軌跡機能 G1、G2、G3、...の丸み付き隙間	M	+		PM-NC
ADISPOS	A	早送り G0 の丸み付き隙間	M	+		PM-NC
ADISPOSA	P	IPOBRKA の許容範囲の大きさ	M	+	+	PM-NC
AFISOF	P	自動フィルタチェーン切り替えの解除	M	+	-	PM-NC
AFISON	P	自動フィルタチェーン切り替えの起動	M	+	-	PM-NC
ALF	A	高速リトラクト角度	M	+		PM-NC
AMIRROR	G	プログラマブルミラーリング	s	+		PM-NC
および	K	論理積		+		PM-NC
ANG	A	輪郭角度	s	+		PM-NC
AP	A	極角度	m/s	+		PM-NC
APR	K	読み出し/表示のアクセス保護		+		PM-NC
APRB	K	読み出しのアクセス権、OPI		+		PM-NC
APRP	K	読み出しのアクセス権、パートプログラム		+		PM-NC
APW	K	書き込みのアクセス保護		+		PM-NC
APWB	K	書き込みのアクセス権、OPI		+		PM-NC
APWP	K	書き込みのアクセス権、パートプログラム		+		PM-NC
APX	K	指定した言語要素を実行するためのアクセス権の定義		+		PM-NC
AR	A	開口角度	m/s	+		PM-NC
AROT	G	プログラマブル座標回転	s	+		PM-NC
AROTS	G	立体角によるプログラマブルフレームの回転	s	+		PM-NC
AS	K	マクロ定義		+		PM-NC
ASCALE	G	プログラマブルスケーリング	s	+		PM-NC
ASIN	F	算術機能、アークサイン		+	+	PM-NC
ASPLINE	G	A スプライン	M	+		PM-NC
ATAN2	F	アークタンジェント 2		+	+	PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
ATOL	A	コンプレッサ機能、旋回スミージング、およびスミージングタイプの軸別許容範囲	M	+		PM-NC
ATRANS	G	追加プログラマブルゼロオフセット	s	+		PM-NC
AUXFUDEL	P	グローバルリストから、特定チャネルの補助機能を削除		+	-	FM-B
AUXFUDELG	P	グローバルリストから、特定チャネルの補助機能グループのすべての補助機能を削除		+	-	FM-B
AUXFUMSEQ	P	M 補助機能の出力処理を決定		+	-	FM-B
AUXFUSYNC	P	補助機能のグローバルリストの文字列としての、チャネル別 SERUPRO と ASUB の完全なパートプログラムブロックを生成		+	-	FM-B
AX	K	可変軸識別子	m/s	+		PM-NC
AXCTSWE	P	軸コンテナの回転		+	-	PM-NC
AXCTSWEC	P	軸コンテナの回転の有効化を取り消し		+	+	PM-NC
AXCTSWED	P	軸コンテナの回転(セットアップ用の命令タイプ)		+	-	PM-NC
AXIS	K	軸識別子、軸アドレス		+		PM-NC
AXNAME	F	入力文字列を軸識別子に変換		+	-	PM-NC
AXSTRING	F	主軸番号を文字列に変換		+	-	PM-NC
AXTOCHAN	P	特定のチャネルのための軸の要求。NC プログラムとシンクロナイズドアクションで実行可能です。		+	+	PM-NC
AXTOSPI	F	軸識別子を主軸インデックスへ変換		+	-	PM-NC
B	A	軸名称	m/s	+		PM-NC
B2	A	工具オリエンテーション: RPY 角またはオイラー角	s	+		PM-NC
B3	A	工具オリエンテーション: 方向ベクトルの成分	s	+		PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
B4	A	工具オリエンテーション:ブロック開始時の面法線ベクトルの第2成分	s	+		PM-NC
B5	A	工具オリエンテーション:ブロック終了時の面法線ベクトルの第2成分	s	+		PM-NC
B6	A	工具オリエンテーション:テーパの回転軸の方向ベクトルの第2成分	s	+		PM-NC
B7	A	工具オリエンテーション:テーパの円筒面の中間位置決めの第2ベクトル要素	s	+		PM-NC
B_AND	O	ビットごとの AND		+		PM-NC
B_OR	O	ビットごとの OR		+		PM-NC
B_NOT	O	ビットごとの否定		+		PM-NC
B_XOR	O	ビットごとの排他的 OR		+		PM-NC
BAUTO	G	隣接する3点による最初のスプライン区間の定義	M	+		PM-NC
BLOCK	K	キーワード TO との併用により、間接サブプログラム呼び出しで処理するプログラム部分を定義します		+		PM-NC
BLSYNC	K	次のブロック切り替えでのみ、割り込みルーチンの処理を起動します		+		PM-NC
BNAT ⁶⁾	G	最初のスプラインブロックへの自然遷移	M	+		PM-NC
BOOL	K	データタイプ:TRUE/FALSE または 1/0 のブール値です		+		PM-NC
BOUND	F	値が定義された数値範囲内にあるかどうかをテストします。複数の値が等しい場合は、テスト値を返します。		+	+	PM-NC
BRISK ⁶⁾	G	スムージングをおこなわない高速の軌跡加減速度	M	+		PM-NC
BRISKA	P	プログラム指令軸に対して最大の軌跡加減速度を起動します		+	-	PM-NC
BSPLINE	G	B スプライン	M	+		PM-NC
BTAN	G	最初のスプラインブロックへの接線方向の遷移	M	+		PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
C	A	軸名称	m/s	+		PM-NC
C2	A	工具オリエンテーション: RPY 角またはオイラー角	s	+		PM-NC
C3	A	工具オリエンテーション: 方向ベクトルの第 3 成分	s	+		PM-NC
C4	A	工具オリエンテーション: ブロック開始時の面法線ベクトルの第 3 成分	s	+		PM-NC
C5	A	工具オリエンテーション: ブロック終了時の面法線ベクトルの第 3 成分	s	+		PM-NC
C6	A	工具オリエンテーション: テーパの回転軸の方向ベクトルの第 3 成分	s	+		PM-NC
C7	A	工具オリエンテーション: テーパの円筒面の中間位置決めの第 3 ベクトル要素	s	+		PM-NC
CAC	K	アブソリュート位置へアプローチします		+		PM-NC
CACN	K	テーブルの記述値で負方向へアブソリュートのアプローチをおこないます		+		PM-NC
CACP	K	テーブルの記述値で正方向へアブソリュートのアプローチをおこないます		+		PM-NC
CADAPTOF	P	負荷調整の無効化		+	-	PM-NC
CADAPTON	P	負荷調整の有効化		+	-	PM-NC
CALCDAT	F	3 または 4 点から、円弧の半径と中心点を計算します		+	-	PM-NC
CALCFIR	P	FIR 加々速度フィルタのダイナミック応答モードへの設定		+	-	PM-NC
CALCPOSI	F	プロテクションゾーン違反、ワーキングエリアリミット、およびソフトウェアリミットのチェック		+	-	PM-NC
CALL	K	間接サブプログラム呼び出し		+		PM-NC
CALLPATH	P	サブプログラム呼び出しのプログラム指令可能な検索パス		+	-	PM-NC
CANCEL	P	モーダルシンクロナイズドアクションをキャンセルします		+	-	FM-SA

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
CANCELSUB	P	現在のサブプログラムレベルのキャンセル		+	+	FM-SA
CASE	K	条件付きプログラム分岐		+		PM-NC
CDC	K	位置への直接アプローチ		+		PM-NC
CDOF ⁶⁾	G	衝突検出をオフにします	M	+		PM-NC
CDOF2	G	3次元外周加工中の衝突検出をオフにします	M	+		PM-NC
CDON	G	衝突検出を起動します	M	+		PM-NC
CFC ⁶⁾	G	輪郭の一定送り速度	M	+		PM-NC
CFIN	G	外側半径ではなく、内側半径のみでの一定送り速度	M	+		PM-NC
CFINE	F	FRAME 変数への仕上げオフセットの割り当て		+	-	PM-NC
CFTCP	G	工具中心点の一定送り速度(中心点軌跡)	M	+		PM-NC
CHAN	K	データの有効範囲を指定します		+		PM-NC
CHANDATA	P	チャンネルデータアクセス用のチャンネル番号の設定		+	-	PM-NC
CHAR	K	データタイプ:ASCII 文字		+		PM-NC
CHF	A	面取り; 値 = 面取り長さ	S	+		PM-NC
CHKDM	F	マガジン内での一義性のチェック		+	-	FM-TM
CHKDNO	F	固有の D 番号のチェック		+	-	PM-NC
CHR	A	面取り; 値 = 移動方向の面取り長さ		+		PM-NC
CIC	K	ステップ値によるアプローチ位置		+		PM-NC
CIP	G	中間点経由の円弧補間	M	+		PM-NC
CLEARM	P	チャンネル協調の 1 個/複数のマークをリセット		+	+	PM-NC
CLRINT	P	割り込みの解除		+	-	PM-NC
CMIRROR	F	座標軸のミラーリング		+	-	PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
COARSEA	K	「汎用イグザクトストップ」に達すると移動終了	M	+		PM-NC
COLLPAIR	F	干渉ペアの要素であるかどうかをチェック		+		PM-NC
COMPCAD	G	コンプレッサ機能 COMPCAD を有効にします。	M	+		PM-NC
COMPCURV	G	コンプレッサ機能 COMPCURV を有効にします	M	+		PM-NC
COMPLETE		データの読み出しと書き込みの制御命令です		+		PM-NC
COMPOF ⁶⁾	G	NC ブロック圧縮を無効にします	M	+		PM-NC
COMPON	G	コンプレッサ機能 COMPON を有効にします	M	+		PM-NC
COMPSURF	G	コンプレッサ機能 COMPSURF を有効にします	M	+		PM-NC
CONTDCON	P	テーブル輪郭の解釈を有効にします		+	-	PM-NC
CONTPRON	P	輪郭解析の起動		+	-	PM-NC
CORROF	P	すべての有効な重畳移動を解除		+	-	PM-NC
CORRTC	F	旋回工具ホルダのオフセットベクトルまたは方向ベクトルを、機械の計測に従って変更します。		+	-	PM-NC
CORRTAFO	F	機械のキネマティックモデルの方向軸のオフセットベクトルまたは方向ベクトルの変更		+	-	PM-NC
COS	F	コサイン (三角関数)		+	+	PM-NC
COUPDEF	P	ELG グループ/主軸同期グループの定義		+	-	PM-NC
COUPDEL	P	ELG グループの解除		+	-	PM-NC
COUPOF	P	ELG グループ/主軸同期グループのペアの無効化		+	-	PM-NC
COUPOFS	P	スレーブ主軸の停止により、ELG グループ/主軸同期ペアを解除		+	-	PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
COUPON	P	ELG グループ/主軸同期グループのペアの有効化		+	-	PM-NC
COUPONC	P	以前のプログラミングで、ELG グループ/主軸同期ペアの起動をおこないます		+	-	PM-NC
COUPRES	P	ELG グループのリセット		+	-	PM-NC
CP ⁶⁾	G	軌跡移動	M	+		PM-NC
CPBC	K	汎用連結:ブロック切り替え条件		+	+	FM-A
CPDEF	K	汎用連結:連結モジュールの作成		+	+	FM-A
CPDEL	K	汎用連結:連結モジュールの削除		+	+	FM-A
CPFMOF	K	汎用連結:完全なスイッチオフでのスレーブ軸の動作		+	+	FM-A
CPFMON	K	汎用連結:スイッチオンでのスレーブ軸の動作		+	+	FM-A
CPFMSON	K	汎用連結:同期モード		+	+	FM-A
CPFPOS	K	汎用連結:スレーブ軸の同期位置		+	+	FM-A
CPFRS	K	汎用連結:座標系基準システム		+	+	FM-A
CPLA	K	汎用連結:マスタ軸の定義		+	-	FM-A
CPLCTID	K	汎用連結:カーブテーブルの番号		+	+	FM-A
CPLDEF	K	汎用連結:マスタ軸の定義と連結モジュールの作成		+	+	FM-A
CPLDEL	K	汎用連結:連結モジュールのマスタ軸の削除		+	+	FM-A
CPLDEN	K	汎用連結:連結係数の分母		+	+	FM-A
CPLINSC	K	汎用連結:マスタ軸の入力値の単位変換係数		+	+	FM-A
CPLINTR	K	汎用連結:マスタ軸の入力値のオフセット値		+	+	FM-A
CPLNUM	K	汎用連結:連結係数の分子		+	+	FM-A
CPLOF	K	汎用連結:連結モジュールのマスタ軸のスイッチオフ		+	+	FM-A

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
CPLON	K	汎用連結:連結モジュールのマスタ軸のスイッチオン		+	+	FM-A
CPLOUTSC	K	汎用連結:連結の出力値の単位変換係数		+	+	FM-A
CPLOUTTR	K	汎用連結:連結の出力値のオフセット値		+	+	FM-A
CPLPOS	K	汎用連結:マスタ軸の同期位置		+	+	FM-A
CPLSETVAL	K	汎用連結:連結基準		+	+	FM-A
CPMALARM	K	汎用連結:特別な連結関連アラーム出力のマスク		+	+	FM-A
CPMBRAKE	K	汎用連結:特定の停止信号と停止命令に対するスレーブ軸の応答		+	-	FM-A
CPMPRT	K	汎用連結:プログラムテストによるブロック検索実行での、パートプログラム開始時の連結動作		+	+	FM-A
CPMRESET	K	汎用連結:RESET に対する連結動作		+	+	FM-A
CPMSTART	K	汎用連結:パートプログラム開始時の連結動作		+	+	FM-A
CPMVDI	K	汎用連結:特定の NC/PLC インターフェース信号に対する、スレーブ軸の応答		+	+	FM-A
CPOF	K	汎用連結:連結モジュールのスイッチオフ		+	+	FM-A
CPON	K	汎用連結:連結モジュールのスイッチオン		+	+	FM-A
CPRECOF ⁶⁾	G	プログラマブル輪郭精度を解除します	M	+		PM-NC
CPRECON	G	プログラマブル輪郭精度を有効にします	M	+		PM-NC
CPRES	K	汎用連結:同期主軸連結の設定データを有効化		+	-	FM-A
CPROT	P	チャネル別プロテクションゾーンの有効化/無効化		+	-	PM-NC
CPROTDEF	P	チャネル別プロテクションゾーンの定義		+	-	PM-NC
CPSETTYPE	K	汎用連結:連結タイプ		+	+	FM-A
CPSYNCOF	K	汎用連結:位置同期制御「汎用」のしきい値		+	+	FM-A

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
CPSYNCO2	K	汎用連結:位置同期制御「汎用」のしきい値 2		+	+	FM-A
CPSYNCOV	K	汎用連結:速度同期制御「汎用」のしきい値		+	+	FM-A
CPSYNFIP	K	汎用連結:位置同期制御「精密」のしきい値		+	+	FM-A
CPSYNFIP2	K	汎用連結:位置同期制御「精密」のしきい値 2		+	+	FM-A
CPSYNFIV	K	汎用連結:速度同期制御「精密」のしきい値		+	+	FM-A
CR	A	円弧半径	S	+		PM-NC
CROT	F	現在の座標系の回転		+	-	PM-NC
CROTS	F	立体角によるプログラマブルフレームの回転(指定軸で回転)	S	+	-	PM-NC
CRPL	F	任意の平面のフレーム回転		+	-	FM-B
CSCALE	F	複数軸のスケーリング係数		+	-	PM-NC
CSPLINE	F	3 次スプライン	M	+		PM-NC
CT	G	接線方向の遷移をおこなう円弧	M	+		PM-NC
CTAB	F	カーブテーブルからマスタ軸位置に応じたスレーブ軸位置を定義		+	+	PM-NC
CTABDEF	P	テーブルの定義の有効化		+	-	PM-NC
CTABDEL	P	カーブテーブルのクリア		+	-	PM-NC
CTABEND	P	テーブルの定義の無効化		+	-	PM-NC
CTABEXISTS	F	番号 n のカーブテーブルをチェック		+	+	PM-NC
CTABFNO	F	メモリで引き続き有効なカーブテーブルの数		+	+	PM-NC
CTABFPOL	F	メモリで引き続き有効な多項式の数		+	+	PM-NC
CTABFSEG	F	メモリで引き続き有効なカーブセグメントの数		+	+	PM-NC
CTABID	F	n 番目のカーブテーブルのテーブル番号を返します		+	+	PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
CTABINV	F	カーブテーブルからスレーブ軸位置に応じたマスタ軸位置の定義		+	+	PM-NC
CTABISLOCK	F	番号 n のカーブテーブルのロック状態を返します		+	+	PM-NC
CTABLOCK	P	解除と上書きのロック		+	+	PM-NC
CTABMEMTYP	F	カーブテーブル番号 n が作成されたメモリを返します。		+	+	PM-NC
CTABMPOL	F	メモリで引き続き有効な多項式の最大数		+	+	PM-NC
CTABMSEG	F	メモリで引き続き有効なカーブセグメントの最大数		+	+	PM-NC
CTABNO	F	SRAM または DRAM の、定義したカーブテーブルの合計数		+	+	FM-A
CTABNOMEM	F	SRAM または DRAM の、定義したカーブテーブルの合計数		+	+	PM-NC
CTABPERIOD	F	番号 n のカーブテーブルのテーブルの周期性を返します		+	+	PM-NC
CTABPOL	F	メモリですでに使用している多項式の数		+	+	PM-NC
CTABPOLID	F	番号 n のカーブテーブルが使用するカーブ多項式の数		+	+	PM-NC
CTABSEG	F	メモリですでに使用しているカーブセグメントの数		+	+	PM-NC
CTABSEGID	F	番号 n のカーブテーブルが使用するカーブセグメントの数		+	+	PM-NC
CTABSEV	F	カーブテーブルのセグメントのスレーブ軸の終了値を返します		+	+	PM-NC
CTABSSV	F	カーブテーブルのセグメントのスレーブ軸の開始値を返します		+	+	PM-NC
CTABTEP	F	カーブテーブル終了位置のマスタ軸の値を返します		+	+	PM-NC
CTABTEV	F	カーブテーブル終了位置の従動軸の値を返します		+	+	PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
CTABTMAX	F	カーブテーブルのスレーブ軸の最大値を返します		+	+	PM-NC
CTABTMIN	F	カーブテーブルのスレーブ軸の最小値を返します		+	+	PM-NC
CTABTSP	F	カーブテーブル開始位置のマスタ軸の値を返します		+	+	PM-NC
CTABTSV	F	カーブテーブル開始位置の従動軸の値を返します		+	+	PM-NC
CTABUNLOCK	P	削除のロックと上書きのロックを無効にします		+	+	PM-NC
CTOL	A	コンプレッサ機能、旋回スモーキング、およびスモーキングタイプの輪郭許容範囲	M	+	-	PM-NC
CTOLG0	A	早送り移動の輪郭許容範囲	M	+	-	PM-NC
CTTRANS	F	複数軸のゼロオフセット		+	-	PM-NC
CUT2D ⁶⁾	G	2 次元 TRC	M	+		PM-NC
CUT2DD	G	差分工具に対する 2½次元 TRC	M	+		PM-NC
CUT2DF	G	現在のフレームを基準にした 2D TRC(傾斜面)	M	+		PM-NC
CUT2DFD	G	現在のフレームを基準にした差分工具に対する 2½次元 TRC (傾斜面)	M	+		PM-NC
CUT3DC	G	外周加工の 3 次元 TRC	M	+		PM-NC
CUT3DCC	G	3 次元径補正を使用した、限界面を考慮した外周加工のための 3D TRC:加工面の輪郭	M	+		PM-NC
CUT3DCCD	G	工具中心点軌跡で標準工具とは違う工具を使用した、限界面を考慮した外周加工のための 3D TRC:限界面への切り込み	M	+		PM-NC
CUT3DCD	G	外周加工の差分工具に対する 3 次元 TRC	M	+		PM-NC
CUT3DF	G	向きが変化する正面削りの 3 次元 TRC	M	+		PM-NC
CUT3DFD	G	向きが変化する正面削りの差分工具に対する 3 次元 TRC	M	+		PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
CUT3DFF	G	向きが一定の正面削りの 3 次元 TRC。 工具オリエンテーションは、G17～G19 で定義した方向で、場合によっては、フ レームによって回転します。	M	+		PM-NC
CUT3DFS	G	向きが一定の正面削りの 3 次元 TRC。 工具オリエンテーションは、G17～G19 で定義され、フレームには影響されませ ん。	M	+		PM-NC
CUTCONOF ⁶⁾	G	工具径補正の無効化	M	+		PM-NC
CUTCONON	G	工具径補正の有効化	M	+		PM-NC
CUTMOD	A	旋回工具のオフセットデータ変更の起動 (方向付け可能な工具ホルダと接続)	M	+		PM-NC
CUTMODK	A	旋回工具のオフセットデータ変更の起動 (キネマティック結合により定義された 方向変換と接続)	M	+		PM-NC
CYCLE60	C (T)	文字彫りサイクル		+		PM-NC
CYCLE61	C (T)	正面削り		+		PM-NC
CYCLE62	C (T)	輪郭呼び出し		+		PM-NC
CYCLE63	C (T)	輪郭ポケットフライス加工		+		PM-NC
CYCLE64	C (T)	輪郭ポケットの前加工		+		PM-NC
CYCLE70	C (T)	ねじフライス削り		+		PM-NC
CYCLE72	C (T)	軌跡のフライス削り		+		PM-NC
CYCLE76	C (T)	長方形スピゴットのフライス削り		+		PM-NC
CYCLE77	C (T)	円形スピゴットのフライス削り		+		PM-NC
CYCLE78	C (T)	ねじのフライス削り		+		PM-NC
CYCLE79	C (T)	ポリゴン		+		PM-NC
CYCLE81	C (T)	穴あけ、センタリング		+		PM-NC
CYCLE82	C (T)	穴あけ、座ぐりフライス		+		PM-NC
CYCLE83	C (T)	深穴ドリル		+		PM-NC
CYCLE84	C (T)	フローティングチャックのないタッピン グ		+		PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
CYCLE85	C (T)	リーマ加工		+		PM-NC
CYCLE86	C (T)	ボーリング		+		PM-NC
CYCLE92	C (T)	突っ切り		+		PM-NC
CYCLE95	C (T)	輪郭に沿った切削		+		PM-NC
CYCLE98	C (T)	連続ねじ		+		PM-NC
CYCLE99	C (T)	ねじ切り		+		PM-NC
CYCLE116	C (M)	円弧の中心点と半径の計算		+		PM-MC
CYCLE119	C (M)	空間内での位置を特定		+		PM-MC
CYCLE150	C (M)	計測結果の表示/ログ		+		PM-MC
CYCLE435	C (T)	目立て工具位置の計算		+		PM-NC
CYCLE495	C (T)	成形研削		+		PM-NC
CYCLE750	C (A)	CYCLE751 ... CYCLE759 (実際のファンクションコールに対する MMC 命令を含む)の内部動作サイクル		-		FM-A
CYCLE751	C (A)	オートチューニングセッションを開く/実行/閉じる		M		FM-A
CYCLE752	C (A)	オートチューニングセッションへの軸の追加		M		FM-A
CYCLE753	C (A)	オートチューニングモードの選択		M		FM-A
CYCLE754	C (A)	言語ブロックの追加/削除		M		FM-A
CYCLE755	C (A)	データレコードのバックアップ/リストア		M		FM-A
CYCLE756	C (A)	オートチューニング結果の有効化		M		FM-A
CYCLE757	C (A)	オートチューニングデータの保存		M		FM-A
CYCLE758	C (A)	パラメータ値の変更		M		FM-A
CYCLE759	C (A)	パラメータ値の読み取り		M		FM-A
CYCLE782	C (T)	ロードの調整		+		PM-NC
CYCLE800	C (T)	旋回		+		PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
CYCLE801	C (T)	格子またはフレーム		+		PM-NC
CYCLE802	C (T)	任意の位置		+		PM-NC
CYCLE806	C (T)	補間旋削		+		PM-NC
CYCLE830	C (T)	深穴ドリル 2		+		PM-NC
CYCLE832	C (T)	高速設定		+		PM-NC
CYCLE840	C (T)	フローティングチャックを使用したタッピング		+		PM-NC
CYCLE899	C (T)	オープン溝のフライス削り		+		PM-NC
CYCLE930	C (T)	溝		+		PM-NC
CYCLE940	C (T)	アンダーカット形状		+		PM-NC
CYCLE951	C (T)	切削		+		PM-NC
CYCLE952	C (T)	輪郭の溝削り		+		PM-NC
CYCLE961	C (M)	ワークのコーナー(内側または外側)の位置を特定し、ゼロオフセットとして挿入します。		+		PM-MC
CYCLE971	C (M)	工具プローブの調整、工具計測の長さや半径(フライス削りのみ)		+		PM-MC
CYCLE973	C (M)	ワークの表面または溝でワークプローブを調整(旋削のみ)		+		PM-MC
CYCLE974	C (M)	選択した計測軸でワーク原点を決定し、1 点計測で工具オフセットを決定します(旋削のみ)。		+		PM-MC
CYCLE976	C (M)	特定の軸および方向の端または作業平面の校正リングまたは校正ボールでワークプローブを校正		+		PM-MC
CYCLE977	C (M)	平面の直径、幅、中心を決定		+		PM-MC
CYCLE978	C (M)	この計測サイクルは、ワーク座標系で端面の位置を計測するために使用します。		+		PM-MC
CYCLE979	C (M)	平面の中心を決定して円弧の半径を計測します。		+		PM-MC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
CYCLE982	C (M)	工具プローブの調整、旋削穴あけおよび フライス工具の計測(旋削のみ)		+		PM-MC
CYCLE994	C (M)	選択した計測軸で、2 点計測でワーク原 点を決定します(旋削のみ)。		+		PM-MC
CYCLE995	C (M)	工作機械の主軸の角度の計測		+		PM-MC
CYCLE996	C (M)	回転軸とキネマティックトランスフォー メーションの座標変換データの特 定		+		PM-MC
CYCLE997	C (M)	ボールの直径と中心を特定し、配置され た 3 個のボールの中心を測定		+		PM-MC
CYCLE998	C (M)	作業平面で参照される平面の回転位置を 特定し、ワーク座標系の端面の角度を特 定します。		+		PM-MC
CYCLE4071	C (T)	反転点での切り込みによる長手方向の研 削		+		PM-NC
CYCLE4072	C (T)	反転点での切り込みによる長手方向の研 削とキャンセル信号		+		PM-NC
CYCLE4073	C (T)	連続切り込みによる長手方向の研削		+		PM-NC
CYCLE4074	C (T)	連続切り込みによる長手方向の研削とキ ャンセル信号		+		PM-NC
CYCLE4075	C (T)	反転点での切り込みによる平面研削		+		PM-NC
CYCLE4077	C (T)	反転点での切り込みによる平面研削とキ ャンセル信号		+		PM-NC
CYCLE4078	C (T)	連続切り込みによる平面研削		+		PM-NC
CYCLE4079	C (T)	断続的な切り込みによる平面研削		+		PM-NC
CYCLE9960	C (M)	キネマティックの完全な計測		+		PM-MC

5.1 命令

命令 D ... F

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1)2)3)4)5) 詳細は、一覧 (ページ 1433)を参照してください。						
D	A	工具オフセット番号		+		PM-NC
D0	A	D0 の場合、工具のオフセットは無効		+		PM-NC
DAC	K	ノンモーダルのアブソリュート指令の軸別直径指定	s	+		PM-NC
DC	K	回転軸のアブソリュート指令、位置への直接アプローチ	s	+		PM-NC
DCI	K	データクラス I (= Individual)の割り当て (SINUMERIK 828D のみ)		+		PM-NC
DCM	K	データクラス M (= Manufacturer)の割り当て (SINUMERIK 828D のみ)		+		PM-NC
DCU	K	データクラス U (= User)の割り当て (SINUMERIK 828D のみ)		+		PM-NC
DEF	K	変数定義		+		PM-NC
DEFAULT	K	CASE 分岐の分岐		+		PM-NC
DEFINE	K	マクロ定義のキーワード		+		PM-NC
DELAYFSTOF	P	停止遅延領域の終了区間の定義	M	+	-	PM-NC
DELAYFSTON	P	停止遅延区間の開始区間の定義	M	+	-	PM-NC
DELDL	F	追加オフセットの削除		+	-	PM-NC
DELDTG	P	残移動距離削除		-	+	FM-SA
DELETE	P	指定したファイルの削除。ファイル名称は、パスとファイル識別子を使用して指定できます。		+	-	PM-NC
DELMLOWNER	F	工具の所有マガジンロケーションの削除		+	-	FM-TM
DEMLRES	F	マガジンロケーション予約の削除		+	-	FM-TM
DELMT	P	マルチ工具の削除		+	-	FM-TM
DELOBJ	F	キネマティック結合、プロテクションゾーン、プロテクションゾーン要素、干渉ペア、座標変換データからの要素の削除		+		PM-NC
DELT	P	工具の削除		+	-	FM-TM
DELTC	P	工具ホルダデータレコードを削除		+	-	FM-TM

