SIEMENS

SINUMERIK

SINUMERIK 840D sl / 828D Fundamentals

Programming Manual

Valid for

Control
SINUMERIK 840D sl / 840DE sl / 828D

CNC software version 4.8 SP3

08/2018
6FC5398-1BP40-6BA2

Legal information

Warning notice system

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring only to property damage have no safety alert symbol. These notices shown below are graded according to the degree of danger.

⚠ DANGER
indicates that death or severe personal injury **will** result if proper precautions are not taken.

⚠ WARNING
indicates that death or severe personal injury **may** result if proper precautions are not taken.

⚠ CAUTION
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠ WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
GERMANY

Document order number: 6FC5398-1BP40-6BA2
℗ 08/2018 Subject to change

Preface

SINUMERIK documentation

The SINUMERIK documentation is organized into the following categories:

- General documentation/catalogs
- User documentation
- Manufacturer/service documentation

Additional information

You can find information on the following topics at the following address (https://support.industry.siemens.com/cs/de/en/view/108464614):

- Ordering documentation/overview of documentation
- Additional links to download documents
- Using documentation online (find and search in manuals/information)

If you have any questions regarding the technical documentation (e.g. suggestions, corrections), please send an e-mail to the following address (mailto:docu.motioncontrol@siemens.com).

mySupport/Documentation

At the following address (https://support.industry.siemens.com/My/ww/en/documentation), you can find information on how to create your own individual documentation based on Siemens' content, and adapt it for your own machine documentation.

Training

At the following address (http://www.siemens.com/sitrain), you can find information about SITRAIN (Siemens training on products, systems and solutions for automation and drives).

FAQs

You can find Frequently Asked Questions in the Service&Support pages under Product Support (https://support.industry.siemens.com/cs/de/en/ps/faq).

SINUMERIK

You can find information about SINUMERIK at the following address (http://www.siemens.com/sinumerik).

Target group

This publication is intended for:

- Programmers
- Project engineers

Benefits

With the programming manual, the target group can develop, write, test, and debug programs and software user interfaces.

Standard scope

This Programming Manual describes the functionality of the standard scope. Extensions or changes made by the machine tool manufacturer are documented by the machine tool manufacturer.

Other functions not described in this documentation might be executable in the control. This does not, however, represent an obligation to supply such functions with a new control or when servicing.

Furthermore, for the sake of clarity, this documentation does not contain all detailed information about all product types and cannot cover every conceivable case of installation, operation or maintenance.

Technical Support

Country-specific telephone numbers for technical support are provided in the Internet at the following address (https://support.industry.siemens.com/sc/ww/en/sc/2090) in the "Contact" area.

Information on structure and contents

Programming Manual, Fundamentals/Job Planning

The description of the NC programming is divided into two manuals:

1. **Fundamentals**
   This "Fundamentals" Programming Manual is intended for use by skilled machine operators with the appropriate expertise in drilling, milling and turning operations. Simple programming examples are used to explain the commands and statements which are also defined according to DIN 66025.
2. **Job planning**
   The Programming Manual "Advanced" is intended for use by technicians with in-depth, comprehensive programming knowledge. By virtue of a special programming language, the SINUMERIK control enables the user to program complex workpiece programs (e.g. for free-form surfaces, channel coordination, ...) and makes programming of complicated operations easy for technologists.

Availability of the described NC language elements

All NC language elements described in the manual are available for the SINUMERIK 840D sl. The availability regarding SINUMERIK 828D should be taken from Table "Operations: Availability for SINUMERIK 828D (Page 452)".

Table of contents

Fundamental safety instructions 1

1.1 General safety instructions

⚠ WARNING
Danger to life if the safety instructions and residual risks are not observed
If the safety instructions and residual risks in the associated hardware documentation are not observed, accidents involving severe injuries or death can occur. • Observe the safety instructions given in the hardware documentation. • Consider the residual risks for the risk evaluation.

⚠ WARNING
Malfunctions of the machine as a result of incorrect or changed parameter settings
As a result of incorrect or changed parameterization, machines can malfunction, which in turn can lead to injuries or death. • Protect the parameterization (parameter assignments) against unauthorized access. • Handle possible malfunctions by taking suitable measures, e.g. emergency stop or emergency off.

1.2 Warranty and liability for application examples

Application examples are not binding and do not claim to be complete regarding configuration, equipment or any eventuality which may arise. Application examples do not represent specific customer solutions, but are only intended to provide support for typical tasks.

As the user you yourself are responsible for ensuring that the products described are operated correctly. Application examples do not relieve you of your responsibility for safe handling when using, installing, operating and maintaining the equipment.

1.3 Industrial security

Note

Industrial security

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the Internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit:

Industrial security (http://www.siemens.com/industrialsecurity)

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at:

Industrial security (http://www.siemens.com/industrialsecurity)

Further information is provided on the Internet:

Industrial Security Configuration Manual (https://support.industry.siemens.com/cs/ww/en/view/108862708)

⚠ WARNING

Unsafe operating states resulting from software manipulation

Software manipulations (e.g. viruses, trojans, malware or worms) can cause unsafe operating states in your system that may lead to death, serious injury, and property damage.

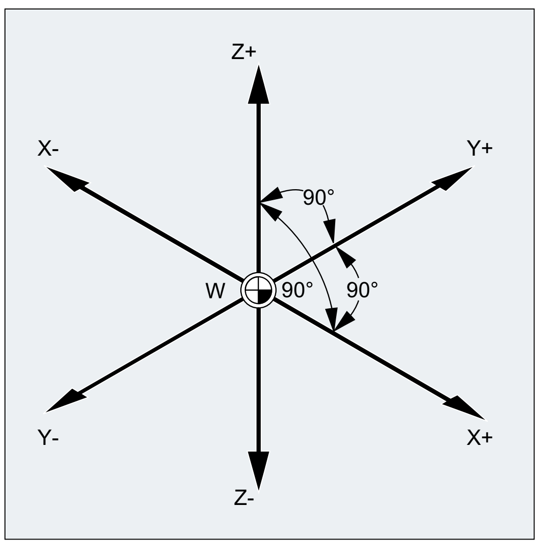
- Keep the software up to date.
- Incorporate the automation and drive components into a holistic, state-of-the-art industrial security concept for the installation or machine.
- Make sure that you include all installed products into the holistic industrial security concept.
- Protect files stored on exchangeable storage media from malicious software by with suitable protection measures, e.g. virus scanners.
- Protect the drive against unauthorized changes by activating the "know-how protection" drive function.

Fundamental Geometrical Principles 2

2.1 Workpiece positions

2.1.1 Workpiece coordinate systems

In order that the machine or the control can work with the positions specified in the NC program, these position specifications have to be made in a reference system that can be transferred to the directions of motion of the machine axes. Cartesian (i.e. clockwise, perpendicular) coordinate systems in accordance with DIN 66217 are used as workpiece coordinate system for machine tools.



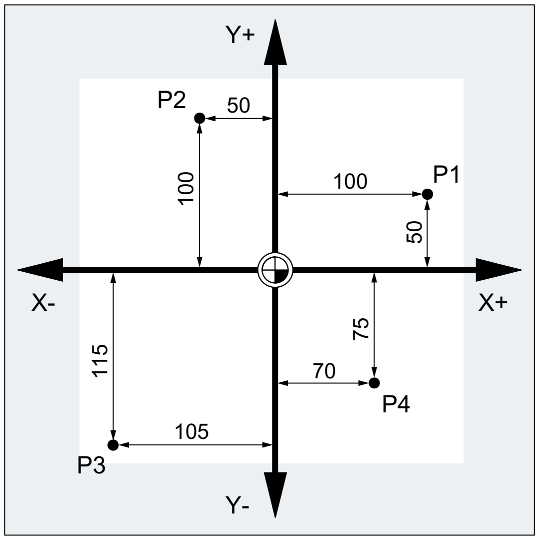
The workpiece zero (W) is the origin of the workpiece coordinate system.

2.1.2 Cartesian coordinates

The axes in the coordinate system are assigned dimensions. In this way, it is possible to clearly describe every point in the coordinate system and therefore every workpiece position through the direction (X, Y and Z) and three numerical values. The workpiece zero always has the coordinates X0, Y0, and Z0.

Position specifications in the form of Cartesian coordinates

To simplify things, we will only consider one plane of the coordinate system in the following example, the X/Y plane:

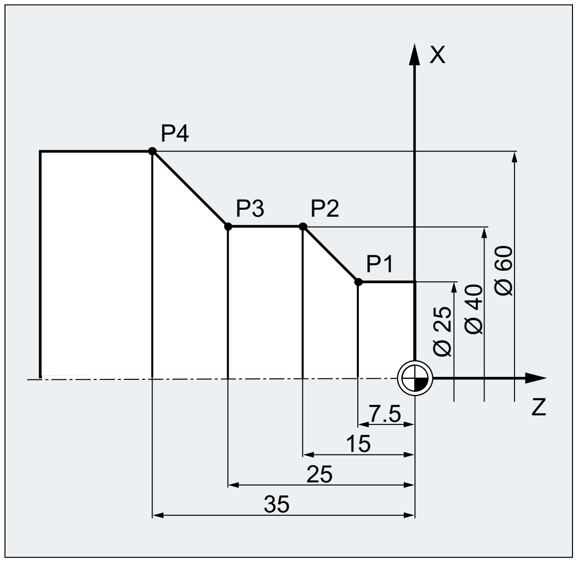


Points P1 to P4 have the following coordinates:

Position	Coordinates
P1	X100 Y50
P2	X-50 Y100
P3	X-105 Y-115
P4	X70 Y-75

Example: Workpiece positions for turning

With lathes, one plane is sufficient to describe the contour:

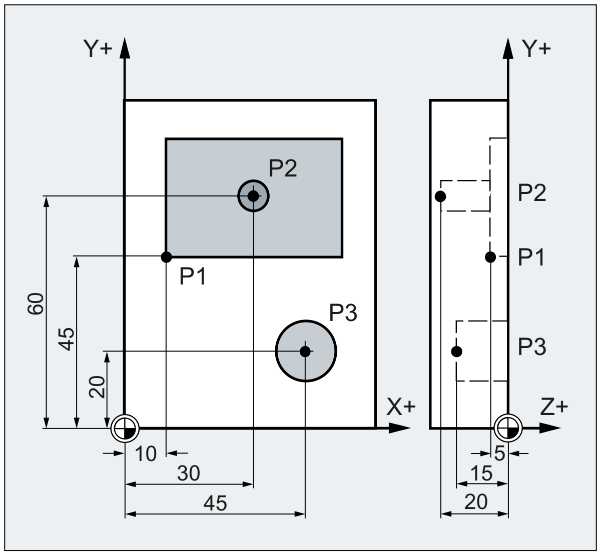


Points P1 to P4 have the following coordinates:

Position	Coordinates
P1	X25 Z-7.5
P2	X40 Z-15
P3	X40 Z-25
P4	X60 Z-35

Example: Workpiece positions for milling

For milling, the feed depth must also be described, i.e. the third coordinate (in this case Z) must also be assigned a numerical value.



Points P1 to P3 have the following coordinates:

Position	Coordinates
P1	X10 Y45 Z-5
P2	X30 Y60 Z-20
P3	X45 Y20 Z-15

2.1.3 Polar coordinates

Polar coordinates can be used instead of Cartesian coordinates to describe workpiece positions. This is useful when a workpiece or part of a workpiece has been dimensioned with radius and angle. The point from which the dimensioning starts is called the "pole".

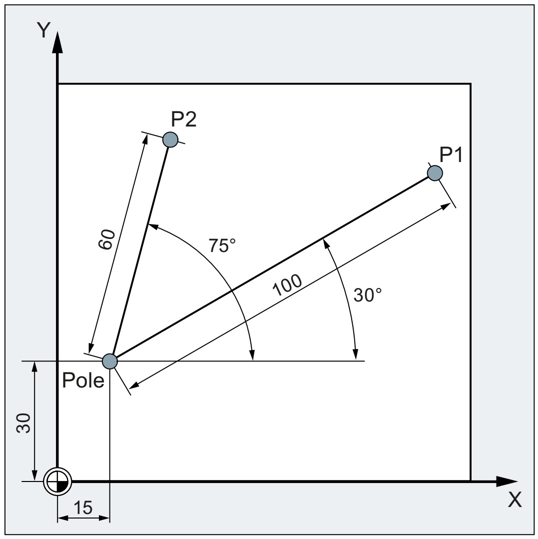
Position specifications in the form of polar coordinates

Polar coordinates are made up of the **polar radius** and the **polar angle**.

The polar radius is the distance between the pole and the position.

The polar angle is the angle between the polar radius and the horizontal axis of the working plane. Negative polar angles are in the clockwise direction, positive polar angles in the counterclockwise direction.

Example



Points P1 and P2 can then be described – with reference to the pole – as follows:

Position	Polar coordinates
P1	RP=100 AP=30
P2	RP=60 AP=75
RP: Polar radius AP: Polar angle	

2.1.4 Absolute dimensions

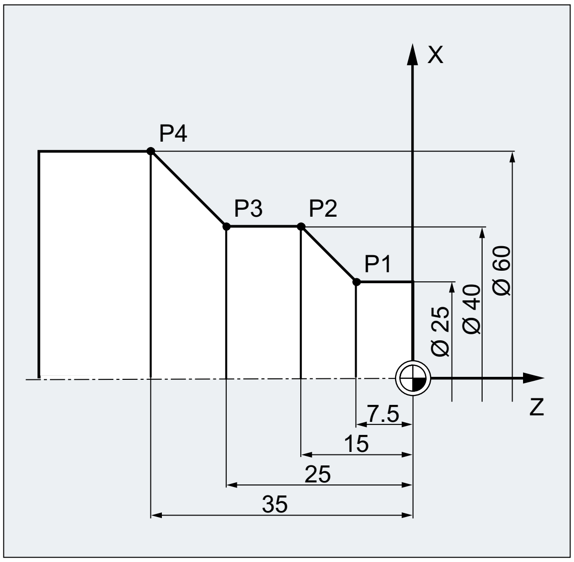
Position specifications in absolute dimensions

With absolute dimensions, all the position specifications refer to the currently valid zero point.

Applied to tool movement this means:

the position, to which the tool is to travel.

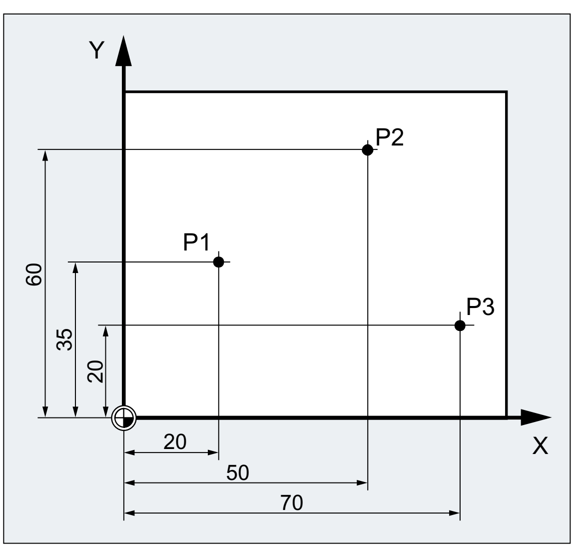
Example: Turning



In absolute dimensions, the following position specifications result for points P1 to P4:

Position	Position specification in absolute dimensions
P1	X25 Z-7.5
P2	X40 Z-15
P3	X40 Z-25
P4	X60 Z-35

Example: Milling



In absolute dimensions, the following position specifications result for points P1 to P3:

Position	Position specification in absolute dimensions
P1	X20 Y35
P2	X50 Y60
P3	X70 Y20

2.1.5 Incremental dimension

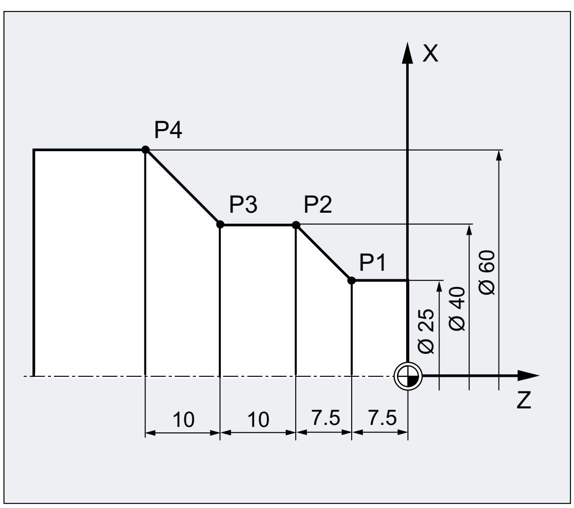
Position specifications in incremental dimensions

In production drawings, the dimensions often do not refer to a zero point, but rather to another workpiece point. So that these dimensions do not have to be converted, they can be specified in incremental dimensions. In this method of dimensional notation, a position specification refers to the previous point.

Applied to tool movement this means:

The incremental dimensions describe the distance the tool is to travel.

Example: Turning



In incremental dimensions, the following position specifications result for points P2 to P4:

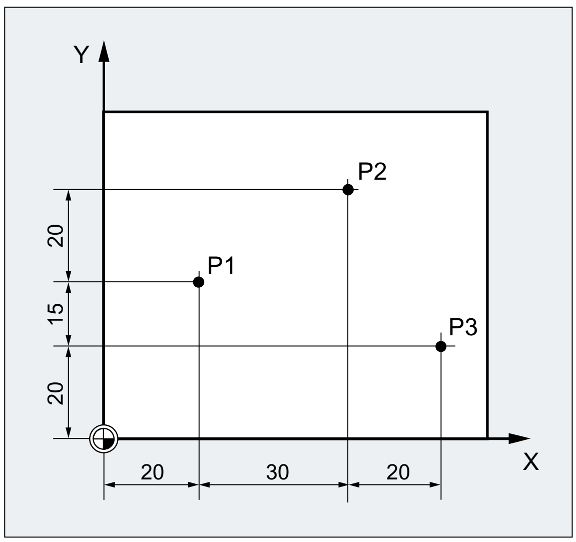
Position	Position specification in incremental dimensions	The specification refers to:
P2	X15 Z-7.5	P1
P3	Z-10	P2
P4	X20 Z-10	P3

Note

With `DIAMOF` or `DIAM90` active, the set distance in incremental dimensions (`G91`) is programmed as a radius dimension.

Example: Milling

The position specifications for points P1 to P3 in incremental dimensions are:



In incremental dimensions, the following position specifications result for points P1 to P3:

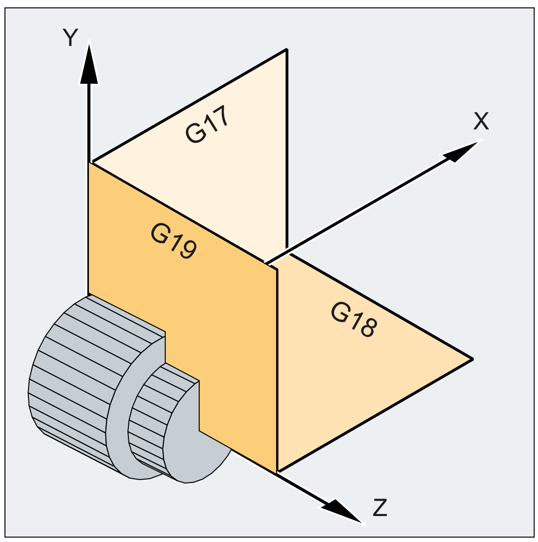
Position	Position specification in incremental dimensions	The specification refers to:
P1	X20 Y35	Zero point
P2	X30 Y20	P1
P3	X20 Y -35	P2

2.2 Working planes

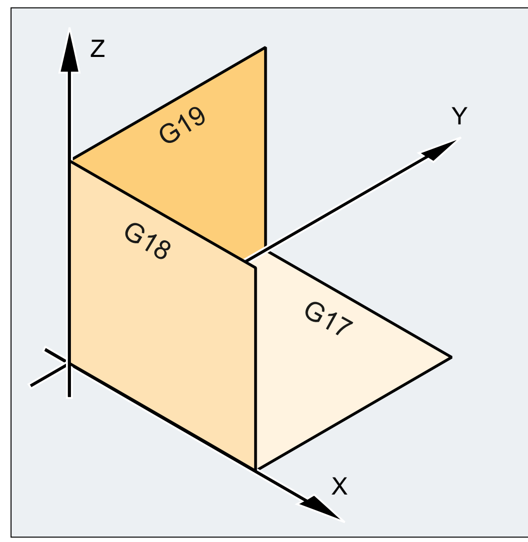
An NC program requires information about the machining plane, because only then can the control, for example, correct the tool correction values correctly. The specification of the working plane is also required for certain types of circular-path programming and polar coordinates.

The working plane is specified in the base Cartesian workpiece coordinate system with two coordinate axes. The third coordinate axis is perpendicular to this work plane and determines the infeed direction of the tool (e.g. for 2D machining).

Working planes for turning/milling



Working planes for turning



Working planes for milling

Activating a work plane

The working planes are activated defined in the NC program with the G commands `G17`, `G18` and `G19`. The relationship is defined as follows:

G command	Working plane	Abscissa	Ordinate	Applicate infeed direction
`G17`	X/Y	X	Y	Z
`G18`	Z/X	Z	X	Y
`G19`	Y/Z	Y	Z	X

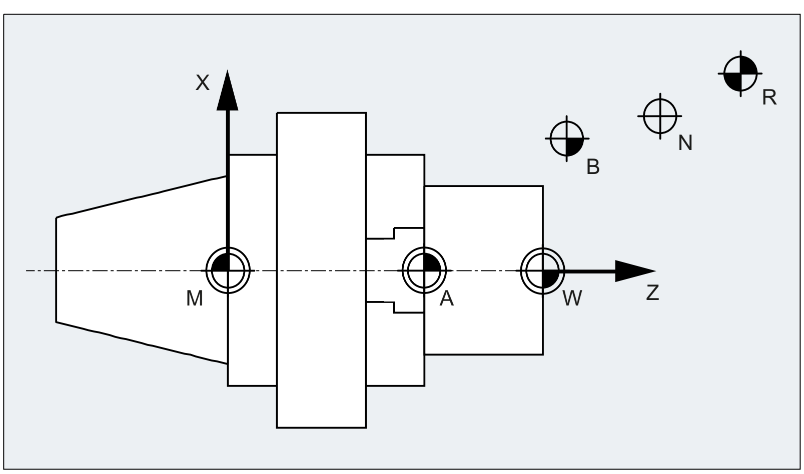
2.3 Zero points and reference points

Various zero points and reference points are defined on an NC machine:

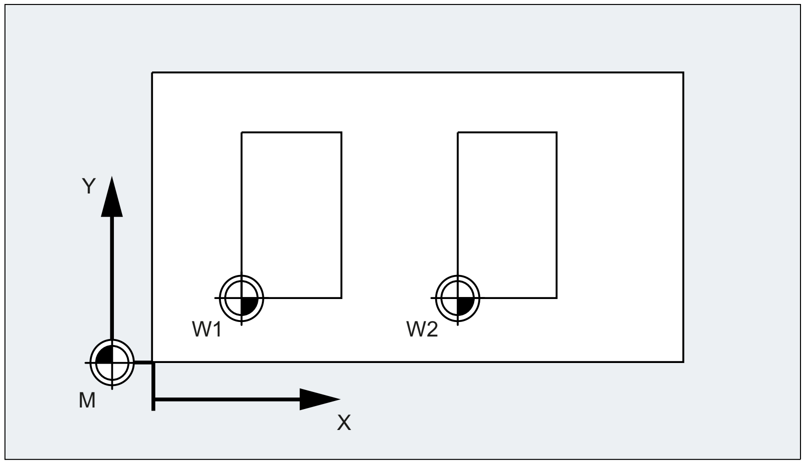
Zero points		
	M	Machine zero The machine zero defines the machine coordinate system (MCS). All other reference points refer to the machine zero.
	W	Workpiece zero = program zero The workpiece zero defines the workpiece coordinate system in relation to the machine zero.
	A	Blocking point Can be the same as the workpiece zero (only for lathes).

Reference points		
	R	Reference point Position defined by output cam and measuring system. The distance to the machine zero **M** must be known so that the axis position at this point can be set exactly to this value.
	B	Starting point Can be defined by the program. The 1st tool starts machining here.
	T	Toolholder reference point Is on the toolholder. By entering the tool lengths, the control calculates the distance between the tool tip and the toolholder reference point.
	N	Tool change point

Zero points and reference points for turning



Zero points for milling



2.4 Coordinate systems

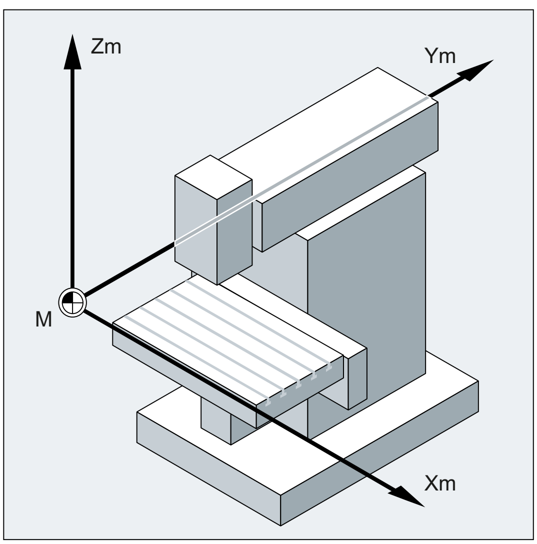
A distinction is made between the following coordinate systems:

- Machine coordinate system (MCS) (Page 28) with the machine zero **M**
- Basic coordinate system (BCS) (Page 30)
- Basic zero system (BZS) (Page 32)
- Settable zero system (SZS) (Page 33)
- Workpiece coordinate system (WCS) (Page 34) with the workpiece zero **W**

2.4.1 Machine coordinate system (MCS)

The machine coordinate system comprises all the physically existing machine axes.

Reference points and tool and pallet changing points (fixed machine points) are defined in the machine coordinate system.