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1)2)3)4)5) 詳細は、一覧 (ページ 1433)を参照してください。						
DELTOOLENV	F	工具環境を記述したデータの削除		+	-	PM-NC
DIACYCOFA	K	軸別モーダル直径指定:サイクル内はオフ	M	+		FM-A
DIAM90	G	G90 では直径指定、G91 では半径指定	M	+		PM-NC
DIAM90A	K	G90 と AC では軸別モーダル直径指定、G91 と IC では半径指定	M	+		PM-NC
DIAMCHAN	K	すべての軸をマシンデータの軸機能から直径/半径指定チャンネル状態へ移行		+		PM-NC
DIAMCHANA	K	直径/半径指定チャンネル状態へ移行		+		PM-NC
DIAMCYCOF	G	チャンネル別の直径指定:サイクル内はオフ	M	+		FM-A
DIAMOF ⁶⁾	G	直径指定:オフ 通常的位置指定、工作機械メーカーに問い合わせてください	M	+		PM-NC
DIAMOFA	K	軸別モーダル直径指定:オフ 通常的位置指定、工作機械メーカーに問い合わせてください	M	+		PM-NC
DIAMON	G	直径指定:ON	M	+		PM-NC
DIAMONA	K	軸別モーダル直径指定:オン 適用、工作機械メーカーに問い合わせてください	M	+		PM-NC
DIC	K	ノンモーダルのインクレメンタル指令の軸別直径指定	s	+		PM-NC
DILF	A	後退距離(長さ)	M	+		PM-NC
DISABLE	P	割り込みのオフ		+	-	PM-NC
DISC	A	工具径補正の挿入円のオーバーシュート	M	+		PM-NC
DISCL	A	高速切り込み移動の終点と加工平面の距離		+		PM-NC
DISPLOF	PA	実行中のブロック表示をマスクします		+		PM-NC
DISPLON	PA	実行中のブロック表示のマスクは無効です		+		PM-NC
DISPR	A	再位置決めの軌跡距離	s	+		PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1)2)3)4)5) 詳細は、一覧 (ページ 1433)を参照してください。						
DISR	A	再位置決めの距離	S	+		PM-NC
DISRP	A	滑らかなアプローチと後退時の、後退平面と加工平面との間の距離		+		PM-NC
DITE	A	ねじの切り上げの軌跡	M	+		PM-NC
DITS	A	ねじの切り始めの軌跡	M	+		PM-NC
DIV	K	整数の除算		+		PM-NC
DL	A	ロケーションに応じた追加工具補正の選択(DL、合計セットアップオフセット)	M	+		PM-NC
DO	K	シンクロナイズドアクション:条件が満たされる場合にアクションを起動する		-	+	FM-SA
DRFOF	P	ハンドルオフセット(DRF)の解除	M	+	-	PM-NC
DRIVE	G	速度に応じた軌跡加減速度	M	+		PM-NC
DRIVEA	P	プログラム指令軸で膝形加減速特性を起動します。		+	-	PM-NC
DRVPRD	P	ドライブパラメータを読み取ります。		+	-	PM-NC
DRVPCR	P	ドライブパラメータを書き込みます。		+	-	PM-NC
DYNFINISH	G	仕上げのダイナミック応答	M	+		PM-NC
DYNNORM ⁶⁾	G	標準ダイナミック応答	M	+		PM-NC
DYNPOS	G	位置決めモードのダイナミック応答、タッピング	M	+		PM-NC
DYNPREC	G	滑らかな仕上げのダイナミック応答	M	+		PM-NC
DYNROUGH	G	荒削りのダイナミック応答	M	+		PM-NC
DYNSEMIFIN	G	荒仕上げのダイナミック応答	M	+		PM-NC
DZERO	P	工具オフセットユニットのすべての D 番号を無効としてマーク		+	-	PM-NC
EAUTO	G	最後の 3 点による最後のスプライン区間の定義	M	+		PM-NC
EGDEF	P	電子ギヤの定義		+	-	PM-NC
EGDEL	P	スレーブ軸の連結定義を解除します		+	-	PM-NC
EGOFC	P	電子ギヤの一括解除		+	-	PM-NC
EGOFS	P	電子ギヤの選択解除		+	-	PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1)2)3)4)5) 詳細は、一覧 (ページ 1433)を参照してください。						
EGON	P	電子ギヤの起動		+	-	PM-NC
EGONSYN	P	電子ギヤの起動		+	-	PM-NC
EGONSYNE	P	アプローチモードを指定して電子ギヤを起動		+	-	PM-NC
ELSE	K	NC プログラム:IF 条件を満たさない場合のプログラムの分岐		+	-	PM-NC
ELSE	K	シンクロナイズドアクション:条件が満たされなかった場合にアクションを起動する		-	+	FM-SA
ENABLE	P	割り込みのオン		+	-	PM-NC
ENAT ⁶⁾	G	次の移動ブロックへの自然遷移	M	+		PM-NC
ENDFOR	K	FOR カウンタループの終了行		+		PM-NC
ENDIF	K	IF 分岐の終了行		+		PM-NC
ENDLABEL	K	REPEAT によるパートプログラムの繰り返し終了ラベル		+		PM-NC、FM-B
ENDLOOP	K	プログラムの無限ループ LOOP の終了行		+		PM-NC
ENDPROC	K	開始行 PROC のあるプログラムの終了行		+		
ENDWHILE	K	WHILE ループの終了行		+		PM-NC
ESRR	P	ドライブのドライブ自律 ESR 後退のパラメータ設定		+		PM-NC
ESRS	P	ドライブのドライブ自律 ESR 停止のパラメータ設定		+		PM-NC
ETAN	G	スプライン開始位置での次の移動ブロックへの接線方向の遷移	M	+		PM-NC
EVERY	K	FALSE から TRUE への条件の遷移時にシンクロナイズドアクションを実行		-	+	FM-SA
EX	K	指数表記の値の割り当てのキーワード		+		PM-NC
EXECSTRING	P	実行するパートプログラム行を含む文字列変数の転送		+	-	PM-NC
EXECTAB	P	移動テーブルから要素を実行		+	-	PM-NC
EXECUTE	P	プログラム実行のオン		+	-	PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1)2)3)4)5) 詳細は、一覧 (ページ 1433)を参照してください。						
EXP	F	指数関数 ex		+	+	PM-NC
EXTCALL	A	外部サブプログラムの実行		+	+	PM-NC
EXTCLOSE	P	書き込みのために開かれていた外部機器/ファイルを閉じる		+	-	PM-NC
EXTERN	K	パラメータ転送を含むサブプログラムの宣言		+		PM-NC
EXTOPEN	P	チャンネルの外部機器/ファイルを書き込みのために開く		+	-	PM-NC
F	A	送り速度値 (G4 と組み合わせて、ドウェル時間も F でプログラム指令します)		+	+	PM-NC
FA	K	軸の送り速度	M	+	+	PM-NC
FAD	A	滑らかなアプローチと後退の送り速度		+		PM-NC
FALSE	K	論理定数:不正		+	+	PM-NC
FB	A	ノンモーダル送り速度		+		PM-NC
FCTDEF	P	多項式関数の定義		+	-	PM-NC
FCUB	G	3 次スプラインによる可変送り速度	M	+		PM-NC
FD	A	ハンドルオーバーライドの軌跡送り速度	S	+		PM-NC
FDA	K	ハンドルオーバーライドの軸送り速度	S	+		PM-NC
FENDNORM ⁶⁾	G	コーナー減速のオフ	M	+		PM-NC
FFWOF ⁶⁾	G	フィードフォワード制御のオフ	M	+		PM-NC
FFWON	G	フィードフォワード制御のオン	M	+		PM-NC
FGREF	K	回転軸の基準半径または旋回軸の軌跡基準係数(ベクトル補間)	M	+		PM-NC
FGROUP	P	軌跡送り速度を使用する単数/複数の軸の定義		+	-	PM-NC
FI	K	フレームデータにアクセスするためのパラメータ:仕上げオフセット		+		PM-NC
FIFOCTRL	G	先読みバッファの制御	M	+		PM-NC
FILEDATE	P	ファイルへの最後の書き込みアクセスの日付を返します		+	-	PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1)2)3)4)5) 詳細は、一覧 (ページ 1433)を参照してください。						
FILEINFO	P	FILEDATE、FILESIZE、FILESTAT、および FILETIME を含む要約情報を返します		+	-	PM-NC
FILESIZE	P	現在のファイルサイズを返します		+	-	PM-NC
FILESTAT	P	読み出し、書き込み、実行、表示、削除 (rwxs)の権限に関するファイル状態を返します		+	-	PM-NC
FILETIME	P	ファイルへの最後の書き込みアクセスの時刻を返します		+	-	PM-NC
FINEA	K	「精密イグザクトストップ」へ到達時に移動終了	M	+		PM-NC
FL	K	同期軸の速度制限	M	+		PM-NC
FLIM	A	最大軌跡速度の設定	M	+	-	PM-NC
FLIN	G	可変毎分送り速度	M	+		PM-NC
FMA	K	複数の軸送り速度	M	+		PM-NC
FNORM ⁶⁾	G	DIN 66025 に準拠した標準送り速度	M	+		PM-NC
FOC	K	ノンモーダルトルク/推力の制限	s	-	+	FM-SA
FOCOF	K	モーダルトルク/推力制限のスイッチオフ	M	-	+	FM-SA
FOCON	K	モーダルトルク/推力制限のオン	M	-	+	FM-SA
FOR	K	実行回数が一定のカウンタループ		+		PM-NC
FP	A	固定点:アプローチする固定点の番号	s	+		PM-NC
FPO	K	多項式でプログラム指令した送り速度特性		+		PM-NC
FPR	P	回転軸識別子		+	-	PM-NC
FPRAOF	P	毎回転送り速度の解除		+	-	PM-NC
FPRAON	P	毎回転送り速度の起動		+	-	PM-NC
FRAME	K	座標系の定義のデータタイプ		+		PM-NC
FRC	A	丸み付けと面取りの送り速度	s	+		PM-NC
FRCM	A	丸み付けと面取りの送り速度、モーダル	M	+		PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1)2)3)4)5) 詳細は、一覧(ページ 1433)を参照してください。						
FROM	K	条件が一度満たされると、シンクロナイズドアクションが動作中であるかぎり、動作を実行します		-	+	FM-SA
FTOC	P	精密工具補正の変更		-	+	FM-SA
FTOCOF ⁶⁾	G	オンライン精密工具補正のオフ	M	+		PM-NC
FTOCON	G	オンライン精密工具補正のオン	M	+		PM-NC
FXS	K	突き当て点停止のオン	M	+	+	PM-NC
FXST	K	突き当て点停止のトルク制限	M	+	+	PM-NC
FXSW	K	突き当て点停止の監視範囲		+	+	PM-NC
FZ	K	1 刃当たり送り速度	M	+		PM-NC

命令 G ... L

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
G0	G	早送りによる直線補間(早送り移動)	M	+		PM-NC
G1 ⁶⁾	G	送り速度による直線補間(直線補間)	M	+		PM-NC
G2	G	右回りの円弧補間	M	+		PM-NC
G3	G	左回りの円弧補間	M	+		PM-NC
G4	G	ドウェル時間、事前設定	s	+		PM-NC
G5	G	傾斜プランジ研削	s	+		PM-NC
G7	G	傾斜プランジ研削時の補正動作	s	+		PM-NC
G9	G	イグザクトストップ - 減速	s	+		PM-NC
G17 ⁶⁾	G	作業平面 X/Y の選択	M	+		PM-NC
G18	G	作業平面 Z/X の選択	M	+		PM-NC
G19	G	作業平面 Y/Z の選択	M	+		PM-NC
G25	G	ワーキングエリアリミットの下限	s	+		PM-NC
G26	G	ワーキングエリアリミットの上限	s	+		PM-NC
G33	G	固定リードのねじ切り	M	+		PM-NC
G34	G	単調増加可変リードねじ切り	M	+		PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
G35	G	単調減少可変リードねじ切り	M	+		PM-NC
G40 ⁶⁾	G	工具径補正のオフ	M	+		PM-NC
G41	G	輪郭の左側の工具径補正	M	+		PM-NC
G42	G	輪郭の右側の工具径補正	M	+		PM-NC
G53	G	現在のゼロオフセットのマスク(ノンモーダル)	s	+		PM-NC
G54	G	1 番目の設定可能ゼロオフセット	M	+		PM-NC
G55	G	2 番目の設定可能ゼロオフセット	M	+		PM-NC
G56	G	3 番目の設定可能ゼロオフセット	M	+		PM-NC
G57	G	4 番目の設定可能ゼロオフセット	M	+		PM-NC
G58 (840D sl)	G	アブソリュートプログラマブルゼロオフセット(荒削りオフセット)	s	+		PM-NC
G58 (828D)	G	5 番目の設定可能ゼロオフセット	M	+		PM-NC
G59 (840D sl)	G	追加のプログラマブルゼロオフセット(仕上げオフセット)	s	+		PM-NC
G59 (828D)	G	6 番目の設定可能ゼロオフセット	M	+		PM-NC
G60 ⁶⁾	G	イグザクトストップ - 減速	M	+		PM-NC
G62	G	工具径補正が有効なときの内側コーナのコーナー減速(G41、G42)	M	+		PM-NC
G63	G	フローティングチャックを使用したタッピング	s	+		PM-NC
G64	G	連続軌跡モード	M	+		PM-NC
G70	G	ジオメトリ指定のインチ寸法(長さ)	M	+	+	PM-NC
G71 ⁶⁾	G	ジオメトリ指定のメトリック寸法(長さ)	M	+	+	PM-NC
G74	G	レファレンス点復帰	s	+		PM-NC
G75	G	固定点アプローチ	s	+		PM-NC
G90 ⁶⁾	G	アブソリュート指令	m/s	+		PM-NC
G91	G	インクレメンタル指令	m/s	+		PM-NC
G93	G	インバースタイム送り(rpm)	M	+		PM-NC
G94 ⁶⁾	G	毎分送り速度 F(mm/min または inch/min、および° /min 単位)	M	+		PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
G95	G	毎回転送り速度 F(mm/rev または inch/rev 単位)	M	+		PM-NC
G96	G	毎回転送り速度(G95 に対して)および収束一定制御	M	+		PM-NC
G97	G	毎回転送り速度および一定主軸速度(周速一定制御オフ)	M	+		PM-NC
G110	G	最後のプログラム指令位置に対する極のプログラミング	S	+		PM-NC
G111	G	現在のワーク座標系の原点に対する極のプログラミング	S	+		PM-NC
G112	G	最後に有効な極に対する極のプログラミング	S	+		PM-NC
G140 ⁶⁾	G	G41/G42 で定義した SAR アプローチ方向	M	+		PM-NC
G141	G	輪郭の左側への SAR アプローチ方向	M	+		PM-NC
G142	G	輪郭の右側への SAR アプローチ方向	M	+		PM-NC
G143	G	接線に応じて SAR アプローチ方向を決定	M	+		PM-NC
G147	G	直線による滑らかなアプローチ	S	+		PM-NC
G148	G	直線による滑らかな後退	S	+		PM-NC
G153	G	基本フレームを含む現在のフレームのマスク	S	+		PM-NC
G247	G	4 分円による滑らかなアプローチ	S	+		PM-NC
G248	G	4 分円による滑らかな後退	S	+		PM-NC
G290 ⁶⁾	G	SINUMERIK モードへの切り替えのオン	M	+		FM-TM
G291	G	ISO2/3 モードへの切り替えのオン	M	+		FM-TM
G331	G	リジッドタッピング、正符号のリード、右回り	M	+		PM-NC
G332	G	リジッドタッピング、負符号のリード、左回り	M	+		PM-NC
G335	G	右回りの凸型ねじの旋削	M	+		PM-NC
G336	G	左回りでの凸ねじの旋削	M	+		PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
G340 ⁶⁾	G	空間アプローチブロック(平面上と奥行きへ同時に移動(ヘリカル))	M	+		PM-NC
G341	G	最初に垂直軸(z)の切り込み、その後平面内にアプローチ	M	+		PM-NC
G347	G	半円による滑らかなアプローチ	s	+		PM-NC
G348	G	半円による滑らかな後退	s	+		PM-NC
G450 ⁶⁾	G	遷移円	M	+		PM-NC
G451	G	等距離の交点	M	+		PM-NC
G460 ⁶⁾	G	アプローチと後退ブロックの干渉検出の適用	M	+		PM-NC
G461	G	TRC ブロックへの円弧の挿入	M	+		PM-NC
G462	G	TRC ブロックへの直線の挿入	M	+		PM-NC
G500 ⁶⁾	G	すべての設定可能フレームの解除、基本フレームは有効です	M	+		PM-NC
G505 ... G599	G	5 ... 99 番目の設定可能なゼロオフセット	M	+		PM-NC
G601 ⁶⁾	G	精密イグザクトストップでブロック切り替え	M	+		PM-NC
G602	G	汎用イグザクトストップでブロック切り替え	M	+		PM-NC
G603	G	IPO のブロック終点でブロック切り替え	M	+		PM-NC
G621	G	すべてのコーナーのコーナー減速	M	+		PM-NC
G641	G	距離条件に応じたスムージングによる連続軌跡モード(=プログラム指令可能な丸み付け隙間)	M	+		PM-NC
G642	G	定義した許容範囲内のスムージングによる連続軌跡モード	M	+		PM-NC
G643	G	定義した許容範囲内のスムージングによる連続軌跡モード(ブロック内部)	M	+		PM-NC
G644	G	最大ダイナミック応答によるスムージングによる連続軌跡モード	M	+		PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
G645	G	スムージング、および定義許容範囲内で 接線方向のブロック遷移をおこなう連続 軌跡モード	M	+		PM-NC
G700	G	ジオメトリ仕様と加工仕様(長さ、送り 速度) のインチ寸法	M	+	+	PM-NC
G710 ⁶⁾	G	ジオメトリ仕様と加工仕様(長さ、送り 速度) のメトリック寸法	M	+	+	PM-NC
G810 ⁶⁾ 、...、 G819	G	OEM ユーザー用に予約された G グループ		+		PM-NC
G820 ⁶⁾ 、...、 G829	G	OEM ユーザー用に予約された G グループ		+		PM-NC
G931	G	移動時間により指定された送り速度、一 定軌跡速度を解除	M	+		
G942	G	毎分送り速度と周速一定制御、または主 軸速度の解除	M	+		
G952	G	毎回転送り速度と周速一定制御、または 主軸速度の解除	M	+		
G961	G	毎分送り速度(G94 に対して)および周速 一定制御	M	+		PM-NC
G962	G	毎分送り速度と毎回転送り速度、および 周速一定制御	M	+		PM-NC
G971	G	毎分送り速度および一定主軸速度(周速 一定制御オフ)	M	+		PM-NC
G972	G	毎分送り速度または毎回転送り速度およ び一定主軸速度(周速一定制御オフ)	M	+		PM-NC
G973	G	主軸速度制限のない毎回転送り速度およ び一定主軸速度(ISO モードでは、LIMS なしの G97)	M	+		PM-NC
GEOAX	P	ジオメトリ軸 1～3 に新しいチャネル軸 を割り当て		+	-	PM-NC
GET	P	チャネル間で有効な軸の入れ替え		+	+	PM-NC
GETACTT	F	同じ名称の工具グループから、動作中の 工具を取得		+	-	FM-TM

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
GETACTTD	F	絶対 D 番号に関連する T 番号を取得		+	-	PM-NC
GETD	P	チャンネル間で軸の直接入れ替え		+	-	PM-NC
GETDNO	F	工具(T)の刃先(CE)の D 番号を返します		+	-	PM-NC
GETEXET	P	ロードした T 番号の読み出し		+	-	FM-TM
GETFREELOC	P	当該の工具用のマガジンの空きスペースを検出		+	-	FM-TM
GETSELT	P	選択した T 番号を返します		+	-	FM-TM
GETT	F	工具名称に対応する T 番号を取得		+	-	FM-TM
GETTCOR	F	工具長または工具長成分、またはその両方の読み出し		+	-	PM-NC
GETTENV	F	T 番号、D 番号、および DL 番号の読み出し		+	-	PM-NC
GETVARAP	F	システム/ユーザー変数への読み取りアクセス権		+	-	PM-NC
GETVARDFT	F	システム/ユーザー変数の読み取り初期値		+	-	PM-NC
GETVARLIM	F	システム/ユーザー変数の読み取り制限値		+	-	PM-NC
GETVARPHU	F	システム/ユーザー変数の読み取り物理単位		+	-	PM-NC
GETVARTYP	F	システム/ユーザー変数の読み取りデータタイプ		+	-	PM-NC
GFRAME0 ... GFRAME100	G	チャンネル内のデータ管理の研削フレーム <n>の起動	M	+		PM-NC
GOTO	K	最初に前方に、次に後方に(最初にプログラム末尾方向、次にプログラム先頭方向に)ジャンプする命令		+		PM-NC
GOTOB	K	後方(プログラム先頭方向)にジャンプ		+		PM-NC
GOTOC	K	GOTO と同じ。ただし、アラーム 14080 「Jump destination not found」(ジャンプ先が見つかりません)はマスクされます		+		PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
GOTOF	K	前方(プログラム末尾方向)にジャンプ		+		PM-NC
GOTOS	K	プログラム先頭へジャンプ		+		PM-NC
BP	K	位置属性の間接プログラミング用キーワード		+		PM-NC
GROUP_ADDEND	C (T)	試験切削追加の終了		+		PM-NC
GROUP_BEGIN	C (T)	プログラムグループの先頭		+		PM-NC
GROUP_END	C (T)	プログラムグループの終了		+		PM-NC
GWPSOF	P	砥石周速度一定(GWPS)の選択解除	S	+	-	PM-NC
GWPSON	P	砥石周速度一定制御(GWPS)の選択	S	+	-	PM-NC
H...	A	PLC への補助機能出力		+	+	PM-NC、FM-B
HOLES1	C (T)	列並びの穴		+		PM-NC
HOLES2	C (T)	円弧並びの穴		+		PM-NC
I	A	補間パラメータ	S	+		PM-NC
I1	A	中間点座標	S	+		PM-NC
IC	K	インクレメンタル指令	S	+		PM-NC
ICYCOF	P	テクノロジーサイクルのすべてのブロックを、ICYCOF 指令後に 1 つの補間サイクルで処理します		+	+	FM-SA
ICYCON	P	テクノロジーサイクルの各ブロックを、ICYCON 指令後に個別の補間サイクルで処理します。		+	+	FM-SA
ID	K	モーダルシンクロナイズドアクションの識別子	M	-	+	FM-SA
IDS	K	モーダル静的シンクロナイズドアクションの識別子		-	+	FM-SA
IF	K	パートプログラム/テクノロジーサイクルの条件分岐の開始		+	+	PM-NC
INDEX	F	入力文字列の文字のインデックスを定義		+	-	PM-NC
INICF	K	NEWCONF 変数の初期化		+		PM-NC
INIPO	K	電源投入時の変数の初期化		+		PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
INIRE	K	リセット時の変数の初期化		+		PM-NC
INIT	P	特定のチャンネルで実行する特定の NC プログラムの選択		+	-	PM-NC
INITIAL		すべての領域にわたる INI ファイルの生成		+		PM-NC
INT	K	データタイプ:符号付き整数		+		PM-NC
INTERSEC	F	2 個の輪郭要素の交点を計算		+	-	PM-NC
INVCCW	G	インボリュート曲線の起動、左回り	M	+		PM-NC
INVCW	G	インボリュート曲線の起動、右回り	M	+		PM-NC
INVFRAME	F	フレームから逆フレームを計算		+	-	FM-B
IP	K	可変補間パラメータ		+		PM-NC
IPOBRKA	P	減速カーブ適用による移動条件	M	+	+	
IPOENDA	K	「補間停止」へ到達時に移動終了	M	+		PM-NC
IPTRLOCK	P	次の運転機能ブロックで検索不可プログラム区間の開始をおこないます。	M	+	-	PM-NC
IPTRUNLOCK	P	中断時に、実行中のブロックで検索不可プログラム区間の終了を設定します。	M	+	-	PM-NC
IR	A	凸ねじを旋削する際の円弧中心点の座標 (X 軸)		+		PM-NC
ISAXIS	F	ジオメトリ軸 1 がパラメータとして指定されているかどうかをチェック		+	-	PM-NC
ISD	A	切り込み深さ	M	+		PM-NC
ISFILE	F	NC アプリケーションメモリにファイルがあるかどうかをチェック		+	-	PM-NC
ISNUMBER	F	入力文字列を数字に変換できるかどうかをチェック		+	-	PM-NC
ISOCALL	K	ISO 言語でプログラム指令したプログラムの間接呼び出し		+		PM-NC
ISVAR	F	NC で宣言された変数が転送パラメータに含まれるかどうかをチェック		+	-	PM-NC
J	A	補間パラメータ	S	+		PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
J1	A	中間点座標	S	+		PM-NC
JERKA	P	プログラム指令軸に対して、マシンデータで設定した加減速動作を起動		+	-	
JERKLIM	K	最大軸加々速度の調整	M	+		PM-NC
JERKLIMA	K	最大スレーブ軸加々速度の低減、または増加	M	+	+	PM-NC
JR	A	凸ねじを旋削する際の円弧中心点の座標 (Y 軸)		+		PM-NC
K	A	補間パラメータ	S	+		PM-NC
K1	A	中間点座標	S	+		PM-NC
KONT	G	工具補正時の、輪郭周りの移動	M	+		PM-NC
KONTC	G	曲率の変化が連続的な多項式によるアプローチ/後退	M	+		PM-NC
KONTT	G	連続の接線の多項式によるアプローチ/後退	M	+		PM-NC
KR	A	凸ねじを旋削する際の円弧中心点の座標 (Z 軸)		+		PM-NC
L	A	サブプログラム番号	S	+	+	PM-NC
LEAD	A	リード角 1 番目の基本工具の向き 2 番目のオリエンテーション多項式	M	+		PM-NC
LEADOF	P	軸間連動機能のオフ		+	+	PM-NC
LEADON	P	軸間連動機能のオン		+	+	PM-NC
LENTOAX	F	動作中の工具の工具長 L1、L2、および L3 の、横座標、縦座標、および垂直座標への割り当てに関する情報を提供		+	-	PM-NC
LFOF ⁶⁾	G	ねじ切りの高速リトラクトのオフ	M	+		PM-NC
LFON	G	ねじ切りの高速リトラクトのオン	M	+		PM-NC
LFPOS	G	POLFMASK または POLFMLIN を使用して宣言した軸の、POLF でプログラム指令したアブソリュート軸位置への後退	M	+		PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
LFTXT ⁶⁾	G	高速リトラクトの後退移動の平面を、軌跡タンジェントと現在の工具方向から特定します	M	+		PM-NC
LFWP	G	高速リトラクトの後退移動の平面を現在の作業平面(G17/G18/G19)で特定します	M	+		PM-NC
LIFTFAST	K	高速リトラクト		+		PM-NC
LIMS	K	G96/G961 および G97 の速度制限	M	+		PM-NC
LLI	K	変数の下限値		+		PM-NC
LN	F	自然対数		+	+	PM-NC
LOCK	P	ID によりシンクロナイズドアクションを解除 (テクノロジーサイクルを停止)		-	+	FM-SA
LONGHOLE	C (T)	長穴		+		PM-NC
LOOP	K	無限ループの開始		+		PM-NC

命令 M ... R

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
M0		プログラムストップ		+	+	PM-NC
M1		オプションルストップ		+	+	PM-NC
M2		プログラムの終了、メインプログラム(M30 と同じ)		+	+	PM-NC
M3		CW 主軸回転		+	+	PM-NC
M4		CCW 主軸回転		+	+	PM-NC
M5		主軸停止		+	+	PM-NC
M6		工具交換		+	+	PM-NC
M17		サブプログラム終了		+	+	PM-NC
M19		主軸を SD43240 で入力された位置に位置決め		+	+	PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
M30		プログラムの終了、メインプログラム (M2 と同じ)		+	+	PM-NC
M40		自動ギヤ切り替え		+	+	PM-NC
M41 ... M45		ギヤ選択 1 ... 5		+	+	PM-NC
M70		軸モードへの移行		+	+	PM-NC
MASLDEF	P	マスタ/スレーブ軸グループの定義		+	+	PM-NC
MASLDEL	P	マスタ/スレーブ軸グループを連結解除し、グループ定義を解除		+	+	PM-NC
MASLOF	P	一時的に連結を解除		+	+	PM-NC
MASLOFS	P	自動スレーブ軸停止により、一時的に連結を解除		+	+	PM-NC
MASLON	P	一時的に連結を起動		+	+	PM-NC
MATCH	F	文字列中の文字列の検索		+	-	PM-NC
MAXVAL	F	2 個の変数の、大きい方の値 (算術機能)		+	+	PM-NC
MCALL	K	モータルサブプログラム呼び出し		+		PM-NC
MEAC	K	残移動距離削除なしの連続軸計測	S	+	+	PM-NC
MEAFRAME	F	計測点からのフレーム計算		+	-	PM-NC
MEAS	A	残移動距離を削除する計測	S	+		PM-NC
MEASA	K	残移動距離を削除する軸計測	S	+	+	PM-NC
MEASURE	F	ワーク計測と工具計測の計算方式		+	-	FM-TE
MEAW	A	残移動距離を削除しない計測	S	+		PM-NC
MEAWA	K	残移動距離を削除しない軸計測	S	+	+	PM-NC
Mi	K	フレームデータへのアクセス:ミラーリング		+		PM-NC
MINDEX	F	入力文字列の文字のインデックスを定義		+	-	PM-NC
MINVAL	F	2 個の変数の、小さい方の値 (算術機能)		+	+	PM-NC
MIRROR	G	プログラマブルミラーリング	S	+		PM-NC
MMC	P	パートプログラムから、対話用ウィンドウを対話形式で HMI 上に呼び出します		+	-	PM-NC
MOD	K	モジュロ除算		+		PM-NC
MODAXVAL	F	モジュロ回転軸のモジュロ位置を特定		+	-	PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
MOV	K	軸位置決めの起動		-	+	FM-SA
MOVT	A	工具方向への移動動作の終点の指定				FM-B
MSG	P	プログラマブルメッセージ	M	+	-	PM-NC
MVTOOL	P	工具を移動するための言語命令		+	-	FM-TM
N	A	NC 補助ブロック番号		+		PM-NC
NAMETOINT	F	システム変数インデックスの特定		+		PM-NC
NC	K	データの有効範囲を指定します		+		PM-NC
NEWCONF	P	変更したマシンデータを適用(「マシンデータの有効化」に相当します)		+	-	PM-NC
NEWMT	F	新しいマルチツールの作成		+	-	FM-TM
NEWT	F	新しい工具の作成		+	-	FM-TM
NORM ⁶⁾	G	工具補正の始点と終点の標準設定	M	+		PM-NC
NOT	K	論理否定		+		PM-NC
NPROT	P	機械別プロテクションゾーンのオン/オフ		+	-	PM-NC
NPROTDEF	P	機械別プロテクションゾーンの定義		+	-	PM-NC
NUMBER	F	入力文字列を数字に変換		+	-	PM-NC
OEMIPO1	G	OEM 補間 1	M	+		PM-NC
OEMIPO2	G	OEM 補間 2	M	+		PM-NC
OF	K	CASE 分岐のキーワード		+		PM-NC
OFFN	A	プログラム指令輪郭の許容量	M	+		PM-NC
OMA1	A	OEM アドレス 1	M	+		PM-NC
OMA2	A	OEM アドレス 2	M	+		PM-NC
OMA3	A	OEM アドレス 3	M	+		PM-NC
OMA4	A	OEM アドレス 4	M	+		PM-NC
OMA5	A	OEM アドレス 5	M	+		PM-NC
または	K	論理演算子、論理和		+		PM-NC
ORIXES	G	機械軸または回転軸の直線補間	M	+		PM-NC
ORIXPOS	G	仮想回転軸の回転軸位置による回転角度	M	+		PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
ORIC ⁶⁾	G	外側コーナーでの向きの変更を、挿入する円弧ブロックに重畳します	M	+		PM-NC
ORICONCCW	G	円錐面での CCW 方向の補間	M	+		PM-NC、FM-TR
ORICONCW	G	円錐面での CW 方向の補間	M	+		PM-NC、FM-TR
ORICONIO	G	中間旋回設定による円錐面の補間	M	+		PM-NC、FM-TR
ORICONTO	G	接線方向の遷移による円錐面の補間 (最終旋回)	M	+		PM-NC、FM-TR
ORICURVE	G	工具の 2 つの接点の移動を指定した旋回補間	M	+		PM-NC、FM-TR
ORID	G	円弧ブロックの前に向きの変更を実行します	M	+		PM-NC
ORIEULER ⁶⁾	G	オイラー角による旋回角度	M	+		PM-NC
ORIMKS	G	機械座標系の工具オリエンテーション	M	+		PM-NC
ORIPATH	G	軌跡に対する工具オリエンテーション	M	+		PM-NC
ORIPATHS	G	軌跡に対する工具オリエンテーション、旋回処理の不連続変化をスムージングします	M	+		PM-NC
ORIPANE	G	平面での補間 (ORIVECT に相当)、大半径円弧補間	M	+		PM-NC
ORIRESET	P	3 つまでの旋回軸による工具オリエンテーションの初期設定		+	-	PM-NC
ORIROTA ⁶⁾	G	アブソリュート回転方向への回転角度	M	+		PM-NC
ORIROTC	G	軌跡タンジェントに対する接線方向の回転ベクトル	M	+		PM-NC
ORIROTR	G	旋回の開始と終了間の平面に対する回転角度	M	+		PM-NC
ORIROTT	G	配向ベクトルの変更に対する回転角度	M	+		PM-NC
ORIRPY	G	RPY 角による旋回角度(XYZ)	M	+		PM-NC
ORIRPY2	G	RPY 角による旋回角度(ZYX)	M	+		PM-NC
ORIS	A	向きの変更	M	+		PM-NC
ORISOF ⁶⁾	G	旋回処理のスムージングのオフ	M	+		PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
ORISOLH	F	方向の計算		+		PM-NC
ORISON	G	旋回処理のスムージングのオン	M	+		PM-NC
ORIVECT ⁶⁾	G	大円弧補間(ORIPLANE と同じ)	M	+		PM-NC
ORIVIRT1	G	仮想旋回軸による旋回角度(定義 1)	M	+		PM-NC
ORIVIRT2	G	仮想旋回軸による旋回角度(定義 1)	M	+		PM-NC
ORIWKS ⁶⁾	G	ワーク座標系の工具オリエンテーション	M	+		PM-NC
OS	K	揺動のオン/オフ		+		PM-NC
OSB	K	揺動:起点	M	+		PM-NC
OSC	G	連続工具オリエンテーションのスムージング	M	+		PM-NC
OSCILL	K	Axis:1～3 つの切り込み軸	M	+		PM-NC
OSCTRL	K	揺動オプション	M	+		PM-NC
OSD	G	セッティングデータでスムージング距離が指定された工具オリエンテーションのスムージング	M	+		PM-NC
OSE	K	揺動の終点	M	+		PM-NC
OSNSC	K	揺動:スパークアウトサイクル数	M	+		PM-NC
OSOF ⁶⁾	G	工具オリエンテーションのスムージングのオフ	M	+		PM-NC
OSP1	K	揺動:左の反転点	M	+		PM-NC
OSP2	K	揺動:右の反転点	M	+		PM-NC
OSS	G	ブロック終点の工具オリエンテーションのスムージング	M	+		PM-NC
OSSE	G	ブロックの始点と終点の工具オリエンテーションのスムージング	M	+		PM-NC
OST	G	セッティングデータの角度許容範囲(°単位)の指定による工具オリエンテーションのスムージング(プログラム指令の旋回処理からの最大誤差)	M	+		PM-NC
OST1	K	揺動:左の反転点の停止点	M	+		PM-NC
OST2	K	揺動:右の反転点の停止点	M	+		PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
OTOL	A	コンプレッサ機能、旋回スミージング、およびスミージングタイプの旋回許容範囲	M	+	-	PM-NC
OTOLG0	A	早送り移動の向き許容範囲	M	+	-	PM-NC
OVR	K	速度オーバライド	M	+		PM-NC
OVRA	K	軸速度オーバライド	M	+	+	PM-NC
OVRRAP	K	早送りオーバライド	M	+		PM-NC
P	A	サブプログラム繰り返しの回数		+		PM-NC
PACCLIM	A	最大軌跡加速度の設定	M	+	-	PM-NC
PAROT	G	ワークにワーク座標系を配置	M	+		PM-NC
PAROTOF ⁶⁾	G	ワークに対するフレーム回転を解除	M	+		PM-NC
PCALL	K	絶対パスとパラメータ転送によるサブプログラム呼び出し		+		PM-NC
PDELAYOF	G	遅延のあるパンチングオフ	M	+		PM-NC
PDELAYON ⁶⁾	G	遅延のあるパンチングオン	M	+		PM-NC
PHI	K	円錐の方向軸を中心とした向きの回転角度		+		PM-NC
PHU	K	変数の物理単位		+		PM-NC
PL	A	1.B スプライン:ノードクリアランス 2.多項式補間 多項式補間のパラメータ区間の長さ	S	+		PM-NC
PM	K	毎分		+		PM-NC
PO	K	多項式補間の多項式係数	S	+		PM-NC
POCKET3	C (T)	長方形ポケットのフライス加工		+		PM-NC
POCKET4	C (T)	円形ポケットのフライス削り		+		PM-NC
POLF	K	高速リトラクトの後退位置	M	+		PM-NC
POLFA	P	\$AA_ESR_TRIGGER による単独軸の後退開始位置	M	+	+	PM-NC
POLFMASK	P	軸間の関係なしに個別に後退用の軸を有効化	M	+	-	PM-NC
POLFMLIN	P	軸間の直線補間で後退用の軸を有効化	M	+	-	PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
POLY	G	多項式補間	M	+		PM-NC
POLYPATH	P	AXIS と VECT の両方の軸グループに対して多項式補間を選択可能	M	+	-	PM-NC
PON	G	パンチングオン	M	+		PM-NC
PONS	G	補間サイクルのパンチングのオン	M	+		PM-NC
POS	K	軸の位置決め		+	+	PM-NC
POSA	K	ブロック境界を越える軸の位置決め		+	+	PM-NC
POSM	P	マガジンの位置決め		+	-	FM-TM
POSMT	P	工具ホルダのロケーション番号へのマルチツールの位置決め		+	-	FM-TM
POSP	K	複数区間の軸位置決め(揺動)		+		PM-NC
POSRANGE	F	現在補間している軸の位置指令が、事前定義された基準位置の範囲にあるかどうかを特定		+	+	FM-SA
POT	F	二乗 (算術機能)		+	+	PM-NC
PR	K	毎回転		+		PM-NC
PREPRO	PA	解析でのサブプログラムの識別		+		PM-NC
PRESETON	P	原点確立状態をリセットして現在値を設定		+	+	PM-NC
PRESETONS	P	原点確立状態をリセットせずに現在値を設定		+	+	PM-NC
PRIO	K	割り込み処理の優先度を設定するためのキーワード		+		PM-NC
PRLOC	K	ローカル変更後にのみ、リセット時の変数の初期化		+		PM-NC
PROC	K	プログラムの最初の命令		+		PM-NC
PROTA	P	干渉モデルの再計算の要求		+		PM-NC
PROTD	F	2つのプロテクションゾーンの間の距離の計算		+		PM-NC
PROTS	P	プロテクションゾーンステータスの設定		+		PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
psi	K	円錐の開口角度		+		PM-NC
PTP	G	ポイントツーポイント移動(PTP 移動)	M	+		PM-NC
PTPGO	G	G0 の場合はポイントツーポイント移動のみ、G0 以外の場合は軌跡移動の CP	M	+		PM-NC
PTPWOC	G	オリエンテーションの変更による補正移動なしのポイントツーポイント移動	M	+		PM-NC
PUNCHACC	P	ニブリングの、移動に応じた加減速度		+	-	PM-NC
PUTFTOC	P	並列目立ての工具仕上げオフセット		+	-	PM-NC
PUTFTOCF	P	FCTDEF で定義した並列目立て用機能に対応した工具仕上げオフセット		+	-	PM-NC
PW	A	B スプライン、点の重み	S	+		PM-NC
QU	K	高速追加 (補助)機能の出力		+		PM-NC
R...	A	設定可能アドレス識別子として、および数値拡張子付き算術変数		+		PM-NC
RAC	K	ノンモーダルのアブソリュート指令の軸別半径指定	S	+		PM-NC
RDISABLE	P	読み込み停止		-	+	FM-SA
READ	P	指定したファイルの複数の行を読み出し、読み出した情報を配列に格納		+	-	PM-NC
REAL	K	データタイプ:符号付きフローティングポイント変数(実数)		+		PM-NC
REDEF	K	システム変数の再定義、ユーザー変数、および NC 言語命令		+		PM-NC
RELEASE	P	軸入れ替えのために機械軸を解放		+	+	PM-NC
REP	K	配列のすべての要素を同じ値で初期化するためのキーワード		+		PM-NC
REPEAT	K	プログラムループの繰り返し		+		PM-NC
REPEATB	K	プログラム行の繰り返し		+		PM-NC
REPOSA	G	すべての軸の直線再位置決め	S	+		PM-NC
REPOSH	G	半円による再位置決め	S	+		PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
REPOSHA	G	全ジオメトリ軸の半円による再位置決め	s	+		PM-NC
REPOSL	G	直線再位置決め	s	+		PM-NC
REPOSQ	G	4 分円の再位置決め	s	+		PM-NC
REPOSQA	G	全ジオメトリ軸の 4 分円による再位置決め	s	+		PM-NC
RESETMON	P	指令値適用の言語命令		+	-	FM-TM
RET	P	サブプログラム終了		+	+	PM-NC
RETB	P	サブプログラム終了		+	+	PM-NC
RIC	K	ノンモーダルのインクレメンタル指令の軸別半径指定	s	+		PM-NC
RINDEX	F	入力文字列の文字のインデックスを定義		+	-	PM-NC
RMB	G	ブロックの始点への再位置決め	M	+		PM-NC
RMBBL	G	ブロックの始点への再位置決め	s	+		PM-NC
RME	G	ブロック終点への再位置決め	M	+		PM-NC
RMEBL	G	ブロック終点への再位置決め	s	+		PM-NC
RMI ⁶⁾	G	中断点への再位置決め	M	+		PM-NC
RMIBL ⁶⁾	G	中断点への再位置決め	s	+		PM-NC
RMN	G	最も近い軌跡点への再位置決め	M	+		PM-NC
RMNBL	G	最も近い軌跡点への再位置決め	s	+		PM-NC
RND	A	輪郭のコーナーの丸み付け	s	+		PM-NC
RNDM	A	モーダル丸み付け	M	+		PM-NC
ROT	G	プログラマブル座標回転	s	+		PM-NC
ROTS	G	立体角によるプログラマブルフレームの回転	s	+		PM-NC
ROUND	F	小数点以下の四捨五入		+	+	PM-NC
ROUNDUP	F	入力値の切り上げ		+	+	PM-NC
RP	A	極半径	m/s	+		PM-NC
RPL	A	平面の回転	s	+		PM-NC
RT	K	フレームデータにアクセスするためのパラメータ:回転		+		PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
RTLIOF	G	直線補間のない G0 (単独軸補間)	M	+		PM-NC
RTLION ⁶⁾	G	直線補間による G0	M	+		PM-NC

命令 S ... Z

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
S	A	主軸速度 (G4 の場合、G96/G961 の意味が異なります)	m/s	+/-	+/-	PM-NC
SAVE	PA	サブプログラムの呼び出し時の情報保存の属性		+/-		PM-NC
SBLOF	P	シングルブロックのマスク		+/-	-	PM-NC
SBLON	P	シングルブロックのマスクを無効化します		+/-	-	PM-NC
SC	K	フレームデータにアクセスするためのパラメータ: スケーリング		+/-		PM-NC
SCALE	G	プログラマブルスケーリング	s	+/-		PM-NC
SCC	K	G96/G961/G962 に対して径方向軸を選択して割り当て。軸識別子は、ジオメトリ軸、チャネル軸、または機械軸タイプの場合があります		+/-		PM-NC
SCPARA	K	サーボパラメータセットをプログラム指令		+/-	+/-	PM-NC
SD	A	スプラインの次数	s	+/-		PM-NC
SET	K	配列のすべての要素を、リスト値で初期化するためのキーワード		+/-		PM-NC
SETAL	P	アラームの設定		+/-	+/-	PM-NC
SETDNO	F	工具(T)の刃先(CE)の D 番号の割り当て		+/-	-	PM-NC
SETINT	K	NC 入力がある場合に、どの割り込みルーチンを起動するかを定義		+/-		PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
SETM	P	専用チャネルのマークの設定		+/-	+/-	PM-NC
SETMS	P	マシンデータで定義したメイン主軸に設定		+/-	-	PM-NC
SETMS(n)	P	主軸 n をメイン主軸として設定		+/-		PM-NC
SETMTH	P	メイン工具ホルダ番号を設定		+/-	-	FM-TM
SETPIECE	P	主軸に割り当てられたすべての工具に部品番号を設定		+/-	-	FM-TM
SETTA	P	摩耗グループから工具を起動		+/-	-	FM-TM
SETTCOR	F	すべての補足条件を考慮して、工具成分を変更		+/-	-	PM-NC
SETTIA	P	摩耗グループから工具を解除します		+/-	-	FM-TM
SF	A	ねじ切りの始点オフセット	M	+/-		PM-NC
SIN	F	正弦(三角関数)		+/-	+/-	PM-NC
SIRELAY	F	SIRELIN、SIRELOUT、および SIRELTIME でパラメータ設定した安全機能の起動		-	+/-	FM-SI
SIRELIN	P	ファンクションブロックの入力変数を初期化		+/-	-	FM-SI
SIRELOUT	P	ファンクションブロックの出力変数を初期化		+/-	-	FM-SI
SIRELTIME	P	ファンクションブロックのタイマーを初期化		+/-	-	FM-SI
SLOT1	C (T)	直線溝		+/-		PM-NC
SLOT2	C (T)	円周溝		+/-		PM-NC
SOFT	G	加々速度一定加減速の軌跡加減速度	M	+/-		PM-NC
SOFTA	P	プログラム指令軸で加々速度制限付きの軸加減速を起動します。		+/-	-	PM-NC
SON	G	ニブリングのオン	M	+/-		PM-NC
SONS	G	補間サイクルのニブリングのオン	M	+/-		PM-NC
SPATH ⁶⁾	G	FGROUP 軸の軌跡基準は円弧長です	M	+/-		PM-NC
SPCOF	P	メイン主軸または主軸を位置制御から速度制御に切り替え	M	+/-	-	PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
SPCON	P	メイン主軸または主軸を速度制御から位置制御に切り替え	M	+/-	-	PM-NC
SPI	F	主軸番号を軸識別子に変換		+/-	-	PM-NC
SPIF1 ⁶⁾	G	パンチング/ニブリングのバイト 1 に対する 高速 NC 入力/出力	M	+/-		FM-TE
SPIF2	G	パンチング/ニブリングのバイト 2 に対する 高速 NC 入力/出力	M	+/-		FM-TE
SPLINEPATH	P	スプライングループの定義		+/-	-	PM-NC
SPN	A	ブロックごとの軌跡区間数	s	+/-		PM-NC
SPOF ⁶⁾	G	ストロークのオフ、 ニブリング、パンチングのオフ	M	+/-		PM-NC
SPOS	K	主軸位置決め	M	+/-	+/-	PM-NC
SPOSA	K	ブロック境界を越える主軸位置決め	M	+/-		PM-NC
SPP	A	軌跡区間の長さ	M	+/-		PM-NC
SPRINT	F	フォーマットされた入力文字列を返します。		+/-		PM-NC
SQRT	F	平方根 (算術機能)		+/-	+/-	PM-NC
SR	A	シンクロナイズドアクションの揺動後退軌跡	s	+/-		PM-NC
SRA	K	シンクロナイズドアクション用の軸の外部入力による揺動後退軌跡	M	+/-		PM-NC
ST	A	シンクロナイズドアクションの揺動スパークアウト時間	s	+/-		PM-NC
STA	K	シンクロナイズドアクションの軸の揺動スパークアウト時間	M	+/-		PM-NC
START	P	複数のチャンネルで、実行中のプログラムから選択プログラムを同時に起動します		+/-	-	PM-NC
STARTFIFO ⁶⁾	G	実行; 全解析メモリを同時にすべて使用	M	+/-		PM-NC
STAT		関節継手の位置	s	+/-		PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
STOLF	A	G0 許容範囲係数	M	+/-		PM-NC
STOPFIFO	G	加工の停止; STARTFIFO を検出するか、解析メモリがいっぱいになるか、プログラムが終了するまで、解析メモリを使用します。	M	+/-		PM-NC
STOPRE	P	解析されたすべてのブロックがメインランで実行されるまで先読み停止		+/-	-	PM-NC
STOPREOF	P	先読み停止の無効化		-	+/-	FM-SA
STRING	K	データタイプ:文字列		+/-		PM-NC
STRINGIS	F	NC 言語と NC サイクル名称の現在の適用範囲、ユーザー変数、マクロ、およびこの命令に特有のラベル名称が存在しているか、有効か、定義済みか、または動作中かをチェックします。		+/-	-	PM-NC
STRLEN	F	文字列長の定義		+/-	-	PM-NC
SUBSTR	F	入力文字列の文字のインデックスを定義		+/-	-	PM-NC
SUPA	G	現在のワークオフセット(プログラム指令オフセット、システムフレーム、ハンドルオフセット(DRF)、外部ゼロオフセット、および重畳移動を含む)のマスク	s	+/-		PM-NC
SUPD	G	有効な工具オフセットのマスク	s	+/-	-	PM-NC
SVC	K	工具切削速度	M	+/-		PM-NC
SYNFCT	P	シンクロナイズドアクションの条件に応じて多項式を評価		-	+/-	FM-SA
SYNR	K	実行タイミングに同期して変数を読み出します		+/-		PM-NC
SYNRW	K	実行タイミング同期して変数の読み出し、および書き込みをおこないます		+/-		PM-NC
SYNW	K	実行タイミングに同期して変数を書き込みます		+/-		PM-NC
t	A	工具の呼び出し (マシンデータで指定した場合にのみ交換; 指定しない場合は、M6 命令が必要)		+/-		PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
TAN	F	タンジェント(三角関数)		+/-	+/-	PM-NC
TANG	P	法線方向制御:連結の定義		+/-	-	PM-NC
TANGDEL	P	法線方向制御:連結の解除		+/-	-	PM-NC
TANGOF	P	法線方向制御:連結の解除		+/-	-	PM-NC
TANGON	P	法線方向制御:連結の有効化		+/-	-	PM-NC
TCA (828D: _TCA)	P	工具状態を問わない工具選択/工具交換		+/-	-	FM-TM
TCARR	A	工具ホルダ(番号「m」)の要求		+/-		PM-NC
TCI	P	工具を工具バッファからマガジンへロード		+/-	-	FM-TM
TCOABS ⁶⁾	G	現在の工具オリエンテーションから工具長成分を特定	M	+/-		PM-NC
TCOFR	G	動作中のフレームの向きから工具長成分を特定	M	+/-		PM-NC
TCOFRX	G	工具の選択時に、動作中のフレームの工具オリエンテーションを特定、X 方向の工具点	M	+/-		PM-NC
TCOFRY	G	工具の選択時に、動作中のフレームの工具オリエンテーションを特定、Y 方向の工具点	M	+/-		PM-NC
TCOFRZ	G	工具の選択時に、動作中のフレームの工具オリエンテーションを特定、Z 方向の工具点	M	+/-		PM-NC
THETA	A	回転角度	s	+/-		PM-NC
TILT	A	傾斜角	M	+/-		PM-NC
TLIFT	P	法線方向制御:挿入ブロック生成の有効化		+/-	-	PM-NC
TML	P	マガジンロケーション番号による工具の選択		+/-	-	FM-TM
TMOF	P	工具監視の解除		+/-	-	PM-NC
TMON	P	工具監視の起動		+/-	-	PM-NC
TO	K	FOR カウンタループの終了値を指定		+/-		PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
TOFF	A	インデックスで指定されたジオメトリ軸に平行な、有効工具長成分の方向への工具長補正	M	+/-		PM-NC
TOFFL	A	工具長成分 L1、L2、または L3 の方向への工具長補正	M	+/-		PM-NC
TOFFLR	A	同時の工具長補正と工具半径補正	M	+/-		PM-NC
TOFFOF	P	オンライン工具長補正の解除		+/-	-	PM-NC
TOFFON	P	オンライン工具長補正の起動		+/-	-	PM-NC
TOFFR	A	工具径補正	M	+/-		PM-NC
TOFRAME	G	フレームを回転して、WCS の Z 軸を工具オリエンテーションに平行に配置します	M	+/-		PM-NC
TOFRAMEX	G	フレームを回転して、WCS の X 軸を工具オリエンテーションに平行に配置します	M	+/-		PM-NC
TOFRAMEY	G	フレームを回転して、WCS の Y 軸を工具オリエンテーションに平行に配置します	M	+/-		PM-NC
TOFRAMEZ	G	TOFRAME と同じです	M	+/-		PM-NC
TOLOWER	F	文字列中の文字を小文字に変換		+/-	-	PM-NC
TOOLENV	F	メモリに格納された工具データの評価内容について現在の状態を保存		+/-	-	PM-NC
TOOLGNT	F	工具グループの工具数の決定		+/-	-	FM-TM
TOOLGT	F	工具グループの工具の T 番号を決定		+/-	-	FM-TM
TOROT	G	フレームを回転して、WCS の Z 軸を工具オリエンテーションに平行に配置します	M	+/-		PM-NC
TOROTOF ⁶⁾	G	工具方向のフレーム回転のオフ	M	+/-		PM-NC
TOROTX	G	フレームを回転して、WCS の X 軸を工具オリエンテーションに平行に配置します	M	+/-		PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
TOROTY	G	フレームを回転して、WCS の Y 軸を工具オリエンテーションに平行に配置します	M	+/-		PM-NC
TOROTZ	G	TOROT と同じです	M	+/-		PM-NC
TOUPPER	F	文字列中の文字を大文字に変換		+/-	-	PM-NC
TOWBCS	G	基本座標系(BCS)の摩耗値	M	+/-		PM-NC
TOWKCS	G	キネマティックトランスフォーメーションをおこなう工具ヘッドの座標系の摩耗値(工具回転による機械座標系とは異なります)	M	+/-		PM-NC
TOWMCS	G	機械座標系(WCS)の摩耗値	M	+/-		PM-NC
TOWSTD ⁶⁾	G	工具長補正の初期設定値	M	+/-		PM-NC
TOWTCS	G	工具座標系の摩耗値(工具ホルダの工具ホルダ基準点)	M	+/-		PM-NC
TOWWCS	G	ワーク座標系(WCS)内の摩耗値	M	+/-		PM-NC
tr	K	フレーム変数のオフセット成分		+/-		PM-NC
TRAANG	P	傾斜軸座標変換		+/-	-	PM-NC
TRACON	P	座標変換重畳		+/-	-	PM-NC
TRACYL	P	円筒:円筒補間		+/-	-	PM-NC
TRAFOOF	P	チャンネルで動作中の座標変換を解除		+/-	-	PM-NC
TRAFOON	P	キネティックチェーンによって定義された座標変換を有効化		+/-	-	PM-NC
TRAILOF	P	非同期連結移動のオフ		+/-	+/-	PM-NC
TRAILON	P	非同期連結移動のオン		+/-	+/-	PM-NC
TRANS	G	アブソリュートプログラマブルゼロオフセット	s	+/-		PM-NC
TRANSMIT	P	極座標変換(正面加工)		+/-	-	PM-NC
TRAORI	P	4 軸座標変換、5 軸座標変換、汎用座標変換		+/-	-	PM-NC
TRUE	K	論理定数:TRUE		+/-		PM-NC
TRUNC	F	小数点以下の切り捨て		+/-	+/-	PM-NC

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧(ページ 1433)を参照してください。						
TU		軸角度	S	+/-		PM-NC
TURN	A	ヘリカルのターン数	S	+/-		PM-NC
ULI	K	変数の上限値		+/-		PM-NC
UNLOCK	P	ID によるシンクロナイズドアクションが有効(テクノロジーサイクルを続行)		-	+/-	FM-SA
UNTIL	K	REPEAT ループ終了の条件		+/-		PM-NC
UPATH	G	FGROUP 軸の軌跡基準は曲線パラメータです	M	+/-		PM-NC
var	K	キーワード:パラメータ転送のタイプ		+/-		PM-NC
VELOLIM	K	最高軸速度または主軸速度を調整	M	+/-	-	PM-NC
VELOLIMA	K	最大スレーブ軸速度の低減、または増加	M	+/-	+/-	PM-NC
WAITC	P	軸/主軸に対して関連ブロック切り替えの条件が満たされるまで待機		+/-	-	PM-NC
WAITE	P	別のチャンネルのプログラム終了を待機		+/-	-	PM-NC
WAITENC	P	軸位置の原点同期または復元まで待機		+/-	-	PM-NC
WAITM	P	指定したチャンネルのマークを待機; 先行ブロックをイグザクトストップで終了		+/-	-	PM-NC
WAITMC	P	指定したチャンネルのマークを待機; 他のチャンネルがマークに到達していない場合にのみイグザクトストップ		+/-	-	PM-NC
WAITP	P	位置決め軸の移動終了まで待機		+/-	-	PM-NC
WAITS	P	主軸位置への到達を待機		+/-	-	PM-NC
WALCSO ⁶⁾	G	ワーク座標系のワーキングエリアリミットの選択解除	M	+/-	-	PM-NC
WALCS1	G	WCS 作業領域リミットグループ 1 が有効	M	+/-	-	PM-NC
WALCS2	G	WCS 作業領域リミットグループ 2 が有効	M	+/-	-	PM-NC
WALCS3	G	WCS 作業領域リミットグループ 3 が有効	M	+/-	-	PM-NC
WALCS4	G	WCS 作業領域リミットグループ 4 が有効	M	+/-	-	PM-NC

5.1 命令

命令	タイプ ¹⁾	意味	W ²⁾	TP ³⁾	SA ⁴⁾	説明の参照先 ⁵⁾
1) 2) 3) 4) 5) 詳細は、一覧 (ページ 1433) を参照してください。						
WALCS5	G	WCS 作業領域リミットグループ 5 が有効	M	+/-	-	PM-NC
WALCS6	G	WCS 作業領域リミットグループ 6 が有効	M	+/-	-	PM-NC
WALCS7	G	WCS 作業領域リミットグループ 7 が有効	M	+/-	-	PM-NC
WALCS8	G	WCS 作業領域リミットグループ 8 が有効	M	+/-	-	PM-NC
WALCS9	G	WCS 作業領域リミットグループ 9 が有効	M	+/-	-	PM-NC
WALCS10	G	WCS 作業領域リミットグループ 10 が有効	M	+/-	-	PM-NC
WALIMOF	G	BCS ワーキングエリアリミットのオフ	M	+/-	-	PM-NC
WALIMON ⁶⁾	G	BCS ワーキングエリアリミットのオン	M	+/-	-	PM-NC
WHEN	K	条件を満たすたびに、動作を 1 回実行します。		-	+/-	FM-SA
WHENEVER	K	条件を満たしている間は、各補間サイクルで動作を周期的に実行します。		-	+/-	FM-SA
WHILE	K	WHILE プログラムループの開始		+/-		PM-NC
WRITE	P	テキストをファイルシステムに書込みます。 指定されたファイルの最後にブロックを追加します。		+/-	-	PM-NC
WRTPR	P	OPI 変数での文字列の書き込み(WRTPR)		+/-	-	PM-NC
X	A	軸名称	m/s	+/-	+/-	PM-NC
XOR	O	排他的論理和		+/-		PM-NC
Y	A	軸名称	m/s	+/-	+/-	PM-NC
Z	A	軸名称	m/s	+/-	+/-	PM-NC

凡例

1) 命令のタイプ:

A アドレス

値の割り当て先の識別子(たとえば、OVR=10).値の割り当てはおこなわないで、機能をオン/オフにするアドレスもあります(例えば、CPLON と CPLOF)。

C (A) AST サイクル

AST (= Automatic Servo Tuning)による自動ポスト最適化(チューニング)の予約済み NC プログラム。パラメータは、特定の最適化の状況に対応するために使用されます。これらのパラメータは呼び出し時に転送されます。

C 計測サイクル

(M) 円筒形のワークの内部の直径を決定するような、一般的に有効な専用の測定動作がプログラムされる予約 NC プログラム。パラメータは特定の測定状況に対応するために使用されます。これらのパラメータは呼び出し時に転送されます。

C (T) 加工種別のサイクル

ねじのタッピングまたはポケットのフライス削りなどの、一般的に有効な専用の加工運転がプログラム指令された予約 NC プログラム。特定の加工状況に対する調整は、サイクル呼び出し時にサイクルに転送されるパラメータによって実行されます。

F 予約機能(戻り値を提供します)

予約機能の呼び出しは、演算式のオペランドにすることができます。

G G 命令

G 命令は G グループに分けられます。グループのうち、1 つの G 命令のみを 1 ブロックにプログラム指令できます。G 命令は、モーダル(同じグループの別の命令によって取り消されるまで)、または、プログラム指令ブロックに対してのみ有効(ノンモーダル)です。

K キーワード

ブロックの構文を定義する識別子。キーワードに値は割り当てられず、キーワードで NC 機能をオン/オフすることはできません。

例:フロー制御(IF, ELSE, ENDIF, WHEN, ...)、プログラムの実行(GOTOB, GOTO, RET ...)

O 演算子

算術演算、比較演算、または論理演算の演算子。

P 予約処理(戻り値を提供しません)

5.1 命令

PA プログラム属性

プログラム属性は、サブプログラムの定義行の終わりに指定します。

PROC <プログラム名称> (...) <プログラム属性>

プログラム属性は、サブプログラムの実行時の動作を決定します。

2) 命令の効果:

M モーダル

s ノンモーダル

3) パートプログラムのプログラマブル属性

+ プログラム指令可能

- プログラム指令不可

M 工作機械メーカーによってのみプログラム指令可能

4) シンクロナイズドアクションのプログラム指令属性:

+ プログラム指令可能

- プログラム指令不可

t テクノロジサイクルでのみプログラム指令可能

5) 命令の詳細説明があるマニュアルの参照先:

FM-A 『機能マニュアル 軸と主軸』

FM-B 機能マニュアル『基本機能』

FM-SA 『機能説明書』シンクロナイズドアクション

FM-SI 機能マニュアル Safety Integrated

FM-TE 機能マニュアル『テクノロジー』

FM-TM 機能マニュアル 工具管理機能

FM-TR 機能マニュアル『座標変換』

PM-MC 『プログラミングマニュアル、測定サイクル』

PM-NC プログラミングマニュアル『NC プログラミング』

6) プログラム開始点での初期設定(他で何もプログラム指令していなければ、コントローラの出荷時設定値です)。

5.2 アドレス

5.2.1 アドレス文字

文字	意味	数値拡張子
A	設定可能なアドレス識別子	x
B	設定可能なアドレス識別子	x
C	設定可能なアドレス識別子	x
D	工具長補正、工具刃先の選択/選択解除	
E	設定可能なアドレス識別子	x
F	送り速度 秒単位のドウェル時間	x
G	G 命令	
H	H 機能	x
I	設定可能なアドレス識別子	x
J	設定可能なアドレス識別子	x
K	設定可能なアドレス識別子	x
L	サブプログラム名称、サブプログラム呼び出し	
M	M 機能	x
N	サブブロック番号	
O	未使用	
P	プログラム実行の回数	
Q	設定可能なアドレス識別子	x
R	変数識別子(R 変数) 設定可能なアドレス識別子(数値拡張子なし)	x
S	主軸値 主軸回転でのドウェル時間	x x
T	工具番号	x
U	設定可能なアドレス識別子	x
V	設定可能なアドレス識別子	x
W	設定可能なアドレス識別子	x

5.2 アドレス

文字	意味	数値拡張子
X	設定可能なアドレス識別子	x
Y	設定可能なアドレス識別子	x
Z	設定可能なアドレス識別子	x
%	ファイル転送の開始文字と区切り文字	
:	メインブロック番号	
/	スキップ識別子	

5.2.2 固定アドレス

軸拡張子を含まない固定アドレス

アドレス識別子	アドレスタイプ	モーダル/ノンモーダル	G70/G71	G700/G710	G90/G91	IC	AC	DC、ACN、ACP	CIC、CAC、CDC、CACN、CACP	QU	割り当てる値のデータタイプ
D	オフセット番号	M								X	符号なしのINT
F	送り速度、ドウェル時間	m、s	X							X	符号なしのREAL
FLIM	最大軌跡速度	M									符号なしのREAL
G	G 命令	G 機能の一覧を参照してください									符号なしのINT
H	補助機能	s								X	M: 符号なしのINT H: REAL

アドレス識別子	アドレスタイプ	モード ル/ ノン モード	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC、 ACN、 ACP	CIC、 CAC、 CDC、 CACN 、 CACP	QU	割り当てる値 のデータタイ プ
L	サブプログラ ム番号	s									符号なしの INT
M	補助機能	s								X	M: 符号なしの INT H: REAL
N	ブロック番 号	s									符号なしの INT
OVR	オーバライ ド	M									符号なしの REAL
OVRRAP	早送り速度 のオーバラ イド	M									符号なしの REAL
P	サブプログラ ム繰り返 しの回数	s									符号なしの INT
PACCLIM	最大軌跡加 速度	M									符号なしの REAL
S	主軸、ドゥ ェル時間	m、s								X	符号なしの REAL
SCC	径方向軸の G96 /G961/ G962 への 割り当て	M									REAL
SPOS	主軸位置決 め	M				X	X	X			REAL

5.2 アドレス

アドレス識別子	アドレスタイプ	モード ル/ ノン モード	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC、 ACN、 ACP	CIC、 CAC、 CDC、 CACN 、 CACP	QU	割り当てる値 のデータタイ プ
SPOSA	ブロック境 界を越える 主軸位置決 め	M				X	X	X			REAL
T	工具番号	M								X	符号なしの INT

軸拡張子を含む固定アドレス

アドレス識別子	アドレスタイプ	モード ル/ ノン モード	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC、 ACN、 ACP	CIC、 CAC、 CDC、 CACN 、 CACP	QU	割り当てる値 のデータタイ プ
ACC	軸加減速度	M									符号なしの REAL
ACCLIMA	スレーブ軸 の軸の加減 速度制限	M									符号なしの REAL
AX	可変軸識別 子	¹⁾	X	X	X	X	X	X			REAL
FA	軸の送り速 度	M	X							X	符号なしの REAL
FDA	ハンドルオー ーバライド の軸送り速 度	s	X								符号なしの REAL
FGREF	基準半径	M	X	X							符号なしの REAL
FL	軸の送り速 度制限	M	X								符号なしの REAL

アドレス識別子	アドレスタイプ	モータ ル/ ノンモ ーダル	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC、 ACN、 ACP	CIC、 CAC、 CDC、 CACN 、 CACP	QU	割り当てる値 のデータタイ プ
FMA	軸の同期送り速度	M									符号なしの REAL
FOC	トルクを制限したノン モータ移動	s									REAL
FOCOF	トルクを制限したノン モータ移動がオフ	M									REAL
FOCON	トルクを制限したノン モータ移動がオン	M									REAL
FXS	突き当て点 停止のオン	M									符号なしの INT
FXST	突き当て点 停止のトル ク制限	M									REAL
FXSW	突き当て点 停止の監視 範囲	M									REAL
IP	可変補間パ ラメータ	s	X	X	X	X	X				REAL
JERKLIM	軸加々速度 制限	M									符号なしの REAL
JERKLIMA	スレーブ軸 の軸加々速 度制限	M									符号なしの REAL

5.2 アドレス

アドレス識別子	アドレスタイプ	モード ル/ノ ンモ ーダル	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC、 ACN、 ACP	CIC、 CAC、 CDC、 CACN 、 CACP	QU	割り当てる値 のデータタイ プ
MEAC	繰り返し計測	S									INT モードと 1～4 つのトリ ガ事象
MEASA	残移動距離 を削除する 軸計測	S									INT モードと 1～4 つのトリ ガ事象
MEAWA	残移動距離 を削除しな い軸計測	S									INT モードと 1～4 つのトリ ガ事象
MOV	軸位置決め の起動	M	X	X	X	X	X	X	X		REAL
OS	揺動のオン/オフ	M									符号なしの INT
OSB	揺動の起点	M	X	X	X	X	X	X			REAL
OSCILL	揺動のため の軸割り当 て、揺動を 起動	M									軸: 1～3 つの切り 込み軸
OSCTRL	揺動オプシ ョン	M									符号なしの INT: 設定オプシ ョン、符号なし の INT:リセット オプション
OSE	揺動の終了 位置	M	X	X	X	X	X	X			REAL