If programming is performed directly in the machine coordinate system (possible with some G commands), the physical axes of the machine respond directly. Any workpiece clamping that is present is not taken into account.

Note

If there are various machine coordinate systems (e.g. 5-axis transformation), then an internal transformation is used to map the machine kinematics on the coordinate system in which the programming is performed.

Three-finger rule

The orientation of the coordinate system relative to the machine depends on the machine type. The axis directions follow the so-called "three-finger rule" of the **right** hand (according to DIN 66217).

Seen from in front of the machine, the middle finger of the right hand points in the opposite direction to the infeed of the main spindle. Therefore:

- the thumb points in the +X direction
- the index finger points in the +Y direction
- the middle finger points in the +Z direction

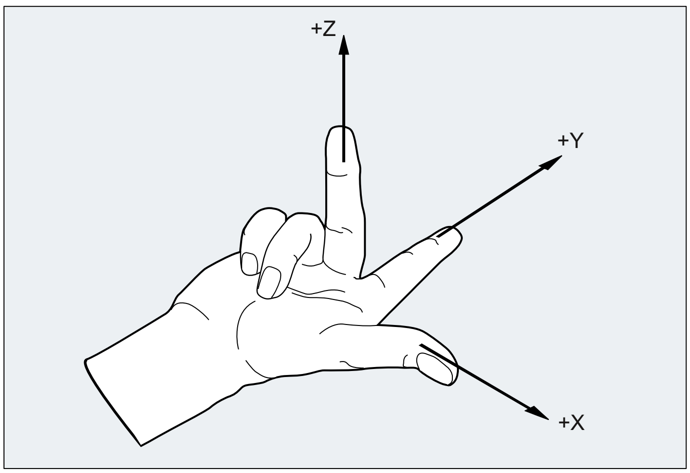
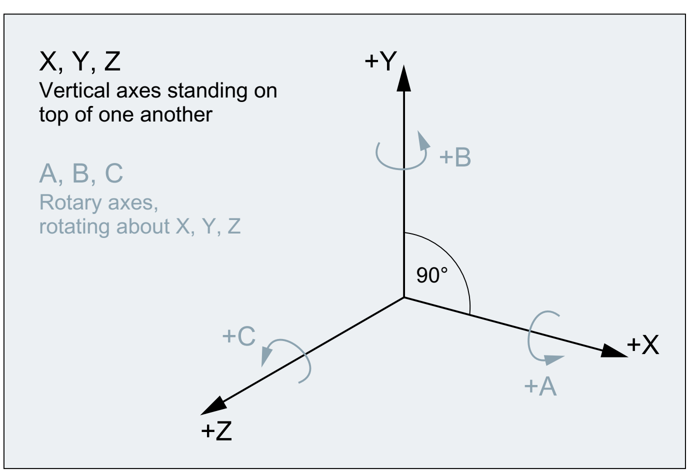


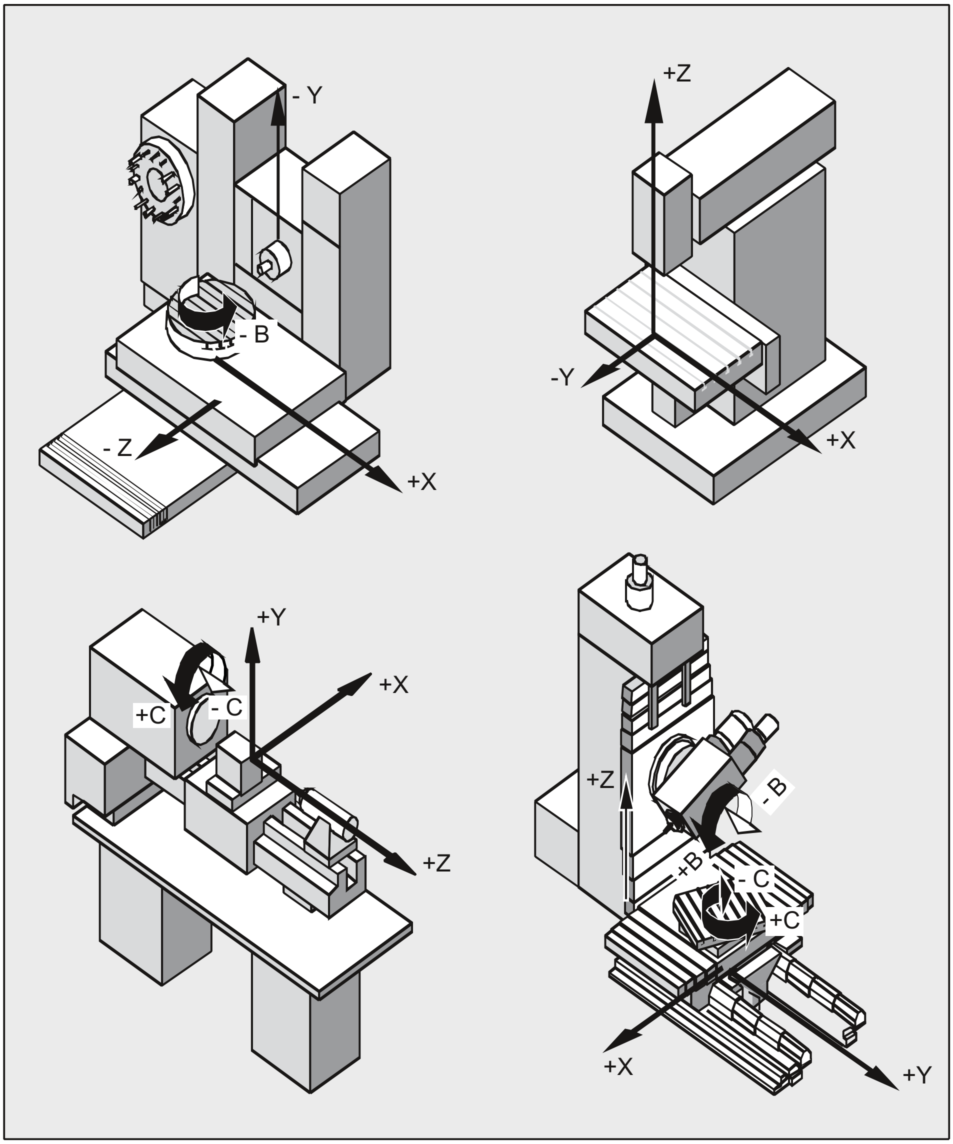
Figure 2-1 "Three-finger rule"

Rotary motions around the coordinate axes X, Y and Z are designated A, B and C. If the rotary motion is in a clockwise direction when looking in the positive direction of the coordinate axis, the direction of rotation is positive:



Position of the coordinate system in different machine types

The position of the coordinate system resulting from the "three-finger rule" can have a different orientation for different machine types. Here are a few examples:



2.4.2 Basic coordinate system (BCS)

The basic coordinate system (BCS) consists of three mutually perpendicular axes (geometry axes) as well as other special axes, which are not interrelated geometrically.

Machine tools without kinematic transformation

BCS and MCS always coincide when the BCS can be mapped onto the MCS without kinematic transformation (e.g. 5-axis transformation, TRANSMIT/TRACYL/TRAANG).

On such machines, machine axes and geometry axes can have the same names.

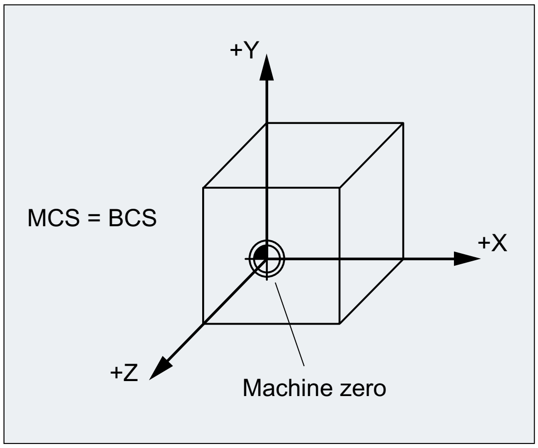


Figure 2-2 MCS = BCS without kinematic transformation

Machine tools with kinematic transformation

BCS and MCS do not coincide when the BCS is mapped onto the MCS with kinematic transformation (e.g. 5-axis transformation, TRANSMIT/TRACYL/TRAANG).

On such machines the machine axes and geometry axes must have different names.

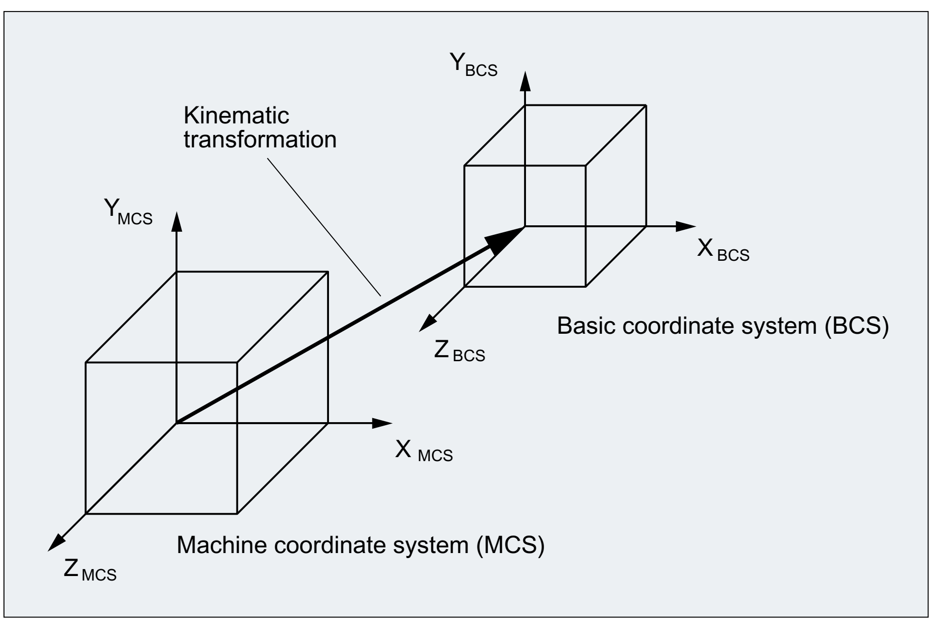


Figure 2-3 Kinematic transformation between the MCS and BCS

Machine kinematics

The workpiece is always programmed in a two- or three-dimensional, right-angled coordinate system (WCS). However, such workpieces are being programmed ever more frequently on machine tools with rotary axes or linear axes not perpendicular to one another. Kinematic transformation is used to represent coordinates programmed in the workpiece coordinate system (rectangular) in real machine movements.

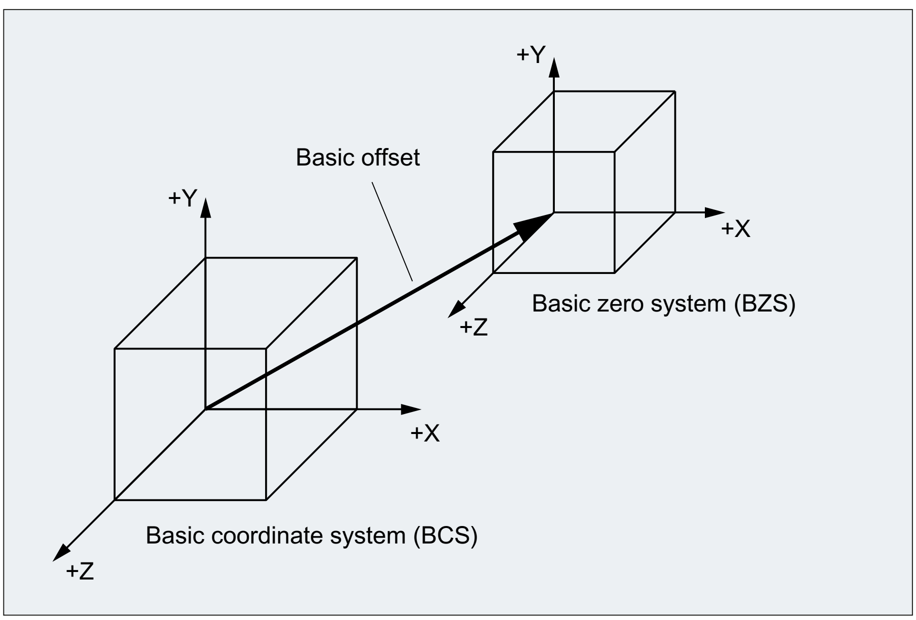
References

Function Manual Expansion Functions; M1: Kinematic transformation

Function Manual, Special Functions; F2: Multi-axis transformations

2.4.3 Basic zero system (BZS)

The basic zero system (BZS) is the basic coordinate system with a basic offset.



Basic offset

The basic offset describes the coordinate transformation between BCS and BZS. It can be used, for example, to define the palette window zero.

The basic offset comprises:

- External zero offset
- DRF offset
- Overlaid movement

- Chained system frames
- Chained basic frames

References

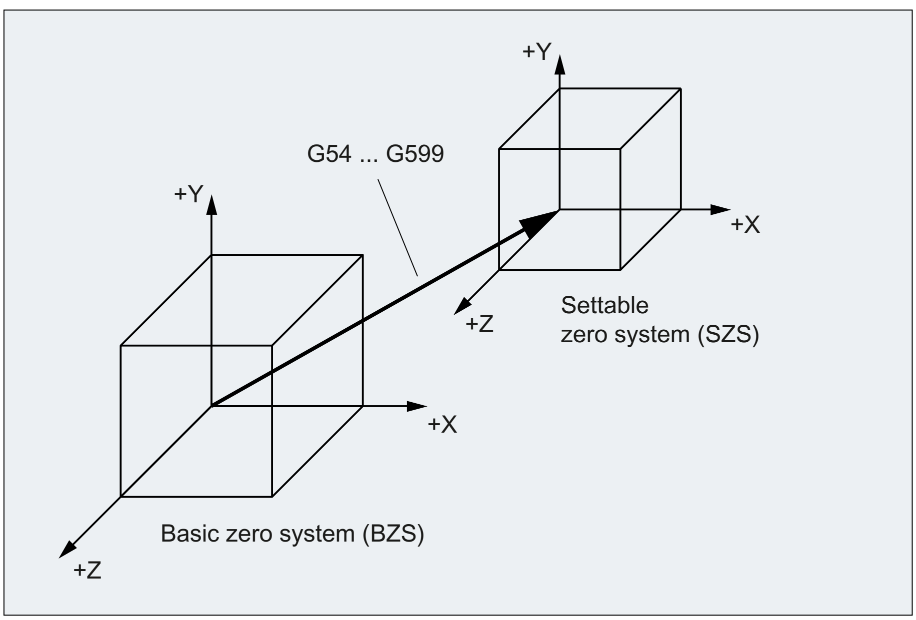
Function Manual, Basic Functions; Axes, Coordinate Systems, Frames (K2)

2.4.4 Settable zero system (SZS)

Settable zero offset

The "settable zero system" (SZS) results from the basic zero system (BZS) through the settable zero offset.

Settable zero offsets are activated in the NC program with the G commands `G54`...`G57` and `G505`...`G599` as follows:



If no programmable coordinate transformations (frames) are active, then the "settable zero system" is the workpiece coordinate system (WCS).

Programmable coordinate transformations (frames)

Sometimes it is useful or necessary within an NC program, to move the originally selected workpiece coordinate system (or the "settable zero system") to another position and, if required, to rotate it, mirror it and/or scale it. This is performed using programmable coordinate transformations (frames).

See Section: "Coordinate transformations (frames)"

Note

Programmable coordinate transformations (frames) always refer to the "settable zero system".

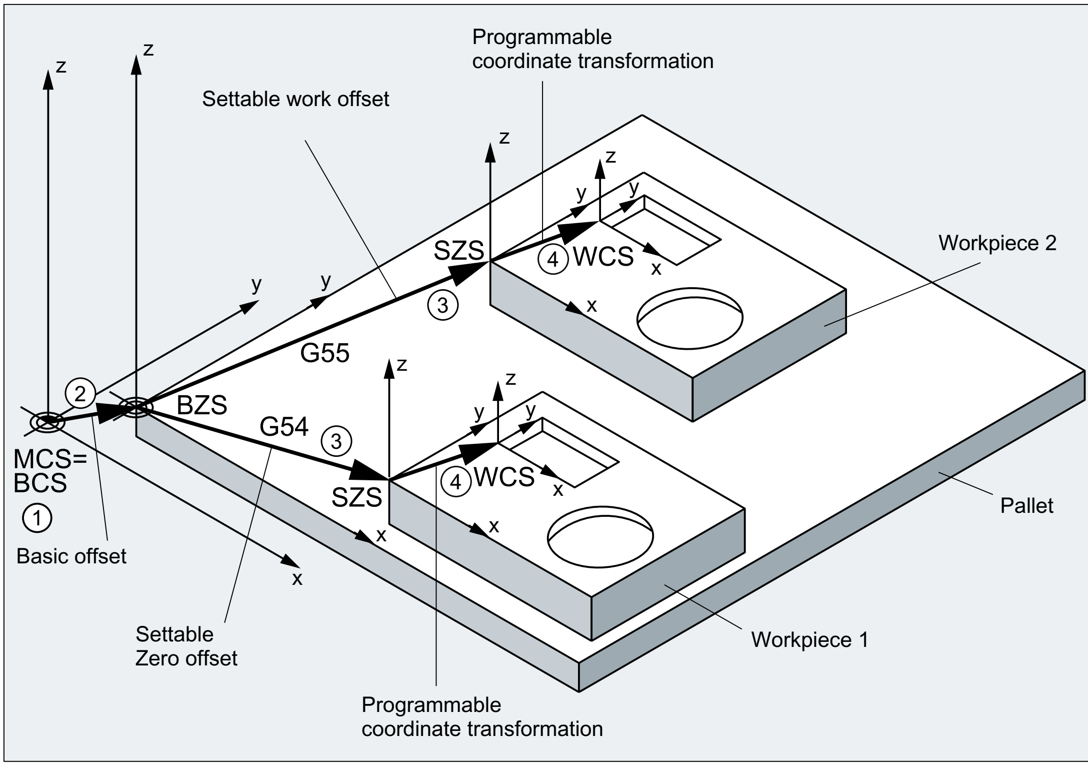
2.4.5 Workpiece coordinate system (WCS)

The geometry of a workpiece is described in the workpiece coordinate system (WCS). In other words, the data in the NC program refers to the workpiece coordinate system.

The workpiece coordinate system is always a Cartesian coordinate system and assigned to a specific workpiece.

2.4.6 What is the relationship between the various coordinate systems?

The example in the following figure should help clarify the relationships between the various coordinate systems:



① A kinematic transformation is not active, i.e. the machine coordinate system and the basic coordinate system coincide.

② The basic zero system (BZS) with the pallet zero result from the basic offset.

③ The "settable zero system" (SZS) for Workpiece 1 or Workpiece 2 is specified by the settable zero offset `G54` or `G55`.

④ The workpiece coordinate system (WCS) results from programmable coordinate transformation.

3 Fundamental Principles of NC Programming

Note

DIN 66025 is the guideline for NC programming.

3.1 Name of an NC program

Rules

Each NC program must be assigned a program name (designation) when it is created. The program name can be chosen freely providing the following rules are observed:

- Permissible characters:
  - Letters: A ... Z, a ... z
  - Numbers: 0 ... 9
  - Underscore: _
- The first two characters should either be two letters or an underscore followed by a letter.

  Note

  If this condition is satisfied, then an NC program can be called as subprogram from another program just by specifying the program name. However, if the program name starts with digits, the subprogram call is then only possible via the CALL statement.

- Maximum length: 24 characters

Note

Uppercase/lowercase letters

The SINUMERIK NC language does **not** distinguish between uppercase and lowercase letters.

Note

Impermissible program names

To avoid problems with Windows applications, the following program names may **not** be used:

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

Further restrictions, see "Names (Page 408)".

Control-internal extensions

The program name assigned when the program is created is expanded within the control with the addition of a prefix and a suffix:

- Prefix: _N_
- Suffix:
  - Main programs: _MPF
  - Subprograms: _SPF

Files in punch tape format

Externally created program files that are read via the V.24 interface must be present in punch tape format.

The following additional rules apply for the program name of a file in punch tape format:

- First character: %
- Then a four-character file extension: _xxx

Examples:

- %_N_SHAFT123_MPF
- %Flange3_MPF

References

For detailed information on downloading, creating and storing NC programs, see:

Turning, milling and grinding operating manual; "Manage programs" section

3.2 Structure and contents of an NC program

3.2.1 Blocks and block components

Blocks

An NC program consists of a sequence of NC blocks. Each block contains the data for the execution of a step in the workpiece machining.

Block components

NC blocks consist of the following components:

- Commands (statements) according to DIN 66025
- Elements of the NC high-level language

Commands according to DIN 66025

The commands according to DIN 66025 consist of an address character and a digit or sequence of digits representing an arithmetic value.

Address character (address)

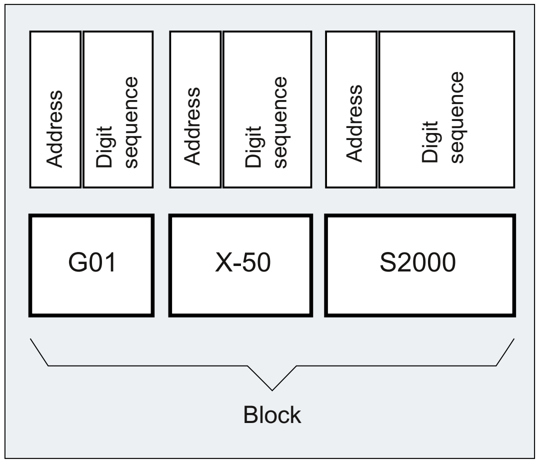
The address character (generally a letter) defines the meaning of the command.

Examples:

Address character	Meaning
G	G commands (preparatory functions)
X	Position data for the X axis
S	Spindle speed

Digit sequence

The digit sequence is the value assigned to the address character. The sequence of digits can contain a sign and decimal point. The sign always appears between the address letter and the sequence of digits. Positive signs (+) and leading zeroes (0) do not have to be specified.



Elements of the NC high-level language

As the command set according to DIN 66025 is no longer adequate for the programming of complex machining sequences in modern machine tools, it has been extended by the elements of the NC high-level language.

These include, for example:

- Commands of the NC high-level language
  In contrast to the commands according to DIN 66025, the commands of the NC high-level language consist of several address letters, e.g.
  - `OVR` for speed override
  - `SPOS` for spindle positioning
- Identifiers (defined names) for:
  - System variables
  - User-defined variables
  - Subprograms
  - Keywords
  - Jump markers
  - Macros

  Note

  An identifier must be unique and cannot be used for different objects.

- Comparison operators
- Logic operators
- Arithmetic functions
- Check structures

References:
Programming Manual, Job Planning; Section: "Flexible NC programming"

Effectiveness of commands

Commands are either modal or non-modal:

- Modal
  Modal commands retain their validity with the programmed value (in all following blocks) until:
  - A new value is programmed under the same command
  - A command is programmed that revokes the effect of the previously valid command
- Non-modal
  Non-modal commands only apply for the block in which they were programmed.

End of program

The last block in the execution sequence contains a special word for the end of program: `M2`, `M17` or `M30`.

3.2.2 Block rules

Start of block

NC blocks can be identified at the start of the block by block numbers. These consist of the character "N" and a positive integer, e.g.
`N40 ...`

The order of the block numbers is arbitrary, however, block numbers in rising order are recommended.

Note

Block numbers must be unique within a program in order to achieve an unambiguous result when searching.

End of block

A block ends with the character LF (LINE FEED = new line).

Note

The LF character does not have to be written. It is generated automatically by the line change.

Block length

A block can contain a maximum of **512 characters** (including the comment and end-of-block character LF).

Note

Three blocks of up to 66 characters each are normally displayed in the current block display on the screen. Comments are also displayed. Messages are displayed in a separate message window.

Order of the statements

In order to keep the block structure as clear as possible, the statements in a block should be arranged in the following order:

`N… G… X… Y… Z… F… S… T… D… M… H…`

Address	Meaning
`N`	Address of block number
`G`	Preparatory function
`X,Y,Z`	Positional data
`F`	Feedrate
`S`	Speed
`T`	Tool
`D`	Tool offset number
`M`	Additional function
`H`	Auxiliary function

Note

Certain addresses can be used repeatedly within a block, e.g.

`G…, M…, H…`

3.2.3 Value assignments

Values can be assigned to the addresses. The following rules apply:

- An "=" sign must be inserted between the address and the value if:
  - The address comprises more than one letter.
  - The value includes more than one constant.

  The "=" sign can be omitted if the address is a single letter and the value consists of only one constant.
- Signs are permitted.
- Separators are permitted after the address letter.

Examples:

`X10`	Value assignment (10) to address X, "=" not required
`X1=10`	Value assignment (10) to address (X) with numeric extension (1), "=" required
`X=10*(5+SIN(37.5))`	Value assignment by means of a numeric expression, "=" required

Note

A numeric extension must always be followed by one of the special characters "=", "(", "[", ")", "]", ",", or an operator, in order to distinguish an address with numeric extension from an address letter with a value.

3.2.4 Comments

To make an NC program easier to understand, comments can be added to the NC blocks.

A comment is at the end of a block and is separated from the program section of the NC block by a semicolon (";").

Example 1:

Program code	Comment
`N10 G1 F100 X10 Y20`	`; Comment to explain the NC block`

Example 2:

Program code	Comment
`N10`	`; Company G&S, order no. 12A71`
`N20`	`; Program written by H. Smith, Dept. TV 4 on November 21, 1994`
`N50`	`; Section no. 12, housing for submersible pump type TP23A`

Note

Comments are stored and appear in the current block display when the program is running.

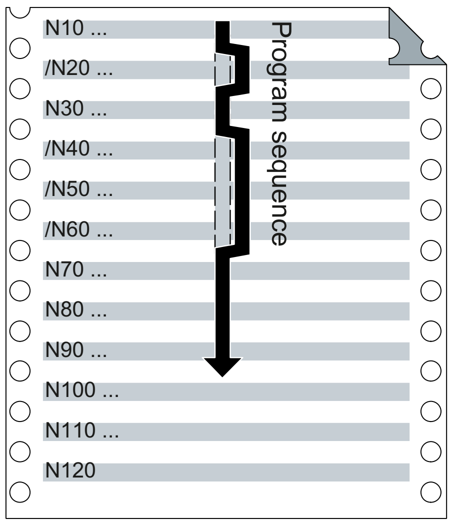
3.2.5 Skipping blocks

NC blocks which are not to be executed in every program pass (e.g. execute a trial program run), can be skipped.

Programming

Blocks which are to be skipped are marked with an oblique "/" in front of the block number. Several consecutive blocks can also be skipped. The statements in the skipped blocks are not executed; the program continues with the next block which is not skipped.

Example:



Program code	Comment
N10	; Is executed
/N20 …	; Skipped
N30 …	; Is executed
/N40 …	; Skipped
N70 …	; Is executed

Skip levels

Blocks can be assigned to skip levels (max. 10) which can be activated via the user interface.

Programming is performed by assigning a forward slash, followed by the number of the skip level. Only one skip level can be specified for each block.

Example:

Program code	Comment
/ ...	; Block is skipped (1st skip level)
/0 ...	; Block is skipped (1st skip level)
/1 N010...	; Block is skipped (2nd skip level)
/2 N020...	; Block is skipped (3rd skip level)
...	
/7 N100...	; Block is skipped (8th skip level)
/8 N080...	; Block is skipped (9th skip level)
/9 N090...	; Block is skipped (10th skip level)

Note

The number of skip levels that can be used depends on a display machine data item.

Note

System and user variables can also be used in conditional jumps in order to control program execution.

4 Creating an NC program

4.1 Basic procedure

The programming of the individual operation steps in the NC language generally represents only a small proportion of the work in the development of an NC program.

Programming of the actual instructions should be preceded by the planning and preparation of the operation steps. The more accurately you plan in advance how the NC program is to be structured and organized, the faster and easier it will be to produce a complete program, which is clear and free of errors. Clearly structured programs are especially advantageous when changes have to be made later.

As every part is not identical, it does not make sense to create every program in the same way. However, the following procedure has shown itself to be suitable in the most cases.

Procedure

1. **Prepare the workpiece drawing**
   - Define the workpiece zero
   - Draw the coordinate system
   - Calculate any missing coordinates
2. **Define the machining sequence**
   - Which tools are used when and for the machining of which contours?
   - In which order will the individual elements of the workpiece be machined?
   - Which individual elements are repeated (possibly also rotated) and should be stored in a subroutine?
   - Are there contour sections in other part programs or subroutines that could be used for the current workpiece?
   - Where are zero offsets, rotating, mirroring and scaling useful or necessary (frame concept)?

3. **Create a machining plan**
   Define all machining operations step-by-step, e.g.
   - Rapid traverse movements for positioning
   - Tool change
   - Define the machining plane
   - Retraction for checking
   - Switch spindle, coolant on/off
   - Call up tool data
   - Feed
   - Path correction
   - Approaching the contour
   - Retraction from the contour
   - etc.
4. **Compile machining steps in the programming language**
   - Write each individual step as an NC block (or NC blocks).
5. **Combine the individual steps into a program**

4.2 Available characters

The following characters are available for writing NC programs:

- Uppercase characters:
  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
- Lowercase characters:
  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
- Numbers:
  0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Special characters:
  See the table below.

Special characters	Meaning
%	Program start character (used only for writing programs on an external PC)
(	For bracketing parameters or expressions
)	For bracketing parameters or expressions
[	For bracketing addresses or indexes
]	For bracketing addresses or indexes
<	Less than
>	Greater than
:	Main block, end of label, chain operator
=	Assignment, part of equation
/	Division, block suppression
*	Multiplication
+	Addition
-	Subtraction, minus sign
"	Double quotation marks, identifier for character string
'	Single quotation marks, identifier for special numerical values: hexadecimal, binary
$	System variable identifiers
s_	Underscore, belonging to letters
?	Reserved
!	Reserved
.	Decimal point
,	Comma, parameter separator
;	Comment start
&	Format character, same effect as space character
LF	End of block
Tab character	Separator
Blank	Separator (blank)

Note

Take care to differentiate between the letter "O" and the digit "0".

Note

No distinction is made between uppercase and lowercase characters (exception: tool call).

Note

Non-printable special characters are treated like blanks.

4.3 Program header

The NC blocks that are placed in front of the actual motion blocks for the machining of the workpiece contour are called the program header.

The program header contains information/statements regarding:

- Tool change
- Tool offsets
- Spindle motion
- Feed control
- Geometry settings (zero offset, selection of the working plane)

Program header for turning

The following example shows the typical structure of an NC program header for turning:

Program code	Comment
N10 G0 G153 X200 Z500 T0 D0	; Retract toolholder before tool turret is rotated.
N20 T5	; Swing in tool 5.
N30 D1	; Activate cutting edge data set of the tool.
N40 G96 S300 LIMS=3000 M4 M8	; Constant cutting rate (Vc) = 300 m/min, speed limitation = 3000 rpm, direction of rotation counterclockwise, cooling on.
N50 DIAMON	; X axis will be programmed in the diameter.
N60 G54 G18 G0 X82 Z0.2	; Call zero offset and working plane, approach starting position.
...	

Program header for milling

The following example shows the typical structure of an NC program header for milling:

Program code	Comment
N10 T="SF12"	; Alternative: T123
N20 M6	; Trigger tool change.
N30 D1	; Activate cutting edge data set of the tool.
N40 G54 G17	; Zero offset and working plane.
N50 G0 X0 Y0 Z2 S2000 M3 M8	; Approach to the workpiece, spindle and coolant on.
...	

If tool orientation / coordinate transformation is being used, any transformations still active should be deleted at the start of the program:

Program code	Comment
N10 CYCLE800()	; Resetting of the swiveled plane
N20 TRAFOOF	; Resetting of TRAORI, TRANSMIT, TRACYL, ...

Program code	Comment
...	

4.4 Program examples

4.4.1 Example 1: First programming steps

Program example 1 is to be used to perform and test the first programming steps on the NC.

Procedure

1. Create a new part program (name)
2. Edit the part program
3. Select the part program
4. Activate single block
5. Start the part program

References:
Operating Manual for the existing user interface

Note

In order that the program can run on the machine, the machine data must have been set appropriately (→ machine manufacturer!).

Note

Alarms can occur during program verification. These alarms have to be reset first.

Program example 1

Program code	Comment
`N10  MSG("THIS IS MY NC PROGRAM")`	`; Message "THIS IS MY NC PROGRAM" displayed in the alarm line.`
`N20 F200 S900 T1 D2 M3`	`; Feedrate, spindle, tool, tool offset, spindle clockwise.`
`N30 G0 X100 Y100`	`; Approach position in rapid traverse.`
`N40 G1 X150`	`; Rectangle with feedrate, straight line in X.`
`N50 Y120`	`; Straight line in Y.`
`N60 X100`	`; Straight line in X.`
`N70 Y100`	`; Straight line in Y.`
`N80 G0 X0 Y0`	`; Retraction in rapid traverse.`
`N100 M30`	`; End of block.`

4.4.2 Example 2: NC program for turning

Program example 2 is intended for the machining of a workpiece on a lathe. It contains radius programming and tool radius compensation.

Note

In order that the program can run on the machine, the machine data must have been set appropriately (→ machine manufacturer!).

Dimension drawing of the workpiece

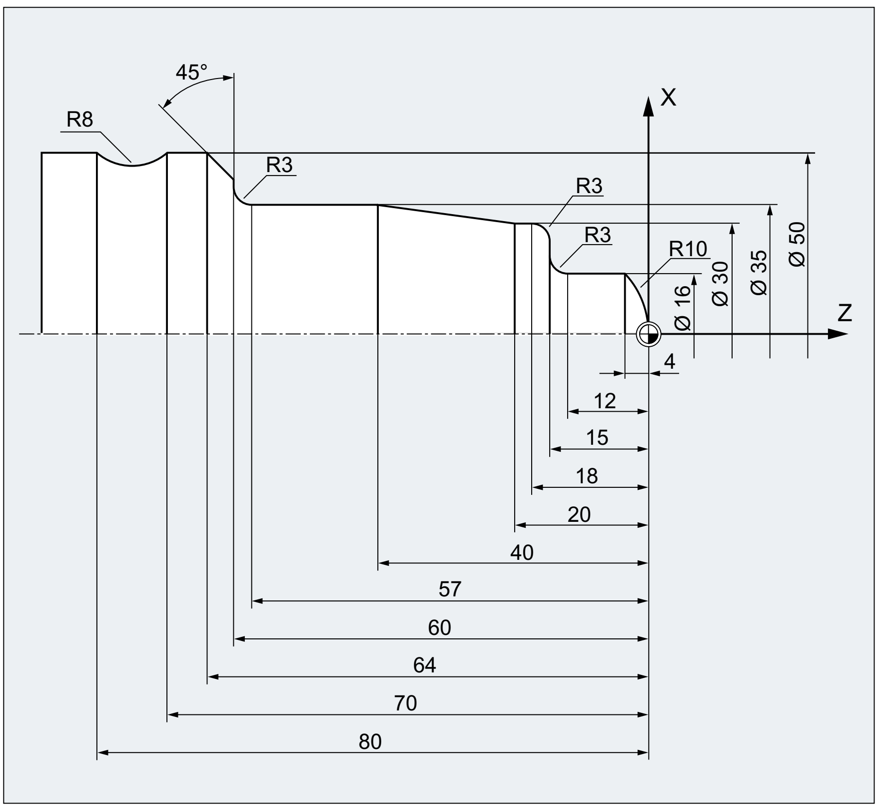


Figure 4-1 Top view

Program example 2

Program code	Comment
N5 G0 G53 X280 Z380 D0	; Starting point.
N10 TRANS X0 Z250	; Zero offset
N15 LIMS=4000	; Speed limitation (G96).
N20 G96 S250 M3	; Select constant cutting rate.
N25 G90 T1 D1 M8	; Select tool selection and offset.
N30 G0 G42 X-1.5 Z1	; Set tool with tool radius compensation.

Program code	Comment
N35 G1 X0 Z0 F0.25	
N40 G3 X16 Z-4 I0 K-10	; Turn radius 10.
N45 G1 Z-12	
N50 G2 X22 Z-15 CR=3	; Turn radius 3.
N55 G1 X24	
N60 G3 X30 Z-18 I0 K-3	; Turn radius 3.
N65 G1 Z-20	
N70 X35 Z-40	
N75 Z-57	
N80 G2 X41 Z-60 CR=3	; Turn radius 3.
N85 G1 X46	
N90 X52 Z-63	
N95 G0 G40 G97 X100 Z50 M9	; Deselect tool radius compensation and approach tool change location.
N100 T2 D2	; Call tool and select offset.
N105 G96 S210 M3	; Select constant cutting rate.
N110 G0 G42 X50 Z-60 M8	; Set tool with tool radius compensation.
N115 G1 Z-70 F0.12	; Turn diameter 50.
N120 G2 X50 Z-80 I6.245 K-5	; Turn radius 8.
N125 G0 G40 X100 Z50 M9	; Retract tool and deselect tool radius compensation.
N130 G0 G53 X280 Z380 D0 M5	; Approach tool change location.
N135 M30	; End of program.

4.4.3 Example 3: NC program for milling

Program example 3 is intended for the machining of a workpiece on a vertical milling machine. It contains surface and side milling as well as drilling.

Note

In order that the program can run on the machine, the machine data must have been set appropriately (→ machine manufacturer!).

Dimension drawing of the workpiece

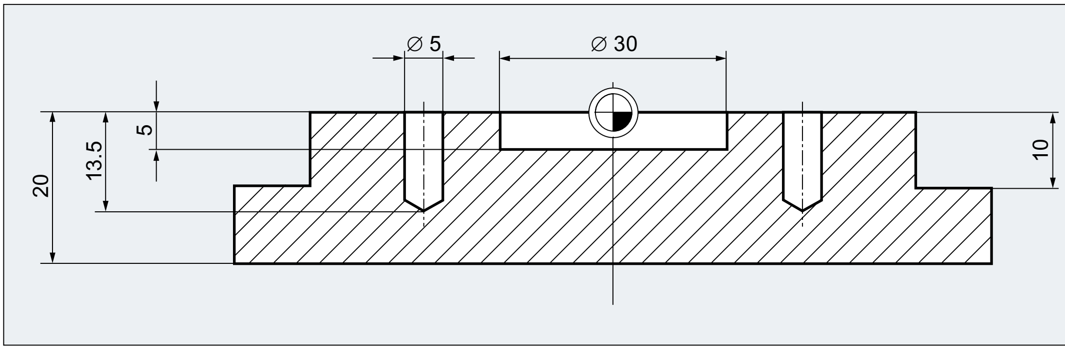


Figure 4-2 Side view

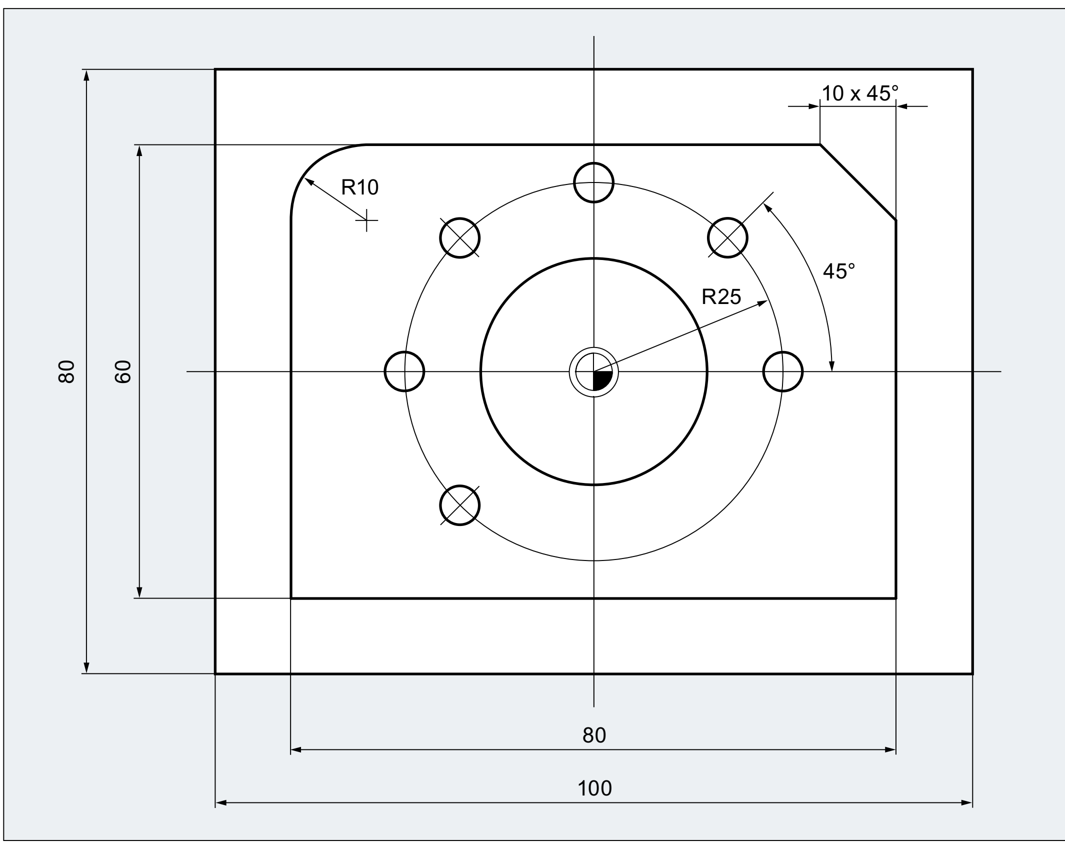


Figure 4-3 Top view

Program example 3

Program code	Comment
N10 T="PF60"	; Preselection of the tool with name PF60.
N20 M6	; Load the tool into the spindle.

Program code	Comment
N30 S2000 M3 M8	; Speed, direction of rotation, cooling on.
N40 G90 G64 G54 G17 G0 X-72 Y-72	; Basic settings of the geometry and approach starting point.
N50 G0 Z2	; Z axis at safety clearance.
N60 G450 CFTCP	; Behavior with active G41/ G42.
N70 G1 Z-10 F3000	; Milling tool at contact depth with feedrate = 3000 mm/ min.
N80 G1 G41 X-40	; Activation of the milling tool radius compensation.
N90 G1 X-40 Y30 RND=10 F1200	; Travel to the contour with feedrate = 1200 mm/min.
N100 G1 X40 Y30 CHR=10	
N110 G1 X40 Y-30	
N120 G1 X-41 Y-30	
N130 G1 G40 Y-72 F3000	; Deselection of the milling tool radius compensation.
N140 G0 Z200 M5 M9	; Retraction of the milling tool, spindle + cooling off.
N150 T="SF10"	; Preselection of the tool with name SF10.
N160 M6	; Load the tool into the spindle.
N170 S2800 M3 M8	; Speed, direction of rotation, cooling on.
N180 G90 G64 G54 G17 G0 X0 Y0	; Basic settings for the geometry and approach starting point.
N190 G0 Z2	
N200 POCKET4(2,0,1,-5,15,0,0,0,0,0,800,1300,0,21,5,,,2,0.5)	; Call of the pocket milling cycle.
N210 G0 Z200 M5 M9	; Retraction of the milling tool, spindle + cooling off.
N220 T="ZB6"	; Call center drill 6 mm.
N230 M6	
N240 S5000 M3 M8	
N250 G90 G60 G54 G17 X25 Y0	; Exact stop G60 for exact positioning.
N260 G0 Z2	
N270 MCALL CYCLE82(2,0,1,-2.6,,0)	; Modal call of the drilling cycle.
N280 POSITION:	; Jump mark for repetition.
N290 HOLES2(0,0,25,0,45,6)	; Position pattern for drilling.
N300 ENDLABEL:	; End identifier for repetition.

Program code	Comment
N310 MCALL	; Resetting of the modal call.
N320 G0 Z200 M5 M9	
N330 T="SPB5"	; Call drill D 5 mm.
N340 M6	
N350 S2600 M3 M8	
N360 G90 G60 G54 G17 X25 Y0	
N370 MCALL CYCLE82(2,0,1,-13.5,,0)	; Modal call of the drilling cycle.
N380 REPEAT POSITION	; Repetition of the position description from centering.
N390 MCALL	; Resetting of the drilling cycle.
N400 G0 Z200 M5 M9	
N410 M30	; End of program.

5 Tool change

Tool change method

In chain, rotary-plate and box magazines, a tool change normally takes place in two stages:

1. The tool is sought in the magazine with the T command.
2. The tool is then loaded into the spindle with the M command.

In circular magazines on turning machines, the T command carries out the entire tool change, that is, locates and inserts the tool.

Note

The tool change method is set via a machine data (→ machine manufacturer).

Conditions

Together with the tool change:

- The tool offset values stored under a D number have to be activated.
- The appropriate working plane has to be programmed (basic setting: G18). This ensures that the tool length compensation is assigned to the correct axis.

Tool management (option)

The programming of the tool change is performed differently for machines with active tool management (option) than for machines without active tool management. The two options are therefore described separately.

5.1 Tool change without tool management

5.1.1 Tool change with T command

There is a direct tool change when the T command is programmed.

Application

For turning machines with circular magazine.

Syntax

Tool selection:
`T<number>`
`T=<number>`
`T<n>=<number>`

Tool deselection:
`T0`
`T0=<number>`

Meaning

`T`:	Command for tool selection including tool change and activation of the tool offset	
`<n>`:	Spindle number as address extension **Note:** The possibility of programming a spindle number as address extension depends on the configuration of the machine; → see machine manufacturer's specifications.	
`<number>`:	Number of the tool	
	Range of values:	0 - 32000
`T0`:	Command for deselection of the active tool	

Example

Program code	Comment
`N10 T1 D1`	`; Loading of tool T1 and activation of the tool offset D1.`
`...`	
`N70 T0`	`; Deselect tool T1.`
`...`	

5.1.2 Tool change with M6

The tool is selected when the `T` command is programmed. The tool only becomes active with `M6` (including tool offset).

Application

For milling machines with chain, rotary-plate or box magazines.

Syntax

Tool selection:

```
T<number>
T=<number>
T<n>=<number>
```

Tool change:

```
M6
```

Tool deselection:

```
T0
T0=<number>
```

Meaning

T:	Command for the tool selection
<n>:	Spindle number as address extension **Note:** The possibility of programming a spindle number as address extension depends on the configuration of the machine; → see machine manufacturer's specifications.
<number>:	Number of the tool Range of values: 0 - 32000
M6:	M function for the tool change (according to DIN 66025) M6 activates the selected tool (T…) and the tool offset (D...).
T0:	Command for deselection of the active tool

Example

```
Program code                 Comment
N10 T1 M6                    ; Loading of tool T1.
N20 D1                       ; Selection of tool length offset.
N30 G1 X10 ...               ; Machining with T1.
...
N70 T5                       ; Preselection of tool T5.
N80 ...                      ; Machining with T1.
...
N100 M6                      ; Loading of tool T5.
N110 D1 G1 X10 ...           ; Machining with tool T5.
...
```

5.2 Tool change with tool management (option)

Tool management

The optional "Tool management" function ensures that at any given time the correct tool is in the correct location and that the data assigned to the tool are up to date. It also allows fast tool changes and avoids both scrap by monitoring the tool service life and machine downtimes by using spare tools.

Tool name

On a machine tool with active tool management, the tools must be assigned a name and number for clear identification (e.g. "Drill", "3").

The tool call can then be via the tool name, e.g.
`T="Drill"`

Note

The tool name may not contain any special characters.

5.2.1 Tool change with T command with active tool management (option)

There is a direct tool change when the T command is programmed.

Application

For turning machines with circular magazine.

Syntax

Tool selection:
```
T=<location>
T=<name>
T<n>=<location>
T<n>=<name>
```

Tool deselection:
```
T0
```

Meaning

`T=`:	Command for tool change and activation of the tool offset The following specifications are possible:	
	`<location>`:	Number of the magazine location
	`<name>`:	Name of tool **Note:** The correct notation (uppercase/lowercase) must be used when programming a tool name.
`<n>`:	Spindle number as address extension **Note:** The possibility of programming a spindle number as an address extension depends on the configuration of the machine; → see machine manufacturer's specifications.	
`T0`:	Command for the tool deselection (magazine location not occupied)	

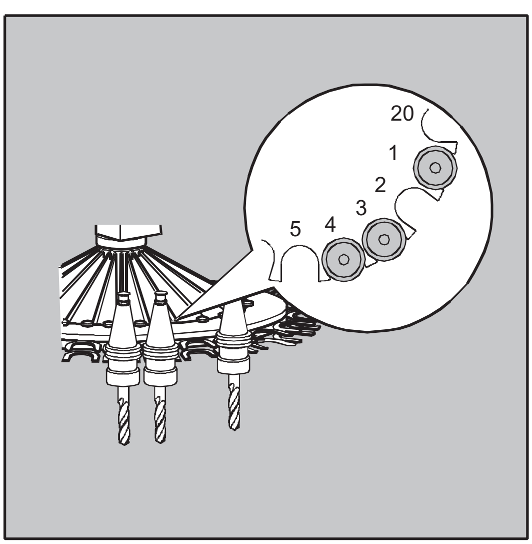
Note

If the selected magazine location is not occupied in a tool magazine, the command acts as for `T0`. The selection of the next occupied magazine location can be used to position the empty location.

Example

A circular magazine has locations 1 to 20 with the following tool assignment:

Location	Tool	Tool group	State
1	Drill, duplo no. = 1	T15	Blocked
2	Not occupied		
3	Drill, duplo no. = 2	T10	Enabled
4	Drill, duplo no. = 3	T1	Active
5 ... 20	Not occupied		



The following tool call is programmed in the NC program:
`N10 T=1`

The call is processed as follows:

1. Magazine location 1 is considered and the tool identifier determined.
2. The tool management recognizes that this tool is blocked and therefore cannot be used.
3. A tool search for T="drill" is initiated in accordance with the search method set:
   "Find the active tool; or else, select the one with the next highest duplo number."
4. The following usable tool is then found:
   "Drill", duplo no. 3 (at magazine location 4)
   This completes the tool selection process and the tool change is initiated.

   Note

   If the "Select the first available tool from the group" search method is employed, the sequence must first be defined within the tool group being loaded. In this case group T10 is loaded, as T15 is blocked.

   When the strategy "Take the first tool with "active" status from the group" is applied, T1 is loaded.

5.2.2 Tool change with M6 with active tool management (option)

The tool is selected when the `T` command is programmed. The tool only becomes active with `M6` (including tool offset).

Application

For milling machines with chain, rotary-plate or box magazines.

Syntax

Tool selection:
```
T=<location>
T=<name>
T<n>=<location>
T<n>=<name>
```

Tool change:
```
M6
```

Tool deselection:
```
T0
```

Meaning

`T=`:	Command for the tool selection The following specifications are possible:	
	`<location>`:	Number of the magazine location
	`<name>`:	Name of tool **Note:** The correct notation (uppercase/lowercase) must be used when programming a tool name.
`<n>`:	Spindle number as address extension **Note:** The possibility of programming a spindle number as an address extension depends on the configuration of the machine; → see machine manufacturer's specifications.	
`M6`:	M function for the tool change (according to DIN 66025) `M6` activates the selected tool (`T…`) and the tool offset (`D...`).	
`T0`:	Command for the tool deselection (magazine location not occupied)	

Note

If the selected magazine location is not occupied in a tool magazine, the command acts as for `T0`. The selection of the next occupied magazine location can be used to position the empty location.