アドレス識別子	アドレスタイプ	モータ ル/ ノンモ ータ ル	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC、 ACN、 ACP	CIC、 CAC、 CDC、 CACN 、 CACP	QU	割り当てる値 のデータタイ プ
OSNSC	スパークア ウトサイク ル数(揺動)	M									符号なしの INT
OSP1	左の反転点 (揺動)	M	X	X	X	X	X	X			REAL
OSP2	右の反転点 (揺動)	M	X	X	X	X	X	X			REAL
OST1	左の反転点 での停止時 間(揺動)	M									REAL
OST2	右の反転点 での停止時 間(揺動)	M									REAL
OVRA	軸オーバラ イド	M	X								符号なしの REAL
PO	多項式係数	s	X	X		X	X	X			符号なしの REAL
POLF	高速リトラ クト位置	M	X	X							符号なしの REAL
POS	軸の位置決 め	M	X	X	X	X	X	X	X		REAL
POSA	ブロック境 界を越える 軸位置決め	M	X	X	X	X	X	X	X		REAL
POSP	複数区間の 軸位置決め (揺動)	M	X	X	X	X	X	X			REAL: 終了位置 Real: 区間長 INT:オプショ ン

5.2 アドレス

アドレス識別子	アドレスタイプ	モーダル/ノンモーダル	G70/G71	G700/G710	G90/G91	IC	AC	DC、ACN、ACP	CIC、CAC、CDC、CACN、CACP	QU	割り当てる値のデータタイプ
STA	スパークアウト時間(軸)	M									符号なしのREAL
SRA	外部入力によるスパークアウト軌跡(軸)	M									符号なしのREAL
VELOLIM	軸の速度制限	M									符号なしのREAL
VELOLIMA	スレーブ軸の軸の速度制限	M									符号なしのREAL

1) アブソリュート終点:モーダル、インクレメンタル終点:ノンモーダル、それ以外の場合は、構文を特定する G 機能に応じてモーダル/ノンモーダルが決まります。

5.2.3 設定可能アドレス

アドレス識別子(初期設定)	アドレスタイプ	モーダル/ノンモーダル	G90/G91	IC	AC	DC、ACN、ACP	CIC、CAC、CDC、CACN、CACP	PR, PM	QU	最大数	割り当てる値のデータタイプ
軸値と終点											
X、Y、Z、A、B、C	軸	1)	X	X	X	X				8	REAL
AP	極角度	m/s 1)	X	X	X					1	REAL

アドレス識別子(初期設定)	アドレスタイプ	モデル/ノンモデル	G90/G91	IC	AC	DC、ACN、ACP	CIC、CAC、CDC、CACN、CACP	PR、PM	QU	最大数	割り当てる値のデータタイプ
RP	極半径	m/s ¹⁾	X	X	X					1	符号なしのREAL
工具オリエンテーション											
A2、B2、C2	オイラー角または RPY 角	s								3	REAL
A3、B3、C3	方向ベクトルのコンポーネント	s								3	REAL
A4、B4、C4	ブロック始点での面法線ベクトルの構成部分	s								3	REAL
A5、B5、C5	ブロック終点での面法線ベクトルの構成部分	s								3	REAL
A6、B6、C6	円錐の回転軸の方向ベクトルの構成部分	s								3	REAL
A7、B7、C7	円錐面の中間の向きのベクトル成分	s								3	REAL
LEAD	リード角	M								1	REAL
THETA	回転角度、工具方向を中心とする回転	M		X	X					1	REAL
TILT	傾斜角	M								1	REAL
ORIS	向きの変更(軌跡に対して)	M								1	REAL

5.2 アドレス

アドレス識別子(初期設定)	アドレスタイプ	モード ダル/ノ ン モード ダル	G90 /G9 1	IC	AC	DC、 ACN、 ACP	CIC、 CAC、 CDC、 CACN 、 CACP	PR、 PM	QU	最大 数	割り当てる値 のデータタイ プ
補間パラメータ											
i、j、k	補間パラメータ 中間点座標	s		x ²⁾	x ²⁾					3	REAL
I1、J1、K1		s	X	X	X					3	REAL
RPL	平面の回転	s								1	REAL
CR	円弧半径	s								1	符号なしの REAL
AR	開口角度	s								1	符号なしの REAL
TURN	ヘリカルのター ン数	s								1	符号なしの INT
PL	パラメータ区 間の長さ	s								1	符号なしの REAL
PW	重み	s								1	符号なしの REAL
SD	スプラインの 次数	M								1	符号なしの INT
TU	軸角度	s								1	符号なしの INT
STAT	関節継手の位 置	M								1	符号なしの INT
SF	ねじ切りの始 点オフセット	M								1	REAL
DISCL	安全距離 SAR	s								1	符号なしの REAL

アドレス識別子(初期設定)	アドレスタイプ	モーダル/ノンモーダル	G90/G91	IC	AC	DC、ACN、ACP	CIC、CAC、CDC、CACN、CACP	PR、PM	QU	最大数	割り当てる値のデータタイプ
DISR	クリアランス/SAR クリアランスの再位置決め	S								1	符号なしの REAL
DISPR	再位置決めの軌跡距離	S								1	符号なしの REAL
ALF	高速リトラクト角度	M								1	符号なしの INT
DILF	高速リトラクト角度	M								1	REAL
FP	固定点:アプローチする固定点の番号	S								1	符号なしの INT
RNDM	モーダル丸み付け	M								1	符号なしの REAL
RND	ノンモーダル丸み付け	S								1	符号なしの REAL
CHF	ノンモーダル面取り	S								1	符号なしの REAL
CHR	オリジナルの移動方向の面取り	S								1	符号なしの REAL
ANG	輪郭角度	S								1	REAL
ISD	切り込み深さ	M								1	REAL
DISC	工具径補正の挿入円のオーバーシュート	M								1	符号なしの REAL
OFFN	通常の輪郭オフセット	M								1	REAL
DITS	ねじの切り始めの軌跡	M								1	REAL

5.2 アドレス

アドレス識別子(初期設定)	アドレスタイプ	モード ノン モード	G90 /G91	IC	AC	DC、 ACN、 ACP	CIC、 CAC、 CDC、 CACN、 CACP	PR、 PM	QU	最大 数	割り当てる値 のデータタイプ
DITE	ねじの切り上げの軌跡	M								1	REAL
ユーナの丸み付け条件											
ADIS	丸み付き隙間	M								1	符号なしの REAL
ADISPOS	早送りの丸み付き隙間	M								1	符号なしの REAL
計測											
MEAS	タッチトリガプローブによる計測	s								1	符号なしの INT
MEAW	残移動距離を削除しない、タッチトリガプローブによる計測	s								1	符号なしの INT
軸、主軸動作											
LIMS	主軸速度の制限	M								1	符号なしの REAL
COARSEA	ブロック切り替え動作:軸の汎用イグザクトストップ	M									
FINEA	ブロック切り替え動作:軸の精密イグザクトストップ	M									

アドレス識別子(初期設定)	アドレスタイプ	モデル/ノンモデル	G90/G91	IC	AC	DC、ACN、ACP	CIC、CAC、CDC、CACN、CACP	PR、PM	QU	最大数	割り当てる値のデータタイプ
IPOENDA	ブロック切り替え動作:軸の補間停止	M									
DIACYCOFA	径方向軸: サイクルで軸の直径指定をオフ	M									
DIAM90A	径方向軸:G90での軸の直径指定	M									
DIAMCHAN	径方向軸: すべての径方向軸を直径指定のチャネル状態へ移行	M									
DIAMCHAN A	径方向軸:直径/半径指定チャネル状態へ移行	M									
DIAMOFA	径方向軸:軸の直径指定をオフ	M									
DIAMONA	径方向軸:軸の直径指定をオン	M									
BP	桁 : 位置属性の間接プログラミング	M									
送り速度											

5.2 アドレス

アドレス識別子(初期設定)	アドレスタイプ	モード ノン モード	G90 /G91	IC	AC	DC、 ACN、 ACP	CIC、 CAC、 CDC、 CACN、 CACP	PR、 PM	QU	最大 数	割り当てる値 のデータタイプ
FAD	滑らかな送り 移動の速度	S						X		1	符号なしの REAL
FD	ハンドルオー バライドの軌 跡送り速度	S								1	符号なしの REAL
FRC	丸み付けと面 取りの送り速 度	S								1	符号なしの REAL
FRCM	丸み付けと面 取りの送り速 度、モード	M								1	符号なしの REAL
FB	ノンモード 送り速度	S								1	符号なしの REAL
ニブリング/パンチング											
SPN	ブロック毎の 軌跡区間数	S								1	INT
SPP	軌跡区間の長 さ	M								1	REAL
研削加工											
ST	スパークアウ ト時間	S								1	符号なしの REAL
SR	後退軌跡	S								1	符号なしの REAL
工具選択											
TCARR	工具ホルダ	M								1	INT
工具管理機能											

アドレス識別子(初期設定)	アドレスタイプ	モデル/ノンモデル	G90/G91	IC	AC	DC、ACN、ACP	CIC、CAC、CDC、CACN、CACP	PR、PM	QU	最大数	割り当てる値のデータタイプ
DL	合計工具オフセット	M								1	INT
OEM アドレス											
OMA1	OEM アドレス 1	M		X	X	X				1	REAL
OMA2	OEM アドレス 2	M		X	X	X				1	REAL
OMA3	OEM アドレス 3	M		X	X	X				1	REAL
OMA4	OEM アドレス 4	M		X	X	X				1	REAL
OMA5	OEM アドレス 5	M		X	X	X				1	REAL
その他											
CUTMOD	旋回工具のオフセットデータ変更(旋回工具ホルダとの組み合わせ)	M									INT
CUTMODK	旋回工具のオフセットデータ変更(キネティックチェーンによって定義されている方向座標変換との組み合わせ)	M									STRING

5.3 G 命令

アドレス識別子(初期設定)	アドレスタイプ	モード モデル/ ノン モデル	G90 /G9 1	IC	AC	DC、 ACN、 ACP	CIC、 CAC、 CDC、 CACN 、 CACP	PR、 PM	QU	最大 数	割り当てる値 のデータタイ プ
TOFF	指定ジオメトリ軸に平行な工具長補正	M									REAL
TOFFL	工具長成分 L1、L2、または L3 の方向への工具長補正	M									REAL
TOFFR	工具径補正	M									REAL
TOFFLR	同時の工具長補正と工具半径補正	M									REAL

- 1) アブソリュート終点:モデル、インクレメンタル終点:ノンモデル。それ以外の場合は、構文を特定する G 命令に応じてモデル/ノンモデルが決まります。
- 2) 円弧の中心点を指すときは、IPO パラメータがインクレメンタルに機能します。このパラメータは、AC によるアブソリュートモードでプログラム指令できます。パラメータに他の意味(ねじリードなど)がある場合は、アドレス指令は無視されます。

5.3 G 命令

5.3.1 G 命令

G 命令は G グループに分けられます。パートプログラムまたはシンクロナイズドアクションでは、ブロックに G グループの G 命令のみを書き込むことができます。G 命令はモデルまたはノンモデルにすることができます。

モデル:同じ G グループの別の G 命令がプログラム指令されるまで

5.3.2 G グループ 1: モーダルで有効な動作命令

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G0	1	早送り	+	M		
G1	2	直線補間(直線補間)	+	M	X	
G2	3	右回りの円弧補間	+	M		
G3	4	左回りの円弧補間	+	M		
CIP	5	中間点経由の円弧補間	+	M		
ASPLINE	6	A スプライン	+	M		
BSPLINE	7	B スプライン	+	M		
CSPLINE	8	3 次スプライン	+	M		
POLY	9	多項式補間	+	M		
G33	10	固定リードのねじ切り	+	M		
G331	11	タッピング	+	M		
G332	12	後退(タッピング)	+	M		
OEMIPO1	13	予備	+	M		
OEMIPO2	14	予備	+	M		
CT	15	接線方向の遷移をおこなう円弧	+	M		
G34	16	単調増加可変リードねじ切り	+	M		
G35	17	単調減少可変リードねじ切り	+	M		
INVCW	18	右回りのインボリュート補間	+	M		
INVCCW	19	左回りのインボリュート補間	+	M		
G335	20	右回りの凸型ねじの旋削	+	M		
G336	21	左回りでの凸ねじの旋削	+	M		

5.3.3 G グループ 2: ノンモーダルで有効な動作、ドウェル時間

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G4	1	ドウェル時間、事前設定	-	S		
G63	2	非同期タッピング	-	S		

5.3 G 命令

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G74	3	同期制御によるレファレンス点復帰	-	S		
G75	4	固定点アプローチ	-	S		
REPOSL	5	直線再位置決め	-	S		
REPOSQ	6	4 分円の再位置決め	-	S		
REPOSH	7	半円の再位置決め	-	S		
REPOSA	8	すべての軸の直線再位置決め	-	S		
REPOSQA	9	全ジオメトリ軸の 4 分円による再位置決め	-	S		
REPOSHA	10	全ジオメトリ軸の半円による再位置決め	-	S		
G147	11	直線による輪郭へのアプローチ	-	S		
G247	12	4 分円による輪郭へのアプローチ	-	S		
G347	13	半円による輪郭へのアプローチ	-	S		
G148	14	直線による輪郭からの移動	-	S		
G248	15	4 分円による輪郭からの移動	-	S		
G348	16	半円による輪郭からの移動	-	S		
G5	17	傾斜プランジ研削	-	S		
G7	18	傾斜プランジ研削時の補正動作	-	S		

5.3.4 G グループ 3:プログラマブルフレーム、作業領域リミットおよび極座標プログラミング

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
TRANS	1	TRANSLATION:プログラマブルオフセット	-	S		
ROT	2	ROTATION:プログラマブル座標回転	-	S		
SCALE	3	SCALE:プログラマブルスケーリング	-	S		
MIRROR	4	ミラーリング:プログラマブルミラーリング	-	S		
ATRANS	5	追加平行移動:追加プログラマブル平行移動	-	S		
AROT	6	追加座標回転:プログラマブル座標回転	-	S		
ASCALE	7	追加スケーリング:プログラマブルスケーリング	-	S		
AMIRROR	8	追加ミラーリング:プログラマブルミラーリング	-	S		

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
-	9	未使用	-	-		
G25	10	最小作業領域リミット/主軸速度制限	-	S		
G26	11	最大作業領域リミット/主軸速度制限	-	S		
G110	12	最後のプログラム指令位置に対する極のプログラミング	-	S		
G111	13	現在のワーク座標系の原点に対する極のプログラミング	-	S		
G112	14	最後に有効な極に対する極のプログラミング	-	S		
G58	15	アブソリュートプログラマブルゼロオフセット	-	S		
G59	16	追加プログラマブルゼロオフセット	-	S		
ROTS	17	立体角による座標回転	-	S		
AROTS	18	立体角による追加座標回転	-	S		

5.3.5 G グループ 4:FIFO

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
STARTFIFO	1	FIFO の起動 実行と同時に解析メモリの書き込み	+	M	X	
STOPFIFO	2	STOP FIFO 加工の停止; STARTFIFO を検出までに解析メモリがいっぱいになるか、FIFO がいっぱいになるか、またはプログラムが終了するまで、解析メモリを使用します。	+	M		
FIFOCTRL	3	解析メモリの自動制御を適用します	+	M		

5.3 G 命令

5.3.6 G グループ 6:平面の選択

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G17	1	平面の選択 1. – 2.ジオメトリ軸	+	M	X	
G18	2	平面の選択 3. – 1.ジオメトリ軸	+	M		
G19	3	平面の選択 2. – 3.ジオメトリ軸	+	M		

5.3.7 G グループ 7:工具径補正

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G40	1	工具径補正なし	+	M	X	
G41	2	輪郭の左側の工具径補正	-	M		
G42	3	輪郭の右側の工具径補正	-	M		

5.3.8 G グループ 8:設定可能なゼロオフセット

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G500	1	設定可能ゼロオフセットの解除(G54～G57、G505～G599)	+	M	X	
G54	2	1 番目の設定可能ゼロオフセット	+	M		
G55	3	2 番目の設定可能ゼロオフセット	+	M		
G56	4	3 番目の設定可能ゼロオフセット	+	M		
G57	5	4 番目の設定可能ゼロオフセット	+	M		
G505	6	5 番目の設定可能ゼロオフセット	+	M		
...	...	...	+	M		

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G599	100	99 番目の設定可能ゼロオフセット	+	M		
この G グループの各 G 命令を使用して、設定可能なユーザーフレーム\$P_UIFR[]を起動します。 G54 はフレーム\$P_UIFR[1]に、G505 はフレーム\$P_UIFR[5]に対応します。 設定可能なユーザーフレームの数と、この G グループの G 命令の数は、マシンデータ MD28080 \$MC_MM_NUM_USER_FRAMES を使用して設定できます。						

5.3.9 G グループ 9:フレームと工具オフセットのマスク

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G53	1	現在のフレームのマスク: TOROT および TOFRAME 用のシステムフレームを含む プログラマブルフレームと 有効な設定可能フレーム(G54～G57、G505～G599)	-	S		
SUPA	2	現在位置設定、接触計測、外部ゼロオフセット、ハンドルオフセット(DRF)を含む PAROT、[外部ゼロオフセット]、および重畳移動のシステムフレームのマスクを含む G153 と同じ働きです。	-	S		
G153	3	すべてのチャンネル別基本フレームまたは NCU グローバル基本フレーム、またはその両方を含む G53 と同じ働きです。	-	S		
SUPD	4	有効な工具オフセットのマスク	-	S		

5.3 G 命令

5.3.10 G グループ 10:イグザクトストップ-連続軌跡モード

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G60	1	イグザクトストップ	+	M	X	
G64	2	連続軌跡モード	+	M		
G641	3	距離条件に応じたスムージングによる連続軌跡モード(=プログラム指令可能な丸み付け隙間)	+	M		
G642	4	定義した許容範囲内のスムージングによる連続軌跡モード	+	M		
G643	5	定義した許容範囲内のスムージングによる連続軌跡モード(ブロック内部)	+	M		
G644	6	最大ダイナミック応答によるスムージングによる連続軌跡モード	+	M		
G645	7	スムージング、および定義許容範囲内で接線方向のブロック遷移をおこなう連続軌跡モード	+	M		

5.3.11 G グループ 11:イグザクトストップ、ノンモーダル

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G9	1	イグザクトストップ	-	S		

5.3.12 G グループ 12:イグザクトストップでのブロック切り替え条件(G60/G9)

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G601	1	精密イグザクトストップでブロック切り替え	+	M	X	
G602	2	汎用イグザクトストップでブロック切り替え	+	M		
G603	3	IPO のブロック終点でブロック切り替え	+	M		

5.3.13 G グループ 13:インチ/メトリック単位のワーク計測

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G70	1	インチ単位で入力(長さ)	+	M		
G71	2	メトリック単位 mm で入力(長さ)	+	M	X	
G700	3	インチ単位、inch/min で入力 (長さ+速度+システム変数)	+	M		
G710	4	メトリック単位、mm/min で入力 (長さ+速度+システム変数)	+	M		

5.3.14 G グループ 14:アブソリュート/インクレメンタルのワーク計測

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G90	1	アブソリュート指令	+	M	X	
G91	2	インクレメンタル指令	+	M		

5.3.15 G グループ 15:送り速度タイプ

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G93	1	インバースタイム送り (rpm)	+	M		
G94	2	毎分送り速度(mm/min、inch/min 単位)	+	M	X	
G95	3	毎回転送り速度(mm/rev、inch/rev 単位)	+	M		
G96	4	毎回転送り速度(G95 に対して)および収束一定制御	+	M		
G97	5	毎回転送り速度および一定主軸速度(周速一定制御オフ)	+	M		
G931	6	移動時間により指定された送り速度、一定軌跡速度を解除	+	M		

5.3 G 命令

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G961	7	毎分送り速度(G94 に対して)および周速一定制御	+	M		
G971	8	毎分送り速度および一定主軸速度(周速一定制御オフ)	+	M		
G942	9	毎分送り速度と周速一定制御、または一定主軸速度	+	M		
G952	10	毎回転送り速度と周速一定制御、または一定主軸速度	+	M		
G962	11	毎分送り速度と毎回転送り速度、および周速一定制御	+	M		
G972	12	毎分送り速度または毎回転送り速度および一定主軸速度(周速一定制御オフ)	+	M		
G973	13	主軸速度制限のない毎回転送り速度および一定主軸速度(ISO モードでは、LIMS なしの G97)	+	M		

5.3.16 G グループ 16:内側と外側の曲率での送り速度オーバーライド

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
CFC	1	内側の半径と外側の半径に対して有効な輪郭の一定送り速度	+	M	X	
CFTCP	2	工具中心点の一定送り速度(中心点軌跡)	+	M		
CFIN	3	内側半径のみの一定送り速度、外側半径の場合は加減速度あり	+	M		

5.3.17 G グループ 17:アプローチと後退の動作、工具補正

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
NORM	1	始点と終点の標準位置です	+	M	X	
KONT	2	始点と終点の、輪郭まわりの移動	+	M		

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
KONTT	3	一定接線によるアプローチ/後退	+	M		
KONTC	4	一定曲率によるアプローチ/後退	+	M		

5.3.18 G グループ 18:コーナの動作、工具補正

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G450	1	挿入円 (工具がワークのコーナを、円弧軌跡を描いて移動します)	+	M	X	
G451	2	等間隔の軌跡の交点 (工具がワークのコーナから後退します)	+	M		

5.3.19 G グループ 19:スプライン開始時の曲線遷移

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
BNAT	1	最初のスプラインブロックへの自然遷移	+	M	X	
BTAN	2	最初のスプラインブロックへの接線方向の遷移	+	M		
BAUTO	3	隣接する 3 点による最初のスプライン区間の定義	+	M		

5.3.20 G グループ 20:スプライン終了時の曲線遷移

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
ENAT	1	次の移動ブロックへの自然遷移	+	M	X	
ETAN	2	次の移動ブロックへの接線方向の遷移	+	M		
EAUTO	3	最後の 3 点による最後のスプライン区間の定義	+	M		

5.3 G 命令

5.3.21 G グループ 21:加減速方法

G 命令	No. ¹⁾	意味	MD20150 2)	W ³⁾	STD ⁴⁾	
					SAG	MH
BRISK	1	スムージングをおこなわない高速の軌跡加減速度	+	M	X	
SOFT	2	加々速度一定の滑らかな軌跡加減速度	+	M		
DRIVE	3	速度に応じた軌跡加減速度	+	M		

5.3.22 G グループ 22:工具補正タイプ

G 命令	No. ¹⁾	意味	MD20150 2)	W ³⁾	STD ⁴⁾	
					SAG	MH
CUT2D	1	2½D TRC	+	M	X	
CUT2DF	2	現在のフレームを基準にした 2½D TRC(傾斜面)	+	M		
CUT3DC	3	外周加工の 3 次元 TRC	+	M		
CUT3DF	4	向きが変化する正面削りの 3 次元 TRC	+	M		
CUT3DFS	5	向きが一定の正面削りの 3 次元 TRC。工具オリエンテーションは、G17～G19 で定義され、フレームには影響されません。	+	M		
CUT3DFF	6	向きが一定の正面削りの 3 次元 TRC。工具オリエンテーションは、G17～G19 で定義した方向で、場合によっては、フレームによって回転します。	+	M		
CUT3DCC	7	3 次元径補正を使用した、限界面を考慮した外周加工のための 3D TRC:加工面の輪郭	+	M		
CUT3DCCD	8	工具中心点軌跡で標準工具とは違う工具を使用した、限界面を考慮した外周加工のための 3D TRC:限界面への切り込み	+	M		
CUT2DD	9	差分工具に対する 2½次元 TRC	+	M		
CUT2DFD	10	現在のフレームを基準にした差分工具に対する 2½次元 TRC (傾斜面)	+	M		

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
CUT3DCD	11	外周加工の差分工具に対する 3 次元 TRC	+	M		
CUT3DFD	12	向きが変化する正面削りの差分工具に対する 3 次元 TRC	+	M		

5.3.23 G グループ 23:輪郭の内側の干渉監視

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
CDOF	1	干渉検出のオフ	+	M	X	
CDON	2	干渉検出のオン	+	M		
CDOF2	3	3 次元外周加工のための衝突検出のオフ	+	M		

5.3.24 G グループ 24:フィードフォワード

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
FFWOF	1	フィードフォワード制御のオフ	+	M	X	
FFWON	2	フィードフォワード制御のオン	+	M		

5.3.25 G グループ 25:工具オリエンテーションの基準

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
ORIWKS	1	ワーク座標系(WCS)の工具オリエンテーション	+	M	X	
ORIMKS	2	機械座標系(MCS)の工具オリエンテーション	+	M		

5.3 G 命令

5.3.26 G グループ 26:REPOS のための再位置決め(モーダル)

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
RMB	1	ブロックの始点への再位置決め	-	M		
RMI	2	中断点への再位置決め	-	M	X	
RME	3	ブロック終点への再位置決め	-	M		
RMN	4	最も近い軌跡点への再位置決め	-	M		

5.3.27 G グループ 27:外側コーナで向きを変更するための工具補正

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
ORIC	1	外側コーナーでの向きの変更を、挿入する円弧ブロックに重畳します	+	M	X	
ORID	2.	円弧ブロックの前に向きの変更を実行します	+	M		

5.3.28 G グループ 28:作業領域リミット

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
WALIMON	1	作業領域リミットのオン	+	M	X	
WALIMOF	2	作業領域リミットのオフ	+	M		

5.3.29 G グループ 29:半径指定/直径指定

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
DIAMOF	1	モーダルのチャネル別直径指定のオフ解除によりチャネル別半径指定を起動します。	+	M	X	
DIAMON	2	モーダルの影響を受けないチャネル別直径指定のオン 効果は、プログラム指令の指令モード(G90/G91)に影響されません。	+	M		
DIAM90	3	モーダルの影響を受けるチャネル別直径指定のオン 効果は、プログラム指令した指令モード(G90/G91)に影響されます。	+	M		
DIAMCYCOF	4	サイクル処理のときのモーダルのチャネル別直径指定のオフ	+	M		

5.3.30 G グループ 30:NC ブロック圧縮

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
COMPOF	1	NC ブロック圧縮のオフ	+/-	M	X	
COMPON	2	コンプレッサ機能 COMPON のオン	+/-	M		
COMPCURV	3	コンプレッサ機能 COMPCURV のオン	+/-	M		
COMPCAD	4	コンプレッサ機能 COMPCAD のオン	+/-	M		
COMPSURF	5	コンプレッサ機能 COMPSURF のオン	+/-	M		

5.3 G 命令

5.3.31 G グループ 31:OEM G 命令

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G810	1	OEM G 命令	-	M		
G811	2	OEM G 命令	-	M		
G812	3	OEM G 命令	-	M		
G813	4	OEM G 命令	-	M		
G814	5	OEM G 命令	-	M		
G815	6	OEM G 命令	-	M		
G816	7	OEM G 命令	-	M		
G817	8	OEM G 命令	-	M		
G818	9	OEM G 命令	-	M		
G819	10	OEM G 命令	-	M		
2 つの G 命令が、OEM ユーザー用に予約されています。これにより、OEM ユーザーは各種機能をプログラム指令して、カスタマイズできます。						

5.3.32 G グループ 32:OEM G 命令

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G820	1	OEM G 命令	-	M		
G821	2	OEM G 命令	-	M		
G822	3	OEM G 命令	-	M		
G823	4	OEM G 命令	-	M		
G824	5	OEM G 命令	-	M		
G825	6	OEM G 命令	-	M		
G826	7	OEM G 命令	-	M		
G827	8	OEM G 命令	-	M		
G828	9	OEM G 命令	-	M		

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G829	10	OEM G 命令	-	M		

2 つの G 命令が、OEM ユーザー用に予約されています。これにより、OEM ユーザーは各種機能をプログラム指令して、カスタマイズできます。

5.3.33 G グループ 33:設定可能な精密工具補正

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
FTOCOF	1	オンライン精密工具補正のオフ	+	M	X	
FTOCON	2	オンライン精密工具補正のオン	-	M		

5.3.34 G グループ 34:工具オリエンテーションのスムージング

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
OSOF	1	工具オリエンテーションのスムージングのオフ	+	M	X	
OSC	2	連続工具オリエンテーションのスムージング	+	M		
OSS	3	ブロック終点の工具オリエンテーションのスムージング	+	M		
OSSE	4	ブロックの始点と終点の工具オリエンテーションのスムージング	+	M		
OSD	5	軌跡長の指定によるブロック内部のスムージング	+	M		
OST	6	角度許容範囲の指定によるブロック内部のスムージング	+	M		

5.3 G 命令

5.3.35 G グループ 35:パンチングとニブリング

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
SPOF	1	ストローク US のオフ、パンチングとパンチングのオフ	+	M	X	
SON	2	ニブリングのオン	+	M		
PON	3	パンチングオン	+	M		
SONS	4	補間サイクルのニブリングのオン	-	M		
PONS	5	補間サイクルのパンチングのオン	-	M		

5.3.36 G グループ 36:遅延のあるパンチング

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
PDELAYON	1	遅延のあるパンチングオン	+	M	X	
PDELAYOF	2	遅延のあるパンチングオフ	+	M		

5.3.37 G グループ 37:送り速度プロファイル

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
FNORM	1	DIN 66025 に準拠した標準送り速度	+	M	X	
FLIN	2	可変毎分送り速度	+	M		
FCUB	3	3 次スプラインによる可変送り速度	+	M		

5.3.38 G グループ 38:パンチング/ニブリングへの高速入力/出力の割り当て

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
SPIF1	1	パンチング/ニブリングのバイト 1 に対する高速 NC 入力/出力	+	M	X	
SPIF2	2	パンチング/ニブリングのバイト 2 に対する高速 NC 入力/出力	+	M		

5.3.39 G グループ 39:プログラム指令可能な輪郭精度

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
CPRECOF	1	プログラム指令可能な輪郭精度のオフ	+	M	X	
CPRECON	2	プログラム指令可能な輪郭精度のオン	+	M		

5.3.40 G グループ 40:工具径補正の抑制

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
CUTCONOF	1	工具径補正の抑制のオフ	+	M	X	
CUTCONON	2	工具径補正抑制のオン	+	M		

5.3.41 G グループ 41:中断可能なねじ切り

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
LFOF	1	中断可能なねじ切りのオフ	+	M	X	
LFON	2	中断可能なねじ切りのオン	+	M		

5.3 G 命令

5.3.42 G グループ 42:工具ホルダ

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
TCOABS	1	現在の工具オリエンテーションから工具長成分を特定	+	M	X	
TCOFR	2	動作中のフレームの向きから工具長成分を特定	+	M		
TCOFRZ	3	工具の選択時に、動作中のフレームの工具オリエンテーションを特定、Z 方向の工具点	+	M		
TCOFRY	4	工具の選択時に、動作中のフレームの工具オリエンテーションを特定、Y 方向の工具点	+	M		
TCOFRX	5	工具の選択時に、動作中のフレームの工具オリエンテーションを特定、X 方向の工具点		M		

5.3.43 G グループ 43:SAR アプローチ方向

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G140	1	G41/G42 で定義した SAR アプローチ方向	+	M	X	
G141	2	輪郭の左側への SAR アプローチ方向	+	M		
G142	3	輪郭の右側への SAR アプローチ方向	+	M		
G143	4	接線に応じて SAR アプローチ方向を決定	+	M		

5.3.44 G グループ 44:SAR 軌跡分割

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G340	1	空間のアプローチブロック;つまり、1 ブロックの平面内の切り込み深さとアプローチ	+	M	X	
G341	2	最初に垂直軸(Z)に切り込み、その後平面内にアプローチ	+	M		

5.3.45 G グループ 45:FGROUP 軸の軌跡基準

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
SPATH	1	FGROUP 軸の軌跡基準は円弧長です	+	M	X	
UPATH	2	FGROUP 軸の軌跡基準は曲線パラメータです	+	M		

5.3.46 G グループ 46:高速リトラクトのレベルの選択

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
LFTXT	1	軌跡タンジェントと現在の工具オリエンテーションから平面を特定します。	+	M	X	
LFWP	2	現在の作業平面から平面(G17/G18/G19)を特定します	+	M		
LFPOS	3	位置への軸の後退	+	M		

5.3.47 G グループ 47:外部 NC コードに対応したモード切り替え

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G290	1	SINUMERIK 言語モードの起動	+	M	X	
G291	2	ISO 言語モードの起動	+	M		

5.3 G 命令

5.3.48 G グループ 48:工具径補正によるアプローチと後退の動作

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
G460	1	アプローチと後退ブロックの干渉検出のオン	+	M	X	
G461	2	TRC ブロックに交点がない場合に、円弧で境界ブロックを拡張	+	M		
G462	3	TRC ブロックに交点がない場合に、直線で境界ブロックを拡張	+	M		

5.3.49 G グループ 49:ポイントツーポイント移動

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
CP	1	軌跡移動	+	M	X	
PTP	2	ポイントツーポイント移動(同期軸移動)	+	M		
PTPG0	3	G0 の場合はポイントツーポイント移動のみ、G0 以外の場合は軌跡移動の CP	+	M		
PTPWOC	4	オリエンテーションの変化により生じる補正移動なしのポイントツーポイント移動	+	M		

5.3.50 G グループ 50:旋回のプログラミング

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
ORIEULER	1	オイラー角による旋回角度	+	M	X	
ORIRPY	2	RPY 角度による旋回角度(回転順序 XYZ)	+	M		
ORIVIRT1	3	仮想旋回軸による旋回角度(定義 1)	+	M		
ORIVIRT2	4	仮想旋回軸による旋回角度(定義 2)	+	M		
ORIAPOS	5	仮想旋回軸の回転軸位置による旋回角度	+	M		
ORIRPY2	6	RPY 角による旋回角度(回転順序 ZYX)	+	M		

5.3.51 G グループ 51:旋回プログラミングの補間タイプ

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
ORIVECT	1	大円弧補間(ORIPLANE と同じ)	+	M	X	
ORIAxes	2	機械軸または旋回軸の直線補間	+	M		
ORIPATH	3	軌跡を表わす、工具オリエンテーションの軌道	+	M		
ORIPLANE	4	平面上の補間(ORIVECT と同じです)	+	M		
ORICONCW	5	円錐面での右回り方向の補間	+	M		
ORICONCC W	6	円錐面での左回り方向への補間	+	M		
ORICONIO	7	中間旋回設定による円錐面の補間	+	M		
ORICONTO	8	接線方向の遷移による円錐面の補間	+	M		
ORICURVE	9	向きの追加空間曲線による補間	+	M		
ORIPATHS	10	軌跡に対する工具オリエンテーション、旋回処理の不備をスムージングします	+	M		

5.3.52 G グループ 52:ワークに対するフレーム回転

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
PAROTOF	1	ワークに対するフレーム回転のオフ	+	M	X	
PAROT	2	ワークに対するフレーム回転のオン ワーク上でワーク座標系を配置します。	+	M		

5.3.53 G グループ 53:工具に対するフレーム回転

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
TOROTOF	1	工具に対するフレーム回転のオフ	+	M	X	
TOROT	2	フレームを回転して、WCS の Z 軸を工具オリエンテーションに平行に配置します	+	M		

5.3 G 命令

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
TOROTZ	3	TOROT と同じです	+	M		
TOROTY	4	フレームを回転して、WCS の Y 軸を工具オリエンテーションに平行に配置します	+	M		
TOROTX	5	フレームを回転して、WCS の X 軸を工具オリエンテーションに平行に配置します	+	M		
TOFRAME	6	フレームを回転して、WCS の Z 軸を工具オリエンテーションに平行に配置します	+	M		
TOFRAMEZ	7	TOFRAME と同じです	+	M		
TOFRAMEY	8	フレームを回転して、WCS の Y 軸を工具オリエンテーションに平行に配置します	+	M		
TOFRAMEX	9	フレームを回転して、WCS の X 軸を工具オリエンテーションに平行に配置します	+	M		

5.3.54 G グループ 54:多項式プログラミングのベクトル回転

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
ORIROTA	1	アブソリュートベクトル回転	+	M	X	
ORIROTR	2	相対ベクトル回転	+	M		
ORIROTT	3	接線方向のベクトル回転	+	M		
ORIROTC	4	軌跡タンジェントに対する接線方向の回転ベクトル	+	M		

5.3.55 G グループ 55:直線補間有り/無しの早送り

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
RTLION	1	直線補間による早送り移動のオン	+	M	X	
RTLIOF	2	直線補間による早送り移動のオフ 早送り移動は、単独軸補間でおこなわれます。	+	M		

5.3.56 G グループ 56:工具の摩耗を考慮

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
TOWSTD	1	工具長補正の初期設定値	+	M	X	
TOWMCS	2	機械座標系(MCS)の摩耗値	+	M		
TOWWCS	3	ワーク座標系(WCS)の摩耗値	+	M		
TOWBCS	4	基本座標系(BCS)の摩耗値	+	M		
TOWTCS	5	工具座標系の摩耗値(工具ホルダの工具ホルダ基準点 T)	+	M		
TOWKCS	6	キネマティックトランスフォーメーションをおこなう工具ヘッドの座標系の摩耗値 (工具回転による機械座標系とは異なります)	+	M		

5.3.57 G グループ 57:コーナ減速

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
FENDNORM	1	コーナー減速のオフ	+	M	X	
G62	2	工具径補正が有効なときの内側コーナのコーナ減速(G41/G42)	+	M		
G621	3	すべてのコーナーのコーナー減速	+	M		

5.3.58 G グループ 59:軌跡補間のダイナミック応答モード

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
DYNNORM	1	標準ダイナミック応答	+	M	X	
DYNPOS	2	位置決めモード、タッピング	+	M		
DYNROUGH	3	荒削り	+	M		
DYNSEMIFIN	4	中仕上げ	+	M		

5.3 G 命令

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
DYNFINISH	5	仕上げ	+	M		
DYNPREC	6	滑らかな仕上げ	+	M		

5.3.59 G グループ 60:作業領域リミット

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
WALCS0	1	ワーク座標系の作業領域リミットのオフ	+	M	X	
WALCS1	2	WCS 作業領域リミットグループ 1 が有効	+	M		
WALCS2	3	WCS 作業領域リミットグループ 2 が有効	+	M		
WALCS3	4	WCS 作業領域リミットグループ 3 が有効	+	M		
WALCS4	5	WCS 作業領域リミットグループ 4 が有効	+	M		
WALCS5	6	WCS 作業領域リミットグループ 5 が有効	+	M		
WALCS6	7	WCS 作業領域リミットグループ 6 が有効	+	M		
WALCS7	8	WCS 作業領域リミットグループ 7 が有効	+	M		
WALCS8	9	WCS 作業領域リミットグループ 8 が有効	+	M		
WALCS9	10	WCS 作業領域リミットグループ 9 が有効	+	M		
WALCS10	11	WCS 作業領域リミットグループ 10 が有効	+	M		

5.3.60 G グループ 61:工具オリエンテーションのスミージング

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
ORISOF	1	工具オリエンテーションのスミージングのオフ	+	M	X	
ORISON	2	工具オリエンテーションのスミージングのオン	+	M		

5.3.61 G グループ 62:REPOS のための再位置決め(ノンモーダル)

G 命令	No. ¹⁾	意味	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
RMBBL	1	ブロックの始点への再位置決め	-	S		
RMIBL	2	中断点への再位置決め	-	S	X	
RMEBL	3	ブロック終点への再位置決め	-	S		
RMNBL	4	最も近い軌跡点への再位置決め	-	S		

5.3.62 G グループ 64:研削フレーム

G 命令	No. ¹⁾	意味 チャンネル\$P_GFRAME =の有効な研削フレーム	MD20150 ²⁾	W ³⁾	STD ⁴⁾	
					SAG	MH
GFRAME[0]	1	データ管理\$P_GFR[0]の研削フレーム(ヌルフレーム)	+	M	X	
GFRAME[1]	2	データ管理\$P_GFR[1]の研削フレーム	+	M		
GFRAME[2]	3	データ管理\$P_GFR[2]の研削フレーム	+	M		
...	...		+	M		
GFRAME[100]	101	データ管理\$P_GFR[100]の研削フレーム	+	M		

5.4 予約処理

凡例

- 1) 内部番号(PLC インタフェースの番号など)
 - 2) MD20150 \$MC_GCODE_RESET_VALUES を使用して、電源投入時、リセット、またはパートプログラム終了時の G グループのリセット設定として G 命令が設定ができます:
 + 設定可能
 - 設定不可
 - 3) G 命令の有効性:
 M モーダル
 s ノンモーダル
 - 4) リセット設定。次のマシンデータを参照してください。
 - MD20149GCODE_RESET_S_VALUES (G グループのリセット設定(固定))
 - MD20150 \$MC_GCODE_RESET_VALUES (G グループのリセット設定)
 - MD20151GCODE_RESET_S_MODE (G グループのリセット応答(固定))
 - MD20152 \$MC_GCODE_RESET_MODE (G グループのリセット応答)
 - MD20154 \$MC_EXTERN_GCODE_RESET_VALUES (ISO モードの G グループのリセット位置)
 - MD20156 \$MC_EXTERN_GCODE_RESET_MODE (外部 G グループのリセット応答)
- SAG 初期設定 **Siemens AG**
 MM 初期設定 **Machine M** メーカー(工作機械メーカーの仕様書を参照してください)

5.4 予約処理

予約処理の呼び出しにより、NC の予約機能の実行が開始されます。予約処理は、予約機能とは異なり、戻り値を返しません。

座標系					
識別子	パラメータ				意味
	1 番目	2 番目	3 番目～15 番目	4 番目～16 番目	

座標系					
識別子	パラメータ				意味
PRESETON	AXIS *): 機械軸の 軸識別子	REAL: プリセット オフセット G700/G710 コンテキス ト	1 と同様...	2 と同様...	原点確立状態をリセットして、 プログラム指令した軸に現在値 を設定します。
PRESETONS	AXIS *): 機械軸の 軸識別子	REAL: プリセット オフセット G700/G710 コンテキス ト	1 と同様...	2 と同様...	原点確立状態をリセットせず に、プログラム指令した軸に現 在値を設定します。
DRFOF					チャンネルに割り当てたすべての 軸の DRF オフセットを削除しま す。

- 一般的には、基準点が一つであれば、ジオメトリ軸または付加軸の識別子を機械軸識別子の代わりに使用することもできます。

軸グループ					
識別子	パラメータ				意味
GEOAX	1 番目	2 番目	3 番目/5 番 目	4 番目/6 番 目	平行座標系の選択
	INT: ジオメトリ 軸の番号 1...3	AXIS: チャンネル軸 識別子	1 と同様	2 と同様	

5.4 予約処理

軸グループ				
識別子		パラメータ		意味
FGROUP	1 番目～8 番目		可変 F 基準値:軌跡送り速度の基準とする軸の定義 最大軸数:8 F 基準値の初期設定は、パラメータなしの FGROUP ()で有効になります。	
	AXIS: チャンネル軸識別子			
SPLINEPATH	1 番目	2 番目～9 番目		スプライングループの定義 最大軸数:8
	INT: スプライングループ(1に固定)	AXIS: 追加の識別子のジオメトリ		
POLYPATH	1 番目	2 番目		選択した軸グループに対する多項式補間の起動
	STRING	STRING		

連結移動							
識別子	パラメータ						意味
	1 番目	2 番目	3 番目	4 番目	5 番目	6 番目	
TANG	AXIS:軸 名称 スレー ブ軸	AXIS: マスタ 軸 1	AXIS: マスタ 軸 2	REAL: 連結係 数	CHAR: オプシ ョン: "B":BCS での法 線方向 制御 "W":WC S での法 線方向 制御	CHAR 調整: "S":標準 "P": 丸み付 け隙 間、角 度許容 範囲に よる自 動調整	法線方向制御:連結の定義 法線方向制御は、2 つのマスタ 軸を指定して特定します。連結 係数で、接線の角度変更とスレ ーブ軸の変化の関係を指定しま す。この係数は通常、1 です。
TANGON	AXIS:軸 名称 スレー ブ軸	REAL:オ フセッ ト角度	REAL: 丸み付 け隙間	REAL: 角度許 容範囲			法線方向制御:連結の有効化
TANGOF	AXIS:軸 名称 スレー ブ軸						法線方向制御:連結の解除
TLIFT	AXIS:追 従軸						法線方向制御:挿入ブロック生 成の有効化
TRAILON	AXIS:ス レーブ 軸	AXIS:マ スタ軸	REAL:連 結係数				非同期連結移動のオフ
TRAILOF	AXIS:ス レーブ 軸	AXIS:マ スタ軸					連結軸移動の解除
TANGDEL	AXIS:ス レーブ 軸						法線方向制御:連結の解除

5.4 予約処理

カーブテーブル						
識別子	パラメータ					意味
	1 番目	2 番目	3 番目	4 番目	5 番目	
CTABDEF	AXIS: スレーブ 軸	AXIS: マスタ軸	INT: テーブル 番号	INT: 定義範囲 の端での 動作	STRING: 保存先の 指定	テーブル定義のオン 後続の移動ブロックでカーブ テーブルを定義します
CTABEND	AXIS: スレーブ 軸	AXIS: マスタ軸	INT: テーブル 番号	INT: 定義範囲 の端での 動作		テーブル定義のオフ
CTABDEL	INT: テーブル 番号 n	INT: テーブル 番号 m	STRING: 保存先の 指定			カーブテーブルのクリア
CTABLOCK	INT: テーブル 番号 n					番号 n のカーブテーブルをロ ックします。つまり、このテ ーブルは削除/上書きできませ ん。
CTABUNLOCK	INT: テーブル 番号 n					CTABLOCK で保護された番号 n のテーブルをロック解除しま す。
LEADON	AXIS: スレーブ 軸	AXIS: マスタ軸	INT: テーブル 番号			軸間連動機能のオン
LEADOF	AXIS: スレーブ 軸	AXIS: マスタ軸				軸間連動機能のオフ

軸の加減速方法		
識別子	パラメータ	意味
	1 番目～8 番目	
BRISKA	AXIS	プログラム指令軸で階段上の軸加減速を起動します。
SOFTA	AXIS	プログラム指令軸で加々速度制限付きの軸加減速を起動します。
DRIVEA	AXIS	プログラム指令軸で膝形加減速特性を起動します。
JERKA	AXIS	プログラム指令軸でマシンデータ \$MA_AX_JERK_ENABLE で設定した加減速動作を起動します。

毎回転送り速度			
識別子	パラメータ		意味
FPRAON	1 番目	2 番目	軸の毎回転送り速度のオン
	AXIS: 毎回転送り速度を起動する軸	AXIS: 毎回転送り速度を計算する軸/主軸 軸をプログラム指令していない場合は、毎回転送り速度がメイン主軸で計算されます。	
FPRAOF	1 番目～n 番目		軸の毎回転送り速度のオフ 毎回転送り速度は、複数の軸で一度に解除できます。1 ブロックに使用可能な限りの軸数をプログラム指令できます。
	AXIS: 毎回転送り速度を解除する軸		

5.4 予約処理

毎回転送り速度			
識別子	パラメータ		
FPR	1 番目		G95 での軌跡の毎回転送り速度を計算する回転軸または主軸を選択します。 FPR による設定はモータルです。
	AXIS: 毎回転送り速度を計算する軸/主軸 軸をプログラム指令していない場合は、毎回転送り速度がメイン主軸で計算されます。		

座標変換				
識別子	パラメータ			意味
	1 番目	2 番目	3 番目	
TRACYL	REAL: 作業直径	INT: 座標変換の番号		円筒:円筒補間 チャンネル毎に複数の座標変換を設定できます。座標変換番号で、有効にする座標変換を指定します。2 番目のパラメータを省略した場合は、MD で設定した座標変換グループが有効になります。
TRANSMIT	INT: 座標変換の番号			極座標変換:極座標変換 チャンネル毎に複数の座標変換を設定できます。座標変換番号で、有効にする座標変換を指定します。このパラメータを省略した場合は、マシンデータで定義した座標変換グループを有効にします。

座標変換				
識別子	パラメータ			意味
	1 番目	2 番目	3 番目	
TRAANG	REAL: 角度	INT: 座標変換の 番号		傾斜軸座標変換 チャンネル毎に複数の座標変換を設定できます。座標変換番号で、有効にする座標変換を指定します。2 番目のパラメータを省略した場合は、MD で設定した座標変換グループが有効になります。 角度をプログラム指令しない場合 (TRAANG (,2)または TRAANG)、最後に指令した角度がモデルに適用されます。
TRAORI	INT: 座標変換の 番号			4 軸、5 軸座標変換 チャンネル毎に複数の座標変換を設定できます。座標変換番号で、有効にする座標変換を指定します。
TRACON	INT: 座標変換の 番号	REAL:その他 のパラメータは、マシンデータにより異なります		座標変換重畳 パラメータの意味は、重畳のタイプに応じて異なります。
TRAFOOF				座標変換の解除
TRAFOON	STRING: 座標変換データセット の名称	REAL: 基準または 作業直径 (TRACYL のみ)	BOOL: 溝壁補正あり/なし (TRACYL のみ)	キネティックチェーンによって定義された座標変換を有効化

5.4 予約処理

主軸			
識別子	パラメータ		意味
	1	2 番目～n 番目	
SPCON	INT: 主軸番号	INT: 主軸番号	位置制御された主軸運転に切り替えます。
SPCOF	INT: 主軸番号	INT: 主軸番号	速度制御された主軸運転に切り替えます。
SETMS	INT: 主軸番号		現在のチャンネルにメイン主軸としての主軸を宣言します。 SETMS()を使用すると、マシンデータの設定が自動的に適用されるため、パラメータを設定する必要はありません。

研削加工		
識別子	パラメータ	
	1 番目	
GWPSON	INT: 主軸番号	砥石周速度一定(制御)のオン 主軸番号をプログラム指令しない場合は、動作中の工具の主軸に対して砥石周速度を選択します。
GWPSOF	INT: 主軸番号	砥石周速度一定(制御)のオフ 主軸番号をプログラム指令しない場合は、動作中の工具の主軸に対して砥石周速度を選択解除します。
TMON	INT: T 番号	研削用工具監視のオン T 番号をプログラム指令しない場合は、動作中の工具に対して監視が有効になります。
TMOF	INT: T 番号	工具監視オフ T 番号をプログラム指令しない場合は、動作中の工具に対して監視を解除します。

切削					
識別子	パラメータ				意味
	1 番目	2 番目	3 番目	4 番目	
CONTPRON	REAL [,11]: 輪郭テーブル	CHAR:加工 タイプ	INT: レリーフカ ットの数	INT: 演算の状態	輪郭解析の起動 次のステップで呼び出す輪郭プログラムまたは NC ブロックを、個々の移動に分割し、輪郭テーブルに格納します。 レリーフカットの数が返されます。
CONTDCON	REAL [, 6]: 輪郭テーブル	INT: 加工方向			輪郭の解読 各ブロックを 1 テーブル行として、名称を付けたテーブルに輪郭のブロックを格納し、符号化してメモリに保存します。
EXECUTE	INT:エラー 状態				プログラムの実行の起動 これにより、基準点編集モードから、またはプロテクションゾーンのセットアップ後に、元の通常のプログラム実行に切り替わります。

テーブルの実行		
識別子	パラメータ	意味
	1 番目	
EXECTAB	REAL [11]: 移動テーブルの要素	移動テーブルから要素を実行

5.4 予約処理

プロテクションゾーン						
識別子	パラメータ					意味
	1 番目	2 番目	3 番目	4 番目	5 番目	
CPROTDEF	INT: プロテクションゾーンの番号	BOOL: TRUE : 工具基準プロテクションゾーン	INT: 0: 4 番目と 5 番目のパラメータを評価しない 1: 4 番目のパラメータを評価する 2: 5 番目のパラメータを評価する 3: 4 番目と 5 番目のパラメータを評価する	REAL:正方向への制限	REAL:負方向への制限	チャンネル別プロテクションゾーンの定義

プロテクションゾーン						
識別子	パラメータ					意味
	1 番目	2 番目	3 番目	4 番目	5 番目	
NPROTDEF	INT: プロテクションゾーンの番号	BOOL: TRUE : 工具基準プロテクションゾーン	INT: 0: 4 番目と 5 番目のパラメータを評価しない 1: 4 番目のパラメータを評価する 2: 5 番目のパラメータを評価する 3: 4 番目と 5 番目のパラメータを評価する	REAL:正方向への制限	REAL:負方向への制限	機械別プロテクションゾーンの定義

5.4 予約処理

プロテクションゾーン						
識別子	パラメータ					意味
	1 番目	2 番目	3 番目	4 番目	5 番目	
CPROT	INT: プロテク ションゾ ーンの番 号	INT:オプ ション 0:プロテ クション ゾーンの オフ 1:プロテ クション ゾーンを 起動待 ちにする 2:プロテ クション ゾーンの オン 3:オプシ ョナルス トップに よるプロ テクショ ンゾーンの 起動待ち (有効な プロテク ションゾ ーンのみ)	REAL:1 番 目のジオ メトリ軸 のプロテ クション ゾーンの オフセッ ト	REAL:2 番 目のジオ メトリ軸 のプロテ クション ゾーンの オフセッ ト	REAL:3 番 目のジオ メトリ軸 のプロテ クション ゾーンの オフセッ ト	チャンネル 別プロテ クション ゾーンの オン/オフ

プロテクションゾーン						
識別子	パラメータ					意味
	1 番目	2 番目	3 番目	4 番目	5 番目	
NPROT	INT: プロテクシ ョンゾーン の番号	INT:オプシ ョン 0:プロテク ションゾー ンのオフ 1:プロテク ションゾー ンを起動待 ちにする 2:プロテク ションゾー ンのオン 3:オプショ ナルストッ プによるプ ロテクショ ンゾーンの 起動待ち(有 効なプロテ クションゾ ーンのみ)	REAL:1 番目 のジオメト リ軸のプロ テクション ゾーンのオ フセット	REAL:2 番目 のジオメト リ軸のプロ テクション ゾーンのオ フセット	REAL:3 番目 のジオメト リ軸のプロ テクション ゾーンのオ フセット	機械別プロテクショ ンゾーンのオン/オ フ

先読み/シングルブロック		
識別子	パラメータ	意味
STOPRE		解析されたすべてのブロックがメインランで実行 されるまで先読み停止
SBLOF		シングルブロック処理をマスクします
SBLON		シングルブロック処理のマスクを取り消します

5.4 予約処理

割り込み		
識別子	パラメータ	意味
	1 番目	
DISABLE	INT: 割り込み入力 の番号	指定したハードウェア入力に割り当てた割り込みルーチンを解除します。高速リトラクトは実行されません。ハードウェア入力と割り込みルーチンの SETINT による割り当ては保持されるため、ENABLE で再度有効にできます。
ENABLE	INT: 割り込み入力 の番号	DISABLE で解除した割り込みルーチンの割り当てを再度有効にします。
CLRINT	INT: 割り込み入力 の番号	割り込みルーチンと属性の、割り込み入力への割り当てを解除します。割り込みルーチンを解除します。割り込みの発生時には応答はありません。

シンクロナイズドアクション		
識別子	パラメータ	意味
	1 番目～n 番目	
CANCEL	INT: シンクロナイズドアクションの番号	指定した ID のモーダルシンクロナイズドアクションを中止します。複数の ID をコンマで区切って指定できます。
CANCELSUB		現在のサブプログラムレベルのキャンセル

関数定義					
識別子	パラメータ				意味
	1 番目	2 番目	3 番目	4 番目～7 番目	
FCTDEF	INT: 関数番号	REAL: 下限値	REAL: 上限値	REAL: 係数 a0～a3	多項式関数の定義 これは、SYFCT または PUTFTOCF で評価されます。

通信						
識別子	パラメータ					意味
	1 番目	2 番目				
MMC	STRING: 命令	CHAR: 応答モード *) "N":応答なし "S":同期応答 "A":非同期応答				NC プログラムでウインドウ画面を設定するための、HMI 命令インタプリタに対する命令
	1 番目	2 番目	3 番目	4 番目	5 番目	
DRVPRD	VAR INT: 状態	AXIS: 機械軸名称	INT: ドライブパラメータ番号	INT: ドライブパラメータインデックス	VAR REAL: ドライブパラメータの読み取り値	パートプログラムレベルにおけるドライブパラメータの読み取り
DRVPCR	VAR INT: 状態	AXIS: 機械軸名称	INT: ドライブパラメータ番号	INT: ドライブパラメータインデックス	REAL: 書き込むドライブパラメータの値	パートプログラムレベルにおけるドライブパラメータの書き込み

） 実行中のコンポーネント(チャンネル、NC など)からの要求により、命令に応答します。

5.4 予約処理

プログラム協調					
識別子		パラメータ		意味	
INIT	1 番目	2 番目	3 番目		チャンネルで実行するNCプログラムの選択
	INT: チャンネル番号 または MD20000 *のチャンネル名称)	STRING: パスの指定	CHAR: 応答モード**)		
	1 番目～n 番目				
START	INT: チャンネル番号 または MD20000 のチャンネル名称*)		複数のチャンネルで、実行中のプログラムから選択プログラムを同時に起動します この命令は、そのチャンネルには無効です		
WAITE	INT: チャンネル番号 または MD20000 のチャンネル名称*)		複数の他のチャンネルのプログラム終了を待機します		
	1 番目	2 番目～n 番目			
WAITM	INT: マーク番号	INT: チャンネル番号 または MD20000 のチャンネル名称*)		指定したチャンネルのマークに到達するまで待機します その前のブロックをイグザクトストップで終了します	
WAITMC	INT: マーク番号	INT: チャンネル番号 または MD20000 のチャンネル名称*)		指定したチャンネルのマークに到達するまで待機します イグザクトストップは、他のチャンネルがマークに到達していない場合にのみ開始します	

プログラム協調		
識別子	パラメータ	意味
	1 番目～n 番目	
SETM	INT: マーク番号	チャンネル協調用の複数のマークを設定します これはそのチャンネル内の処理には影響しません
CLEARM	INT: マーク番号	チャンネル協調用の複数のマークを削除します これはそのチャンネル内の処理には影響しません
	1 番目～n 番目	
WAITP	AXIS: 軸識別子	以前に POSA でプログラム指令した指定された位置決め軸が、そのプログラム指令された終点に達するまで待機します
WAITS	INT: 主軸番号	以前に SPOSA でプログラム指令した指定された主軸が、そのプログラム指令された終点に達するまで待機します

5.4 予約処理

プログラム協調					
識別子	パラメータ				意味
RET	1 番目	2 番目	3 番目	4 番目	PLC に機能を出力しないサブプログラムの終点です 最初のパラメータ(ジャンプ先)を指定した場合、最初に呼び出しブロックの後のブロックへの復帰ジャンプがおこなわれます。次に、プログラム指令(RET または RETB)に応じて次の方法に従ってジャンプ先が検索されます。 • RET: プログラムの末尾方向に検索します。見つからなかった場合は、プログラムの先頭方向に検索がおこなわれます。 • RETB: プログラムの先頭方向に検索します。見つからなかった場合は、プログラムの末尾方向に検索がおこなわれます。
	INT (または STRING): 復帰ジャンプ先(ブロック番号/マーク)	INT: 0: 1 番目のパラメータのジャンプ先への復帰ジャンプ > 0: 後続のブロックに戻る	INT: スキップするサブプログラムレベルの数	BOOL: メインプログラムの最初のブロックに戻る	
RETB	INT (または STRING): 復帰ジャンプ先(ブロック番号/マーク)	INT: 0: 1 番目のパラメータのジャンプ先への復帰ジャンプ > 0: 後続のブロックに戻る	INT: スキップするサブプログラムレベルの数	BOOL: メインプログラムの最初のブロックに戻る	
	1 番目～n 番目				
GET	AXIS: 軸識別子***)				機械軸(軸)の割り当て 指定した軸は、RELEASE を使用して他のチャンネルで解放してください
GETD	AXIS: 軸識別子***)				機械軸(軸)を直接割り当て 指定した軸は RELEASE を使用して解放しないでください

プログラム協調					
識別子	パラメータ				意味
RELEASE	AXIS: 軸識別子***)				機械軸(複数軸)の解放
	1 番目	2 番目	3 番目	4 番目	
PUTFTOC	REAL: オフセッ ト値	INT: パラメー タ番号	INT: チャネル 番号 または MD20000 *のチャネ ル名称 ¹⁾	INT:主軸番 号	精密工具補正の変更
PUTFTOCF	INT: 関数の番 号	VAR REAL: 指令値	INT:パラ メータ番 号	INT: チャネル 番号 または MD20000 *のチャネ ル名称 ¹⁾	FCTDEF で定義した関数に従って精密工 具補正を変更します (3 次までの多項式) ここで使用する番号は FCTDEF で指定し てください。
AXTOCHAN	1 番目	2 番目	3 番目～n 番目	4 番目～ m 番目	別のチャンネルに移行する軸
	AXIS: 軸識別子	INT: チャネル 番号 または MD20000 *のチャネ ル名称 ¹⁾	1 と同様...	2 と同様...	

¹⁾ チャンネル番号の代わりに、MD20000 \$MC_CHAN_NAME で定義したチャンネル名称をプログラム指令することもできます。

**) 実行中のコンポーネント(チャンネル、NC など)からの要求により、命令に応答します。

***) SPI 機能を使用して、軸の代わりに主軸をプログラム指令することができます。例: GET(SPI(1))

5.4 予約処理

データアクセス		
識別子	パラメータ	意味
CHANDATA	1 番目	チャンネルデータアクセス用のチャンネル番号を設定します(初期化ブロックでのみ使用できます)。その後のアクセスは CHANDATA で設定したチャンネルを参照します。
	INT: チャンネル番号	
NEWCONF		変更したマシンデータを有効にします

メッセージ			
識別子	パラメータ		意味
	1 番目	2 番目	
MSG	STRING: メッセージ	INT: 実行	任意の文字列をメッセージとして操作画面に出力します
WRTPR	STRING: 文字列	INT: 実行	OPI 変数での文字列の書き込み(WRTPR)

ファイルアクセス						
識別子	パラメータ					意味
READ	1 番目	2 番目	3 番目	4 番目	5 番目	ファイルシステムからブロックを読み取ります
	VAR INT: エラー	CHAR[160]: ファイル名称	INT: 読み取るファイル区間の開始行	INT: 読み取る行数	VAR CHAR[255]: 読み取った情報を保存する配列変数	

ファイルアクセス						
識別子	パラメータ					意味
WRITE	1 番目	2 番目	3 番目	4 番目		ファイルシステム (または外部機器/フ ァイル)にブロック を書き出します
	VAR INT: エラー	CHAR[160]: ファイル名 称	STRING: 外部出力用 の機器/ファ イル	CHAR[200]: ブロック		
DELETE	1 番目	2 番目				ファイルを削除しま す。
	VAR INT: エラー	CHAR[160]: ファイル名 称				

アラーム			
識別子	パラメータ		意味
	1 番目	2 番目	
SETAL	INT: アラーム番 号(サイクル アラーム)	STRING: 文字列	アラームの設定 アラーム番号の他に、4 個までのパラメータを含む文字列を指定できます。 以下の予約パラメータを使用できます。 %1 =チャンネル番号 %2 =ブロック番号、ラベル %3 =サイクルアラームのテキストインデックス %4 =その他のアラームパラメータ

工具管理機能				
識別子	パラメータ			意味
	1 番目	2 番目		

5.4 予約処理

工具管理機能							
識別子	パラメータ						意味
DELDL	INT: T 番号	INT: D 番号					工具刃先(または、D を指定していない場合は工具)のすべての追加オフセットを削除します。
DELT	STRING[32]: 工具識別子	INT: 予備工具番号					工具の削除 予備工具番号は省略できます。
DELTC	INT: データセット番号 n	INT: データセット番号 m					工具ホルダのデータセット番号 n～m を削除します
DZERO							チャンネルに割り当てた工具オフセットユニットのすべての工具の D 番号を、無効に設定します。
	1 番目	2 番目	3 番目	4 番目	5 番目	6 番目	
GETFREELOC	VAR INT: マガジン番号(戻り値)	VAR INT: ロケーション番号(戻り値)	INT: T 番号	INT: 参照マガジン番号	CHAR: 4 番目のパラメータによる指定	INT: 予約モード	工具用の空きロケーションを検索します
	1 番目	2 番目					
GETSELT	VAR INT: T 番号(戻り値)	INT: 主軸番号					主軸用に事前選択した工具の T 番号を返します
GETEXET	VAR INT: T 番号(戻り値)	INT: 主軸番号					NC プログラムからみた有効な工具の T 番号を返します

工具管理機能						
識別子	パラメータ					意味
GETTENV	STRING: 工具環境の 名称	INT ARRAY[3]: 戻り値				工具環境に保存され た T、D、および DL 番号を読み取ります
	1 番目	2 番目	3 番目	4 番目		
POSM	INT: 位置決めの ロケーションの番号	INT: 移動する マガジンの 番号	INT: 内部マガ ジンのロ ケーションの番号	INT:内部マ ガジンの マガジン 番号		マガジンの位置決め
RESETMON	VAR INT: 状態=運転の 結果(戻り 値)	INT:内部の T 番号	INT: 工具の D 番号	INT: ビット指 定任意選 択パラメ ータ		工具の現在位置を指 令値に設定します
SETDNO	1 番目 INT: T 番号	2 番目 INT:刃先番 号	3 番目 INT: D 番号			工具(T)の刃先のオ フセット番号(D)を 設定します
SETMTH	1 番目 INT: 工具ホルダ 番号					工具ホルダ番号を設 定します
SETPIECE	1 番目 INT: 減算時に使 用する値	2 番目 INT:主軸番 号				主軸のワークカウン タを減算します 製造工程に関連する 工具のカウント監視 データを更新します
	1 番目	2 番目	3 番目	4 番目		

5.4 予約処理

工具管理機能						
識別子	パラメータ					意味
SETTA	VAR INT: 状態=運転の 結果(戻り 値)	INT:マガジ ン番号	INT:摩耗グ ループ番 号	INT: 工具のサ ブグルー プ		摩耗グループから工 具を起動
SETTIA	VAR INT: 状態=運転の 結果(戻り 値)	INT:マガジ ン番号	INT:摩耗グ ループ番 号	INT: 工具のサ ブグルー プ		摩耗グループから工 具を解除します
TCA	1 番目	2 番目	3 番目			工具状態を問わない 工具選択/工具交換
	STRING[32]: 工具識別子	INT: 予備工具 番号	INT: 工具ホル ダ番号			
TCI	1 番目	2 番目				工具を工具バッファ からマガジンへロー ドします
	INT: バッファの 番号	INT: 工具ホル ダ番号				
MVTOOL	1 番目	2 番目	3 番目	4 番目	5 番目	工具を移動するた めの言語命令
	INT: 状態	INT:マガジ ン番号	INT: ロケーシ ョン番号	INT:移動後 のマガジ ン番号	INT:移動 後のタ ーゲッ トロケ ーショ ン番号	

工具オリエンテーション				
識別子	パラメータ			意味
	1 番目	2 番目	3 番目	
ORIRESET	REAL: 初期設定、1 番目のジオ メトリ軸	REAL: 初期設定、2 番目のジオ メトリ軸	REAL: 初期設定、3 番目のジオ メトリ軸	工具オリエンテーションの初期設定

主軸同期							
識別子	パラメータ						意味
	1 番目	2 番目	3 番目	4 番目	5 番目	6 番目	
COUPDEF	AXIS: スレー ブ主軸	AXIS: マスタ 主軸	REAL: 伝達比の 分子	REAL: 伝達比の 分母	STRING[8] : ブロック 切り替え 動作	STRING[2] : 連結タイ プ	主軸同期グループ を定義します
COUPDEL	AXIS: スレー ブ主軸	AXIS: マスタ 主軸					主軸同期グループ を解除します
COUPRES	AXIS: スレー ブ主軸	AXIS: マスタ 主軸					連結パラメータ を、設定した MD 値と SD 値にリセッ トします

5.4 予約処理

主軸同期							
識別子	パラメータ						意味
	1 番目	2 番目	3 番目	4 番目	5 番目	6 番目	
COUPON	AXIS: スレー ブ主軸	AXIS: マスタ 主軸	REAL: スレーブ 主軸のス イッチオ ン位置				主軸同期連結を開始します スレーブ主軸のス イッチオン位置(回 転の正方向への LS の 0°位置(アブソリ ュートまたはイン クレメンタルで)を 基準とする FS と LS 間の角度オフセッ ト)を指定すると、 連結は指定位置を 通過した時点での みスイッチオンさ れます。
COUPONC	AXIS: スレー ブ主軸	AXIS: マスタ 主軸					主軸同期連結を開始します COUPONC により、 連結をオンにした ときに、スレーブ 主軸の現在有効な 速度が引き継がれ ます (M3/M4 S...)