Example

Program code	Comment
`N10 T=1 M6`	`; Loading of the tool from magazine location 1.`
`N20 D1`	`; Selection of tool length offset.`
`N30 G1 X10 ...`	`; Machining with tool T=1.`
`...`	
`N70 T="Drill"`	`; Preselection of the tool with name "Drill".`
`N80 ...`	`; Machining with tool T=1.`
`...`	
`N100 M6`	`; Loading of the drill.`
`N140 D1 G1 X10 ...`	`; Machining with drill.`
`...`	

5.3 Behavior with faulty T programming

The behavior with faulty T programming depends on the configuration of the machine:

MD22562 TOOL_CHANGE_ERROR_MODE		
Bit	**Value**	**Meaning**
7	0	Basic setting! With the T programming, a check is made immediately as to whether the NC recognizes the T number. If not, an alarm is triggered.
	1	The programmed T number will only be checked following D selection. If the NC does not recognize the tool number, an alarm is issued during D selection. This response is desirable if, for example, tool programming is also intended to achieve positioning and the tool data is not necessarily available (circular magazine).

Tool offsets 6

6.1 General information about the tool offsets

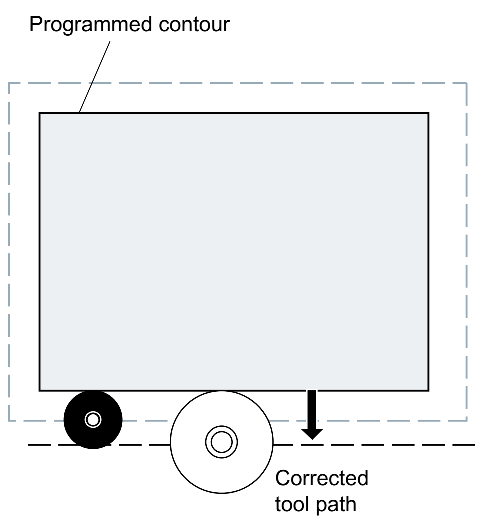
Workpiece dimensions are programmed directly (e.g. according to the production drawing). Therefore, tool data such as milling tool diameter, cutting edge position of the turning tool (counterclockwise/clockwise turning tool) and tool length does not have to be taken into consideration when creating the program.

The control corrects the travel path

When machining a workpiece, the tool paths are controlled according to the tool geometry such that the programmed contour can be machined using any tool.

In order that the control can calculate the tool paths, the tool data must be entered in the tool compensation memory of the control. Only the required tool (`T...`) and the required offset data record (`D...`) are called via the NC program.

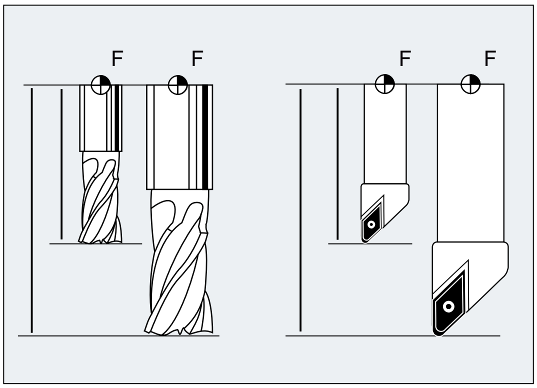
While the program is being processed, the control fetches the offset data it requires from the tool compensation memory and corrects the tool path individually for different tools:



6.2 Tool length compensation

The tool length compensation compensates for the differences in length between the tools used.

The tool length is the distance between the toolholder reference point and the tool tip:



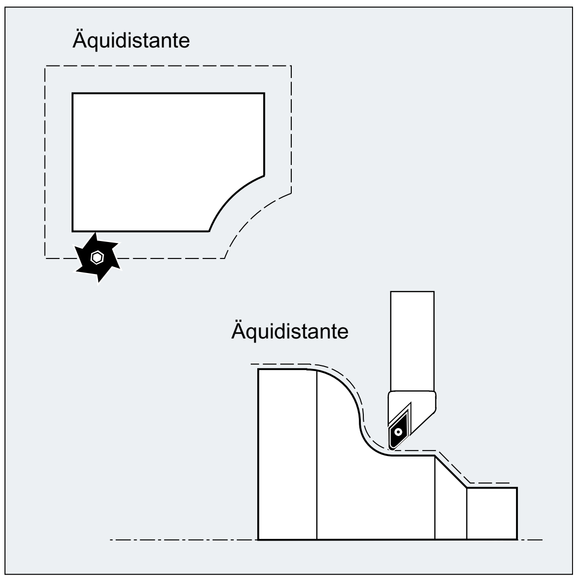
This length is measured and entered in the tool compensation memory of the control together with definable wear values. From this data, the control calculates the traversing movements in the infeed direction.

Note

The offset value for the tool length is dependent upon the spatial orientation of the tool.

6.3 Tool radius compensation

The contour and tool path are not identical. The milling tool or cutting edge center must travel along a path corresponding to the tool radius that is equidistant from the contour (tool center path). To do this, while executing the program, the control shifts the programmed tool center point path – based on the tool radius of the active tool – so that the tool cutting edge precisely traverses along the programmed contour.



The tool radius compensation is described in detail in Section "Tool radius compensation (Page 263)".

See also

2 1/2 D tool offset (CUT2D, CUT2DD, CUT2DF, CUT2DFD) (Page 301)

6.4 Tool compensation memory

The following data must be available in the tool compensation memory of the controller for each tool edge:

- Tool type
- Cutting edge position
- Tool geometry variables (length, radius)

This data is entered as tool parameters (max. 25). Which parameters are required for a tool depends on the tool type. Any tool parameters that are not required must be set to "zero" (corresponds to the default setting of the system).

Note

Values that have been entered once in the compensation memory are included in the processing at each tool call.

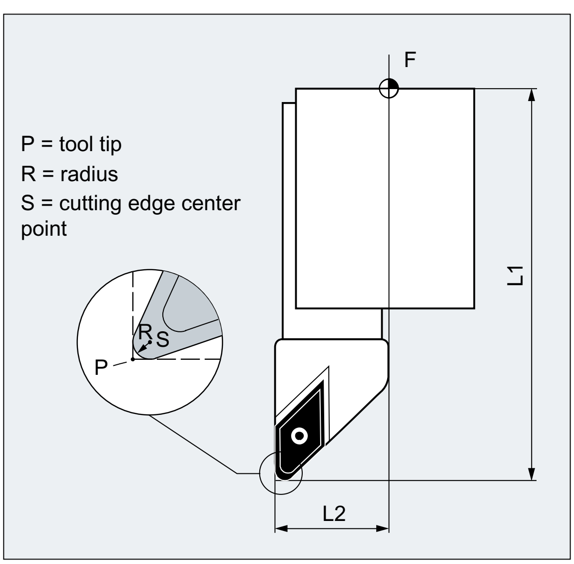
Tool type

The tool type (drill, milling or turning tool) determines which geometry data is necessary and how this is taken into account.

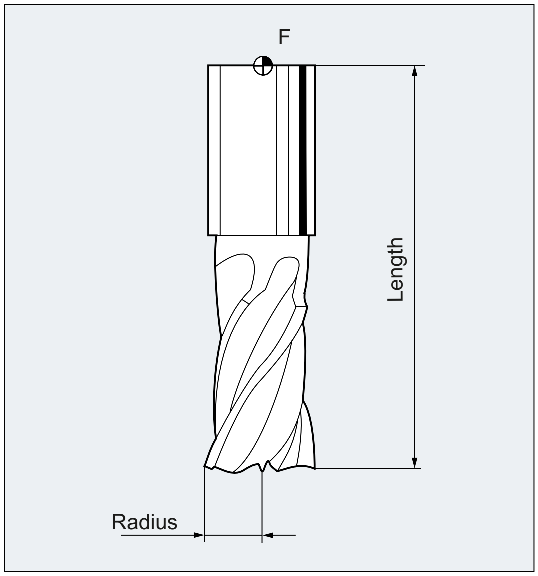
Cutting edge position

The cutting edge position describes the position of the tool tip P in relation to the cutting edge center point S.

The cutting edge position is required together with the cutting edge radius for the calculation of the tool radius compensation for turning tools (tool type 5xx).



Tool geometry variables (length, radius)



The tool geometry variables consist of several components (geometry, wear). The controller computes the components to a certain dimension (e.g. overall length 1, total radius). The respective overall dimension becomes effective when the compensation memory is activated.

How these values are calculated in the axes is determined by the tool type and the current plane (`G17`/`G18`/`G19`).

References

Function Manual, Basic Functions; Tool Offsets (W1); Section "Tool edge"

6.5 Tool types

6.5.1 General information about the tool types

Each tool type is assigned a unique 3-digit number. The assignment of the tool to one of the following technologies or tool groups is done using the first digit (the 100s column):

Tool type	Tool group
1xy	Milling tools (Page 72)
2xy	Drills (Page 74)
3xy	Reserved
4xy	Grinding tools (Page 75)
5xy	Turning tools (Page 76)
6xy	Reserved
7xy	Special tools (Page 78)

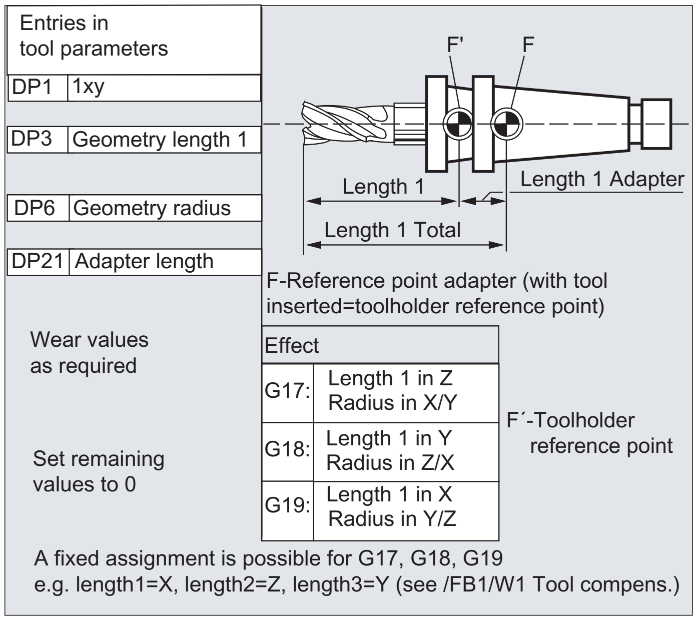
6.5.2 Milling tools

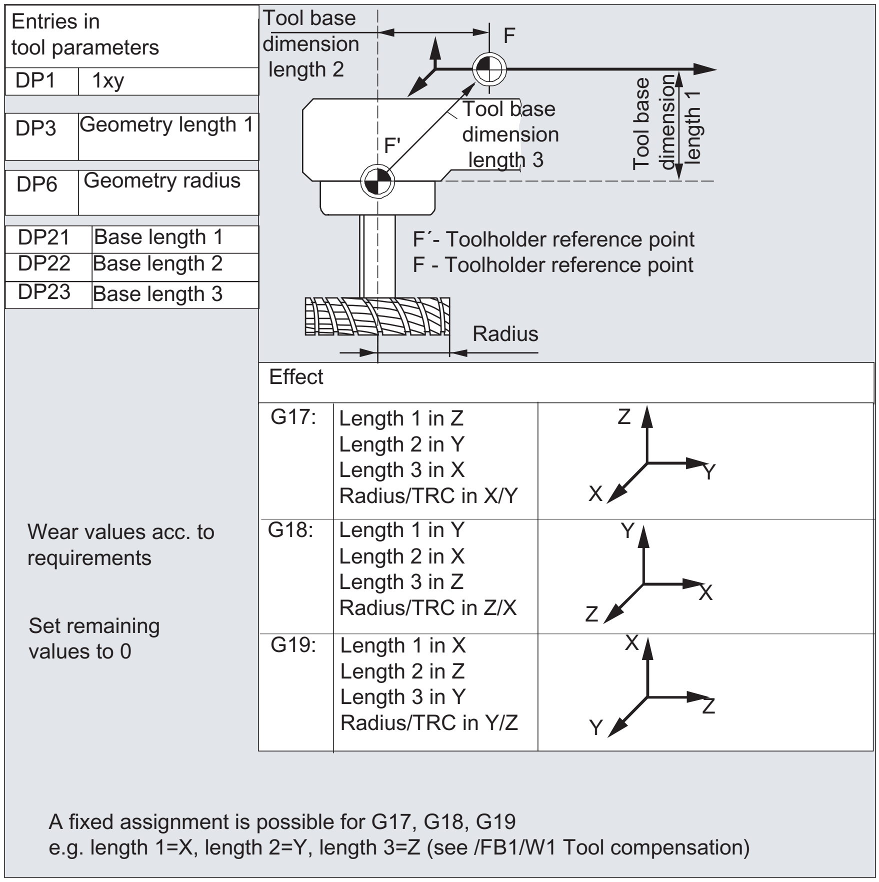
The following tool types are available in the "Milling tools" group:

100	Milling tool according to CLDATA (Cutter Location Data)
110	Ballhead cutter (cylindrical die milling tool)
111	Ballhead cutter (tapered die milling tool)
120	End mill (without corner rounding)
121	End mill (with corner rounding)
130	Angle head cutter (without corner rounding)
131	Angle head cutter (with corner rounding)
140	Facing tool
145	Thread cutter
150	Side mill
151	Saw
155	Bevel cutter (without corner rounding)
156	Bevel cutter (with corner rounding)
157	Tapered die milling tool
160	Drill and thread milling cutter

Tool parameters

The following figures provide an overview of which tool parameters ($TC_DP<x>) for milling tools are entered in the compensation memory:





References

Tool offset

Function Manual Basic Functions; Tool Offset (W1)

6.5.3 Drills

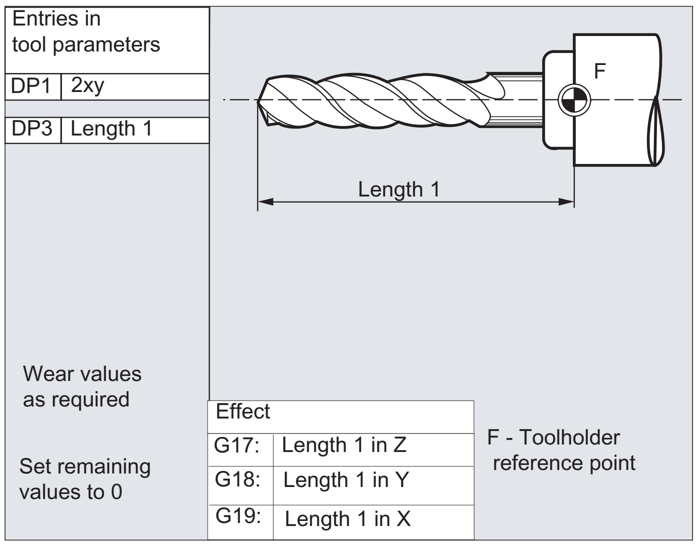
The following tool types are available in the "Drills" group:

200	Twist drill
205	Drill
210	Boring bar
220	Center drill
230	Countersink
231	Counterbore
240	Tap regular thread
241	Tap fine thread

242	Tap Whitworth thread
250	Reamer

Tool parameters

The following figure provides an overview of which tool parameters ($TC_DP<x>) for drills are entered in the compensation memory:



References

Tool offset

Function Manual Basic Functions; Tool Offset (W1)

6.5.4 Grinding tools

The following tool types are available in the "Grinding tools" group:

400	Surface grinding wheel
401	Surface grinding wheel with monitoring
402	Surface grinding wheel without monitoring without base dimension (TOOLMAN)
403	Surface grinding wheel with monitoring without base dimension for grinding wheel peripheral speed GWPS
410	Facing wheel
411	Facing wheel (TOOLMAN) with monitoring
412	Facing wheel (TOOLMAN) without monitoring
413	Facing wheel with monitoring without base dimension for grinding wheel peripheral speed GWPS
490	Dresser

Tool parameters

The following figure provides an overview of which tool parameters ($TC_DP<x>) for grinding tools are entered in the compensation memory:

Entries in the tool parameters	
DP1	403
DP2	Position *
DP3	Length 1
DP4	Length 2
DP6	Radius

* Tool nose position

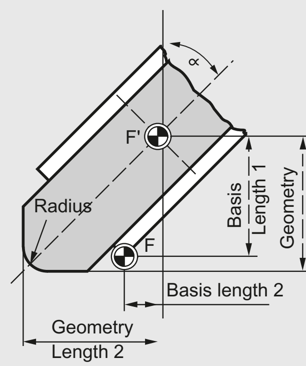
Wear values correspond to the requirement

Other values should be set to 0

Effect	
G17:	Length 1 in Y Length 2 in X Radius in X/Y
G18:	Length 1 in X Length 2 in Z Radius in Z/X
G19:	Length 1 in Z Length 2 in Y Radius in Y/Z

TPG1	Spindle number
TPG2	Chaining rule
TPG3	Minimum wheel radius
TPG4	Min. wheel width
TPG5	Actual wheel width
TPG6	Maximum speed
TPG7	Max. peripheral speed
TPG8	Angle of the inclined wheel
TPG9	Parameter No. for radius calculation

F: Tool carrier reference point



References

Tool offset

Function Manual Basic Functions; Tool Offset (W1)

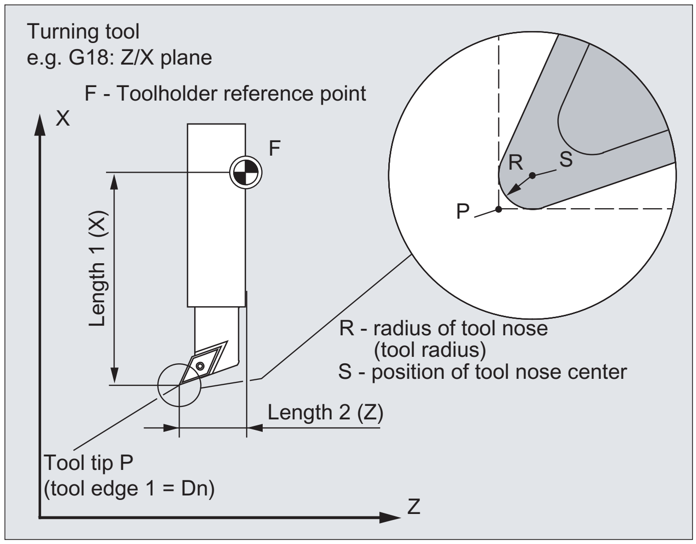
6.5.5 Turning tools

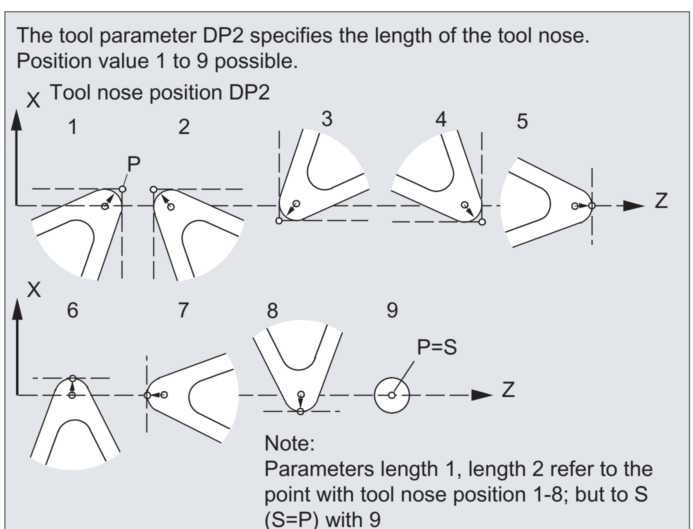
The following tool types are available in the "Turning tools" group:

500	Roughing tool
510	Finishing tool
520	Plunge cutter
530	Parting tool
540	Threading tool
550	Button tool / forming tool (TOOLMAN)
560	Rotary drill (ECOCUT)
580	Probe with cutting edge position parameters

Tool parameters

The following figures provide an overview of which tool parameters ($TC_DP<x>) for turning tools are entered in the compensation memory:





Entries in tool parameters	
DP1	5xy
DP2	1...9
DP3	Length 1
DP4	Length 2
DP6	Radius

Wear values as required

Set remaining values to 0

Effect	
G17:	Length 1 in Y Length 2 in X
G18:	Length 1 in X Length 2 in Z
G19:	Length 1 in Z Length 2 in Y

References

Tool offset

Function Manual Basic Functions; Tool Offset (W1)

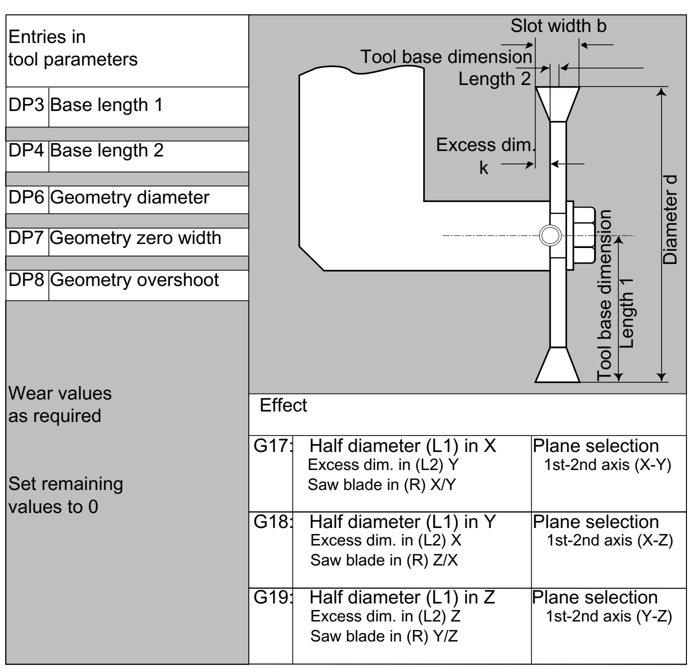
6.5.6 Special tools

The following tool types are available in the "Special tools" group:

700	Slotting saw
710	3D probe
711	Edge probe
712	Mono probe
713	L probe
714	Star probe
725	Calibration tool
730	Stop
731	Spindle sleeves
732	End support

Tool parameters: Slotting saw

The following figure provides an overview of which tool parameters ($TC_DP<x>) for "Slotting saw" tool type are entered in the compensation memory:



References

Tool offset

Function Manual Basic Functions; Tool Offset (W1)

Measuring input

Programming Manual Measuring Cycles

6.5.7 Chaining rule

The geometry tool length compensations, wear and base dimension can be chained for both the left and the right tool nose radius compensation, i.e. if the tool length compensations are changed for the left cutting edge, then the values are also automatically entered for the right cutting edge and vice versa.

References

Function Manual, Extended Functions; Grinding (W4)

6.6 Tool offset call (D)

Cutting edges 1 to 8 (with active tool management 12) of a tool can be assigned different tool offset data blocks (e.g. different offset values for the left and right cutting edge of a grooving tool).

Activation of the offset data (including the data for the tool length offset) of a special cutting edge is performed by calling the D number. When `D0` is programmed, offsets for the tool have no effect.

A tool radius compensation must also be activated via `G41`/`G42`.

Note

Tool length offsets take immediate effect when the D number is programmed. If no D number is programmed, the default setting defined via the machine data is active for a tool change (→ see machine manufacturer's specifications).

Syntax

Activation of a tool offset data block:
```
D<number>
```

Activate the tool radius compensation:
```
G41 ...
G42 ...
```

Deactivation of the tool offsets:
```
D0
G40
```

Meaning

`D`:	Command for the activation of an offset data block for the active tool The tool length offset is applied with the first programmed traverse of the associated length compensation axis. **Notice:** A tool length offset can also take effect without D programming, when the automatic activation of a tool edge has been configured for the tool change (→ see machine manufacturer's specifications).	
`<number>`:	The tool offset data block to be activated is specified via the **<number>** parameter. The type of D programming depends on the configuration of the machine (see paragraph "Type of D programming").	
	Range of values:	0 - 32000
`D0`:	Command for the deactivation of the offset data block for the active tool	
`G41`:	Command for the activation of the tool radius compensation with machining direction **left** of the contour	
`G42`:	Command for the activation of the tool radius compensation with machining direction **right** of the contour	
`G40`:	Command for the deactivation of the tool radius compensation	

Note

The tool radius compensation is described in detail in the "Tool radius compensation" section.

Type of D programming

The type of D programming is defined via machine data.

This can be done as follows:

- D number = cutting edge number
  D numbers ranging from 1 to max. 12 are available for every tool T<number> or T="Name" (with TOOLMAN). These D numbers are assigned directly to the tool cutting edges. An offset data block ($TC_DPx[t,d]) belongs to each D number (= cutting edge number).
- Free selection of D numbers
  The D numbers can be freely assigned to the cutting edge numbers of a tool. The upper limit for the D numbers that can be used is limited by a machine data.
- Absolute D number without reference to the T number
  Independence between D number and T number can be selected in systems without tool management. The reference of T number, cutting edge and offset by the D number is defined by the user. The range of D numbers is between 1 and 32000.

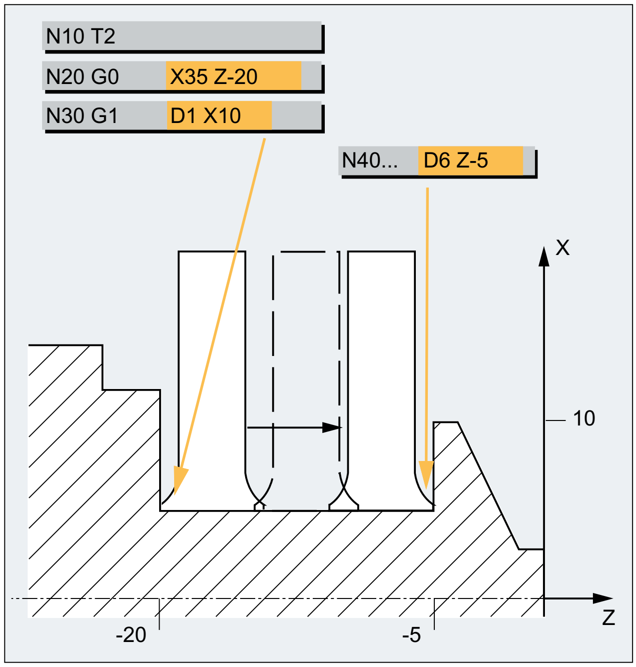
References:
Function Manual, Basic Functions; Tool Offset (W1),
Function Manual, Tool Management, Section: "Variants of D-number assignments"

Examples

Example 1: Tool change with T command (turning)

Program code	Comment
N10 T1 D1	; Load tool T1 and activate tool offset data block D1 of T1.
N11 G0 X... Z...	; The tool length offsets are applied.
N50 T4 D2	; Load tool T4 and activate tool offset data block D2 of T4.
...	
N70 G0 Z... D1	; Activate other cutting edge D1 for tool T4.