主軸同期							
識別子	パラメータ						意味
	1 番目	2 番目	3 番目	4 番目	5 番目	6 番目	
COUPOF	AXIS: スレー ブ主軸	AXIS: マスタ 主軸	REAL: スレーブ 主軸のス イッチオ フ位置(ア ブソリュ ート)	REAL: マスタ主 軸のスィ ッチオフ 位置(アブ ソリュ ート)			主軸同期連結を解 除します 位置を指定した場 合は、指定した位 置をすべて越えて 移動した場合にの み、連結が解除さ れます。 スレーブ主軸は、 連結の解除前の最 後のプログラム指 令速度で回転し続 けます。
COUPOFS	AXIS: スレー ブ主軸	AXIS: マスタ 主軸	REAL: スレーブ 主軸のス イッチオ フ位置(ア ブソリュ ート)				スレーブ主軸の停 止で主軸同期連結 を解除します 位置を指定した場 合は、指定した位 置を通過した場合 にのみ、連結が解 除されます。
WAITC	AXIS: スレー ブ主軸	STRING [8]: ブロッ ク切り 替え動 作	AXIS: スレーブ 主軸	STRING[8] : ブロッ ク切り 替え 動作			主軸(最大 2)の連結 ブロック切り替え の条件が満たされ るまで待機します。 ブロック切り替え 動作を指定しない 場合は、COUPDEF の定義で指定した ブロック切り替え 動作が適用されま す。

5.4 予約処理

電子ギヤ						
識別子	パラメータ					意味
EGDEL	1 番目					スレーブ軸 の連結定義 を解除しま す
	AXIS: スレー ブ軸					
EGDEF	1 番目	2 番目/4 番目/6 番目/8 番目/10 番目	3 番目/5 番目/7 番目/9 番目/11 番目			電子ギヤの 定義
	AXIS: スレー ブ軸	AXIS: マスタ 軸	INT: 連結タ イプ			
EGON	1 番目	2 番目	3 番目/6 番目/9 番目/12 番目/15 番目	4 番目/7 番目/10 番目/13 番目/16 番目	5 番目/8 番目/11 番目/14 番目/17 番目	電子ギヤを 同期せずに 起動します
	AXIS: スレー ブ軸	STRING: ブロッ ク切り 替え動 作	AXIS: マスタ 軸	REAL: 連結係 数の分 子	REAL: 連結係 数の分 母	

電子ギヤ									
識別子	パラメータ								意味
EGONSYN	1 番目	2 番目	3 番目	4 番目/8 番目/12 番目/16 番目/20 番目	5 番目/9 番目/13 番目/17 番目/21 番目	6 番目/10 番目/14 番目/18 番目/22 番目	7 番目/11 番目/15 番目/19 番目/23 番目		電子ギヤを同期して起動します
	AXIS: スレーブ軸	STRING: ブロック切り替え動作	REAL: スレーブ軸の同期位置	AXIS: マスタ軸	REAL: マスタ軸の同期位置	REAL: 連結係数の分子	REAL: 連結係数の分母		
EGONSYNE	1 番目	2 番目	3 番目	4 番目	5 番目/9 番目/13 番目/17 番目/21 番目	6 番目/10 番目/14 番目/18 番目/22 番目	7 番目/11 番目/15 番目/19 番目/23 番目	8 番目/12 番目/16 番目/20 番目/24 番目	電子ギヤを同期してアプローチモードを指定して起動します
	AXIS: スレーブ軸	STRING: ブロック切り替え動作	REAL: スレーブ軸の同期位置	STRING: アプローチモード	AXIS: マスタ軸	REAL: マスタ軸の同期位置	REAL: 連結係数の分子	REAL: 連結係数の分母	
EGOFS	1 番目	2 番目～n 番目							電子ギヤの選択解除
	AXIS: スレーブ軸	AXIS: マスタ軸							
EGOFC	1 番目								電子ギヤを解除します (主軸にのみ使用可能)
	AXIS: スレーブ主軸								

5.4 予約処理

ニブリング					
識別子	パラメータ				意味
	1 番目	2 番目	3 番目	4 番目	
PUNCHAAC	REAL: 穴の間隔の 最小値	REAL: 初期加減速 度	REAL: 穴の間隔の 最大値	REAL: 最終加減速度	移動に依存した加減速度を起 動します

パッシブファイルシステムの情報機能				
識別子	パラメータ			意味
	1 番目	2 番目	3 番目	
FILEDATE	VAR INT: エラーメッ セージ	CHAR[160]: ファイル名 称	VAR CHAR[8]: 「dd.mm.yy 」形式の日 付	ファイルへの最後の書き込みアクセスの日付 を返します
FILETIME	VAR INT: エラーメッ セージ	CHAR[160]: ファイル名 称	VAR CHAR[8]: 「hh.mm.ss 」形式の時 刻	ファイルへの最後の書き込みアクセスの時刻 を返します
FILESIZE	VAR INT: エラーメッ セージ	CHAR[160]: ファイル名 称	VAR INT: ファイルサ イズ	ファイルの現在のサイズを返します

パッシブファイルシステムの情報機能				
識別子	パラメータ			意味
	1 番目	2 番目	3 番目	
FILESTAT	VAR INT: エラーメッセージ	CHAR[160]: ファイル名称	VAR CHAR[5]: 「rwxsd」形式の日付	次の権限の中からファイルの状態を返します。 - 読み取り(r:read) - 書き込み(w:write) - 実行(x:execute) - 表示(s:show) - 削除(d:delete)
FILEINFO	VAR INT: エラーメッセージ	CHAR[160]: ファイル名称	VAR CHAR[32]: 「rwxsd nnnnnnnn dd.mm.yy hh:mm:ss」形式の日付	FILEDATE、FILETIME、FILESIZE、および FILESTAT によって読み出し可能なファイルの すべての情報を返します

軸コンテナ		
識別子	パラメータ	意味
	1 番目～n 番目	
AXCTSWE	AXIS: 軸コンテナ	軸コンテナの回転
AXCTSWED	AXIS: 軸コンテナ	軸コンテナの回転(セットアップ用の命令タイプ)
AXCTSWEC:	AXIS: 軸コンテナ	軸コンテナの回転の有効化を取り消し

マスタ/スレーブ連結		
識別子	パラメータ	意味
	1 番目～n 番目	
MASLON	AXIS: 軸識別子	マスタ/スレーブ連結を起動
MASLOF	AXIS: 軸識別子	マスタ/スレーブ連結を解除

5.4 予約処理

マスタ/スレーブ連結		
識別子	パラメータ	意味
	1 番目～n 番目	
MASLOFS	AXIS: 軸識別子	マスタ/スレーブ連結を解除して、スレーブ主軸を自動的に減速
MASLDEF	AXIS: 軸識別子	マスタ/スレーブ連結の定義 最後の軸がマスタ軸です
MASLDEL	AXIS: 軸識別子	マスタ/スレーブ連結の解除とグループの定義の削除

オンライン工具長補正			
識別子	パラメータ		意味
	1 番目	2 番目	
TOFFON	AXIS: オフセット 方向	REAL: オフセット 方向でのオ フセット値	指定したオフセット方向でのオンライン工具長補正を有効に します
TOFFOF	AXIS: オフセット 方向		指定したオフセット方向でのオンライン工具長補正をリセッ トします

SERUPRO		
識別子	パラメータ	意味
IPTRLOCK		検索不可プログラム区間の開始
IPTRUNLOCK		検索をマスクするプログラム区間の終了

後退		
識別子	パラメータ	意味
	1 番目～n 番目	
POLFMASK	AXIS: ジオメトリ軸または機械軸の名称	(軸間の関係なしに)高速リトラ クト用の軸を有効化

後退				
識別子	パラメータ			意味
POLFMLIN	AXIS: ジオメトリ軸または機械軸の名称			直線高速リトラクト用の軸を有効化
POLFA	1 番目 AXIS: チャンネル軸識別子	2 番目 INT: タイプ	3 番目 REAL: 値	一軸の後退位置

干渉の回避			
識別子	パラメータ		意味
	1 番目		
PROTA	STRING: "R"		干渉モデルの再計算の要求
PROTS	1 番目 CHAR: 状態	2 番目～n 番目 STRING: プロテクションゾーン 名称	プロテクションゾーンの状態の設定

インテリジェント負荷調整					
識別子	パラメータ				意味
CADAPTON	1 番目 INT: 状態	2 番目 AXIS: 機械軸名称	3 番目 INT: 入力変数	4 番目 REAL: 入力変数(オプション)	負荷調整の有効化

5.5 シンクロナイズドアクションでの予約処理

インテリジェント負荷調整					
識別子	パラメータ				意味
CADAPTOF	1 番目	2 番目	3 番目		負荷調整の無効化
	INT: 状態	AXIS: 機械軸名称	INT: 入力変数		

加々速度の調整		
識別子	パラメータ	意味
CALCFIR		FIR 加々速度フィルタのダイナミック応答モードへの設定 CALCFIR は CUST_832 メーカサイクルで使用します。
AFISON		自動フィルタチェーン切り替えの起動
AFISOF		自動フィルタチェーン切り替えの解除

5.5 シンクロナイズドアクションでの予約処理

次の予約処理はシンクロナイズドアクションでのみ使用できます。

同期処理		
識別子	パラメータ	意味
STOPREOF		先読み停止の無効化 STOPREOF 命令を含むシンクロナイズドアクションを実行すると、次の出力ブロック(=メインランのブロック)の後に先読み停止がおこなわれます。先読み停止は、出力ブロックの終了、または STOPREOF 条件が満たされると、キャンセルされます。この場合、STOPREOF 命令を含むすべてのシンクロナイズドアクション動作は、実行済みと解釈されます。
RDISABLE		読み込み停止

5.5 シンクロナイズドアクションでの予約処理

同期処理		
識別子	パラメータ	意味
DELDTG	1.	残移動距離削除
	AXIS: 軸の残移動距離削除をおこなう軸 (任意選択)。軸を省略した場合は、軌跡の距離に対して残移動距離削除がおこなわれます	DELDTG 命令を含むシンクロナイズドアクションを実行すると、次の出力ブロック(=メインランのブロック)の後に先読み停止がおこなわれます。先読み停止は、出力ブロックが終了すると、または DELDTG 条件が満たされると、キャンセルされます。軸の残移動距離削除時の終点までの軸距離は、\$AA_DELT[軸] に格納されます; 残移動距離は\$AC_DELT に格納されます。

テクノロジーサイクルのプログラム協調		
識別子	パラメータ	意味
	1.	
LOCK	INT: 無効にするシンクロナイズドアクションの ID	この ID のシンクロナイズドアクションをロックするか、テクノロジーサイクルを停止します 複数の ID をプログラム指令できます
UNLOCK	INT: ロック解除するシンクロナイズドアクションの ID	この ID のシンクロナイズドアクションをロック解除するか、テクノロジーサイクルを続行します 複数の ID をプログラム指令できます
ICYCON		テクノロジーサイクルの各ブロックを、ICYCON 指令後に個別の補間サイクルで処理します。
ICYCOF		テクノロジーサイクルのすべてのブロックを、ICYCOF 指令後に 1 つの補間サイクルで処理します

5.6 予約機能

多項式関数						
識別子		パラメータ				意味
SYNFCT	1.	2.	3.			シンクロナイズドアクションの条件が満たされた場合は、1 番目の式で特定された多項式がこの入力変数で評価されます。値の上下の範囲を制限し、結果変数が代入されます。
	INT: FCTDEF で定義した多項式関数の番号	VAR REAL: 結果変数 *)	VAR REAL: 入力変数 **)			
FTOC	1.	2.	3.	4.	5.	FCTDEF で定義した機能に従って精密工具補正を変更します(3 次までの多項式)。ここで使用する番号は FCTDEF で指定してください。
	INT: FCTDEF で定義した多項式関数の番号	VAR REAL: 入力変数 **)	INT: 長さ 1、2、3	INT: チャンネル番号	INT: 主軸番号	

*) 結果変数として使用できるのは特別なシステム変数のみです(機能マニュアル『シンクロナイズドアクション』を参照してください)。

**) 入力変数として使用できるのは特別なシステム変数のみです(機能マニュアル『シンクロナイズドアクション』を参照してください)。

5.6 予約機能

予約機能の呼び出しにより、NC の予約機能の実行が開始されます。これは予約処理とは異なり、戻り値を返します。予約機能の呼び出しは、演算式のオペランドにすることができます。

座標系						
識別子	戻り値	パラメータ				意味
		1.	2.	3. - 15.	4. - 16.	
CTTRANS	FRAME	AXIS: 軸識別子	REAL:オフセット	AXIS: 軸識別子	REAL:オフセット	平行移動:複数軸の汎用ゼロオフセット

座標系						
識別子	戻り値	パラメータ				意味
CFINE	FRAME	AXIS: 軸識別子	REAL:オフセ ット	AXIS: 軸識別子	REAL:オフセ ット	平行移動:複数軸の精 密ゼロオフセット
CSCALE	FRAME	AXIS: 軸識別子	REAL: スケーリン グ係数	AXIS: 軸識別子	REAL: スケーリン グ係数	スケーリング:複数軸 のスケーリング係数
		1.	2.	3.と 5.	4.と 6.	
CROT	FRAME	AXIS: 軸識別子	REAL:座標回 転	AXIS: 軸識別子	REAL:座標回 転	座標回転:現在の座標 系の回転 パラメータの最大 数:6 (ジオメトリ軸毎に 1 個の軸識別子と 1 個 の値)
CROTS	FRAME	AXIS: 軸識別子	REAL:立体角 による座標 回転	AXIS: 軸識別子	REAL:立体角 による座標 回転	座標回転:立体角によ る現在の座標系の回 転 パラメータの最大 数:6 (ジオメトリ軸毎に 1 個の軸識別子と 1 個 の値)
CMIRROR		1.	2. – 8.			ミラーリング:座標軸 のミラーリング
	FRAME	AXIS	AXIS			

5.6 予約機能

座標系					
識別子	戻り値	パラメータ			意味
		1.	2.		
CRPL	FRAME	INT: 回転軸	REAL: 回転角度		任意の平面のフレーム回転
ADDFRAME	INT: 0:OK 1:指定したターゲット(文字列)が間違っています 2:ターゲットフレームが設定されていません 3:フレームでの回転は許可されていません	FRAME: 追加で計測または計算されたフレーム	STRING: 指定したターゲットフレーム		文字列で指定したターゲットフレームを計算します 新規フレーム一式が、旧フレーム一式と追加されたフレームの結合になるように、ターゲットフレームが計算されます。
INVFRAME	FRAME	1. FRAME			フレームから逆フレームを計算します フレームと逆フレームをフレーム結合すると、結果は常にゼロフレームになります

座標系						
識別子	戻り値	パラメータ				意味
MEAFRAME	FRAME	1.	2.	3.		空間の 3 つの計測点によるフレーム計算
		REAL[3,3]: 空間の計測点の座標	REAL[3,3]: 指定した点の座標	VAR REAL: フレーム演算の特性に関する情報を返す変数		

ジオメトリ機能					
識別子	戻り値	パラメータ			意味
		1.	2.	3.	
CALCDAT	BOOL: エラー状態	VAR REAL [n, 2]: 点 1～n のテーブル(横軸、縦軸)	INT: 点の数	VAR REAL [3]: 結果:計算した円弧の中心点の横座標、縦座標、および半径	3 または 4 点から、円弧の中心点の座標と半径を計算します これらの点はそれぞれ、違う点にしてください。
INTERSEC	BOOL: エラー状態	VAR REAL [11]: 1 番目の輪郭要素	VAR REAL [11]: 2 番目の輪郭要素	VAR REAL [2]: 交点座標を表わすベクトルが得られます。横座標と縦座標	2 つの輪郭要素の交点座標を計算します。 エラー状態で、交点が見つかったかどうかを示します。

5.6 予約機能

カーブテーブル機能								
識別子	戻り値	パラメータ						意味
		1.	2.	3.	4.	5.	6.	
CTAB	REAL: スレーブ軸の位置	REAL: マスタ軸の位置	INT: テーブル番号	VAR REAL[]: ピッチ結果	AXIS: スレーブ軸のスケーリング	AXIS: マスタ軸のスケーリング		カーブテーブルから指定したマスタ軸位置に対するスレーブ軸位置を特定します。 パラメータ 4/5 をプログラム指令していない場合は、標準のスケーリングで計算されます。
CTABINV	REAL: マスタ軸の位置	REAL: スレーブ軸の位置	REAL: マスタ位置	INT: テーブル番号	VAR REAL[]: ピッチ結果	AXIS: スレーブ軸のスケーリング	AXIS: マスタ軸のスケーリング	カーブテーブルから指定したスレーブ軸位置に対するマスタ軸位置を特定します。 パラメータ 5/6 をプログラム指令していない場合は、標準のスケーリングで計算されます。
CTABID	INT: カーブテーブル番号	INT: メモリの入力番号	STRING: 保存先: "SRAM" 、 "DRAM"					指定したメモリの番号で入力されているカーブテーブル番号を特定します。

カーブテーブル機能								
識別子	戻り値	パラメータ						意味
		1.	2.	3.	4.	5.	6.	
CTABISLOCK	INT: ロック 状態	INT: テーブ ル番号						カーブテーブルのロ ック状態を特定しま す。 > 0:テーブルがロッ クされています 1:CTABLOCK 2:有効な連結 3:CTABLOCK と有効 な連結 0:カーブテーブルは ロックされていませ ん -1:カーブテーブル がありません
CTABEXISTS	INT: 存在し ます	INT: テーブ ル番号						カーブテーブルがス タティックまたはダ イナミック NC メモ リにあるかどうかを 特定します。 0:FALSE 1:TRUE
CTABMEMTYP	INT: 保存先	INT: テーブ ル番号						カーブテーブルの保 存場所を特定します 1:DRAM 0:SRAM -1:カーブテーブル がありません

5.6 予約機能

カーブテーブル機能								
識別子	戻り値	パラメータ						意味
		1.	2.	3.	4.	5.	6.	
CTABPERIOD	INT: 周期性	INT: テーブル 番号						カーブテーブルの周 期性を特定します。 0:周期的ではありません 1:マスタ軸で周期的 です 2:マスタ軸とスレー ブ軸で周期的です -1:カーブテーブル がありません
CTABNO	INT: カーブ テーブル の数							定義した(スタティ ックおよびダイナミ ック NC メモリの) カーブテーブルの 数を特定します。
CTABNOMEM	INT: カーブ テーブル の数	STRING: 保存先: "SRAM" 、 "DRAM"						指定したメモリの定 義したカーブテー ブルの数を特定し ます。
CTABFNO	INT: テーブ ルの数	STRING: 保存先: "SRAM" 、 "DRAM"						指定したメモリの引 き続き有効なカー ブテーブルの数を 特定します。

カーブテーブル機能								
識別子	戻り値	パラメータ						意味
		1.	2.	3.	4.	5.	6.	
CTABSEG	INT: カーブ セグメ ントの 数	STRING: 保存先: "SRAM" 、 "DRAM"	STRING: セグメ ントタ イプ: "L":直線 "P":多項 式					指定したメモリの指 定したセグメントタ イプの使用されるカー ブセグメントの数を 特定します >=0:番号 -1:無効なメモリタ イプ パラメータ 2 をプ ログラム指令しない 場合は、直線セグメ ントと多項式セグメ ントの合計が出力さ れます。
CTABFSEG	INT: カーブ セグメ ントの 数	STRING: 保存先: "SRAM" 、 "DRAM"	STRING: セグメ ントタ イプ: "L":直線 "P":多項 式					指定したメモリの指 定したセグメントタ イプの引き続き有効 なカーブセグメント の数を特定します >=0:番号 -1:無効なメモリタ イプ
CTABSEGID	INT: カーブ セグメ ントの 数	INT: テーブ ル番号	STRING: セグメ ントタ イプ: "L":直線 "P":多項 式					カーブテーブルによ って使用される指定 したセグメントタイ プのカーブセグメン トの数を特定します >=0:番号 -1:カーブテーブル がありません

5.6 予約機能

カーブテーブル機能								
識別子	戻り値	パラメータ						意味
		1.	2.	3.	4.	5.	6.	
CTABMSEG	INT: カーブ セグメ ントの 数	STRING: 保存先: "SRAM" 、 "DRAM"	STRING: セグメ ントタ イプ: "L":直線 "P":多項 式					指定したメモリの指 定したセグメントタ イプのカーブセグメ ントの最大数を特定 します >=0:番号 -1:カーブテーブル がありません
CTABPOL	INT: カーブ 多項式 の数	STRING: 保存先: "SRAM" 、 "DRAM"						指定したメモリの使 用されるカーブ多項 式の数进行特定しま す。 >=0:番号 -1:カーブテーブル がありません
CTABPOLID	INT: カーブ 多項式 の数	INT: テーブ ル番号						カーブテーブルで使 用されるカーブ多項 式の数进行特定しま す >=0:番号 -1:カーブテーブル がありません
CTABFPOL	INT: カーブ 多項式 の数	STRING: 保存先: "SRAM" 、 "DRAM"						指定したメモリのカー ブ多項式の最大数 を特定します。 >=0:番号 -1:カーブテーブル がありません
CTABMPOL	INT: カーブ 多項式 の数	STRING: 保存先: "SRAM" 、 "DRAM"						指定したメモリのカー ブ多項式の最大数 を特定します。 >=0:番号 -1:カーブテーブル がありません

カーブテーブル機能								
識別子	戻り値	パラメータ						意味
		1.	2.	3.	4.	5.	6.	
CTABSSV	REAL: スレーブ軸の位置	REAL: マスタ軸の位置	INT: テーブル番号	VAR REAL[]: ピッチ結果	AXIS: スレーブ軸のスケーリング	AXIS: マスタ軸のスケーリング		指定したマスタ軸値のカーブセグメントの開始位置でのスレーブ軸位置を特定します
CTABSEV	REAL: スレーブ軸の位置	REAL: マスタ軸の位置	INT: テーブル番号	VAR REAL[]: ピッチ結果	AXIS: スレーブ軸のスケーリング	AXIS: マスタ軸のスケーリング		指定したマスタ軸値のカーブセグメントの終了位置でのスレーブ軸位置を特定します
CTABTSV	REAL: スレーブ軸の位置	INT: テーブル番号	VAR REAL[]: テーブルの開始位置でのピッチ結果	AXIS: スレーブ軸				カーブテーブルの開始位置でのスレーブ軸位置を特定します。
CTABTEV	REAL: スレーブ軸の位置	INT: テーブル番号	VAR REAL[]: テーブルの終了位置でのピッチ結果	AXIS: スレーブ軸				カーブテーブルの終了位置でのスレーブ軸位置を特定します。
CTABTSP	REAL: マスタ軸の位置	INT: テーブル番号	VAR REAL[]: テーブルの開始位置でのピッチ結果	AXIS: マスタ軸				カーブテーブルの開始位置でのマスタ軸位置を特定します。

5.6 予約機能

カーブテーブル機能								
識別子	戻り値	パラメータ						意味
		1.	2.	3.	4.	5.	6.	
CTABTEP	REAL: マスタ 軸の位 置	INT: テーブ ル番号	VAR REAL[]: テーブ ルの終 了位置 でのピ ッチ結 果	AXIS: マスタ 軸				カーブテーブルの終 了位置でのマスタ軸 位置を特定します。
CTABTMIN	REAL: 最小値	INT: テーブ ル番号	REAL: マスタ 値区間 の下限	REAL: マスタ 値区間 の上限	AXIS: スレー ブ軸	AXIS: マスタ 軸		カーブテーブルの定 義範囲全体または定 義した区間でのスレ ーブ軸の最小値を特 定します
CTABTMAX	REAL: 最大値	INT: テーブ ル番号	REAL: マスタ 値区間 の下限	REAL: マスタ 値区間 の上限	AXIS: スレー ブ軸	AXIS: マスタ 軸		カーブテーブルの定 義範囲全体または定 義した区間でのスレ ーブ軸の最大値を特 定します
注: カーブテーブル機能は、シンクロナイドアクションでもプログラム指令できます。								

軸機能						
識別子	戻り値	パラメータ				意味
		1.	2.	3.	4.	
AXNAME	AXIS: 軸識別子	STRING []: 入力文字列				入力文字列を軸識別 子に変換
AXSTRING	STRING[]: 軸名称	AXIS: 軸識別子				軸識別子を文字列に 変換

軸機能						
識別子	戻り値	パラメータ				意味
		1.	2.	3.	4.	
ISAXIS	BOOL: 軸が存在する(TRUE)または存在しない(FALSE)	INT: ジオメトリ軸の番号(1～3)				マシンデータ MD20050 \$MC_AXCONF_GEO AX_ASSIGN_TAB に従って、パラメータとして指定したジオメトリ軸 1～3 が存在するかどうかを確認します
SPI	AXIS: 軸識別子	INT: 主軸番号				主軸番号を軸識別子に変換
AXTOSPI	INT: 主軸番号	AXIS: 軸識別子				軸識別子を主軸番号に変換
MODAXVAL	REAL: モジュロ値	AXIS: 軸識別子	REAL: 軸位置			入力した軸位置から残りのモジュロ値を計算します 指定した軸がモジュロではない場合は、軸位置は変更せずに返されます
POSRANGE	BOOL: 位置指令が位置範囲内にある(TRUE)またはない(FALSE)	AXIS: 軸識別子	REAL: 座標系の基準位置	REAL: 位置範囲の幅	INT: 座標系	軸の位置指令が事前定義された基準位置の範囲内にあるかどうかを特定します

5.6 予約機能

工具管理機能					
識別子	戻り値	パラメータ			意味
		1.	2.	3.	
CHKDM	INT: 状態:チェック結果:	INT: マガジン番号	INT: D 番号		D 番号がマガジン内で固有であるかどうかを確認します
CHKDNO	INT: 状態:チェック結果:	INT: 1 番目の工具の T 番号	INT: 2 番目の工具の T 番号	INT: D 番号	D 番号が固有であるかどうかを確認します
GETACTT	INT: 状態	INT: T 番号	STRING [32]: 工具名称		同じ名称を含む工具グループから、動作中の工具を特定します
GETACTTD	INT: 状態:チェック結果:	VAR INT: 検索された T 番号(戻り値)	INT: D 番号		絶対 D 番号に関連付けられた T 番号を特定します
GETDNO	INT: D 番号	INT: T 番号	INT: 刃先番号		工具 T の刃先の D 番号を特定します
GETT	INT: T 番号	STRING [32]: 工具名称	INT: 予備工具番号		工具名称の T 番号を特定します
NEWT	INT: T 番号	STRING [32]: 工具名称	INT: 予備工具番号		新しい工具を設定します(工具データを提供します) 予備工具番号は省略できます。
TOOLENV	INT: 状態	STRING: 名称			指定した名称の工具環境をスタティック NC メモリに保存します

工具管理機能					
識別子	戻り値	パラメータ			意味
		1.	2.	3.	
DELTOOLENV	INT: 状態	STRING: 名称			指定した名称の工具環境を スタティック NC メモリから 削除します 名称を指定しない場合は、 すべての工具環境を削除し ます。
GETTENV	INT: 状態	STRING: 名称	VAR INT: T 番号[0] D 番号[1] DL 番号[2]		指定した名称の工具環境から、 T 番号、D 番号、および DL 番号を特定します

算術					
識別子	戻り値	パラメータ			意味
		1.	2.	3.	
SIN	REAL	REAL			サイン
ASIN	REAL	REAL			アークサイン
COS	REAL	REAL			コサイン
ACOS	REAL	REAL			アークコサイン
TAN	REAL	REAL			タンジェント
ATAN2	REAL	REAL	REAL		アークタンジェント 2
SQRT	REAL	REAL			平方根
POT	REAL	REAL			二乗
TRUNC	REAL	REAL			整数要素
ROUND	REAL	REAL			切り下げ
ROUNDUP	REAL	REAL			切り上げ
ABS	REAL	REAL			絶対値
LN	REAL	REAL			自然対数
EXP	REAL	REAL			指数関数 e^x
MINVAL	REAL	REAL	REAL		2 個のパラメータの小さい 方の値を特定

5.6 予約機能

算術					
識別子	戻り値	パラメータ			意味
		1.	2.	3.	
MAXVAL	REAL	REAL	REAL		2 個のパラメータの大きい方の値を特定
BOUND	REAL: 状態の確認	REAL: 下限値	REAL: 上限値	REAL: 指令値	指令値が制限内にあるかどうかを特定します。
注: 算術機能はシンクロナイドアクションでもプログラム指令できます。これらの算術機能は、メインランで計算されて評価されます。シンクロナイズドアクションパラメータ \$AC_PARAM[<n>] は、計算用やバッファメモリとして使用することもできます。					