Example 2: Different offset values for the left and right cutting edge of a grooving tool



6.7 Change in the tool offset data

Effectiveness

A change in the tool offset data takes effect the next time the T or D number is programmed.

Set tool offset data to be active immediately

The following machine data can be used to specify that entered tool offset data takes effect immediately:

MD9440 $MM_ACTIVATE_SEL_USER

⚠ WARNING
Risk of collision If MD9440 is set, tool offsets resulting from changes in tool offset data **during the part program stop**, are applied when the part program is continued.

6.8 Programmable tool offset (TOFFL, TOFF, TOFFR)

The user can use the commands `TOFFL`/`TOFF` and `TOFFR` to modify the effective tool length or the effective tool radius in the NC program, without changing the tool offset data stored in the compensation memory.

These programmed offsets are deleted again at the end of the program.

Tool length offset

Depending on the type of programming, programmed tool length offsets are assigned either to the tool length components L1, L2 and L3 (`TOFFL`) stored in the compensation memory or to the geometry axes (`TOFF`). The programmed offsets are treated accordingly for a plane change (`G17`/`G18`/`G19` ↔ `G17`/`G18`/`G19`):

- If the offset values are assigned to the tool length components, the directions in which the programmed offsets apply are replaced accordingly.
- If the offset values are assigned to the geometry axes, a plane change does not effect the assignment in relation to the coordinate axes.

Tool radius offset

The command `TOFFR` is available for the programming of a tool radius offset.

Syntax

Tool length offset:

```
TOFFL=<value>
TOFFL[1]=<value>
TOFFL[2]=<value>
TOFFL[3]=<value>
TOFF[<geometry axis>]=<value>
```

Tool radius offset:

```
TOFFR=<value>
```

Meaning

`TOFFL`:	Command for the compensation of the effective tool length `TOFFL` can be programmed with or without index: • Without index: `TOFFL=` The programmed offset value is applied in the same direction as the tool length component **L1** stored in the compensation memory. • With index: `TOFFL[1]=`, `TOFFL[2]=` or `TOFFL[3]=` The programmed offset value is applied in the same direction as the tool length component **L1, L2** or **L3** stored in the offset memory. The commands `TOFFL` and `TOFFL[1]` have an identical effect. **Note:** How these tool length offset values are calculated in the axes is determined by the tool type and the current working plane (`G17`/`G18`/`G19`).
`TOFF`:	Command for the compensation of the tool length in the component parallel to the specified geometry axis `TOFF` is applied in the direction of the tool length component which is effective with non-rotated tool (orientable toolholder or orientation transformation) parallel to the `<geometry axis>` specified in the index. **Note:** A frame does not influence the assignment of the programmed values to the tool length components, i.e. the workpiece coordinate system (WCS) is not used for the assignment of the tool length components to the geometry axes, but rather the tool in the basic tool position.
`<geometry axis>`:	Identifier of the geometry axis
`TOFFR`:	Command for the compensation of the effective tool radius `TOFFR` changes the effective tool radius **with active tool radius compensation** by the programmed offset value.
`<value>`:	Offset value for the tool length or radius Type: REAL

Note

The `TOFFR` command has almost the same effect as the `OFFN` command (see "Tool radius compensation (Page 263)"). There is only a difference with active peripheral curve transformation (TRACYL) and active slot side compensation. In this case, the tool radius is affected by `OFFN` with a negative sign, but by `TOFFR` with a positive sign.

`OFFN` and `TOFFR` can be effective simultaneously. They then generally have an additive effect (except for slot side compensation).

Further syntax rules

- The tool length can be changed simultaneously in all three components. However, commands of the `TOFFL`/`TOFFL[1..3]` group and commands of the `TOFF[<geometry axis>]` may not be used simultaneously in one block.
  `TOFFL` and `TOFFL[1]` may also not be written simultaneously in one block.
- If all three tool length components are not programmed in a block, the components not programmed remain unchanged. In this way, it is possible to build up offsets for several components block-by-block. However, this only applies as long as the tool components have been modified either only with `TOFFL` or only with `TOFF`. Changing the programming type from `TOFFL` to `TOFF` or vice versa deletes any previously programmed tool length offsets (see example 3).

Supplementary conditions

- **Evaluation of setting data**
  The following setting data is evaluated when assigning the programmed offset values to the tool length components:
  SD42940 $SC_TOOL_LENGTH_CONST (change of tool length components on change of planes).
  SD42950 $SC_TOOL_LENGTH_TYPE (assignment of the tool length offset independent of tool type).
  If this setting data has valid values not equal to 0, then these take preference over the contents of G group 6 (plane selection `G17` - `G19`) or the tool type ($TC_DP1[<T no.>, <D no.>]) contained in the tool data, i.e. this setting data influences the evaluation of the offsets in the same way as the tool length components L1 to L3.
- **Tool change**
  All offset values are retained during a tool change (cutting edge change), e.g. they are also effective for the new tool (new cutting edge).

Examples

Example 1: Positive tool length offset

The active tool is a drill with length L1 = 100 mm.

The active plane is `G17`, i.e. the drill points in the Z direction.

The effective drill length is to be increased by 1 mm. The following variants are available for the programming of this tool length offset:
`TOFFL=1`

or
`TOFFL[1]=1`

or
`TOFF[Z]=1`

Example 2: Negative tool length offset

The active tool is a drill with length L1 = 100 mm.

The active plane is `G18`, i.e. the drill points in the Y direction.

The effective drill length is to be decreased by 1 mm. The following variants are available for the programming of this tool length offset:
`TOFFL=-1`

or
`TOFFL[1]=-1`

or
`TOFF[Y]=1`

Example 3: Changing the programming type from TOFFL to TOFF

The active tool is a milling tool. The active plane is `G17`.

Program code	Comment
N10 TOFFL[1]=3 TOFFL[3]=5	; Effective offsets: L1=3, L2=0, L3=5
N20 TOFFL[2]=4	; Effective offsets: L1=3, L2=4, L3=5
N30 TOFF[Z]=1.3	; Effective offsets: L1=0, L2=0, L3=1.3

Example 4: Plane change

Program code	Comment
N10 $TC_DP1[1,1]=120	
N20 $TC_DP3[1,1]= 100	; Tool length L1=100 mm.
N30 T1 D1 G17	
N40 TOFF[Z]=1.0	; Offset in Z direction (corresponds to L1 for G17).
N50 G0 X0 Y0 Z0	; Machine axis position X0 Y0 Z101.
N60 G18 G0 X0 Y0 Z0.	; Machine axis position X0 Y100 Z1.
N70 G17	
N80 TOFFL=1.0	; Offset in L1 direction (corresponds to Z for G17).
N90 G0 X0 Y0 Z0	; Machine axis position X0 Y0 Z101.
N100 G18 G0 X0 Y0 Z0	; Machine axis position X0 Y101 Z0.

In this example, the offset of 1 mm in the Z axis is retained when changing to `G18` in block `N60`; the effective tool length in the Y axis is the unchanged tool length of 100 mm.

However, in block `N100`, the offset is effective in the Y axis when changing to `G18` as it was assigned to tool length L1 in the programming and this length component is effective in the Y axis with `G18`.

Further information

Applications

The "Programmable tool offset" function is especially interesting for ball mills and milling tools with corner radii as these are often calculated in the CAM system to the ball center instead of the ball tip. However, generally the tool tip is measured when measuring the tool and stored as tool length in the compensation memory.

System variables for reading the current offset values

The currently effective offsets can be read with the following system variables:

System variable		Meaning
$P_TOFFL [<n>]	with 0 ≤ n ≤ 3	Reads the current offset value of `TOFFL` (for n = 0) or `TOFFL[1...3]` (for n = 1, 2, 3) in the preprocessing context.
$P_TOFF [<geometry axis>]		Reads the current offset value of `TOFF[<geometry axis>]` in the preprocessing context.
$P_TOFFR		Reads the current offset value of `TOFFR` in the preprocessing context.
$AC_TOFFL[<n>]	with 0 ≤ n ≤ 3	Reads the current offset value of `TOFFL` (for n = 0) or `TOFFL[1...3]` (for n = 1, 2, 3) in the main run context (synchronized actions).
$AC_TOFF[<geometry axis>]		Reads the current offset value of `TOFF[<geometry axis>]` in the main run context (synchronized actions).
$AC_TOFFR		Reads the current offset value of `TOFFR` in the main run context (synchronized actions).

Note

The system variables $AC_TOFFL, $AC_TOFF and AC_TOFFR trigger an automatic preprocessing stop when reading from the preprocessing context (NC program).

7 Spindle motion

7.1 Spindle speed (S), spindle direction of rotation (M3, M4, M5)

The spindle speed and direction of rotation values set the spindle in rotary motion and provide the conditions for chip removal.

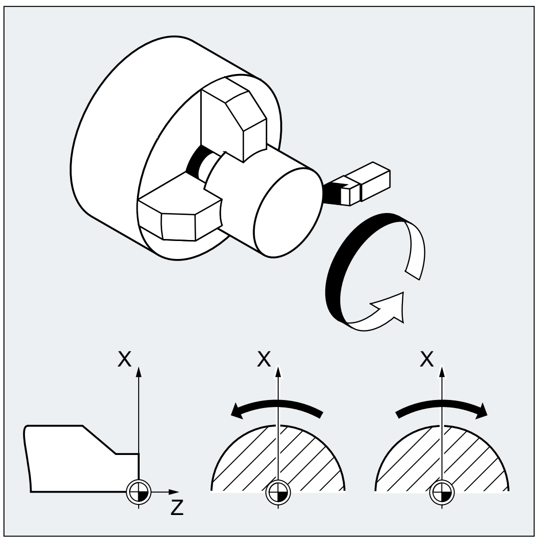


Figure 7-1 Spindle motion during turning

Other spindles may be available in addition to the main spindle (e.g. the counterspindle or an actuated tool on turning machines). As a rule, the main spindle is declared the master spindle in the machine data. This assignment can be changed using an NC command.

Syntax

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

`M3` / `M<n>=3`

`M4` / `M<n>=4`

`M5` / `M<n>=5`

`SETMS(<n>)`	
`...`	
`SETMS`	

Meaning

`S…:`	Spindle speed in rpm for the master spindle
`S<n>=...:`	Spindle speed in rpm for spindle <n>

	Note: The speed specified with `S0=…` applies to the master spindle
`M3`:	Direction of spindle rotation clockwise for master spindle
`M<n>=3`:	Spindle direction of rotation clockwise for spindle <n>
`M4`:	Direction of spindle rotation counter-clockwise for master spindle
`M<n>=4`:	Spindle direction of rotation counter-clockwise for spindle <n>
`M5`:	Spindle stop for master spindle
`M<n>=5`:	Spindle stop for spindle <n>
`SETMS(<n>)`:	Set spindle <n> as master spindle
`SETMS`:	If `SETMS` is programmed without a spindle name, the configured master spindle is used instead.

Note

Up to three S-values can be programmed per NC block, e.g.:

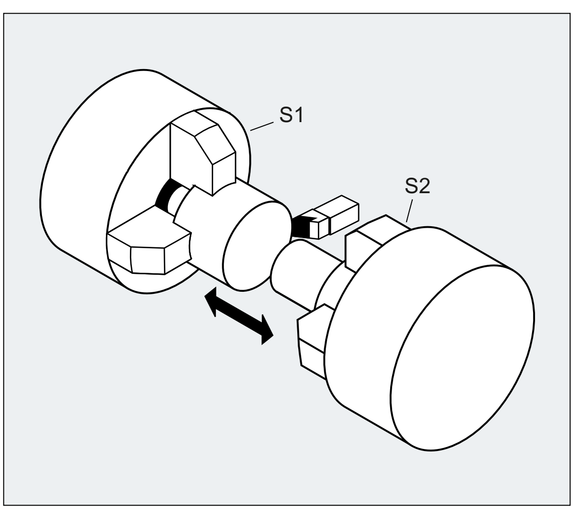
`S... S2=... S3=...`

Note

`SETMS` must be in a separate block.

Example

S1 is the master spindle, S2 is the second spindle. The part is to be machined from two sides. To do this, it is necessary to divide the operations into steps. After the cut-off point, the synchronizing device (S2) takes over machining of the workpiece after the cut off. To do this, this spindle S2 is defined as the master spindle to which G95 then applies.



Program code	Comment
`N10 S300 M3`	`; Speed and direction of rotation for drive spindle = pre-set master spindle.`
`...`	`; Machining of the right-hand workpiece side.`

Program code	Comment
N100 SETMS(2)	; S2 is now the master spindle.
N110 S400 G95 F…	; Speed for new master spindle.
...	; Machining of the left-hand workpiece side.
N160 SETMS	; Switching back to master spindle S1.

Further information

Interpretation of the S value for the master spindle

If function G331 or G332 is active in G group 1 (modally valid motion commands), the programmed S-value will always be interpreted as the speed in rpm. Otherwise, the interpretation of the S-value will depend upon G group 15 (feedrate type): If G96, G961 or G962 is active, the S-value is interpreted as a constant cutting rate in m/min; otherwise, it is interpreted as a speed in rpm.

Changing from G96/G961/G962 to G331/G332 sets the value of the constant cutting rate to zero; changing from G331/G332 to a function within the G group 1 other than G331/G332 sets the speed value to zero. The corresponding S-values have to be reprogrammed if required.

Preset M commands M3, M4, M5

In a block with axis commands, functions M3, M4, M5 are activated **before** the axis movements commence (basic setting on the control).

Example:

Program code	Comment
N10 G1 F500 X70 Y20 S270 M3	; The spindle ramps up to 270 rpm and the movements then executed in X and Y.
N100 G0 Z150 M5	; Spindle stop before the retraction movement in Z.

Note

Machine data can be used to set when axis movements should be executed; either once the spindle has powered up to the setpoint speed, or immediately after the programmed switching operations have been traversed.

Working with multiple spindles

Five spindles (master spindle plus four additional spindles) can be available in one channel at the same time.

One of the spindles is defined in machine data as the **master spindle**. Special functions such as thread cutting, tapping, revolutional feedrate, and dwell time apply to this spindle. For the remaining spindles (e.g. a second spindle and an actuated tool) the numbers corresponding to the speed and the direction of rotation / spindle stop must be specified.

Example:

Program code	Comment
N10 S300 M3 S2=780 M2=4	; Master spindle: 300 rpm, CW rotation 2nd spindle: 780 rpm, CCW rotation

Programmable switchover of master spindle

The `SETMS(<n>)` command can be used in the NC program to define any spindle as the master spindle. `SETMS` must be in a separate block.

Example:

Program code	Comment
N10 SETMS (2)	; Spindle 2 is now the master spindle.

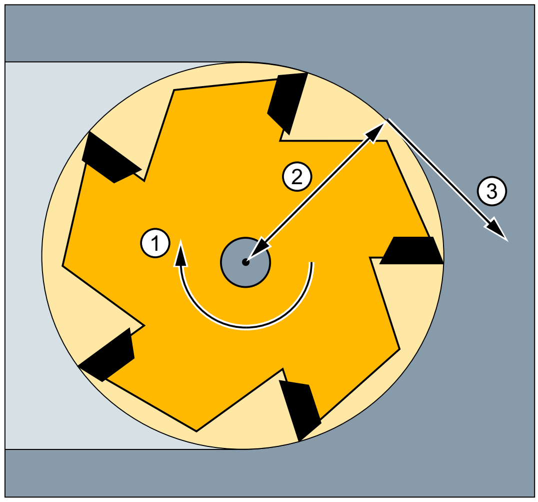
Note

The speed specified with `S...`, along with the functions programmed with `M3`, `M4`, `M5`, now apply to the newly declared master spindle.

If `SETMS` is programmed without a spindle name, the master spindle programmed in the machine data is used instead.

7.2 Tool cutting speed (SVC)

As an alternative to the spindle speed, the tool cutting speed, which is more commonly used in practice, can be programmed for milling operations.



① Spindle speed
② Tool radius
③ Tool cutting speed

The control uses the radius of the active tool to calculate the effective spindle speed from the programmed tool cutting speed:

S = (SVC * 1000) / (R_T * 2π)

with:	S:	Spindle speed in rpm
	SVC:	Tool cutting speed in m/min or ft/min
	R_T:	Radius of the active tool in mm

The tool type ($TC_DP1) of the active tool is not taken into account.

The programmed tool cutting speed is independent of path feedrate F and G function group 15 (feedrate type). The direction of rotation and the spindle start are implemented via M3 and M4, and the spindle stop via M5.

A change to the tool radius data in the offset memory will be applied the next time a tool offset is selected or the next time the active offset data is updated.

Changing the tool or selecting/deselecting a tool offset data set generates a recalculation of the effective spindle speed.

Requirements

Programming of the tool cutting speed requires:

- The geometric ratios of a rotating tool (milling cutter or drilling tool)
- An active tool offset data set

Syntax

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

Meaning

`SVC`:	Keyword for programming of the tool cutting speed
`[<n>]`:	Number of spindle This address extension specifies which spindle the programmed cutting speed is to be applied for. In the absence of an address extension, the rate is always applied to the master spindle. **Note:** A separate cutting speed can be preset for each spindle. **Note:** Programming SVC without an address extension requires that the master spindle has the active tool. If the master spindle changes, the user will need to select a tool accordingly.
`<Value>`:	Value of tool cutting speed Unit: m/min (for G71/G710) or ft/min (for G70/G700)
`T... D...`:	The tool radius must be established in the block with SVC. Thus, a corresponding tool including tool offset data block must either be active or selected in the block. There is no fixed sequence for SVC and T/D selection during programming in the same block.
`S... M3/M4`:	Programming of the spindle speed will effect deselection of the tool cutting speed. **Note:** Switching between SVC programming and S programming is possible at any time, even while the spindle is rotating. In each case, the value that is not active is deleted.

Note

SVC programming is not possible if the following spindle feedrate movements are active:

- Constant cutting speed: G96/G961/G962 S... (Page 99)
- Constant grinding wheel peripheral speed: SUG (Page 105)
- Position spindle: SPOS/SPOSA/M19 (Page 121)
- ; Switch master spindle over to axis mode: M70 (Page 121)

Conversely, programming one of these functions will effect a deselection of SVC (tool cutting speed).

Note

Maximum tool speed

System variable $TC_TP_MAX_VELO[<tool number>] can be used to preset a maximum tool speed (spindle speed).
If no speed limit has been defined, there will be no monitoring.

Note

The tool paths of "standard tools" generated, e.g. using CAD systems which already take the tool radius into account and only contain the deviation from the standard tool in the tool nose radius, are not supported in conjunction with SVC programming.

Examples

The following shall apply to all examples: Tool carrier = spindle (for standard milling)

Example 1: Milling cutter 6 mm radius

Program code	Comment
N10 G0 X10 T1 D1	; Selection of milling tool with, e.g. $TC_DP6[1,1] = 6 (tool radius = 6 mm)
N20 SVC=100 M3	; Cutting speed = 100 m/min
	□ Resulting spindle speed:
	S = (100 m/min * 1000) / (6.0 mm * 2 * 3.14) = 2653.93 rpm
N30 G1 X50 G95 FZ=0.03	; SVC and tooth feedrate
...	

Example 2: Tool selection and SVC in the same block

Program code	Comment
N10 G0 X20	
N20 T1 D1 SVC=100	; Tool and offset data set selection together with SVC in block (no specific sequence).
N30 X30 M3	; Spindle start with CW direction of rotation, cutting speed 100 m/min
N40 G1 X20 F0.3 G95	; SVC and revolutional feedrate

Example 3: Defining cutting speeds for two spindles

Program code	Comment
N10 SVC[3]=100 M6 T1 D1	
N20 SVC[5]=200	; The tool radius of the active tool offset is the same for both spindles. The effective speed is different for spindle 3 and spindle 5.

Example 4:

Assumptions:

Master or tool change is determined by the tool carrier:

MD20124 $MC_TOOL_MANAGEMENT_TOOL CARRIER > 1

In the event of a tool change the old tool offset is retained. A tool offset for the new tool is only activated when `D` is programmed:

MD20270 $MC_CUTTING_EDGE_DEFAULT = - 2

Program code	Comment
N10 $TC_MPP1[9998,1]=2	; Magazine location is tool carrier
N11 $TC_MPP5[9998,1]=1	; Magazine location is tool carrier 1
N12 $TC_MPP_SP[9998,1]=3	; Tool carrier 1 is assigned to spindle 3
N20 $TC_MPP1[9998,2]=2	; Magazine location is tool carrier
N21 $TC_MPP5[9998,2]=4	; Magazine location is tool carrier 4
N22 $TC_MPP_SP[9998,2]=6	; Tool carrier 4 is assigned to spindle 6
N30 $TC_TP2[2]="WZ2"	
N31 $TC_DP6[2,1]=5.0	; Radius = 5.0 mm of T2, offset D1
N40 $TC_TP2[8]="WZ8"	
N41 $TC_DP6[8,1]=9.0	; Radius = 9.0 mm of T8, offset D1
N42 $TC_DP6[8,4]=7.0	; Radius = 7.0 mm of T8, offset D4
...	
N100 SETMTH(1)	; Set master tool carrier number
N110 T="WZ2" M6 D1	; Tool T2 is loaded and offset D1 is activated.
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)	; Set master tool carrier number
N140 T="WZ8"	; Corresponds to T8="WZ8"
N150 M6	; Corresponds to M4=6
	Tool "WZ8" is in the master tool carrier, but because MD20270=-2, the old tool offset remains active.
N160 SVC=50	; S3 = (50 m/min * 1000) / (5.0 mm * 2 * 3.14) = 1592.36 rpm
	The offset applied to tool carrier 1 is still active and assigned to spindle 3.
N170 D4	; Offset D4 of the new tool "WZ8" becomes active (in tool carrier 4).
N180 SVC=300	; S6 = (300 m/min * 1000) / (7.0 mm * 2 * 3.14) = 6824.39 rpm
	Spindle 6 is assigned to tool carrier 4.

Example 5:

Assumptions:

Spindles are tool carriers at the same time:

MD20124 $MC_TOOL_MANAGEMENT_TOOL CARRIER = 0

In the event of a tool change, tool offset data set D4 is selected automatically:

MD20270 $MC_CUTTING_EDGE_DEFAULT = 4

Program code	Comment
N10 $TC_MPP1[9998,1]=2	; Magazine location is tool carrier
N11 $TC_MPP5[9998,1]=1	; Magazine location is tool carrier 1 = spindle 1
N20 $TC_MPP1[9998,2]=2	; Magazine location is tool carrier
N21 $TC_MPP5[9998,2]=3	; Magazine location is tool carrier 3 = spindle 3
N30 $TC_TP2[2]="WZ2"	
N31 $TC_DP6[2,1]=5.0	; Radius = 5.0 mm of T2, offset D1

Program code	Comment
`N40 $TC_TP2[8]="WZ8"`	
`N41 $TC_DP6[8,1]=9.0`	`; Radius = 9.0 mm of T8, offset D1`
`N42 $TC_DP6[8,4]=7.0`	`; Radius = 7.0 mm of T8, offset D4`
`...`	
`N100 SETMS(1)`	`; Spindle 1 = master spindle`
`N110 T="WZ2" M6 D1`	`; Tool T2 is loaded and offset D1 is activated.`
`N120 G1 G94 F1000 M3 SVC=100`	`; S1 = (100 m/min * 1000) / (5.0 mm * 2 * 3.14) = 3184.71 rpm`
`N200 SETMS(3)`	`; Spindle 3 = master spindle`
`N210 M4 SVC=150`	`; S3 = (150 m/min * 1000) / (5.0 mm * 2 * 3.14) = 4777.07 rpm`
	`Refers to tool offset D1 of T="WZ2", S1 continues to turn at previous speed.`
`N220 T="WZ8"`	`; Corresponds to T8="WZ8"`
`N230 M4 SVC=200`	`; S3 = (200 m/min * 1000) / (5.0 mm * 2 * 3.14) = 6369.43 rpm`
	`Refers to tool offset D1 of T="WZ2".`
`N240 M6`	`; Corresponds to M3=6`
	`Tool "WZ8" is in the master spindle, tool offset D4 of the new tool becomes active.`
`N250 SVC=50`	`; S3 = (50 m/min * 1000) / (7.0 mm * 2 * 3.14) = 1137.40 rpm`
	`Offset D4 on master spindle is active.`
`N260 D1`	`; Offset D1 of new tool "WZ8" active.`
`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`
`...`	

Further information

Tool radius

The following tool offset data (associated with the active tool) affect the tool radius when:

- $TC_DP6 (radius - geometry)
- $TC_DP15 (radius - wear)
- $TC_SCPx6 (offset for $TC_DP6)
- $TC_ECPx6 (offset for $TC_DP6)

The following are not taken into account:

- Online radius compensation
- Allowance on the programmed contour (OFFN)

Tool radius compensation (G41/G42)

Although tool radius compensation (G41/G42) and SVC both relate to the tool radius, they are not functionally linked and are independent of one another.

Tapping without compensating chuck (G331, G332)

SVC programming is also possible in conjunction with G331 or G332.

Synchronized actions

SVC cannot be programmed from synchronized actions.

Reading the cutting speed and the spindle speed programming variant

The cutting speed of a spindle and the speed programming variant (spindle speed S or tool cutting speed SVC) can be read using system variables:

- With preprocessing stop in the part program via system variables:

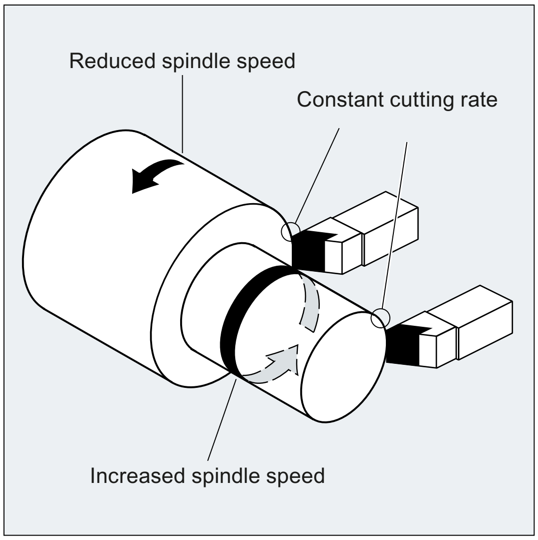
$AC_SVC[<n>]	Effective cutting speed when the current main run block for spindle with number <n> was preprocessed.
$AC_S_TYPE[<n>]	Effective spindle speed programming variant when the current main run block for spindle with number <n> was preprocessed. **Value:** **Meaning:** 1 Spindle speed S in rpm 2 Tool cutting speed SVC in m/min or ft/min

- Without preprocessing stop in the part program via system variables:

$P_SVC[<n>]	Programmed cutting speed for spindle <n>
$P_S_TYPE[<n>]	Programmed spindle speed programming variant for spindle <n> **Value:** **Meaning:** 1 Spindle speed S in rpm 2 Tool cutting speed SVC in m/min or ft/min

7.3 Constant cutting rate (G96/G961/G962, G97/G971/G972, G973, LIMS, SCC)

If the "Constant cutting speed" function is active, the spindle speed is modified as a function of the respective workpiece diameter so that the cutting speed S in m/min or ft/min remains constant at the tool edge.



This results in the following advantages:

- Uniformity and consequently improved surface quality of turned parts
- Machining with less wear on tools

Syntax

Activating/deactivating constant cutting speed for the master spindle:

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

Speed limitation for the master spindle:
```
LIMS=<value>
LIMS[<spindle>]=<value>
```

Other reference axis for G96/G961/G962:
```
SCC[<axis>]
```

Note

`SCC[<axis>]` can be programmed together with `G96`/`G961`/`G962` or in isolation.

Meaning

`G96`:	Revolutional feedrate (as for G95 (Page 107)) and constant cutting speed G95 is activated automatically with G96. If G95 has not been activated previously, a new feedrate value F... has to be specified when G96 is called.	
`G961`:	Linear feedrate (as for G94 (Page 107)) and constant cutting speed	
`G962`:	Linear feedrate or revolutional feedrate and constant cutting speed	
`S...`:	In conjunction with G96, G961 or G962, S... is not interpreted as a spindle speed but rather as a cutting speed. The cutting speed is always applied to the master spindle.	
	Unit:	m/min (for G71/G710) or ft/min (for G70/G700)
	Range of values:	0.1 m/min to 9999 9999.9 m/min
`G97`:	Revolutional feedrate and constant spindle speed (constant cutting speed OFF)	
`G971`:	Linear feedrate and constant spindle speed (constant cutting speed OFF)	
`G972`:	Linear feedrate or revolutional feedrate and constant spindle speed (constant cutting speed OFF)	
`G973`:	Revolutional feedrate without spindle speed limitation and constant spindle speed (G97 without LIMS for ISO mode)	
	Note: After G97 (or G971 ... G973), S... is again interpreted as a spindle speed in rpm. In the absence of a new spindle speed being specified, the last speed set with G96 (respectively G961 or G962) is retained.	
`LIMS`:	Speed limitation for the master spindle (only applied if G96/G961/G97 active) On machines with selectable master spindles, limitations of differing values can be programmed for up to four spindles within one block.	
	`<spindle>`:	Number of spindle
	`<value>`:	Spindle speed upper limit in rpm
`SCC`:	If any of the G96/G961/G962 functions are active, SCC[<axis>] can be used to assign any geometry axis as a reference axis.	

Note

If G96/G961/G962 is selected for the first time, a constant cutting speed S... must be entered; if G96/G961/G962 is selected again, the entry is optional.

Note

The speed limitation programmed with LIMS must not exceed the speed limit programmed with G26 or defined in the setting data.

Note

The reference axis for G96/G961/G962 must be a geometry axis assigned to the channel at the time when SCC[<axis>] is programmed. SCC[<axis>] can also be programmed when any of the G96/G961/G962 functions are active.

Examples

Example 1: Activating the constant cutting speed with speed limitation

Program code	Comment
N10 SETMS (3)	
N20 G96 S100 LIMS=2500	; Constant cutting speed = 100 m/min, max. speed = 2500 rpm
...	
N60 G96 G90 X0 Z10 F8 S100 LIMS=444	; Max. speed = 444 rpm

Example 2: Defining speed limitation for four spindles

Speed limitations are defined for spindle 1 (master spindle) and spindles 2, 3, and 4:

Program code
N10 LIMS=300 LIMS[2]=450 LIMS[3]=800 LIMS[4]=1500
...

Example 3: Y-axis assignment for face cutting with X axis

Program code	Comment
N10 G18 LIMS=3000 T1 D1	; Speed limitation at 3000 rpm
N20 G0 X100 Z200	
N30 Z100	
N40 G96 S20 M3	; Constant cutting speed = 20 m/min, is dependent upon X axis.
N50 G0 X80	
N60 G1 F1.2 X34	; Face cutting in X at 1.2 mm/revolution.
N70 G0 G94 X100	
N80 Z80	
N100 T2 D1	
N110 G96 S40 SCC[Y]	; Y axis is assigned to G96 and G96 is activated (can be achieved in a single block). Constant cutting speed = 40 m/min, is dependent upon X axis.
...	
N140 Y30	
N150 G01 F1.2 Y=27	; Plunge-cutting in Y, feedrate F = 1.2 mm/revolution.
N160 G97	; Constant cutting speed off.
N170 G0 Y100	

Further information

Calculation of the spindle speed

The SZS position of the face axis (radius) is the basis for calculating the spindle speed from the programmed cutting rate.

Note

Frames between WCS and SZS (e.g. programmable frames such as SCALE, TRANS or ROT) are taken into account in the calculation of the spindle speed and can bring about a change in speed (for example, if there is a change in the effective diameter in the case of SCALE).

Speed limitation LIMS

If a workpiece that varies greatly in diameter needs to be machined, it is advisable to specify a speed limit for the spindle with LIMS (maximum spindle speed). This prevents excessively high speeds with small diameters. LIMS is only applied if G96, G961 and G97 are active. LIMS is not applied if G971 is selected. On loading the block into the main run, all programmed values are transferred into the setting data.

Note

The speed limits changed with LIMS in the part program are taken into the setting data and therefore remain saved after the end of program.

However, if the speed limits changed with LIMS are no longer to apply after the end of program, the following definition must be inserted in the GUD block of the machine manufacturer:

REDEF $SA_SPIND_MAX_VELO_LIMS PRLOC

Deactivating the constant cutting rate (G97/G971/G972/G973)

After G97 (or G971 ... G973), S... is again interpreted as a spindle speed in rpm. In the absence of a new spindle speed being specified, the last speed set with G96 (respectively G961 or G962) is retained.

The G96/G961 function can also be deactivated with G94 or G95. In this case, the last speed programmed S... is used for subsequent machining operations.

G97 can be programmed without G96 beforehand. The function then has the same effect as G95; LIMS can also be programmed.

Using G973, the constant cutting rate can be deactivated without activating a spindle speed limitation.

Note

The transverse axis must be defined in machine data.

Rapid traverse G0

With rapid traverse G0, there is no change in speed.

Exception:

If the contour is approached in rapid traverse and the next NC block contains a G1/G2/G3/... path command, the speed is adjusted in the G0 approach block for the next path command.

Other reference axis for G96/G961/G962

If any of the G96/G961/G962 functions are active, SCC[<axis>] can be used to assign any geometry axis as a reference axis. If the reference axis changes, which will in turn affect the TCP (tool center point) reference position for the constant cutting rate, the resulting spindle speed will be reached via the set braking or acceleration ramp.

Axis exchange of the assigned channel axis

The reference axis property for G96/G961/G962 is always assigned to a geometry axis. In the event of an axis exchange involving the assigned channel axis, the reference axis property for G96/G961/G962 is retained in the old channel.

A geometry axis exchange will not affect how the geometry axis is assigned to the constant cutting rate. If the TCP reference position for G96/G961/G962 is affected by a geometry axis exchange, the spindle will reach the new speed via a ramp.

If no new channel axis is assigned as a result of a geometry axis exchange (e.g. GEOAX(0,X)), the spindle speed will be frozen in accordance with G97.

Examples for geometry axis exchange with assignments of the reference axis:

Program code	Comment
N05 G95 F0.1	
N10 GEOAX(1, X1)	; Channel axis X1 becomes the first geometry axis.
N20 SCC[X]	; First geometry axis (X) becomes the reference axis
	; for G96/G961/G962.
N30 GEOAX(1, X2)	; Channel axis X2 becomes the first geometry axis.
N40 G96 M3 S20	; Reference axis for G96 is channel axis X2.

Program code	Comment
N05 G95 F0.1	
N10 GEOAX(1, X1)	; Channel axis X1 becomes the first geometry axis.
N20 SCC[X1]	; X1 and implicitly the first geometry axis (X) becomes
	the reference axis for G96/G961/G962.
N30 GEOAX(1, X2)	; Channel axis X2 becomes the first geometry axis.
N40 G96 M3 S20	; Reference axis for G96 is X2 or X, no alarm.

Program code	Comment
N05 G95 F0.1	
N10 GEOAX(1, X2)	; Channel axis X2 becomes the first geometry axis.
N20 SCC[X1]	; X1 is not a geometry axis, alarm.

Program code	Comment
N05 G0 Z50	
N10 X35 Y30	
N15 SCC[X]	; Reference axis for G96/G961/G962 is X.
N20 G96 M3 S20	; Constant cutting rate ON at 10 mm/min.
N25 G1 F1.5 X20	; Face cutting in X at 1.5 mm/revolution.

Program code	Comment
N30 G0 Z51	
N35 SCC[Y]	; Reference axis for G96 is Y, reduction of spindle speed (Y30).
N40 G1 F1.2 Y25	; Face cutting in Y at 1.2 mm/revolution.

References:
Function Manual, Basic Functions; Transverse Axes (P1) and Feedrates (V1)

7.4 Switching constant grinding wheel peripheral speed (GWPSON, GWPSOF) on/off:

With the predefined procedures GWPSON(...) and GWPSOF(...), the constant grinding wheel peripheral speed (GWPS) for grinding tools (tool type: 400 to 499) is switched on and off.

Syntax

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

Meaning

`GWPSON(...)`:	Switch on the constant grinding wheel peripheral speed
`GWPSOF(...)`:	Switch off the constant grinding wheel peripheral speed
`<TNo>`:	T number **Note:** Only required if the constant grinding wheel peripheral speed is to be switch-ed on or off for an inactive grinding wheel rather than the active grinding wheel that is currently in use.
`S<n>=...`:	Grinding wheel peripheral speed in m/s or ft/s for spindle <n>
`S0=...` or `S...`:	Grinding wheel peripheral speed for the master spindle

Query status

The following system variable can be used to query from the part program whether the constant grinding wheel peripheral speed is active for a specific spindle.

$P_GWPS[<n>] ; where <n> = spindle number

Value	Meaning
0 (= FALSE)	GWPS is **inactive**.
1 (= TRUE)	GWPS is **active**.

7.5 Programmable spindle speed limitation (G25, G26)

The minimum and maximum spindle speeds defined in the machine and setting data can be modified by means of a part program command.

Programmed spindle speed limitations are possible for all spindles of the channel.

Syntax

```
G25 S… S1=… S2=…
G26 S… S1=… S2=…
```

Meaning

`G25`:	**Lower** spindle speed limit
`G26`:	**Upper** spindle speed limit
`S... S1=… S2=…` :	Minimum or maximum spindle speed(s) **Note:** A maximum of three spindle speed limits can be programmed for each block. Range of values: 0.1 to 9999 9999.9 rpm

---

Note

A spindle speed limitation programmed with `G25` or `G26` overwrites the speed limits in the setting data and, therefore, remains stored even after the end of the program.

However, if the speed limits changed with `G25`/`G26` are no longer to apply after the end of program, the following definitions must be inserted in the GUD block of the machine manufacturer:

REDEF $SA_SPIND_MIN_VELO_G25 PRLOC

REDEF $SA_SPIND_MAX_VELO_G26 PRLOC

Example

Program code	Comment
`N10 G26 S1400 S2=350 S3=600`	`; Upper speed limit for master spindle, spindle 2 and spindle 3.`

Feed control 8

8.1 Feedrate (G93, G94, G95, F, FGROUP, FL, FGREF)

These commands are used in the NC program to set the feedrates for all axes involved in the machining sequence.

Syntax

```
G93
G94
G95
F<value>
FGROUP(<axis_1>,<axis_2>,...)
FGREF[<rotary axis>]=<reference radius>
FL[<axis>]=<value>
```

Meaning

G93:	Path feed type: Inverse-time feedrate [rpm]
G94:	Path feed type: Linear feedrate [mm/min], [inch/min] or [degrees/min]
G95:	Path feed type: Revolutional feedrate [mm/revolution] or [inch/revolution] The revolutional feedrate can be derived from a master spindle, any other spindle or a rotary axis.
F<value>	Path feedrate for all or the path axes selected with FGROUP.
FGROUP:	Definition of the path axes to which the F-programmed path feed refers
FGREF:	FGREF is used to program the effective radius (<reference radius>) for each of the rotary axes specified under FGROUP
FL:	Limit velocity for synchronized/path axes The unit set with G94 applies One FL value can be programmed per axis (channel axes, geometry axis or orientation axis)
<axis>:	Name of a channel axis, type: AXIS

Examples

Example 1: Mode of operation of FGROUP

The following example is intended to demonstrate the effect of FGROUP on the path and path feedrate. The variable $AC_TIME contains the time of the block start in seconds. It can only be used in synchronized actions.

Program code	Comment
N100 G0 X0 A0	
N110 FGROUP(X,A)	

Program code	Comment
N120 G91 G1 G710 F100	; Feedrate = 100mm/min or 100 degrees/min
N130 DO $R1=$AC_TIME	
N140 X10	; Feedrate = 100 mm/min, path = 10 mm, R1 = approx. 6 s
N150 DO $R2=$AC_TIME	
N160 X10 A10	; Feedrate = 100 mm/min, path = 14.14 mm, R2 = approx. 8 s
N170 DO $R3=$AC_TIME	
N180 A10	; Feedrate = 100 degrees/min, path = 10 degrees, R3 = approx. 6 s
N190 DO $R4=$AC_TIME	
N200 X0.001 A10	; Feedrate = 100 mm/min, path = 10 mm, R4 = approx. 6 s
N210 G700 F100	; Feedrate = 2540 mm/min or 100 degrees/min
N220 DO $R5=$AC_TIME	
N230 X10	; Feedrate = 2540 mm/min, path = 254 mm, R5 = approx. 6 s
N240 DO $R6=$AC_TIME	
N250 X10 A10	; Feedrate = 2540 mm/min, path = 254.2 mm, R6 = approx. 6 s
N260 DO $R7=$AC_TIME	
N270 A10	; Feedrate = 100 degrees/min, path = 10 degrees, R7 = approx. 6 s
N280 DO $R8=$AC_TIME	
N290 X0.001 A10	; Feedrate = 2540 mm/min, path = 10 mm, R8 = approx. 0.288 s
N300 FGREF[A]=360/(2*$PI)	; Set 1 degree = 1 inch via the effective radius
N310 DO $R9=$AC_TIME	
N320 X0.001 A10	; Feedrate = 2540 mm/min, path = 254 mm, R9 = approx. 6 s
N330 M30	

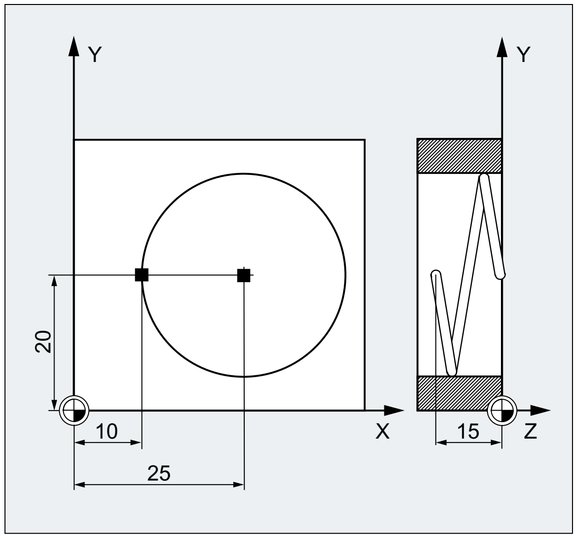
Example 2: Traverse synchronized axes with limit speed FL

The path velocity of the path axes is reduced if the synchronized axis Z reaches the limit velocity.

Program code
N10 G0 X0 Y0
N20 FGROUP(X)
N30 G1 X1000 Y1000 G94 F1000 FL[Y]=500
N40 Z-50

Example 3: Helical interpolation

Path axes X and Y traverse with the programmed feedrate, the infeed axis Z is a synchronized axis.

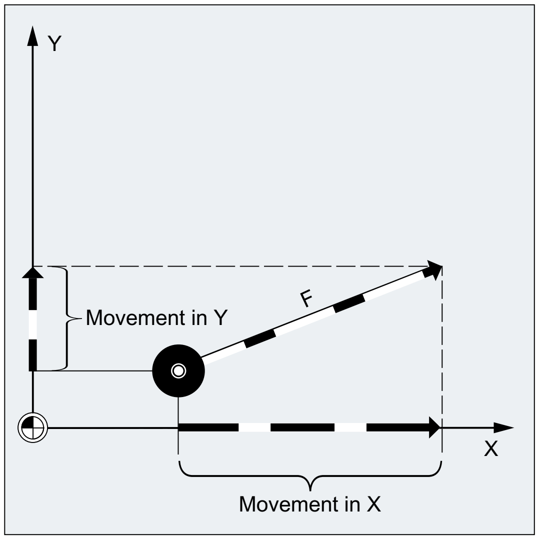


Program code	Comment
N10 G17 G94 G1 Z0 F500	; Feed of the tool.
N20 X10 Y20	; Approach the starting position.
N25 FGROUP(X,Y)	; Axes X/Y are path axes, Z is a synchronized axis.
N30 G2 X10 Y20 Z-15 I15 J0 F1000 FL[Z]=200	; On the circular path, the feedrate is 1,000 mm/min, traversing in the Z direction is synchronized.
...	
N100 FL[Z]=$MA_AX_VELO_LIMIT[0,Z]	; The limit speed is deselected by reading the speed from the MD. Read the value from the MD.
N110 M30	; End of program

Further information

Feedrate for path axes (F)

The path feedrate is generally composed of the individual speed components of all geometry axes participating in the movement and refers to the center point of the cutter or the tip of the turning tool.



The feedrate is specified under address `F`. Depending on the default setting in the machine data, the units of measurement specified with the G commands are either in mm or inch.

One `F` value can be programmed per NC block. The feedrate unit is defined using one of the G commands `G93`/`G94`/`G95`. The feedrate `F` acts only on path axes and remains active until a new feedrate is programmed. Separators are permitted after the address `F`.

Examples:

`F100` or `F 100`

`F.5`

`F=2*FEED`

Feedrate type (G93/G94/G95)

The G commands `G93`, `G94` and `G95` are modal. In the event of switching between `G93`, `G94` and `G95`, the path feedrate value has to be reprogrammed. When machining with rotary axes, the feedrate can also be specified in degrees/min.

Inverse-time feedrate (G93)

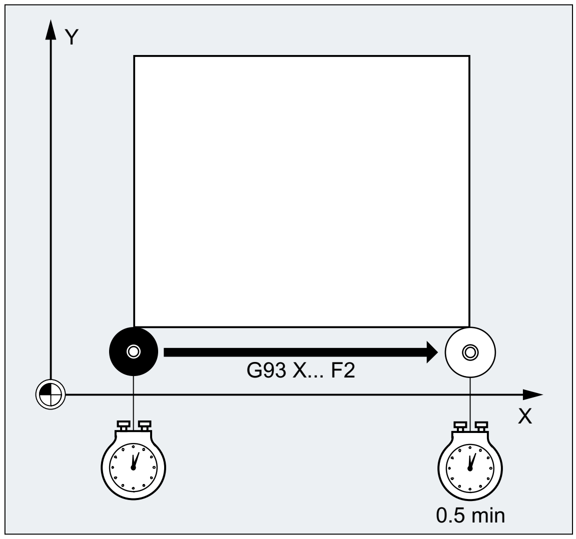
The inverse-time feedrate specifies the time required to execute the motion commands in a block.

Unit: rpm

Example:

`N10 G93 G01 X100 F2`

Means: The programmed path is traversed in 0.5 min.



Note

If the path lengths vary greatly from block to block, a new `F` value should be specified in each block with `G93`. When machining with rotary axes, the feedrate can also be specified in degrees/min.

Feedrate for synchronized axes

The feedrate programmed under address `F` applies to all the path axes programmed in a block but not to the synchronized axes. The synchronized axes are controlled such that they require the same time for their path as the path axes, and all axes reach their end point at the same time.

Limit velocity for synchronized axes (FL)

The `FL` command can be used to program a limit velocity for synchronized axes. In the absence of a programmed `FL`, the rapid traverse velocity applies. `FL` is deselected by assignment to MD (MD36200 $MA_AX_VELO_LIMIT).

Traverse path axis as synchronized axis (FGROUP)

`FGROUP` is used to define whether a path axis should be traversed with path feedrate or as a synchronized axis. In helical interpolation, for example, it is possible to define that only two geometry axes, X and Y, are to be traversed at the programmed feedrate. The infeed axis Z is the synchronized axis in this case.

Example: `FGROUP(X,Y)`

Change FGROUP

The setting made with FGROUP can be changed:

1. By reprogramming FGROUP: e.g. FGROUP(X,Y,Z)
2. By programming FGROUP without a specific axis: FGROUP()
   In accordance with FGROUP(), the initial setting in the machine data applies: Geometry axes are now once again traversed in the path axis grouping.

Note

With FGROUP, axis identifiers must be the names of channel axes.

Units of measurement for feedrate F

In addition to the geometrical settings G700 and G710, the G commands are also used to define the measuring system for the feedrates F. In other words:

- For G700: [inch/min]
- For G710: [mm/min]

Note

G70/G71 have **no** effect on feedrate settings.

Unit of measurement for synchronized axes with limit speed FL

The unit set for F using G command G700/G710 is also valid for FL.

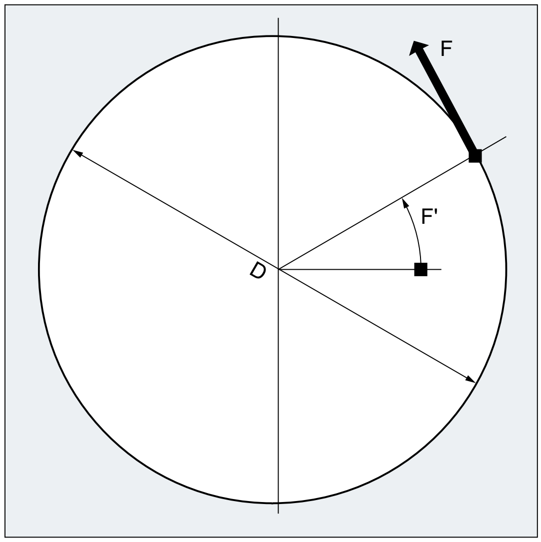
Unit for rotary and linear axes

For linear and rotary axes which are combined with FGROUP and traverse a path together, the feedrate is interpreted in the unit of the linear axes (depending on the default with G94/G95, in mm/min or inch/min and mm/rev or inch/rev).

The tangential velocity of the rotary axis in mm/min or inch/min is calculated according to the following formula:

F[mm/min] = F'[degrees/min] * π * D[mm]/360[degrees]

where:	F:	Tangential velocity
	F':	Angular velocity
	π:	Circle constant
	D:	Diameter



Traverse rotary axes with path velocity F (FGREF)

For machining operations in which the tool or the workpiece or both are moved by a rotary axis, the effective machining feedrate is to be interpreted as a path feed in the usual way by reference to the F value. This requires the specification of an effective radius (reference radius) for each of the rotary axes involved.

The unit of the reference radius depends on the `G70`/`G71`/`G700`/`G710` setting.

All axes involved must be included in the `FGROUP` command to be taken into account in the calculation of the path feedrate.

In order to ensure compatibility with the behavior with no `FGREF` programming, the factor 1 degree = 1 mm is activated on system power up and RESET. This corresponds to a reference radius of FGREF= 360 mm/(2π) = 57.296 mm.

Note

This default is independent of the active basic system (MD10240 $MN_SCALING_SYSTEM_IS_METRIC) and the currently active `G70`/`G71`/`G700`/`G710` setting.

Special situations:

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

With this type of programming, the F value programmed in N110 is evaluated as the rotary axis feedrate in degrees/min, while the feedrate evaluation in N120 is either 100 inch/min or 100 mm/min, dependent upon the currently active G70/G71/G700/G710 setting.

NOTICE
Feedrate difference FGREF evaluation also works if only rotary axes are programmed in the block. The normal F value interpretation as degree/min applies in this case only if the radius reference corresponds to the FGREF default: • For G71/G710: FGREF[A]=57.296 • For G70/G700: FGREF[A]=57.296/25.4

Read reference radius

The value of the reference radius of a rotary axis can be read using system variables:

- In synchronized actions or with preprocessing stop in the part program via system variable:

$AA_FGREF[<axis>]	Current main run value

- Without preprocessing stop in the part program via system variable:

$PA_FGREF[<axis>]	Programmed value

If no values are programmed, the default 360 mm/(2π) = 57.296 mm (corresponding to 1 mm per degree) will be read in both variables.