String 機能					
識別子	戻り値	パラメータ			意味
		1.	2.	3.	
ISNUMBER	BOOL	STRING: 入力文字列			入力文字列を数字に変換できるかどうかを確認します。
NUMBER	REAL	STRING: 入力文字列			入力文字列を数値に変換します。
TOUPPER	STRING	STRING: 入力文字列			入力文字列を大文字に変換します
TOLOWER	STRING	STRING: 入力文字列			入力文字列を小文字に変換します
STRLEN	INT	STRING: 入力文字列			文字列の最後(I/O)までの入力文字列の長さを特定します。
INDEX	INT	STRING: 入力文字列	CHAR: 検索文字		入力文字列の文字の位置を左から右に特定します。 文字列の左からの 1 番目の文字は、インデックス 0 です。

String 機能					
識別子	戻り値	パラメータ			意味
		1.	2.	3.	
RINDEX	INT	STRING: 入力文字列	CHAR: 検索文字		入力文字列の文字の位置を右から左に特定します。 文字列の右からの 1 番目の文字は、インデックス 0 です。
MINDEX	INT	STRING: 入力文字列	STRING: 検索文字		入力文字列の 2 番目のパラメータで指定された文字の位置を左から右に特定します。 入力文字列の左からの 1 番目の文字は、インデックス 0 です。
SUBSTR	STRING	STRING: 入力文字列	INT	INT	開始文字(2 番目のパラメータ)と文字数(3 番目のパラメータ)で定義した入力文字列のサブ文字列を特定します。
SPRINT	STRING	STRING: 入力文字列			フォーマットされた入力文字列を特定します

5.6 予約機能

計測サイクルの機能								
識別子	戻り値	パラメータ						意味
		1.	2.	3.	4.	5.	6.	
CALCPOSI	INT: 状態	REAL[3] : WCS で の開始 位置	REAL[3] : 開始位 置に対 するイン クレメン タル軌跡 の指定	REAL[5] : 監視リ ミット までの 最短距離	REAL[3] : 可能な インク レメン タル距離 に対する 戻り値の 配列	BOOL: 単位系 の変換 (はいいいえ)	INT: リミッ ト監視 のタイ プ	ジオメトリ軸が、指定した起点から開始して、軸リミットに違反せずに定義した軌跡を移動できるかどうかを確認します。 定義した軌跡を、制限に違反せずに移動できない場合は、最大許容値を返します。
GETTCOR	INT: 状態	REAL [11]:	STRING: 工具長 成分:座 標系	STRING: 工具環 境の名 称	INT: 工具の 内部 T 番号	INT: 工具の 刃先番 号(D 番 号)	INT: ロケー ション 関連の オフセ ットの 番号(工 具の DL 番号)	工具長と工具長成分を、工具環境または現在の環境から特定します
LENTOAX	INT: 状態	INT[3]: ジオメ トリ軸 の軸割 り当て	REAL[3] : 座標系 での工 具長の 割り当 てのマ トリク ス	STRING: 割り当 てに使 用する 座標系				動作中の工具の工具長 L1、L2、L3 の、横座標、縦座標、垂直座標への割り当てに関する情報を特定します。 ジオメトリ軸への割り当ては、フレームと有効平面(G17~19)によって決まります。

SETTCOR	INT: 状態	1.	2.	3.	4.	5.	6.	7.	8.	9.	
		REAL [3]: 空間 内の オフ セット ベク トル	STR.: 成分 識別 子	INT: 補正 する 成分 0～11	INT: 書き 込み 操作 のタ イプ 0～3	INT: ジオ メトリ 軸の インデ ックス	STRIN G: 工具 環境 の名 称	INT: 工具 の初期 T 番号	INT: 工具 のD 番号	INT: 工具 のDL 番号	個々の成分の評価に含まれているすべての必要条件を考慮して、工具成分を変更します。

その他の関数											
識別子	戻り値	パラメータ						意味			
		1.	2.	3.	4.	5.	6.				
STRINGIS	INT: 文字列に関する情報	STRING: 確認する要素の名称						指定した文字列を現在の言語仕様の範囲でNCプログラム言語の要素として使用できるかどうかを確認します。			
ISVAR	BOOL: 既知の変数(はい/いいえ)	STRING: 変数の名称						転送パラメータにNCの既知の変数(マシニングデータ、セッティングデータ、システム変数、GUDなどの一般変数)が含まれるかどうかを確認します。			
GETVARTYP	INT: データタイプ	STRING: 変数の名称						システム変数/ユーザー変数のデータタイプを特定します			
GETVARPHU	INT: 物理単位の数値	STRING: 変数の名称						システム変数/ユーザー変数の物理単位を特定します			

5.6 予約機能

その他の関数								
識別子	戻り値	パラメータ						意味
		1.	2.	3.	4.	5.	6.	
GETVARAP	INT: アクセスの保護レベル	STRING: 変数の名称	STRING: アクセスのタイプ					システム変数/ユーザー変数へのアクセス権を特定します
GETVARLIM	INT: 状態	STRING: 変数の名称	CHAR: 読み出す制限値を指定します	VAR REAL: 制限値の戻し				システム変数/ユーザー変数の下限値/上限値を特定します
GETVARDFT	INT: 状態	STRING: 変数の名称	VAR REAL / STRING/ FRAME: 初期値の戻し	INT: 1 番目の寸法のインデックス(任意選択)	INT: 2 番目の寸法のインデックス(任意選択)	INT: 3 番目の寸法のインデックス(任意選択)		システム変数/ユーザー変数の初期値を特定します
COLLPAIR	INT: チェック結果	STRING: 1 番目のプロテクションゾーンの名称	STRING: 2 番目のプロテクションゾーンの名称	BOOL: アラームのマスク(任意選択)				干渉ペアの要素であるかどうかをチェックします
PROTD	REAL: 2 つのプロテクションゾーンの間隔	STRING: 1 番目のプロテクションゾーンの名称	STRING: 2 番目のプロテクションゾーンの名称	VAR REAL: 値の戻し: 3 次元の間隔ベクトル	BOOL: 間隔および間隔ベクトルの検出器(任意選択)			2 つの指定されたプロテクションゾーンの間隔を特定します。

その他の関数								
識別子	戻り値	パラメータ						意味
		1.	2.	3.	4.	5.	6.	
DELOBJ	INT: 異常番号	STRING: 削除する要素 タイプ	INT: 削除する要素 タイプの開始 インデックス (任意選択)	INT: 削除する要素 タイプの終了 インデックス (任意選択)	BOOL: アラームのマ スク (オプション)			キネマティック結 合、プロテクション ゾーン、プロテクシ ョンゾーン要素、干 渉ペア、座標変換デ ータから要素を削除 します
NAMETOINT	INT: システム変数 インデ ックス	STRING: システム変数 配列の 名称	STRING: 文字列/ 名称	BOOL: アラームのマ スク (オプション)				関連するシステム変 数インデックスを文 字列に基づいて決定 します。
ORISOLH	INT: 異常番号	INT: 指令値 の動作 を制御 します	REAL: 第 1 角 度	REAL: 第 2 角 度				機械の回転軸の位置 を設定することに役 立ち、定義された、 ワークに対して動き に依存しない位置に 旋削工具を動かすこ とができます。 必要条件: キネティックチェー ンによってパラメー タ設定された 6 軸座 標変換が有効である こと。

5.7 HMI で現在設定されている言語

その他の関数								
識別子	戻り値	パラメータ						意味
		1.	2.	3.	4.	5.	6.	
CORRTAFO	INT: 異常番号	REAL: 補正ベクトル	INT: 変更対象の要素	INT: 補正モード	BOOL: アラームのマスク (オプション)			機械のキネマティックモデルの旋回軸の補正ベクトルや方向ベクトルを変更します。
CORRTC	INT: 異常番号	REAL: 補正ベクトル	INT: 変更対象の要素	INT: 補正モード	BOOL: アラームのマスク (オプション)			旋回工具ホルダのオフセットベクトルまたは方向ベクトルを、機械の計測に従って変更します。

5.7 HMI で現在設定されている言語

下の表に、操作画面で使用できるすべての言語を記載します。

現在設定されている言語を、以下のシステム変数を使用してパートプログラムとシンクロナイズドアクションで問い合わせることができます。

`$AN_LANGUAGE_ON_HMI = <value>`

<value>	言語	言語コード
1	ドイツ語(ドイツ)	GER
2	フランス語	FRA
3	英語(英国)	ENG
4	スペイン語	ESP
6	イタリア語	ITA
7	オランダ語	NLD
8	中国語(簡体)	CHS
9	スウェーデン語	SVE
18	ハンガリー語	HUN
19	フィンランド語	FIN

5.7 HMI で現在設定されている言語

<value>	言語	言語コード
28	チェコ語	CSY
50	ポルトガル語(ブラジル)	PTB
53	ポーランド語	PLK
55	デンマーク語	DAN
57	ロシア語	RUS
68	スロバキア語	SKY
72	ルーマニア語	ROM
80	中国語(繁体)	CHT
85	韓国語	KOR
87	日本語	JPN
89	トルコ語	TRK

注記

\$AN_LANGUAGE_ON_HMI は以下の場合に更新されます。

- システムの起動後
- NC および/または PLC のリセット後
- M2N の範囲内で別の NC への切り替え後
- HMI での言語の切り替え後

5.7 HMI で現在設定されている言語

付録

A.1 略語の一覧

A	
A	出力(Output)
AFIS	Automatic Filter Switch: 自動フィルタ切り替え
ASCII	American Standard Code for Information Interchange: 情報交換のための米国標準コード
ASIC	Application Specific Integrated Circuit: ユーザー専用の特定の用途のための集積回路
ASUP	非同期サブプログラム(Asynchronous SUBprogram)
AUTO	運転モード「自動」
AUXFU	Auxiliary Function: 補助機能(Auxiliary Function)
AWL	命令の一覧(Statement list)

B	
BAG	モードグループ
BCD	Binary Coded Decimals: バイナリコードで表現された 10 進数
BICO	Binector Connector
BIN	Binary Files: (BINary files)
BKS	基本座標系
BM	操作アラーム
BO	Binector Output
BTSS	操作パネルインタフェース

C	
CLC	クリアランス制御
CNC	Computerized Numerical Control: Computerunterstützte numerische Steuerung

A.1 略語の一覧

C	
COM	Communication
CP	Communication Processor
CPU	Central Processing Unit:(Central Processing Unit)
CST	Configured Stop:設定された停止

D	
DB	データブロック (PLC)
DBB	データブロックバイト (PLC)
DBD	データブロックダブルワード (PLC)
DBW	データブロックワード (PLC)
DBX	データブロックビット (Data block bit) (PLC)
DDS	Drive Data Set:(Drive Data Set)
DIR	Directory:(DIRectory)
DO	Drive Object
DRF	Differential Resolver Function:差動レソルバ機能(手動パルス発生器)
DRY	Dry Run: ドライラン送り速度
DW	データワード (Data word)
DWORD	ダブルワード (Double word) (現在 32 ビット)

E	
E	入力 (Input)
EES	Execution from External Storage
E/A	入力/出力 (Input/Output)
ESR	停止延長と退避
ETC	ETC キー ">"; 同じメニューのソフトキーバーの拡張

F	
FB	ファンクションコール (PLC)
FC	Function Call:ファンクションコール (PLC)
FDD	Feed Disable:送り速度無効

F	
FdStop	Feed Stop:送り停止
FIFO	First In First Out:アドレス指定なしで動作し、データが格納された順序で読み取られるメモリ
FM	故障メッセージ
FUP	コントロールシステムフローチャート(Control system flowchart) (PLC プログラミング方式)
FW	Firmware

G	
GEO	ジオメトリ(GEOmetry)、例: ジオメトリ軸
GP	基本プログラム(Basic Program)(PLC)
GUD	Global User Data:(Global User Data)

H	
HEX	16 進数を示す略語
HiFu	補助機能(Auxiliary Function)
HMI	Human Machine Interface:SINUMERIK 操作画面
HSA	メイン主軸ドライブ
HT	Handheld Terminal
HW	ハードウェア(HardWare)

I	
IBN	セットアップ
INC	Increment:(INCrement)
INI	Initializing Data:(INItializing Data)
IPO	Interpolator

J	
JOG	ジョグ(JOGing):セットアップモード

A.1 略語の一覧

K	
KOP	ラダー図(PLC プログラミング方式)

L	
LED	Light Emitting Diode:(Light Emitting Diode)
LMS	位置検出器
LR	位置コントローラ

M	
Main	Main program:メインプログラム(OB1、PLC)
MCP	Machine Control Panel:機械操作パネル(Machine Control Panel)
MD	マシンデータ (Machine Data)
MDA	Manual Data Automatic:(Manual Data Input)
MDS	Motor Data Set:(Motor Data Set)
MELDW	メッセージワード
MKS	機械座標系
MM	Motor Module
MPF	Main Program File:メインプログラム(NC)
MPI	Multi Point Interface
MSTT	機械操作パネル(Machine Control Panel)

N	
NC	Numerical Control:移動範囲などのブロック解析をおこなう数値制御
NCU	Numerical Control Unit:NC ハードウェアユニット
NCK	Numerical Control Kernel
NCSD	NC Start Disable
NST	インタフェース信号
NV	ゼロオフセット
NX	Numerical Extension:軸の拡張モジュール

O	
OB	PLC のオーガニゼーションブロック (Organization Block in the PLC)
OEM	Original Equipment Manufacturer
OP	Operation Panel:(Operator Panel)

P	
PCU	PC Unit:コンピュータ装置
PG	プログラミング装置(Programming device)
PLC	Programmable Logic Control:コントローラ
PN	PROFINET
PO	Power On
POS	位置/位置決め(Position/positioning)
PPO	パラメータプロセスデータオブジェクト:PROFIBUS DP 伝送用のサイクリックデータメッセージと「可変速ドライブ」プロファイル
PPU	Panel Processing Unit (パネル一体型 CNC のセントラルハードウェア、SINUMERIK 828D など)
PROFIBUS	Process Field Bus:シリアルデータバス
PRT	プログラムテスト
PTP	Point to Point:ポイントツーポイント(Point-to-point)
PZD	プロセスデータ (Process data):PPO のプロセスデータ部分

R	
REF	レファレンス点復帰機能(REFerence point approach function)
REPOS	再位置決め機能(REPOSition function)
RESU	Retrace support
RID	Read In Disable
RP	R 変数(R Parameter)、算術変数、予約ユーザー変数

S	
SA	シンクロナイズドアクション
SBL	Single Block:(Single BLock)

A.1 略語の一覧

S	
SBT	Safe Brake Test
SCC	Safety Control Channel
SCL	Structured Control Language
SD	Settingdatum またはセッティングデータ
SDI	Safe Direction
SERUPRO	Search-Run by Program Test:プログラムテストによるブロック検索
SIC	Safety Info Channel
SKP	Skip:パートプログラムブロックをスキップするための機能
SLP	Safe Limited Position
SLS	Safely Limited Speed
SMI	Sensor Module Integrated
SOS	Safe Operating Stop
SPF	Sub Program File:サブプログラム(NC)
SS1	Safe Stop 1
SS2	Safe Stop 2
STO	Safe Torque Off
STW	コントロールワード
SUG	といし周速制御(Grinding Wheel Peripheral Speed)
SW	ソフトウェア(SoftWare)

T	
TCU	Thin Client Unit
TM	Terminal Module (SINAMICS)
TO	Tool Offset:(Tool Offset)
TOA	Tool Offset Active:工具オフセットの識別子(ファイルタイプ)
TOFF	オンライン工具長補正
TRANSMIT	Transform Milling Into Turning:旋盤でのフライス削り運転用の座標変換

U	
UP	サブプログラム
USB	Universal Serial Bus

V	
VDI	NC と PLC の間の内部通信インタフェース

W	
WKS	ワーク座標系
WPD	Work Piece Directory:(WorkPiece Directory)
WZ	工具(Tool)
WZV	工具管理機能

Z	
ZSW	ステータスワード(ドライブの)

索引

-

- 試し切り追加終了 - GROUP_ADDEND
外部プログラミング, 1367

\$

\$A_PROBE, 704, 711
\$A_PROBE_LIMITED, 712
\$AA_ACC, 145
\$AA_ATOL, 982
\$AA_COUP_ACT
軸間連動機能, 1067
連結移動時, 1042
\$AA_ESR_ENABLE, 1136
\$AA_FGREF, 127
\$AA_FGROUP, 127
\$AA_GOMODE, 204
\$AA_LEAD_SP, 1067
\$AA_LEAD_SV, 1067
\$AA_MM, 704
\$AA_MM1...4, 713
\$AA_MW, 704
\$AA_MW1...4, 713
\$AC_ACT_PROG_NET_TIME, 1146
\$AC_ACTUAL_PARTS, 1150
\$AC_AXCTSWA, 1027
\$AC_AXCTSWE, 1027
\$AC_CTOL, 982
\$AC_CTOL_GO_ABS, 208
\$AC_CUT_INV, 928
\$AC_CUTMOD, 927
\$AC_CUTMOD_ANG, 927
\$AC_CUTMODK, 927
\$AC_CUTTING_TIME, 1145
\$AC_CYCLE_TIME, 1145
\$AC_DELAYFST, 993
\$AC_ESR_TRIGGER, 1136
\$AC_F_TYPE, 162
\$AC_FGROUP_MASK, 127
\$AC_FZ, 162
\$AC_MEA, 704, 712
\$AC_OLD_PROG_NET_TIME, 1146
\$AC_OLD_PROG_NET_TIME_COUNT, 1146
\$AC_OPERATING_TIME, 1145
\$AC_OTOL, 982
\$AC_OTOL_GO_ABS, 208

\$AC_PROG_NET_TIME_TRIGGER, 1147
\$AC_REPOS_PATH_MODE, 1002
\$AC_REQUIRED_PARTS, 1150
\$AC_S_TYPE, 111
\$AC_SMAXVELO, 1005
\$AC_SMAXVELO_INFO, 1005
\$AC_SPECIAL_PARTS, 1150
\$AC_STOLF, 207
\$AC_SVC, 111
\$AC_TOFF, 100
\$AC_TOFFCR, 100
\$AC_TOFFL, 100
\$AC_TOFFR, 100
\$AC_TOTAL_PARTS, 1150
\$AC_TRAFO_CORR_ELEM_P, 845
\$AC_TRAFO_CORR_ELEM_T, 845
\$AC_TRAFO_ORIAX_LOC, 845
\$AN_AXCTAS, 1027
\$AN_AXCTSWA, 1027
\$AN_ESR_TRIGGER, 1136
\$AN_LANGUAGE_ON_HMI, 1582
\$AN_POWERON_TIME, 1145
\$AN_SETUP_TIME, 1145
\$NT_CLOSE_CHAIN_T, 845
\$NT_CNTRL, 845
\$NT_CORR_ELEM_P, 844
\$NT_CORR_ELEM_T, 844
\$NT_NAME, 836
\$NT_ROT_AX_NAME, 923
\$NT_TRAFO_INDEX, 836
\$P_ACTBFRAME, 746
\$P_AD, 927
\$P_AEP, 314
\$P_APDV, 314
\$P_APR, 314
\$P_BFRAME, 746
\$P_CHBFRAME, 746
\$P_CHBFMASK, 747
\$P_CTOL, 982
\$P_CTOL_GO_ABS, 208
\$P_CUT_INV, 928
\$P_CUTMOD, 927
\$P_CUTMOD_ANG, 927
\$P_CUTMOD_ERR, 929
\$P_CUTMODK, 927
\$P_DELAYFST, 993
\$P_F_TYPE, 162
\$P_FGROUP_MASK, 128
\$P_FZ, 162

\$P_GWPS, 118
\$P_IFRAME, 748
\$P_IS_EES_PATH, 640
\$P_NCBFRAME, 746
\$P_NCBFRMASK, 747
\$P_ORI_DIFF, 917
\$P_ORI_POS, 916
\$P_ORI_SOL, 918
\$P_ORI_STAT, 921
\$P_OTOL, 982
\$P_OTOL_GO_ABS, 208
\$P_PATH, 639
\$P_PFRAME, 748
\$P_PROG, 639
\$P_PROGPATH, 639
\$P_S_TYPE, 111
\$P_SIM, 716, 1038, 1151
\$P_STACK, 639
\$P_STOLF, 208
\$P_SUBPAR, 563
\$P_SVC, 111
\$P_TOFF, 99
\$P_TOFFCR, 100
\$P_TOFFL, 99
\$P_TOFFR, 100
\$P_TOOLENV, 939
\$P_TOOLENVN, 939
\$P_WORKAREA_CS_COORD_SYSTEM, 395
\$P_WORKAREA_CS_LIMIT_MINUS, 396
\$P_WORKAREA_CS_LIMIT_PLUS, 396
\$P_WORKAREA_CS_MINUS_ENABLE, 396
\$P_WORKAREA_CS_PLUS_ENABLE, 395
\$PA_ATOL, 982
\$PA_FGREF, 127
\$PA_FGROUP, 128
\$SA_LEAD_TYPE, 1067
\$SN_PA_ACTIV_IMMED, 663
\$TC_CARR_CORR_ELEM, 906
\$TC_CARR1...14, 891
\$TC_CARR13, 899
\$TC_CARR14, 899
\$TC_CARR18...23, 891
\$TC_CARR18[m], 896
\$TC_DP1 ... 25, 846
\$TC_TP_MAX_VELO, 107

*

* (算術機能), 496

/

/ (算術機能), 496

[

「Siemens Industry Online Support」アプリ, 28

+

+ (算術機能), 496

<

< (比較演算子), 498

<< (結合演算子), 506

<= (関係演算子), 498

<> (比較演算子), 498

=

== (比較演算子), 498

>

> (比較演算子), 498

>= (関係演算子), 498

0

0 文字, 504

1

10 進定数, 427

16 進数定数, 428

2

2 進数定数, 429

A

ABS, 496

AC, 169

ACC, 144

ACCLIMA, 974
ACN, 176
ACOS, 496
ACP, 176
ACTBLOCNO, 575
ACTFRAME, 722
ADIS, 330
ADISPOS, 330
ADISPOSA, 717
AFISOF, 985
AFISON, 985
ALF
 ねじ切り時の高速リトラクト, 259
 輪郭からの高速リトラクト, 624
AMIRROR, 366
ANG, 237
ANG1, 237
ANG2, 237
AP, 196
APR, 462
APRB, 462
APRP, 462
APW, 462
APWB, 462
APWP, 462
AR
 円弧軌跡のプログラミング, 218
AROT, 352
AROTS, 358
AS, 553
ASCALE, 363
ASIN, 496
ASPLINE, 679
ATAN2, 496
ATOL, 978
ATRANS, 345
AV, 1076
AX, 1018
AXCTSWE, 1025
AXCTSWEC, 1025
AXCTSWED, 1025
AXIS, 443
AXNAME, 505
AXSTRING, 1018
AXTOCHAN, 1016
AXTOSPI, 1018
A スプライン, 686

B

B_AND, 498
B_NOT, 498

B_OR, 498
B_XOR, 498
BAUTO, 679
BCS, 46
BFRAME, 722
BLOCK, 605
BLSYNC, 617
BNAT, 679
BOOL, 443
BOUND, 475
BP, 490
BRISK, 971
BRISKA, 971
BSPLINE, 679
BTAN, 679
BZS, 47
B スプライン, 687

C

CAC, 677
CACN, 677
CACP, 677
CADAPTOF, 1030
CADAPTON, 1030
CALL, 604
CALLPATH, 608
CASE, 529
CDC, 677
CDOF, 318
CDOF2, 318
CDON, 318
CFC, 150
CFIN, 150
CFINE, 732
CFTCP, 150
CHAN, 443
CHANDATA, 641
CHAR, 443
CHF, 274
CHKDNO, 887
CHR, 274
CIC, 677
CIP, 222
CLEARM, 546
CLRINT, 620
COARSE, 1076
COARSEA, 717
COMPCAD, 691
COMPCURV, 691
COMPLETE, 641
COMPOF, 691

COMPON, 691
COMPSURF, 691
CONTDCON, 1185
CONTPRON, 1178
CORROF, 375
CORRTC, 902
CORRTRAFO, 837
COS, 496
COUPDEF, 1076
COUPDEL, 1076
COUPOF, 1076
COUPOFS, 1076
COUPON, 1076
COUPONC, 1076
COUPRES, 1076
CP, 811
CPBC, 1089
CPDEF, 1088
CPDEL, 1088
CPFMOF, 1092
CPFMON, 1091
CPFMSON, 1090
CPFPOS + CPOF, 1092
CPFPOS + CPON, 1089
CPFRS, 1089
CPLA, 1088
CPLCTID, 1088
CPLDEF, 1088
CPLDEL, 1088
CPLDEN, 1088
CPLINSC, 1095
CPLINTR, 1095
CPLNUM, 1088
CPLOF, 1088
CPLON, 1088
CPLOUTSC, 1096
CPLOUTTR, 1095
CPLPOS, 1089
CPLSETVAL, 1088
CPMALARM, 1096
CPMBRAKE, 1096
CPMPRT, 1095
CPMRESET, 1093
CPMSTART, 1095
CPMVDI, 1096
CPOF, 1088
CPON, 1088
CPRECOF, 983
CPRECON, 983
CPROT, 660
CPROTDEF, 656
CPSETTYPE, 1096
CPSYNCOF, 1096
CPSYNCOF2, 1096
CPSYNCOV, 1096
CPSYNFIP, 1096
CPSYNFIP2, 1096
CPSYNFIV, 1096
CR, 216
CROTS, 358
CSPLINE, 679
CT, 225
CTAB, 1055
CTABDEF, 1043
CTABDEL, 1050
CTABEND, 1043
CTABEXISTS, 1049
CTABFNO, 1060
CTABFPOL, 1060
CTABFSEG, 1060
CTABID, 1053
CTABINV, 1055
CTABISLOCK, 1053
CTABLOCK, 1051
CTABMEMTYP, 1053
CTABMPOL, 1060
CTABMSEG, 1060
CTABNO, 1060
CTABNOMEM, 1060
CTABPERIOD, 1053
CTABPOL, 1060
CTABPOLID, 1060
CTABSEG, 1060
CTABSEGID, 1060
CTABSEV, 1055
CTABSSV, 1055
CTABTEP, 1055
CTABTEV, 1055
CTABTMAX, 1055
CTABTMIN, 1055
CTABTSP, 1055
CTABTSV, 1055
CTABUNLOCK, 1051
CTOL, 978
CTOLG0, 204
CTRANS, 732
CUT2D, 320
CUT2DD, 320
CUT2DF, 320
CUT2DFD, 320
CUT3DC, 865
CUT3DCC, 876
CUT3DCCD, 876
CUT3DCD, 865

- CUT3DF, 869
CUT3DFD, 869
CUT3DFF, 869
CUT3DFS, 869
CUTCONOF, 323
CUTCONON, 323
CUTMOD, 923
CUTMODK, 923
CYCLE4071
 外部プログラミング, 1342
CYCLE4072
 外部プログラミング, 1344
CYCLE4073
 外部プログラミング, 1348
CYCLE4074
 外部プログラミング, 1350
CYCLE4075
 外部プログラミング, 1354
CYCLE4077
 外部プログラミング, 1357
CYCLE4078
 外部プログラミング, 1361
CYCLE4079
 外部プログラミング, 1363
CYCLE435 - ドレッサー座標系の設定
 外部プログラミング, 1286
CYCLE495 - 成形研削
 外部プログラミング, 1287
CYCLE60 - 彫刻
 外部プログラミング, 1220
CYCLE61 - 正面削り
 外部プログラミング, 1223
CYCLE62- 輪郭の呼び出し
 外部プログラミング, 1226
CYCLE63 - 輪郭ポケットのフライス削り/輪郭ポケット
の削り残し/輪郭スピゴットのフライス削り/輪郭スピ
ゴットの削り残し
 外部プログラミング, 1227
CYCLE64 - 輪郭ポケットの前加工
 外部プログラミング, 1231
CYCLE70 - ねじフライス削り
 外部プログラミング, 1234
CYCLE72 - 輪郭フライス削り
 外部プログラミング, 1236
CYCLE76 - 長方形スピゴット
 外部プログラミング, 1241
CYCLE77 - 円形スピゴット
 外部プログラミング, 1244
CYCLE78 - ドリルねじフライス削り
 外部プログラミング, 1247
CYCLE782 - 負荷に従った位置合わせ
 外部プログラミング, 1289
CYCLE79 - 多角形
 外部プログラミング, 1251
CYCLE800 - 旋回平面/旋回工具/工具の位置合わせ
 外部プログラミング, 1292
CYCLE801 - 格子またはフレーム位置決めパターン
 外部プログラミング, 1298
CYCLE802 - 任意の位置
 外部プログラミング, 1300
CYCLE806 - 補間旋削
 外部プログラミング, 1304
CYCLE81 - センタリング
 外部プログラミング, 1254
CYCLE82 - 穴あけ
 外部プログラミング, 1255
CYCLE83 - 深穴ドリル 1
 外部プログラミング, 1258
CYCLE830 - 深穴あけ 2
 外部プログラミング, 1305
CYCLE832 - 高速設定
 外部プログラミング, 1313
CYCLE84 - フローティングチャックなしのタッピング
 外部プログラミング, 1262
CYCLE840 - フローティングチャックありのタッピング
 外部プログラミング, 1317
CYCLE85 - リーマ加工
 外部プログラミング, 1267
CYCLE86 - ボーリング
 外部プログラミング, 1268
CYCLE899 - オープン溝
 外部プログラミング, 1321
CYCLE92 - 突切り
 外部プログラミング, 1270
CYCLE930 - 溝削り
 外部プログラミング, 1324
CYCLE940 - アンダーカット形状 E と形状 F/ねじのア
ンダーカット
 外部プログラミング, 1328
CYCLE95 - 輪郭切削
 外部プログラミング, 1272
CYCLE951 - 旋削
 外部プログラミング, 1331
CYCLE952 - 旋削/旋削削り残し/プランジ切削/プランジ
切削削り残し/プランジ旋削/プランジ旋削削り残し
 外部プログラミング, 1335
CYCLE98 - 連続ねじ
 外部プログラミング, 1275
CYCLE99 - ねじの旋削
 外部プログラミング, 1281
C スプライン, 688

D

D..., 90
D0, 90
DAC, 184
DC, 176
DEF, 443
DEFAULT, 529
DEFINE ... AS, 553
DELAYFSTOF, 991
DELAYFSTON, 991
DELDL, 851
DELETE, 648
DELOBJ, 824
DELTOOLENV, 936
DIACYCOFA, 184
DIAM90, 182
DIAM90A, 184
DIAMCHAN, 184
DIAMCHANA, 184
DIAMCYCOF, 182
DIAMOF, 182
DIAMOF A, 184
DIAMON, 182
DIAMONA, 184
DIC, 184
DILF, 259
DIN 66217, 44
DIN サブプログラム名称, 636
DISABLE, 619
DISC, 298
DISCL, 302
DISPLOF, 575
DISPLON, 575
DISPR, 995
DISR, 302
DISRP, 302
DITE, 256
DITS, 256
DIV, 496
DI, 850
DO, 1108
DRFOF, 378
DRIVE, 971
DRIVEA, 971
DRVPRD, 1035
DRV PWR, 1035
DV, 1076
DYNFINISH, 976
DYN NORM, 976
DYNPOS, 976

DYNPREC, 976
DYNROUGH, 976
DYNSEMIFIN, 976

D 番号

確認, 887
任意割り当て, 887
名称変更, 888

E

Easy XML, 1158
EAUTO, 679
EES, 630
EES 表記, 632
EG
電子ギヤ, 1068
EGDEF, 1069
EGDEL, 1075
EGOFC, 1074
EGOFS, 1074
EGON, 1070
EGONSYN, 1070
EGONSYNE, 1070
ELSE, 540, 1108
ENABLE, 619
ENAT, 679
ENDFOR, 543
ENDIF, 540
ENDLABEL の間にあるパートプログラム区間, 531
ENDLOOP, 542
ENDWHILE, 544
ESR, 1135
ESRR, 1143
ESRS, 1142
ETAN, 679
EVERY, 1108
EXECSTRING, 493
EXECTAB, 1191
EXECUTE, 1194
EXP, 496
EXTCALL
SINUMERIK 840D sl 用, 610
EXTCLOSE, 1163
EXTERN, 597
EXTOPEN, 1163

F

F...
ねじ切り G34 G35, 255

送り速度, 119
直線補間, 209
FA, 139
FAD, 302
FALSE, 443
FB, 156
FCTDEF, 860
FCUB, 966
FD, 146
FDA, 146
FENDNORM, 716
FFWOF, 977
FFWON, 977
FGREF, 119
FGROUP, 119
FIFOCTRL, 988
FILEDATE, 653
FILEINFO, 653
FILESIZE, 653
FILESTAT, 653
FILETIME, 653
FINE, 1076
FINEA, 717
FL, 119
FLIM, 1007
FLIN, 966
FMA, 152
FNORM, 966
FOR, 543
FP, 399
FPO, 966
FPR, 139
FPRAOF, 139
FPRAON, 139
FRAME, 443
FRC, 274
FRCM, 274
FROM, 1108
FTOCOF, 864
FTOCON, 864