For linear axes, the value in both variables is always 1 mm.

Read path axes affecting velocity

The axes involved in path interpolation can be read using system variables:

- In synchronized actions or with preprocessing stop in the part program via system variables:

$AA_FGROUP[<axis>]	Returns the value "1" if the specified axis affects the path velocity in the current main run record by means of the basic setting or through FGROUP programming. Otherwise, the variable returns the value "0".
$AC_FGROUP_MASK	Returns a bit key of the channel axes programmed with FGROUP which are to affect the path velocity.

- Without preprocessing stop in the part program via system variables:

$PA_FGROUP[<axis>]	Returns the value "1" if the specified axis affects the path velocity by means of the basic setting or through FGROUP programming. Otherwise, the variable returns the value "0".
$P_FGROUP_MASK	Returns a bit key of the channel axes programmed with FGROUP which are to affect the path velocity.

Path reference factors for orientation axes with FGREF

With orientation axes the mode of operation of the `FGREF[]` factors is dependent upon whether the change in the orientation of the tool is implemented by means of rotary axis or vector interpolation.

In the case of **rotary axis interpolation**, as is the case with rotary axes, the relevant `FGREF` factors of the orientation axes are calculated individually as reference radius for the axis paths.

In the case of **vector interpolation**, an effective `FGREF` factor, which is calculated as the geometric mean value of the individual `FGREF` factors, is applied.

FGREF[effective] = nth root of [(FGREF[A] * FGREF[B]...)]

where:	A:	Axis identifier of 1st orientation axis
	B:	Axis identifier of 2nd orientation axis
	C:	Axis identifier of 3rd orientation axis
	n:	Number of orientation axes

Example:

Since there are two orientation axes for a standard 5-axis transformation, the effective factor is, therefore, the root of the product of the two axial factors:

FGREF[effective] = square root of [(FGREF[A] * FGREF[B])]

Note

It is, therefore, possible to use the effective factor for orientation axes `FGREF` to define a reference point on the tool to which the programmed path feedrate refers.

8.2 Traverse positioning axes (POS, POSA, POSP, FA, WAITP, WAITMC)

Positioning axes are traversed independently of the path axes at a separate, axis-specific feedrate. There are no interpolation commands. The `POS`/`POSA`/`POSP` commands are used to traverse the positioning axes and coordinate the motion sequences at the same time.

The following are typical examples of positioning axes:

- Pallet feed equipment
- Gauging stations

`WAITP` can be used to identify a position in the NC program where the program is to wait until an axis programmed with `POSA` in a previous NC block reaches its end position.

`WAITMC` loads the next NC block immediately when the specified wait marker is received.

Syntax

```
POS[<axis>]=<position>
POSA[<axis>]=<position>
POSP[<axis>]=(<end position>,<partial length>,<mode>)
FA[<axis>]=<value>
WAITP(<axis>) ; Programming in a separate NC block.
WAITMC(<wait marker>)
```

Meaning

<table>
<tr><td rowspan="4">POS/POSA:</td><td colspan="3">Move positioning axis to specified position
POS and POSA have the same functionality but differ in their block change behavior:
• POS delays the enabling of the NC block until the position has been reached.
• POSA enables the NC block even if the position has not been reached.</td></tr>
<tr><td><axis>:</td><td colspan="2">Name of the axis to be traversed (channel or geometry axis identifier)</td></tr>
<tr><td rowspan="2"><position>:</td><td colspan="2">Axis position to be approached</td></tr>
<tr><td>Type:</td><td>REAL</td></tr>
</table>

POSP:	Move positioning axis to specified end position in sections		
	<end position>:	Axis end position to be approached	
	<partial length>:	Length of a section	
	<mode>:	Approach mode	
		= 0:	For the last two sections, the path remaining until the end position is split into two residual sections of equal size (preset).
		= 1:	The partial length is adjusted so that the total of all calculated partial lengths corresponds exactly to the path up to the end position.
	Note: POSP is used specifically to program oscillating motion. **References:** Programming Manual, Job Planning; Section "Oscillation"		
FA:	Feedrate for the specified positioning axis		
	<axis>:	Name of the axis to be traversed (channel or geometry axis identifier)	
	<value>:	Feedrate	
		Unit:	mm/min or inch/min or degrees/min
	Note: Up to five FA values can be programmed for each NC block.		
WAITP:	Wait for a positioning axis to be traversed The subsequent blocks are not processed until the specified positioning axis programmed in a previous NC block with POSA has reached its end position (with exact stop fine).		
	<axis>:	Name of the axis (channel or geometry axis identifier) for which the WAITP command is to be applied	
	Note: With WAITP, an axis can be made available as an oscillating axis or for traversing as a concurrent positioning axis (via PLC).		
WAITMC:	Wait for the specified wait marker to be received When the wait marker is received, the next NC block is loaded immediately.		
	<wait marker>:	Number of the wait marker	

CAUTION

Travel with POSA

If a command, which implicitly causes a preprocessing stop, is read in a following block, this block is not executed until all other blocks which are already preprocessed and stored have been executed. The previous block is stopped in exact stop (as G9).

Examples

Example 1: Travel with POSA and access to machine status data

The control generates an internal preprocessing stop on access to machine status data ($A...). Machining is stopped until all preprocessed and saved blocks have been executed in full.

Program code	Comment
N40 POSA[X]=100	
N50 IF $AA_IM[X]==R100 GOTOF LABEL1	; Access to machine status data.
N60 G0 Y100	
N70 WAITP(X)	
N80 LABEL1:	
N...	

Example 2: Wait for end of travel with WAITP

Pallet feed equipment

Axis U:	Pallet store
	Transport of workpiece pallet to working area
Axis V:	Transfer line to a gauging station where spot checks are carried out to assist the process

Program code	Comment
N10 FA[U]=100 FA[V]=100	; Axis-specific feedrate specifications for the individual positioning axes U and V.
N20 POSA[V]=90 POSA[U]=100 G0 X50 Y70	; Traverse positioning and path axes.
N50 WAITP(U)	; Program execution does not resume until axis U reaches the end point programmed in N20.
…	

Further information

Travel with POSA

Block step enable or program execution is not affected by `POSA`. The movement to the end position can be performed during execution of subsequent NC blocks.

Travel with POS

The next block is not executed until all axes programmed under `POS` reach their end positions.

Wait for end of travel with WAITP

After a `WAITP`, assignment of the axis to the NC program is no longer valid; this applies until the axis is programmed again. This axis can then be operated as a positioning axis through the PLC, or as a reciprocating axis from the NC program/PLC or HMI.

Block change in the braking ramp with IPOBRKA and WAITMC

An axis is only decelerated if the wait marker has not yet been reached or if another end-of-block criterion is preventing the block change. After a `WAITMC`, the axis starts immediately if no other end-of-block criterion is preventing the block change.

8.3 Position-controlled spindle mode (SPCON, SPCOF)

With the commands SPCON or SPCOF, the position-controlled mode of the spindle is explicitly activated or deactivated.

Note

The switching on of the position control mode with SPCON requires a maximum of three position control cycles.

Syntax

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

Meaning

SPCON:	Activate position-controlled mode The specified spindle is switched over from speed control to position control. SPCON s modal and is retained until SPCOF.
SPCOF:	Deactivate position-controlled mode The specified spindle is switched over from position control to speed control.
<n>,<m>, etc.:	0 ... k spindle numbers Without specification of a spindle number: Master spindle of the channel

Note

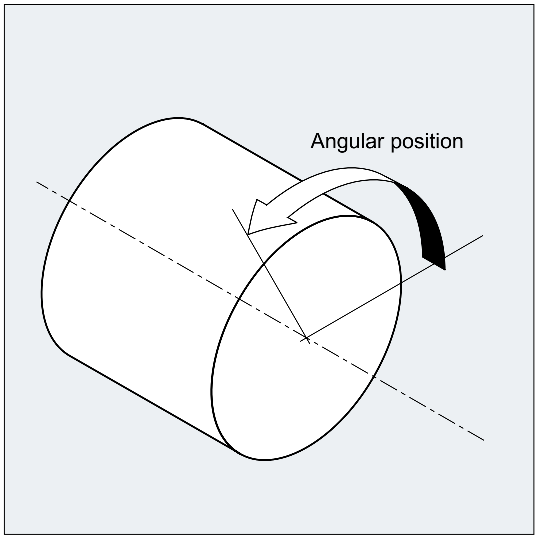
For a synchronous spindle with setpoint coupling, the leading spindle must not be switched to speed-controlled mode with SPCOF.

References

Function Manual, Extended Functions; Section "S3 Synchronous Spindle"

8.4 Positioning spindles (SPOS, SPOSA, M19, M70, WAITS)

`SPOS`, `SPOSA` or `M19` can be used to set spindles to specific angular positions, e.g. during tool change.



`SPOS`, `SPOSA` and `M19` induce a temporary switchover to position-controlled mode until the next `M3`/`M4`/`M5`/`M41` to `M45`.

Positioning in axis mode

The spindle can also be operated as a path axis, synchronized axis or positioning axis at the address defined in the machine data. When the axis identifier is specified, the spindle is in axis mode. `M70` switches the spindle directly to axis mode.

End of positioning

The end-of-motion criterion when positioning the spindle can be programmed using `FINEA`, `CORSEA`, `IPOENDA` or `IPOBRKA`.

The program advances to the next block if the end of motion criteria for all spindles or axes programmed in the current block plus the block change criterion for path interpolation are fulfilled.

Synchronization

In order to synchronize spindle movements, `WAITS` can be used to wait until the spindle position is reached.

Requirements

The spindle to be positioned must be capable of operation in position-controlled mode.

Syntax

Position spindle:

`SPOS=<value>`/`SPOS[<n>]=<value>`

`SPOSA=<value>`/`SPOSA[<n>]=<value>`

`M19`/`M<n>=19`

Switch spindle over to axis mode:

`M70`/`M<n>=70`

Define end-of-motion criterion:

`FINEA`/`FINEA[S<n>]`

`COARSEA`/`COARSEA[S<n>]`

`IPOENDA`/`IPOENDA[S<n>]`

`IPOBRKA`/`IPOBRKA(<axis>[,<instant in time>])` ; Programming in a separate NC block.

Synchronize spindle movements:

`WAITS`/`WAITS(<n>,<m>)` ; Programming in a separate NC block.

Meaning

`SPOS`/`SPOSA`:	Set spindle to specified angle `SPOS` and `SPOSA` have the same functionality but differ in their block change behavior: • `SPOS` delays the enabling of the NC block until the position has been reached. • `SPOSA` enables the NC block even if the position has not been reached.			
	`<n>`:	Number of the spindle to be positioned. If a spindle number is not specified or if the spindle number is set to "0", `SPOS` or `SPOSA` will be applied to the master spindle.		
	`<value>`:	Angular position to which the spindle is to be set.		
		Unit:	Degrees	
		Type:	REAL	
		The following options are available for programming the position approach mode:		
		`=AC(<value>)`:	Absolute dimensions	
			Range of values:	0 … 359,9999
		`=IC(<value>)`:	Incremental dimensions	
			Range of values:	0 … ±99 999,999
		`=DC(<value>)`:	Approach absolute value directly	
		`=ACN(<value>)`:	Absolute dimension, approach in negative direction	
		`=ACP(<value>)`:	Absolute dimension, approach in positive direction	
		`=<value>`:	as `DC(<value>)`	
`M<n>=19`:	Set the master spindle (`M19` or `M0=19`) or spindle number <n> (`M<n>=19`) to the angular position preset with SD43240 $SA_M19_SPOS with the position approach mode preset in SD43250 $SA_M19_SPOSMODE. The NC block is not enabled until the position has been reached.			

M<n>=70:	Switch the master spindle (M70 or M0=70) or spindle number **<n>** (M<n>=70) over to axis mode. No defined position is approached. The NC block is enabled after the switchover has been performed.
FINEA:	Motion end when "Exact stop fine" reached
COARSEA:	Motion end when "Exact stop coarse" reached
IPOENDA:	End of motion on reaching "interpolator stop"
S<n>:	Spindle for which the programmed end-of-motion criterion is to be effective <n>: Spindle number If a spindle is not specified in [S<n>] or a spindle number of "0" is specified, the programmed end-of-motion criterion will be applied to the master spindle.
IPOBRKA:	A block change is possible in the braking ramp. <axis>: Channel axis identifier <instant in time>: Instant in time of the block change with reference to the braking ramp Unit: Percent Range of values: 100 (application point of the braking ramp) to 0 (end of the braking ramp) If a value is not assigned to the <instant in time> parameter, the current value of the setting data is applied: SD43600 $SA_IPOBRAKE_BLOCK_EXCHANGE **Note:** IBOBRKA with an instant in time of "0" is identical to IPOENDA.
WAITS:	Synchronization command for the specified spindle(s) The subsequent blocks are not processed until the specified spindle(s) programmed in a previous NC block with SPOSA has (have) reached its (their) end position(s) (with exact stop fine). WAITS after M5: Wait for the specified spindle(s) to come to a standstill. WAITS after M3/M4: Wait for the specified spindle(s) to reach their setpoint speed. <n>,<m>: Numbers of the spindles to which the synchronization command is to be applied. If a spindle number is not specified or if the spindle number is set to "0", WAITS will be applied to the master spindle.

Note

Three spindle positions are possible for each NC block.

Note

With incremental dimensions IC(<value>), spindle positioning can take place over several revolutions.

Note

If position control was activated with SPCON prior to SPOS, this remains active until SPCOF is issued.

Note

The control detects the transition to axis mode automatically from the program sequence. Explicit programming of M70 in the part program is, therefore, essentially no longer necessary. However, M70 can continue to be programmed, e.g. to increase the legibility of the part program.

Further information

Positioning with SPOSA

The block step enable or program execution is not affected by SPOSA. The spindle positioning can be performed during execution of subsequent NC blocks. The program moves onto the next block if all the functions (except for spindle) programmed in the current block have reached their block end criterion. The spindle positioning operation may be programmed over several blocks (see WAITS).

Note

If a command, which implicitly causes a preprocessing stop, is read in a following block, execution of this block is delayed until all positioning spindles are stationary.

Positioning with SPOS/M19

The block step enabling condition is met when all functions programmed in the block reach their end-of-block criterion (e.g. all auxiliary functions acknowledged by the PLC, all axes at their end point) and the spindle reaches the programmed position.

Velocity of the movements:

The velocity and the delay response for positioning are stored in the machine data. The configured values can be modified by programming or by synchronized actions, see:

- Feedrate for positioning axes / spindles (FA, FPR, FPRAON, FPRAOF) (Page 126)
- Programmable acceleration override (ACC) (option) (Page 131)

Specification of spindle positions:

As the G90/G91 commands are not effective here, the corresponding dimensions apply explicitly, e.g. AC, IC, DC, ACN, ACP. If no specifications are made, traversing automatically takes place as for DC.

Synchronize spindle movements with WAITS

WAITS can be used to identify a point at which the NC program waits until one or more spindles programmed with SPOSA in a previous NC block reach their positions.

Example:

Program code	Comment
N10 SPOSA[2]=180 SPOSA[3]=0	
...	
N40 WAITS(2,3)	; The block waits until spindles 2 and 3 have reached the positions specified in block N10.

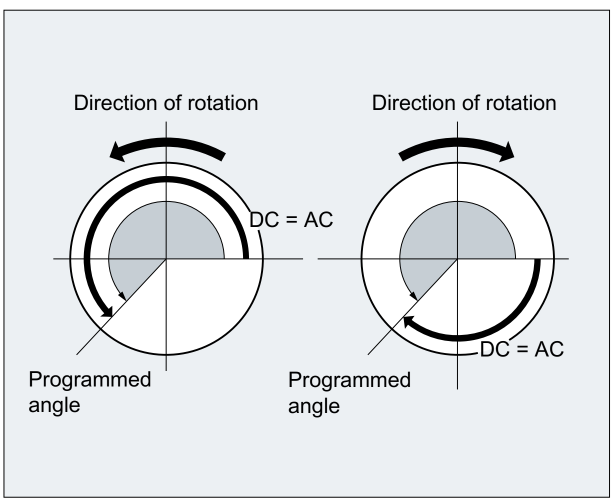
`WAITS` can be used after `M5` to wait until the spindle(s) has (have) stopped. `WAITS` can be used after `M3`/`M4` to wait until the spindle(s) has (have) reached the specified speed/direction of rotation.

Note

If the spindle has not yet been synchronized with synchronization marks, the positive direction of rotation is taken from the machine data (state on delivery).

Position spindle from rotation (M3/M4)

When `M3` or `M4` is active, the spindle comes to a standstill at the programmed value.



There is no difference between `DC` and `AC` dimensioning. In both cases, rotation continues in the direction selected by `M3`/`M4` until the absolute end position is reached. With `ACN` and `ACP`, deceleration takes place if necessary, and the appropriate approach direction is taken. With `IC`, the spindle rotates additionally to the specified value starting at the current spindle position.

Position a spindle from standstill (M5)

The exact programmed distance is traversed from standstill (`M5`).

8.5 Feedrate for positioning axes / spindles (FA, FPR, FPRAON, FPRAOF)

It is also possible to derive the revolutional feedrate for path and synchronized axes or for individual positioning axes/spindles from another rotary axis or spindle.

Positioning axes such as workpiece transport systems, tool turrets and end supports are traversed independently of path and synchronized axes. A separate feedrate is therefore defined for each positioning axis.

A separate axial feedrate can also be programmed for spindles.

Syntax

Feedrate for positioning axis:
```
FA[<axis>]=…
```

Axis feedrate for spindle:
```
FA[SPI(<n>)]=…
FA[S<n>]=…
```

Derive revolutional feedrate for path/synchronized axes:

```
FPR (<rotary axis>)

FPR(SPI(<n>))

FPR(S<n>)
```

Derive rotational feedrate for positioning axes/spindles:

```
FPRAON(<axis>,<rotary axis>)

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

FPRAON(<axis>,S<n>)

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

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

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

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

FPRAOF(<axis>,SPI(<n>), etc.)

FPRAOF(<axis>,S<n>, etc.)
```

Meaning

`FA[...]=...:`	Feedrate for the specified positioning axis or positioning speed (axial feedrate) for the specified spindle	
	Unit:	mm/min or inch/min or degrees/min
	Range of values:	… 999,999.999 mm/min, degrees/min … 39 999.9999 inch/min

FPR(...):	FPR is used to identify the rotary axis (<rotary axis>) or spindle (SPI(<n>)/S<n>) from which the revolutional feedrate for the revolutional feedrate of the path and synchronized axes programmed under G95 is to be derived.
FPRAON(...):	Derive rotational feedrate for positioning axes and spindles The first parameter (<axis>/SPI(<n>)/S<n>) identifies the positioning axis/spindle to be traversed with revolutional feedrate. The second parameter (<rotary axis>/SPI(<n>)/S<n>) identifies the rotary axis/spindle from which the revolutional feedrate is to be derived. **Note:** The second parameter can be omitted, in which case the feedrate will be derived from the master spindle.
FPRAOF(...):	FPRAOF is used to deselect the derived revolutional feedrate for the specified axes or spindles.
<axis>:	Axis identifier (positioning or geometry axis)
SPI(<n>)/S<n>:	Spindle identifier SPI(<n>) and S<n> are identical in terms of function. <n>: Spindle number **Note:** SPI converts spindle numbers into axis identifiers. The transfer parameter (<n>) must contain a valid spindle number.

Note

The programmed feedrate FA[...] is modal.

Up to five feedrates for positioning axes or spindles can be programmed in each NC block.

Note

The derived feedrate is calculated according to the following formula:

Derived feedrate = programmed feedrate * absolute master feedrate

Examples

Example 1: Synchronous spindle coupling

With synchronous spindle coupling, the positioning speed of the following spindle can be programmed independently of the master spindle, e.g. for positioning operations.

Program code	Comment
...	
FA[S2]=100	; Positioning speed of the following spindle (spindle 2) = 100 degrees/min
...	

Example 2: Derived revolutional feedrate for path axes

Path axes X, Y must be traversed at the revolutional feedrate derived from rotary axis A:

Program code
...
N40 FPR(A)
N50 G95 X50 Y50 F500
...

Example 3: Derive revolutional feedrate for master spindle

Program code	Comment
N30 FPRAON(S1,S2)	; The revolutional feedrate for the master spindle (S1) must be derived from spindle 2.
N40 SPOS=150	; Position master spindle.
N50 FPRAOF(S1)	; Deselect derived revolutional feedrate for the master spindle.

Example 4: Derive revolutional feedrate for positioning axis

Program code	Comment
N30 FPRAON(X)	; The revolutional feedrate for positioning axis X must be derived from the master spindle.
N40 POS[X]=50 FA[X]=500	; The positioning axis is traversing at 500 mm/revolution of the master spindle.
N50 FPRAOF(X)	

Further information

FA[…]

The feedrate type is always `G94`. When `G70`/`G71` is active, the unit is metric/inches according to the default setting in the machine data. `G700`/`G710` can be used to modify the unit in the program.

Note

If no `FA` is programmed, the value defined in the machine data applies.

FPR(…)

As an extension of the `G95`command (revolutional feedrate referring to the master spindle), `FPR` allows the revolutional feedrate to be derived from any chosen spindle or rotary axis. `G95 FPR(…)` is valid for path and synchronized axes.

If the rotary axis/spindle specified in the FPR command is operating on position control, then the setpoint linkage is active. Otherwise the actual-value linkage is effective.

FPRAON(…)

`FPRAON` is used to derive the revolutional feedrate for positioning axes and spindles from the current feedrate of another rotary axis or spindle.

FPRAOF(…)

The revolutional feedrate can be deactivated for one or a number of axes/spindles simultaneously with the `FPRAOF` command.

8.6 Programmable feedrate override (OVR, OVRRAP, OVRA)

The velocity of path/positioning axes and spindles can be modified in the NC program.

Syntax

```
OVR=<value>
OVRRAP=<value>
OVRA[<axis>]=<value>
OVRA[SPI(<n>)]=<value>
OVRA[S<n>]=<value>
```

Meaning

`OVR`:	Feedrate modification for path feedrate F
`OVRRAP`:	Feedrate modification for rapid traverse velocity
`OVRA`:	Feedrate modification for positioning feedrate `FA` or for spindle speed `S`
`<axis>`:	Axis identifier (positioning or geometry axis)
`SPI(<n>)`/`S<n>`:	Spindle identifier `SPI(<n>)` and `S<n>` are identical in terms of function. `<n>`: Spindle number **Note:** `SPI` converts spindle numbers into axis identifiers. The transfer parameter `(<n>)` must contain a valid spindle number.
`<value>`:	Feedrate modification in percent The value refers to or is combined with the feedrate override set on the machine control panel. Range of values: … 200%, integers **Note:** With path and rapid traverse override, the maximum velocities set in the machine data is not overshot.

8.7 Programmable acceleration override (ACC) (option)

In critical program sections, it may be necessary to limit the acceleration to below the maximum values, e.g. to prevent mechanical vibrations from occurring.

The programmable acceleration override can be used to modify the acceleration for each path axis or spindle via a command in the NC program. The limit is effective for all types of interpolation. The values defined in the machine data apply as 100% acceleration.

Syntax

```
ACC[<axis>]=<value>
ACC[SPI(<n>)]=<value>
ACC(S<n>)=<value>
```

Deactivate:

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

Syntax

ACC:	Acceleration change for the specified path axis or speed change for the specified spindle.
<axis>:	Channel axis name of path axis
SPI(<n>)/S<n>:	Spindle identifier SPI(<n>) and S<n> are identical in terms of function. <n>: Spindle number **Note:** SPI converts spindle numbers into axis identifiers. The transfer parameter (<n>) must contain a valid spindle number.
<value>:	Acceleration change in percent The value refers to or is combined with the feedrate override set on the machine control panel. Range of values: 1 to 200%, integers

Note

With a greater acceleration rate, the values permitted by the manufacturer may be exceeded.

Example

Program code	**Comment**
N50 ACC[X]=80	; The axis slide in the X direction should only be traversed with 80% acceleration.
N60 ACC[SPI(1)]=50	; Spindle 1 should only accelerate or brake with 50% of the acceleration capacity.

Further information

Acceleration override programmed with ACC

The acceleration override programmed with `ACC[...]` is always taken into consideration on output as in system variable $AA_ACC. Readout in the parts program and in synchronized actions takes place at different times in the NC processing run.

In the part program

The value written in the part program is then only taken into consideration in system variable $AA_ACC as written in the part program if `ACC` has not been changed in the meantime by a synchronized action.

In synchronized actions

The following thus applies: The value written to a synchronized action is then only considered in system variable $AA_ACC as written to the synchronized action if `ACC` has not been changed in the meantime by a part program.

The preset acceleration can also be changed via synchronized actions (see Function Manual, Synchronized Actions).

Example:

Program code
`...`
`N100 EVERY $A_IN[1] DO POS[X]=50 FA[X]=2000 ACC[X]=140`

The current acceleration value can be called with system variable $AA_ACC[<axis>]. Machine data can be used to define whether the last `ACC` value set should apply on RESET/part program end or whether 100% should apply.

8.8 Feedrate with handwheel override (FD, FDA)

The `FD` and `FDA` commands can be used to traverse axes with handwheels during execution of the part program. The programmed settings for traversing the axes are then overlaid with the handwheel pulses evaluated as path or velocity defaults.

Path axes
In the case of path axes, the programmed path feedrate can be overlaid. The handwheel of the 1st geometry axis of the channel is evaluated. The handwheel pulses evaluated per interpolation cycle dependent on the direction of rotation correspond to the path velocity to be overlaid. The path velocity limit values which can be achieved by means of handwheel override are:

- Minimum: 0
- Maximum: Machine data limit values of the path axes involved in traversing

---

Note

Path feedrate

The path feedrate F and the handwheel feedrate FD cannot be programmed in the same NC block.

Positioning axes
In the case of positioning axes, the travel path or velocity can be overlaid as an axial value. The handwheel assigned to the axis is evaluated.