G

G コード
間接プログラミング, 489
G0 許容範囲, 204
G1, 209
G110, 194
G111, 194
G112, 194
G140, 302
G141, 302

G142, 302
G143, 302
G147, 302
G148, 302
G153
ゼロオフセット, 163
フレームの解除, 374
G17, 166
G18, 166
G19, 166
G2, 212
G247, 302
G248, 302
G25
作業領域リミット, 391
主軸速度の制限, 118
G26
作業領域リミット, 391
主軸速度の制限, 118
G290, 1176
G291, 1176
G3, 212
G33, 248
G335, 263
G336, 263
G34, 255
G340, 302
G341, 302
G347, 302
G348, 302
G35, 255
G4, 408
G40, 280
G41, 280
G42, 280
G450, 298
G451, 298
G460, 314
G461, 314
G462, 314
G5, 805
G500
ゼロオフセット, 163
G505～G599, 163
G53
ゼロオフセット, 163
フレームの解除, 374
G54～G57, 163
G58, 349
G59, 349
G60, 327
G601, 327

G602, 327
G603, 327
G62, 716
G621, 716
G64, 330
G641, 330
G642, 330
G643, 330
G644, 330
G645, 330
G7, 805
G70, 178
G700, 178
G71, 178
G710, 178
G74, 398
G75, 399
G810～G819, 715
G820～G829, 715
G9, 327
G90, 169
G91, 171
G93, 119
G94, 119
G95, 119
G96, 112
G961, 112
G962, 112
G97, 112
G971, 112
G972, 112
G973, 112
GEOAX, 1020
GET, 1011
GETACTTD, 889
GETD, 1011
GETDNO, 888
GETTCOR, 939
GETTENV, 937
GETVARAP, 479
GETVARDFT, 483
GETVARDIM, 482
GETVARLIM, 481
GETVARPHU, 479
GETVARTYP, 484
GFRAME0 ... GFRAME100, 380
GOTO, 525
GOTOB, 525
GOTOC, 525
GOTOF, 525
GOTOS, 523

GROUP_ADDEND - 試し切り追加終了
外部プログラミング, 1367
GROUP_BEGIN - プログラムブロック開始
外部プログラミング, 1366
GROUP_END - プログラムブロック終了
外部プログラミング, 1367
GUD, 444
GWPS, 117
GWPSOF, 117
GWPSON, 117
G グループ
テクノロジー, 976
G 命令, 1500

H

HOLES1 - 行位置決めパターン
外部プログラミング, 1199
HOLES2 - 円または円弧位置決めパターン
外部プログラミング, 1200

I

I...
ねじ切り G33, 248
ねじ切り G34 G35, 255
円弧補間, 212
IC, 171
ID, 1108
IDS, 1108
IF, 540
IFRAME, 722
INDEX, 509
INICF, 443
INIPO, 443
INIRE, 443
INIT, 546
INITIAL, 641
INITIAL_INI, 641
INT, 443
INTEGER 定数, 428
INTERSEC, 1189
INVCCW, 232
INVCW, 232
IPOBRKA, 717
IPOENDA, 717
IPOSTOP, 1076
IPTRLOCK, 993
IPTRUNLOCK, 993
IR, 263
ISAXIS, 1018

ISFILE, 652
ISNUMBER, 505
ISOCALL, 606
ISVAR, 476

J

J...
 ねじ切り G34 G35, 255
 円弧補間, 212
JERKLIM, 1006
JERKLIMA, 974
JR, 263

K

K...
 ねじ切り G33, 248
 ねじ切り G34 G35, 255
 円弧補間, 212
KONT, 291
KONTC, 291
KONTT, 291
KR, 263

L

L..., 595
LEAD, 767
LEADOF, 1062
LEADON, 1062
LENTOAX, 962
LF, 54
LFOF, 259
LFON, 259
LFPOS, 259
LFTXT, 259
LFWP, 259
LIFTFAST, 621
LIMS, 112
LINE FEED, 54
LLI, 458
LN, 496
LONGHOLE - 長穴
 外部プログラミング, 1217
LOOP, 542
LUD, 444

M

M..., 384
M0, 384
M1, 384
M17, 580
M19
 M 機能, 384
 主軸の位置決め, 133
M2, 384
M3, 101
M30, 580
M4, 101
M40, 384
M41, 384
M42, 384
M43, 384
M44, 384
M45, 384
M5, 101
M6, 74
M70, 133
MASLDEF, 1105
MASLDEL, 1105
MASLOF, 1105
MASLOFS, 1105
MASLON, 1105
MATCH, 509
MAXVAL, 475
MCALL, 601
MCS, 43
MD10010, 546
MD10210, 902
MD10240, 180
MD10260, 178
MD10280, 547
MD10651, 264
MD10710, 259
MD15800, 440
MD18104;MD 1 8 1 0 4, 935
MD18116, 936
MD18156, 440
MD20360, 945
MD24558;MD 2 4 5 5 8, 947
MD24658;MD 2 4 6 5 8, 947
MD9440, 92
MEAC, 705
MEAFRAME, 739
MEAS, 702
MEASA, 705
MEAW, 702

MEAWA, 705
MINDEX, 509
MINVAL, 475
MIRROR, 366
MMC, 1158
MOD, 496
MODAXVAL, 1018
MPF, 629
MSG, 388
mySupport ドキュメンテーション, 26
M 機能, 384

N

NAMETOINT, 829
NC, 443
NCK 表記, 632
NC プログラミング
 文字セット, 60
NC プログラム
 作成, 59
NC 高機能言語, 53
NEWCONF, 1152
NOC, 1076
NORM, 291
NOT, 498
NPROT, 660
NPROTDEF, 656
NUMBER, 505
NUT, 779

O

OEMIP01/2, 715
OEM アドレス, 715
OEM 機能, 715
OFFN, 280
OMA1 ... OMA5, 715
OpenSSL, 29
OR, 498
ORIAxes, 776
ORIC, 881
ORICONCCW, 779
ORICONCW, 779
ORICONIO, 779
ORICONTO, 779
ORICURVE, 783
ORID, 881
ORIEULER, 776
ORIMKS, 774
ORIPATH, 791

ORIPATHS, 791
ORIPLANE, 779
ORIRESET(A, B, C), 765
ORIROTA, 786
ORIROTC
 工具オリエンテーションの回転, 786
 工具回転の補間時, 792
ORIROTR, 786
ORIROTT, 786
ORIRPY, 776
ORIRPY2, 776
ORIS, 881
ORISOF, 799
ORISOLH, 909
ORISON, 799
ORIVECT, 776
ORIVIRT1, 776
ORIVIRT2, 776
ORIWKS, 774
OS, 1109
OSB, 1109
OSC, 881
OSCILL, 1115
OSCTRL, 1109
OSD, 881
OSE, 1109
OSNSC, 1109
OSOF, 881
OSP1, 1109
OSP2, 1109
OSS, 881
OSSE, 881
OST, 881
OST1, 1109
OST2, 1109
OTOL, 978
OTOLG0, 204
OVR, 143
OVRA, 143
OVRRAP, 143

P

P..., 600
P_ACTFRAME, 748
PACCLIM, 1009
PAROT, 371
PAROTOF, 371
PCALL, 607
PDELAYOF, 1122
PDELAYON, 1122
PFRAME, 722

PHI
円錐面に沿った向き, 779
旋回多項式, 785
PHU, 459
PL
スプライン補間, 679
多項式補間, 692
PLC
-軸, 417
PM, 302
PO, 692
PO[PHI]
円錐面に沿った向き, 779
工具オリエンテーションの回転, 791
旋回多項式, 785
PO[PSI]
円錐面に沿った向き, 779
工具オリエンテーションの回転, 791
旋回多項式, 785
PO[THT]
工具オリエンテーションの回転, 791
旋回多項式, 785
PO[XH]
2つの接点の向きの指定, 783
旋回多項式, 785
PO[YH]
2つの接点の向きの指定, 783
旋回多項式, 785
PO[ZH]
旋回多項式, 785
PO[ZH]:
2つの接点の向きの指定, 783
POCKET3 - 長方形ポケット
外部プログラミング, 1202
POCKET4 - 円形ポケット
外部プログラミング, 1206
POLF
NC 制御による退避, 1136
ねじ切り時の高速リトラクト, 259
POLFA, 1136
POLFMASK
NC 制御による退避, 1136
ねじ切り時の高速リトラクト, 259
POLFMLIN
NC 制御による退避, 1136
ねじ切り時の高速リトラクト, 259
POLY, 692
POLYPATH, 692
PON, 1132
PONS, 1122
POS, 129
POSA, 129

POSFS, 1076
POSP, 129
POT, 496
PR, 302
PREPRO, 579
PRESETON, 736
PRESETONS, 737
PRIO, 617
PRLOC, 443
Process DataShare, 1163
PROTA, 831
PROTD, 833
PROTS, 832
psi
円錐面に沿った向き, 779
旋回多項式, 785
PTP, 810
PTPG0, 810
PTPWOC, 811
PUD, 444
PUNCHACC, 1122
PUTFTOC, 863
PUTFTOCF, 862
PW, 679

Q
QU, 383

R
RAC, 184
READ, 650
REAL, 443
REAL 定数, 428
REDEF, 450
RELEASE, 1011
REP, 468
REPEAT, 531
REPEATB, 531
REPOSA, 995
REPOSH, 995
REPOSHA, 995
REPOSL, 995
REPOSQ, 995
REPOSQA, 995
RET, 581
RET (パラメータ設定可能), 582
RETB (パラメータ設定可能), 589
RG, 439
RIC, 184

RINDEX, 509
RMBBL, 995
RMEBL, 995
RMIBL, 995
RMNBL, 995
RND, 274
RNDM, 274
ROT, 352
ROTS, 358
ROUND, 496
ROUNDUP, 502
RP, 196
RPL, 352
RPY, 770
Run MyScreens, 1158

S

S, 101
SAR, 302
SAVE, 569
SBLOF, 570
SBLON, 570
SCALE, 363
SCC, 112
SCPARA, 1029
SD, 679
SD41610, 846
SD41611, 846
SD42010, 258
SD42440, 172
SD42442, 172
SD42450\$SC_CONTPREC, 984
SD42451 \$SC_CONTPREC_G00_ABS, 984
SD42460 \$SC_MINFEED (CPRECON による最低軌跡送り速度), 985
SD42465, 337
SD42466, 337
SD42475, 797
SD42476, 797
SD42477, 797
SD42900, 854
SD42910, 854
SD42920, 854
SD42930, 855
SD42935, 857
SD42940, 858
SD42984, 927
SD43240, 135
SD43250, 135
SET, 468
SETAL, 1168

SETDNO, 888
SETINT, 617
SETM, 546
SETMS, 101
SETTCOR, 947
SF, 248
Siemens Industry Online Support
 アプリ, 28
SIN, 496
SINUMERIK, 21
SLOT1- 直線溝
 外部プログラミング, 1210
SLOT2 - 円周溝
 外部プログラミング, 1214
SOFT, 971
SOFTA, 971
SON, 1122
SONS, 1122
SPCOF, 132
SPCON, 132
SPF, 629
SPI, 1018
SPIF1, 1122
SPIF2, 1122
SPLINEPATH, 690
SPN; S P N, 1128
SPOF, 1122
SPOS, 133
SPOSA, 133
SPP, 1128
SPRINT, 513
SQRT, 496
SR, 152
SRA, 152
ST, 152
STA, 152
START, 546
STARTFIFO, 988
STAT, 811
STOLF, 204
STOPFIFO, 988
Stopre, 988
STRING, 443
 結合, 506
STRINGIS, 1153
STRLEN, 508
SUBSTR, 510
SUPA
 ゼロオフセット, 163
 フレームの解除, 374
SVC, 105
SYNR, 443

SYNRW, 443
SYNW, 443
SZS, 48
S 値の
 解釈, 103

T

T0, 73
TAN, 496
TANG, 1097
TANGDEL, 1103
TANGOF, 1103
TANGON, 1101
TCARR, 897
TCOABS, 897
TCOFR, 897
TCOFRX, 897
TCOFRY, 897
TCOFRZ, 897
THETA
 工具オリエンテーションの回転, 786
 工具回転の補間時, 792
TILT, 767
TLIFT, 1100
TMOF, 1133
TMON, 1133
TOFF, 94
TOFFL, 94
TOFFLR, 94
TOFFOF, 906
TOFFON, 906
TOFFR, 94
TOFRAME, 371
TOFRAMEX, 371
TOFRAMEY, 371
TOFRAMEZ, 371
TOLOWER, 508
TOOLENV, 933
TOROT, 371
TOROTOF, 371
TOROTX, 371
TOROTY, 371
TOROTZ, 371
TOUPPER, 508
TOWBCS, 856
TOWKCS, 856
TOWMCS, 856
TOWSTD, 856
TOWTCS, 856
TOWWCS, 856

TRAANG
 角度がプログラム指令可能な, 804
TRACON, 808
TRACYL, 801
TRAFOOF, 824
TRAFOON, 836
TRAILOF, 1039
TRAILON, 1039
TRANS, 345
TRANSMIT, 800
TRAORI, 764
TRUE, 443
TRUNC, 496
TU, 816
TURN, 229

U

ULI, 458
UNTIL, 545
User XML, 1158

V

VELOLIM, 1004
VELOLIMA, 974

W

WAITC, 1076
WAITE, 546
WAITENC, 1027
WAITM, 546
WAITMC, 546
WAITP, 129
WAITS, 133
WALCS<n>, 394
WALCS0, 394
WALIMOF, 391
WALIMON, 391
WCS, 48
 ワーク上に配置, 371
WHEN, 1108
WHEN-DO, 1119
WHENEVER, 1108
WHENEVER-DO, 1119
WHILE, 544
WORKPIECE, 1170
WRITE, 644
WRTPR, 389

X

X..., 192
XOR, 498

Y

Y..., 192

Z

Z..., 192

ア

アドレス, 423
 値の割り当て, 55
アドレス指定, 632
アドレス文字, 1485
アブソリュート指令, 37
アプローチ点/角度, 294
アラーム
 NC プログラムで設定, 1168
アンダーカット形状 E と形状 F/ねじのアンダーカット
- CYCLE940
 外部プログラミング, 1328

イ

イグザクトストップ, 327
インクレメンタル指令, 39, 171
インボリュート, 232

オ

オイラー角, 770
オープン溝 - CYCLE899
 外部プログラミング, 1321
オプションナルストップ, 387
オフセット
 工具長, 94, 95
 工具半径, 94
オフセットメモリ, 846
オンライン工具長補正, 906

お

および, 498

カ

カウントループ, 543

キ

キネマティックス
 分解, 896
キネマティックスタイプ, 896

ク

クランプトルク
 -突き当て点, 406
グローバルパートプログラムメモリ (GDIR), 631

コ

コメント, 56

サ

サブプログラム
 パラメータ転送を含まない呼び出し, 595
 パラメータ転送を含む呼び出し, 597
 プログラム指令可能な検索パス, 608
 リターン、パラメータ設定可能 (RET ...), 582
 リターン、パラメータ設定可能 (RETB...), 589
 繰り返し, 600
 呼び出し、モーダル, 601
 呼び出し、間接, 604
 名称, 558
 用途, 557

シ

ジオメトリ軸
 切り替え, 1020
システムフレーム, 745
シミュレーション, 1151
ジャンプ
 ジャンプラベルへ, 525
 プログラムの先頭へ, 523
ジャンプマーク
 プログラム区間の繰り返し, 531
ジャンプラベル
 プログラムのジャンプ, 526

シングルブロック
マスク, 570

ス

スキップレベル, 57
スクロールねじ, 253
スケーリング係数, 363
ストップ
 オプション, 387
 プログラム, 387
ストレートねじ, 253
スプライン
 タイプ, 686
 補間, 679
スプライングループ, 690
スムージング
 旋回処理の, 799
スレーブ軸
 軸間連動機能, 1062

す

すべてのコーナのコーナ減速, 717

セ

ゼロオフセット
 プログラム指令可能, 345
 外部, 734
 設定可能, 163
ゼロオフセットシステム
 ベシック, 47
 設定可能, 48
センタリング - CYCLE81
 外部プログラミング, 1254

チ

チェック
 命令, 538
チャンネル
 軸, 415

テ

ディレクトリパス, 635
データマトリックスコード, 29
テーパねじ, 254
テープ出力フォーマット, 51

テクニカルサポート, 28

ト

ドウェル時間, 408
ドライブパラメータの読み取り/書き込み, 1035
ドライブ名, 634
ドリル, 83
ドリルねじフライス削り - CYCLE78
 外部プログラミング, 1247
トレーニング, 28
ドレッサー座標系の設定 - CYCLE435
 外部プログラミング, 1286

ニ

ニブリング, 1122

ネ

ネストレベル
 チェック命令の, 539

ね

ねじ
 ピッチ, 255
 -回転方向, 249
 -切り G33, 248
 切り G34 G35, 255
 -多条, 248
 連続, 249
ねじの旋削 - CYCLE99
 外部プログラミング, 1281
ねじフライス削り - CYCLE70
 外部プログラミング, 1234

ノ

ノンモーダル, 53

ハ

パス指定, 634
パス指定とパラメータによるサブプログラム, 607
パラメータ
 サブプログラム呼び出しのための転送, 597
 サブプログラム呼び出し時の転送, 562
 マシン, 846

仮, 561
現在値, 561
パンチング, 1122
ハンドル
 オーバライド, 146

フ

ファイル
 -情報, 653
ファイル名, 636
フィードバックを送信, 26
フライス工具, 80
フライス工具の加工点, 868
フライス工具の先端, 868
フライス工具基準点, 868
ブランクの定義, 1170
フレーム, 340
 グローバル, 743
 システム, 745
 -スケーリング, 363
 チャネル別, 744
 フレーム結合, 731
 -ミラーリング, 366
 -結合, 750
 呼び出し, 730
 選択解除, 374
 代入, 731
 -命令, 342
フレーム成分
 FI, 728
 MI, 728
 RT, 728
 SC, 728
 TR, 728
フレーム変数
 座標変換の呼び出し, 720
 値の割り当て, 726
 予約フレーム変数, 722
フローティングチャックなしのタッピング - CYCLE84
 外部プログラミング, 1262
フローティングチャックを使用したタッピング -
CYCLE840
 外部プログラミング, 1317
プログラム
 アドレス指定, 632
 シミュレーション, 1151
 ジャンプ, 525
 ヘッダー, 61
 メモリ, 629
 繰り返し, 600
 実行時間, 1144

終了, 54
初期化, 641
分岐, 529
名称, 50
プログラムストップ, 387
プログラムブロック開始 - GROUP_BEGIN, 1366
プログラムブロック終了 - GROUP_END
 外部プログラミング, 1367
プログラムメモリ
 ファイルタイプ, 629
 標準ディレクトリ, 628
プログラムループ
 IF ループ, 540
 REPEAT ループ, 545
 WHILE ループ, 544
 カウントループ, 543
 ループの終了, 542
プログラム区間
 -繰り返し, 531
プログラム区間の繰り返し
 CALL の間接プログラミングによる, 605
ブロック, 51
 スキップ, 57
 終了, 54
 -終了 LF, 61
 長, 54
 番号, 54
 命令の順序, 55
ブロックスキップ, 57
ブロック表示
 マスクする, 575
プロテクションゾーン, 656

へ

ヘリカル補間, 229

ホ

ポイントツーポイント移動, 810
ボーリング - CYCLE86
 外部プログラミング, 1268

マ

マクロ, 553
マスタ軸
 軸間連動機能, 1062
マスタ値シミュレーション, 1067

メ

メインエントリ, 188
メイン主軸, 414
メタルソー, 89
メッセージ, 388
メモリ
 プログラム, 627
 解析, 988
 作業, 641

モ

モーダル有効, 53

ラ

ラベル, 531

リ

リーマ加工 - CYCLE85
 外部プログラミング, 1267
リンク
 マスタリンク軸, 420
 -軸, 418
 変数, 441

レ

レファレンス点復帰, 398

ワ

ワーク
 カウンタ, 1149
 ディレクトリ, 629
 メインディレクトリ, 629
 原点, 42
 輪郭, 191
ワークチェーン, 843
ワーク座標系, 48

位

位置決め軸, 415
位置指令値フィルタチェーン, 985

位置属性

 間接プログラミング, 490
位置同期制御, 1076

移

移動指令, 191

一

一般データ保護規則 (GDPR), 29

右

右ねじ, 250

円

円または円弧位置決めパターン - HOLES2
 外部プログラミング, 1200
円形スピゴット - CYCLE77
 外部プログラミング, 1244
円形ポケット - POCKET4
 外部プログラミング, 1206
円弧データ
 計算, 1192
円弧軌跡のプログラミング
 開口角度と中心点による, 218
 極座標による, 220
 半径と終点による, 216
 補間タイプ, 212
円弧軌跡のプログラム
 中心点と終点による, 213
円弧補間
 ヘリカル補間, 229
 中間ポイントと終了ポイントによる, 222
円周溝 - SLOT2
 外部プログラミング, 1214
円筒座標, 197
円筒補間, 756

加

加々速度
 オフセット, 1006
 制限, 971
加工平面, 40
加速モード, 971

解

解析

-メモリ, 988

回

回転

配向ベクトルの, 786

回転ベクトルの補間, 787

回転軸

回転角度, 891

距離ベクトル, 891

方向ベクトル, 891

外

外部ゼロオフセット, 734

外部プログラミング, 1366

拡

拡張アドレス表記, 424

格

格子またはフレーム位置決めパターン - CYCLE801

外部プログラミング, 1298

角

角度オフセットによる位置同期制御, 1076

割

割り込みルーチン

プログラム指令可能な移動方向, 624

解除, 620

解除/起動, 619

後退移動, 623

新規割り当て, 619

輪郭からの高速リトラクト, 621

間

間接プログラミング

G コードの, 489

アドレスの, 487

パートプログラム行の, 493

位置属性の, 490

丸

丸み付け, 274

基

基準点, 42

基準半径, 126

基本オフセット, 47

基本ゼロオフセットシステム, 47

基本座標系, 46

機

機械

原点, 42

-軸, 414

機械座標系, 43

機能

予約, 1562

記

記述の範囲, 24

起

起点, 42

起点-目標点, 191

軌

軌跡に対する工具オリエンテーション, 789

軌跡演算, 422

軌跡加速度

最大値, 1009

軌跡軸, 415

軌跡速度

最大値, 1007

極

極, 194

極角度, 36

極座標, 36

極座標変換, 756

極半径, 36

傾

傾斜プランジ研削, 805

傾斜角座標変換(TRAANG)

角度がプログラム指令可能な, 804

形

形状

-軸, 412

径

径方向軸, 190

経

経路

計算, 238

計

計測サイクルパラメータ

CYCLE961, 1402

CYCLE971, 1428

CYCLE973, 1371

CYCLE974, 1374

CYCLE976, 1385

CYCLE977, 1397

CYCLE978, 1388

CYCLE979, 1405

CYCLE982, 1424

CYCLE994, 1379

CYCLE995, 1413

CYCLE996, 1417

CYCLE9960, 1421

CYCLE997, 1410

CYCLE998, 1393

穴

穴あけ - CYCLE82

外部プログラミング, 1255

結

結合

文字列の, 506

検

検索パス

サブプログラム呼び出し, 637

プログラム指令可能な検索パス, 608

検索不可区間の認識と検出, 994

研

研削工具, 84

原

原点

旋削用, 189

原点フレーム, 163

現

現在値連結, 1079

言

言語モード, 1176

固

固定点

アプローチ, 399

後

後退

NC 制御, 1136

ドライブ内蔵, 1143

ねじ切りの方向, 260

向

向きのプログラミング, 776

工

工具

- M6 による交換, 74
- T 番号による交換, 73
- オフセットメモリ, 78
- オリエンテーション, 881
- グループ, 80
- タイプ, 80
- タイプ番号, 80
- パラメータ, 846
- 径補正, 78
- 交換位置, 42
- 刃先, 90
- 先端, 79
- 長さ補正, 77
- 工具オフセット
 - オフセット, 94
 - オフセットメモリ, 846
 - 追加, 850
 - 摩耗値の座標系, 855
- 工具オリエンテーション
 - 軌跡を基準とした, 789
- 工具オリエンテーションの初期設定 ORIRESET, 766
- 工具チェーン, 843
- 工具ホルダ
 - キネマティック, 891
 - 基準点, 42
 - 旋回, 897
- 工具径補正
 - CUT2DF, 323
 - コーナ減速, 716
 - 外側コーナの, 298
- 工具速度
 - 最大値, 107

溝

- 溝削り - CYCLE930
 - 外部プログラミング, 1324

行

- 行位置決めパターン - HOLES1
 - 外部プログラミング, 1199

高

- 高速リトラクト
 - ねじ切り, 259
- 高速設定 - CYCLE832
 - 外部プログラミング, 1313

左

- 左ねじ, 250

座

- 座標
 - 円筒, 197
 - 極座標, 36
 - 直交, 34
- 座標回転
 - プログラマブル, 352
- 座標系
 - ベーシック, 46
 - 一覧, 43
- 座標変換
 - 3 軸、4 軸、5 軸座標変換, 764
 - キネマティックトランスフォーメーション, 752
 - 結合, 808
 - 座標変換重畳, 752
 - 方向座標変換, 751
- 座標変換(フレーム), 48
- 座標変換の制約事項, 823

作

- 作業メモリ, 641
- 作業領域リミット
 - BCS の, 391

三

- 三本指の法則, 44

算

- 算術変数
 - グローバル, 439
 - チャンネル別, 437

残

残り時間
加工の, 1148

始

始点オフセット
ねじ切りの, 248

指

指令値連結, 1079

自

自動軌跡分割, 1127
自動中断ポイント, 995

識

識別子, 50
システム変数の, 60
特別な数値の, 60
文字列の, 60

軸

軸
PLC, 417
コンテナ, 419
-タイプ, 411
チャネル, 415
マスタリンク軸, 420
メイン, 412
リンク, 418
機械, 414
軌跡, 415
形状, 412
交換, 1011
同期, 416
付加, 413
命令, 417
連結移動, 1041
軸(Axes)
位置決め, 415

軸間連動機能, 1062

マスタ軸とスレーブ軸の同期制御, 1065
現在値連結と指令値連結, 1066

実

実行時間
チェック命令の動作, 539

主

主軸
M 機能, 387
メイン, 414
位置決め, 133
回転方向, 101
交換, 1011
速度, 101
速度制限, 118
主軸同期
ペア定義, 1082
連結, 1075

出

出力
外部機器/ファイルへの, 1163

処

処理
予約, 1526
処理時間, 1145

初

初期化
配列の, 468
初期化プログラム, 641

章

章の構成, 22

衝

衝突検出, 318

深

深穴あけ 2 - CYCLE830
外部プログラミング, 1305
深穴ドリル 1 - CYCLE83
外部プログラミング, 1258

刃

刃先
位置, 79
-基準点, 325
中心点, 79
-当該の位置, 325
半径, 79
番号, 91
輪郭工具の数, 322
刃先番号, 887

数

数値拡張子, 424

寸

寸法
オプション, 169
回転軸と主軸の場合, 176
直径の, 182
半径の, 182

成

成形研削 - CYCLE495
外部プログラミング, 1287

正

正面削り, 773
正面削り - CYCLE61
外部プログラミング, 1223

製

製品サポート, 27

切

切り込み深さ, 868
切り上げ, 502
切り替え可能なジオメトリ軸, 1020
切削
サポート機能, 1178
切削 - CYCLE951
外部プログラミング, 1331
切削速度, 105
切削速度(一定), 112

先

先読み, 335
先読み停止
内部, 411

旋

旋回工具ホルダ, 891
旋回軸, 776
旋回平面/旋回工具/工具の位置合わせ - CYCLE800
外部プログラミング, 1292
旋削/旋削削り残し/プランジ切削/プランジ切削削り残し/プランジ旋削/プランジ旋削削り残し - CYCLE952
外部プログラミング, 1335
旋削工具, 86

送

送り速度
インバースタイム, 123
オーバーライド, 148
-オーバーライド, 143
ハンドルオーバーライド, 146
位置決め軸の, 139
規則, 119
軌跡軸の, 122
-速度, 209
単位, 125
同期軸の, 124

速

速度同期制御, 1076
速度連結, 1079

他

他社の Web サイト, 24

多

多角形 - CYCLE79

外部プログラミング, 1251

多項式係数, 694

多項式補間, 692

大

大文字小文字を区別しない, 632

単

単位系, 178

値

値の割り当て, 55

彫

彫刻 - CYCLE60

外部プログラミング, 1220

長

長穴 - LONGHOLE

外部プログラミング, 1217

長方形のスπιゴット - CYCLE76

外部プログラミング, 1241

長方形ポケット - POCKET3

外部プログラミング, 1202

直

直径指定, 182

直交 PTP 移動, 810

直交座標系, 34

直線

-補間, 209

直線溝 - SLOT1

外部プログラミング, 1210

直線軸旋回機構による座標変換, 763

停

停止

NC 制御, 1141

サイクルの終了時, 387

ドライブ内蔵, 1142

停止点, 42

定

定数, 427

適

適用, 22

電

電子ギヤ, 1068

動

動作終了条件

プログラム指令可能な, 717

同

同期

軸, 416

精密, 1079

汎用, 1079

同期揺動

シンクロナイズドアクション, 1119

次の区間切り込み, 1121

切り込みの定義, 1118

切り込み移動, 1119

反転点範囲内の切り込み, 1119

評価、IPO サイクル, 1121

揺動軸と切り込み軸の割り当て, 1118

特

特異点, 776

特殊工具, 88

特殊文字, 60, 61

凸

凸型ねじ, 263

突

突き当て点, 404

突切り - CYCLE92

外部プログラミング, 1270

内

内側コーナのコーナ減速, 717

内部先読み停止, 411

任

任意の位置 - CYCLE802

外部プログラミング, 1300

配

配向ベクトル THETA, 786

配列, 468

定義, 468

要素, 468

配列インデックス, 471

半

半径

有効, 126

半径指定, 182

比

比較演算子, 498

非

非同期揺動, 1109

付

付加軸, 413

負

負荷に従った位置合わせ - (CYCLE782)

外部プログラミング, 1289

分

分母多項式, 697

文

文字セット, 60

文字列

フォーマット, 513

-演算子, 503

文字列の

長さ, 508

変

変数

タイプ変換, 485, 504

ユーザー定義, 443

保

保持ブロック, 994

補

補間旋削 - CYCLE806

外部プログラミング, 1304

補助機能出力

高速, 383

補助機能の特性, 381

連続軌跡モード, 384

補正

工具長, 77

工具半径, 78

平面, 323

方

方向ベクトル, 771

方向座標変換 TRAORI

機械のキネマティックス, 754

向きのプログラミング, 765

向きのプログラミングのタイプ, 766

旋回運動, 753
汎用 5/6 軸座標変換, 755

無

無限ループ, 542

命

命令, 52
軸, 417

面

面取り, 274

目

目標点, 191

揺

揺動
シンクロナイズドアクションによる制御,
区間切り込み,
同期揺動,
非同期,
非同期揺動,
揺動動作
反転点, 1118
反転点での切り込み, 1120
反転範囲, 1118

輪

輪郭
アプローチ/後退, 291
-コード, 1185
テーブル, 1178
-解析, 1178
再位置決め, 995
-要素, 190
輪郭からの高速リトラクト, 621
輪郭のコーナ
丸み付け, 274
面取り, 274
輪郭の呼び出し - CYCLE62
外部プログラミング, 1226

輪郭フライス削り - CYCLE72
外部プログラミング, 1236
輪郭ポケットのフライス削り/輪郭ポケットの削り残し/
輪郭スピゴットのフライス削り/輪郭スピゴットの削り
残し - CYCLE63
外部プログラミング, 1227
輪郭ポケットの予備穴あけ - CYCLE64
外部プログラミング, 1231
輪郭解析
エラーフィードバック信号, 1194
輪郭精度
プログラム指令可能, 983
輪郭切削 - CYCLE95
外部プログラミング, 1272
輪郭定義のプログラミング, 237
輪郭要素
移動, 1191

連

連結;
汎用, 1087
連結のタイプ, 1079
連結移動, 1039
連結移動軸, 1041
連結係数, 1039
連結軸の組み合わせ, 1039
連結状態
軸間連動機能, 1067
連結移動時, 1042
連続ねじ - CYCLE98
外部プログラミング, 1275
連続軌跡モード, 330

論

論理演算子, 498