- Path override
  The handwheel pulses evaluated dependent on the direction of rotation correspond to the axis path to be traveled. Only handwheel pulses in the direction of the programmed position are evaluated.
- Velocity override
  The handwheel pulses evaluated per interpolation cycle dependent on the direction of rotation correspond to the axial velocity to be overlaid. The path velocity limit values which can be achieved by means of handwheel override are:
  - Minimum: 0
  - Maximum: Machine data limit values of the positioning axis

A detailed description of how to set handwheel parameters appears in:

Reference
/FB2/ Function Manual Extended Functions; manual traversing and manual handwheel travel (H1)

Syntax

```
FD=<velocity>
FDA[<axis>]=<velocity>
```

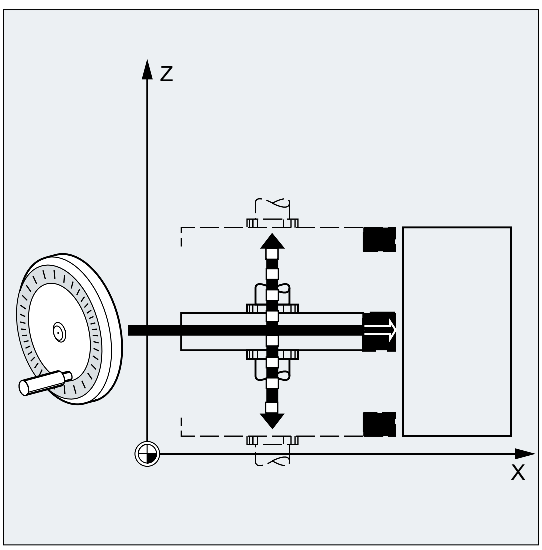
Meaning

`FD=<velocity>`:	Path feedrate and enabling of velocity override with handwheel <velocity>: • Value = 0: Not allowed! • Value ≠ 0: Path velocity
`FDA[<axis>]=<velocity>`:	Axial feedrate <velocity>: • Value = 0: Path default with handwheel • Value ≠ 0: Axial velocity
`<axis>`:	Axis identifier of positioning axis

Note

`FD` and `FDA` are non-modal.

Example



Path definition: The grinding wheel oscillating in the Z direction is traversed to the workpiece in the X direction with the handwheel.

The operator can continue to feed manually until the sparks are flying uniformly. Activating "Delete distance-to-go" switches to the next NC block and machining continues in AUTOMATIC mode.

Further information

Traverse path axes with velocity override (FD=<velocity>)

The following conditions must be met for the part program block in which path velocity override is programmed:

- Path command `G1`, `G2` or `G3` active
- Exact stop `G60` active
- Linear feedrate `G94` active

Feedrate override

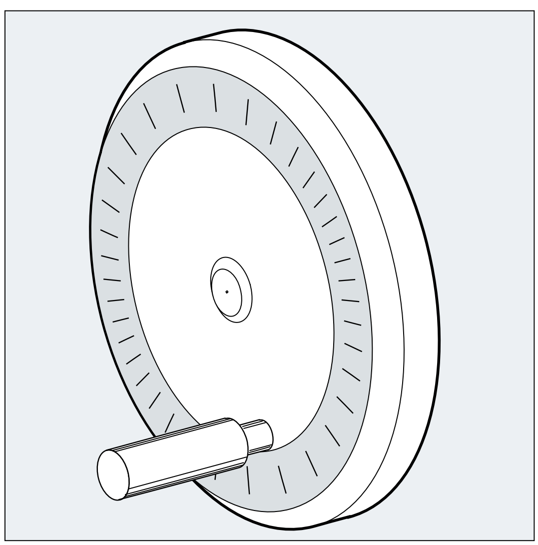
The feedrate override only affects the programmed path velocity and not the velocity component generated with the handwheel (exception: (except if feed override = 0).

Example:

Program code	Description
N10 X… Y… F500	; Feedrate = 500 mm/min
N20 X… Y… FD=700	; Feedrate = 700 mm/min and velocity override with handwheel.
	; Acceleration from 500 to 700 mm/min in N20. The handwheel can be used to vary the speed dependent on the direction of rotation between 0 and the maximum value (machine data).

Traverse positioning axes with path default (FDA[<axis>]=0)

In the NC block with programmed `FDA[<axis>]=0` the feed is set to zero so that the program cannot generate any travel movement. The programmed travel movement to the target position is now controlled exclusively by the operator rotating the handwheel.



Example:

Program code	Description
...	
N20 POS[V]=90 FDA[V]=0	; Target position = 90 mm, axial feedrate = 0 mm/min and path override with handwheel.
	; Velocity of axis V at start of block = 0 mm/min.
	; Path and speed defaults are set using handwheel pulses

Direction of movement, travel velocity

The axes follow the path set by the handwheel in the direction of the sign. Forward and backwards travel is possible dependent on the direction of rotation. The faster the handwheel rotates, the higher the traversing speed.

Traversing range:
The traversing range is limited by the starting position and the programmed end point.

Traverse positioning axis with velocity override (FDA[<axis>]=<velocity>)

In NC blocks with programmed `FDA[…]=…`, the feedrate from the last programmed `FA` value is accelerated or decelerated to the value programmed under `FDA`. Starting from the current feedrate `FDA`, the handwheel can be turned to accelerate the programmed movement to the target position or decelerate it to zero. The values set as parameters in the machine data serve as the maximum velocity.

Example:

Program code	Description
N10 POS[V]=… FA[V]=100	; Axial feedrate = 100 mm/min
N20 POS[V]=100 FAD[V]=200	; Axial target position = 100, axial feedrate = 200 mm/min
	; and velocity override with handwheel.
	; Acceleration from 100 to 200 mm/min in N20. The
	; Handwheel can be used to vary the velocity depending on the direction of rotation
	; between 0 and the maximum value (machine data).

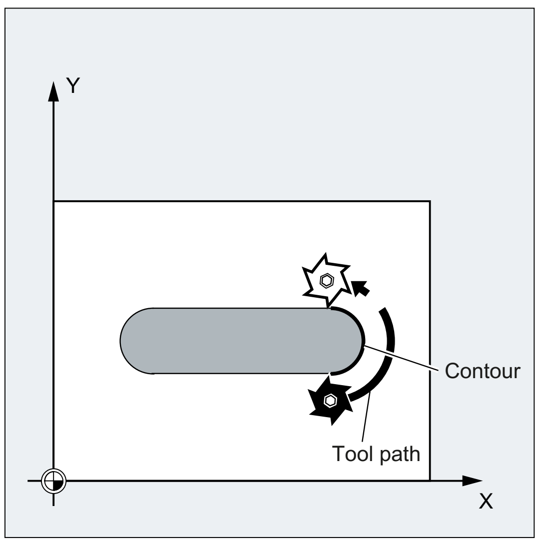
Traversing range:
The traversing range is limited by the starting position and the programmed end point.

8.9 Feedrate optimization for curved path sections (CFTCP, CFC, CFIN)

With activated correction mode G41/G42, the programmed feedrate for the milling tool radius first refers to the milling tool center path (refer to Chapter "Coordinate transformations (frames) (Page 321)").

When you mill a circle (the same applies to polynomial and spline interpolation) the extent to which the feedrate varies at the cutter edge is so significant under certain circumstances that it can impair the quality of the machined part.

Example: Milling a small outside radius with a large tool. The path that the outside of the milling tool must travel is considerably longer than the path along the contour.



Because of this, machining at the contour takes place with a very low feedrate. To prevent adverse effects, the feedrate needs to be controlled accordingly for curved contours.

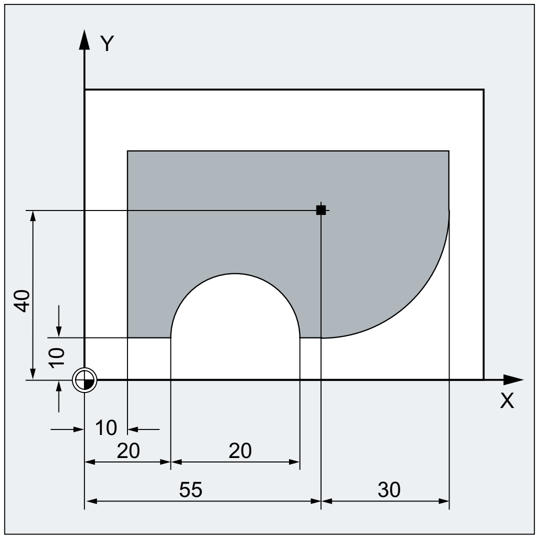
Syntax

```
CFTCP
CFC
CFIN
```

Meaning

`CFTCP`:	Constant feedrate on the milling cutter center path The control keeps the feedrate constant and feedrate overrides are deactivated.
`CFC`:	Constant feedrate at the contour (tool cutting edge). This function is preset per default.
`CFIN`:	Constant feedrate at the tool cutting edge only at concave contours, otherwise on the milling cutter center path. The feedrate is reduced for inside radii.

Example

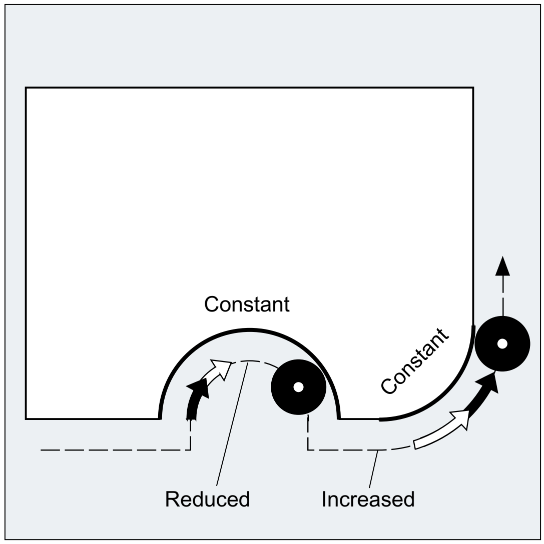


In this example, the contour is first produced with `CFC`-corrected feedrate. During finishing, the cutting base is also machined with `CFIN`. This prevents the cutting base being damaged at the outside radii by a feedrate that is too high.

Program code	Comment
`N10 G17 G54 G64 T1 M6`	
`N20 S3000 M3 CFC F500 G41`	
`N30 G0 X-10`	
`N40 Y0 Z-10`	`; Feed to first cutting depth`
`N50 CONTOUR1`	`; Subprogram call`
`N40 CFIN Z-25`	`; Feed to second cutting depth`
`N50 CONTOUR1`	`; Subprogram call`
`N60 Y120`	
`N70 X200 M30`	

Further information

Constant feedrate on contour with CFC



The feedrate is reduced for inside radii and increased for outside radii. This ensures a constant speed at the tool edge and thus at the contour.

8.10 Several feedrate values in one block (F, ST, SR, FMA, STA, SRA)

The "Multiple feedrates in one block" function can be used to activate different feedrate values for an NC block, a dwell time or a retraction motion-synchronously, dependent on external digital and/or analog inputs.

Syntax

Path motion

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

Axial motion:

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

Meaning

F=...:	The path feedrate is programmed under the address F and remains valid during the absence of an input signal.	
	Effective:	Modal
F2=... to F7=...:	In addition to the path feedrate, up to six further feedrates can be programmed in the block. The numerical expansion indicates the bit number of the input that activates the feedrate when changed:	
	Effective:	Non-modal
ST=...:	Dwell time in s (for grinding technology: sparking-out time)	
	Input bit:	1
	Effective:	Non-modal
SR=...:	Retraction path The unit for the retraction path refers to the current valid unit of measurement (mm or inch).	
	Input bit:	0
	Effective:	Non-modal
FA[<Ax>]=...:	The axial feedrate is programmed under the address FA and remains valid during the absence of an input signal.	
	Effective:	Modal
FMA[2,<Ax>]=... to FMA[7,<Ax>]=...:	In addition to the axial feedrate FA up to six further feedrates per axis can be programmed in the block with FMA. The first parameter indicates the bit number of the input and the second the axis for which the feedrate is to apply.	
	Effective:	Non-modal

`STA[<Ax>]=...`:	Axial dwell time in s (for grinding technology: sparking-out time)	
	Input bit:	1
	Effective:	Non-modal
`SRA[<Ax>]=...`:	Axial retraction path	
	Input bit:	0
	Effective:	Non-modal
`<Ax>`:	Axis for which the feedrate is to apply	

Note

Priority of the signals

The signals are scanned in ascending order starting at input bit 0 (I0). Therefore, the retraction motion has the highest priority and the feedrate F7 the lowest priority. Dwell time and retraction motion end the feedrate motions that were activated with F2 to F7.

The signal with the highest priority determines the current feedrate.

Note

Delete distance-to-go

If input bit 1 is activated for the dwell time or bit 0 for the return path, the distance to go for the path axes or the relevant single axes is deleted and the dwell time or return started.

Note

Retraction path

The unit for the retraction path refers to the current valid unit of measurement (mm or inch).

The reverse stroke is always made in the opposite direction to the current motion. `SR`/`SRA` always programs the value for the reverse stroke. No sign is programmed.

Note

POS instead of POSA

If feedrates, dwell time or return path are programmed for an axis on account of an external input, this axis must not be programmed as POSA axis (positioning axis over multiple blocks) in this block.

Note

Status query

It is also possible to poll the status of an input for synchronous commands of various axes.

Note

LookAhead

Look Ahead is also active for multiple feedrates in one block. In this way, the current feedrate can be restricted by the Look Ahead value.

Examples

Example 1: Path motion

Program code	Comment
`G1 X48 F1000 F7=200 F6=50 F5=25 F4=5 ST=1.5 SR=0.5`	`; Path feedrate = 1000`

	`; Additional path feedrate values:`
	`; 200 (input bit 7)`
	`; 50 (input bit 6)`
	`; 25 (input bit 5)`
	`; 5 (input bit 4)`

	`; Dwell time 1.5 s`
	`; Retraction 0.5 mm`

Example 2: Axial motion

Program code	Comment
`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`	`; Feedrate for axis A = 800`

	`; Additional feedrate values for axis A: 720 (input bit 7)`
	`; 640 (input bit 6)`
	`; 560 (input bit 5)`

	`; Axial dwell time: 1.5 s`
	`; Axial retraction: 0.5 mm`

Example 3: Multiple operations in one block

Program code	Comment
`N20 T1 D1 F500 G0 X100`	`Initial setting`

`N25 G1 X105 F=20 F7=5 F3=2.5 F2=0.5 ST=1.5 SR=0.5`	`; Normal feedrate with F,`
	`; roughing with F7,`
	`; finishing with F3,`
	`; smooth-finishing with F2,`
	`; dwell time 1.5 s,`
	`; retraction path 0.5 mm`

`...`	

8.11 Non-modal feedrate (FB)

The "Non-modal feedrate" function can be used to define a separate feedrate for a single block. After this block, the previous modal feedrate is active again.

Syntax

```
FB=<value>
```

Meaning

`FB`:	Feedrate for current block only
`<VALUE>`:	The programmed value must be greater than zero. Values are interpreted based on the active feedrate type: • G94: feedrate in mm/min or degrees/min • G95: feedrate in mm/rev or inch/rev • G96: constant cutting rate

---

Note

If no traversing motion is programmed in the block (e.g. computation block), the `FB` has no effect.

If no explicit feedrate for chamfering/rounding is programmed, then the value of `FB` also applies for any chamfering/rounding contour element in this block.

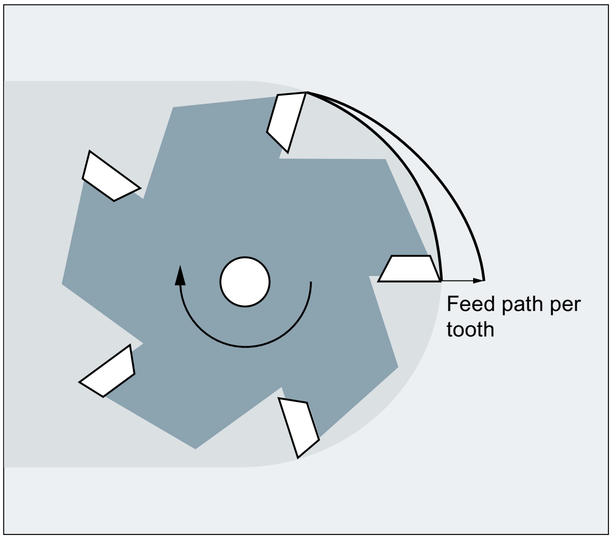
Feedrate interpolations `FLIN`, `FCUB`, etc. are also possible without restriction.

Simultaneous programming of `FB` and `FD` (handwheel travel with feedrate override) or `F` (modal path feedrate) is **not** possible.

Example

Program code	Comment
`N10 G0 X0 Y0 G17 F100 G94`	`;Initial setting`
`N20 G1 X10`	`; Feedrate 100 mm/min`
`N30 X20 FB=80`	`; Feedrate 80 mm/min`
`N40 X30`	`; Feedrate is 100 mm/min again.`
`...`	

8.12 Tooth feedrate (G95 FZ)



The control uses the $TC_DPNT (number of teeth) tool parameter associated with the active tool offset data block to calculate the effective revolutional feedrate for each traversing block from the programmed tooth feedrate.

F = FZ * $TC_DPNT

with:	F:	Revolutional feedrate in mm/rev or inch/rev
	FZ:	Tooth feedrate in mm/tooth or inch/tooth
	$TC_DPNT:	System variable tool parameter: Number of teeth/rev

The tool type ($TC_DP1) of the active tool is not taken into account.

The programmed tooth feedrate is independent of the tool change and the selection/deselection of a tool offset data block; it is retained in modal format.

A change to the $TC_DPNT tool parameter associated with the active tool cutting edge will be applied the next time a tool offset is selected or the next time the active offset data is updated.

Changing the tool or selecting/deselecting a tool offset data block generates a recalculation of the effective revolutional feedrate.

Note

The tooth feedrate refers only to the path (axis-specific programming is not possible).

Syntax

```
G95 FZ...
```

Meaning

G95:	Type of feedrate: Revolutional feedrate in mm/rev or inch/rev (dependent upon G700/G710) For G95 see "Feedrate (G93, G94, G95, F, FGROUP, FL, FGREF) (Page 107)"	
FZ:	Tooth feedrate	
	Activation:	with G95
	Effective:	Modal
	Unit:	mm/tooth or inch/tooth (dependent upon G700/G710)

NOTICE

Tool change/Changing the master spindle

A subsequent tool change or changing the master spindle must be taken into account by the user by means of corresponding programming, e.g. reprogramming FZ.

NOTICE

Tool operations undefined

Technological concerns such as climb milling or conventional milling, front face milling or peripheral face milling, etc., along with the path geometry (straight line, circle, etc.), are not taken into account automatically. Therefore, these factors have to be given consideration when programming the tooth feedrate.

Note

Switchover between G95 F... and G95 FZ...

Switching over between G95 F... (revolutional feedrate) and G95 FZ... (tooth feedrate) will delete the non-active feedrate value in each case.

Note

Derive feedrate with FPR

As is the case with the revolutional feedrate, FPR can also be used to derive the tooth feedrate of any rotary axis or spindle (see "Feedrate for positioning axes / spindles (FA, FPR, FPRAON, FPRAOF) (Page 126)").

Examples

Example 1: Milling cutter with five teeth ($TC_DPNE = 5)

Program code	Comment
N10 G0 X100 Y50	
N20 G1 G95 FZ=0.02	; Tooth feedrate 0.02 mm/tooth
N30 T3 D1	; Load tool and activate tool offset data block.
M40 M3 S200	; Spindle speed 200 rpm

Program code	Comment
N50 X20	; Milling with: FZ = 0.02 mm/tooth effective revolutional feedrate: F = 0.02 mm/tooth * 5 teeth/rev = 0.1 mm/rev or F = 0.1 mm/rev * 200 rpm = 20 mm/min
...	

Example 2: Switchover between G95 F... and G95 FZ...

Program code	Comment
N10 G0 X100 Y50	
N20 G1 G95 F0.1	; Revolutional feedrate 0.1 mm/rev
N30 T1 M6	
N35 M3 S100 D1	
N40 X20	
N50 G0 X100 M5	
N60 M6 T3 D1	; Load tool with e.g. five teeth ($TC_DPNT = 5).
N70 X22 M3 S300	
N80 G1 X3 G95 FZ=0.02	; Change G95 F… to G95 FZ…, tooth feedrate active with 0.02 mm/tooth.
...	

Example 3: Derive tooth feedrate of a spindle (FBR)

Program code	Comment
…	
N41 FPR(S4)	; Tool in spindle 4 (not the master spindle).
N51 G95 X51 FZ=0.5	; Tooth feedrate 0.5 mm/tooth dependent upon spindle S4.
...	

Example 4: Subsequent tool change

Program code	Comment
N10 G0 X50 Y5	
N20 G1 G95 FZ=0.03	; Tooth feedrate 0.03 mm/tooth
N30 M6 T11 D1	; Load tool with e.g. seven teeth ($TC_DPNT = 7).
N30 M3 S100	
N40 X30	; Effective revolutional feedrate 0.21 mm/rev
N50 G0 X100 M5	
N60 M6 T33 D1	; Load tool with e.g. five teeth ($TC_DPNT = 5).
N70 X22 M3 S300	
N80 G1 X3	; Tooth feedrate modal 0.03 mm/tooth, effective revolutional feedrate 0.15 mm/rev
...	

Example 5: Changing the master spindle

Program code	Comment
N10 SETMS (1)	; Spindle 1 is the master spindle.
N20 T3 D3 M6	; Tool 3 is changed to spindle 1.
N30 S400 M3	; Speed S400 of spindle 1 (and therefore T3).
N40 G95 G1 FZ0.03	; Tooth feedrate 0.03 mm/tooth
N50 X50	; Path motion, the effective feedrate is dependent upon:
	- The tooth feedrate FZ
	- The speed of spindle 1
	- The number of teeth of the active tool T3
N60 G0 X60	
...	
N100 SETMS(2)	; Spindle 2 becomes the master spindle.
N110 T1 D1 M6	; Tool 1 is changed to spindle 2.
N120 S500 M3	; Speed S500 of spindle 2 (and therefore T1).
N130 G95 G1 FZ0.03 X20	; Path motion, the effective feedrate is dependent upon:
	- The tooth feedrate FZ
	- The speed of spindle 2
	- The number of teeth of the active tool T1

Note

Following the change in the master spindle (`N100`), a tool actuated by spindle 2 must be substituted (`N110`).

Further information

Changing between G93, G94 and G95

`FZ` can also be programmed when `G95` is not active, although it will have no effect and is deleted when `G95` is selected. In other words, when changing between `G93`, `G94`, and `G95`, in the same way as with `F`, the `FZ` value is also deleted.

Reselection of G95

Reselecting `G95` when `G95` is already active has no effect (unless a change between `F` and `FZ` has been programmed).

Non-modal feedrate (FB)

When `G95 FZ...` (modal) is active, a non-modal feedrate `FB...` is interpreted as a tooth feedrate.

SAVE mechanism

In subprograms with the `SAVE` attribute `FZ` is written to the value prior to the subprogram starting (in the same way as `F`).

Multiple feedrate values in one block

The "Multiple feedrate values in one block" function is not possible with tooth feedrate.

Synchronized actions

`FZ` cannot be programmed from synchronized actions.

Read tooth feedrate and path feedrate type

The tooth feedrate and the path feedrate type can be read using system variables.

- With preprocessing stop in the part program via system variables:

	$AC_FZ	Tooth feedrate effective when the current main run block was preprocessed.	
	$AC_F_TYPE	Path feedrate type effective when the current main run block was preprocessed.	
		Value:	**Meaning:**
		0	mm/min
		1	mm/rev
		2	inch/min
		3	inch/rev
		11	mm/tooth
		33	inch/tooth

- Without preprocessing stop in the part program via system variables:

	$P_FZ	Programmed tooth feedrate	
	$P_F_TYPE	Programmed path feedrate type	
		Value:	**Meaning:**
		0	mm/min
		1	mm/rev
		2	inch/min
		3	inch/rev
		11	mm/tooth
		33	inch/tooth

Note

If `G95` is not active, the $P_FZ and $AC_FZ variables will always return a value of zero.

Geometry settings 9

9.1 Settable zero offset (G54 to G57, G505 to G599, G53, G500, SUPA, G153)

The `G54` to `G57` and `G505` to `G599` commands activate the values of the associated parameterizable zero offsets of the workpiece coordinate system compared with the base coordinate system set from the user interface.

Syntax

Switching on:

```
G54
...
G57
G505
...
G599
```

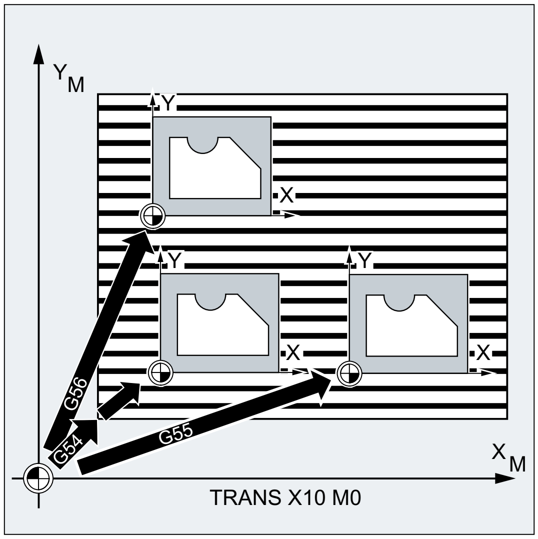
Switching off or suppressing:

```
G500
G53
G153
SUPA
```

Meaning

`G54` to `G57`:	Call of the 1st to 4th settable zero offset (WO)	
`G505` to `G599`:	Call of the 5th to 99th settable zero offset	
`G500`:	Deactivation of the current settable zero offset	
	`G500`=zero frame: (default setting; contains no offset, rotation, mirroring or scaling)	Deactivation of the settable zero offset until the next call, activation of the entire basic frame ($P_ACTBFRAME).
	`G500` not equal to 0:	Activation of the first settable zero offset ($P_UIFR[0]) and activation of the entire basic frame ($P_ACTBFRAME) or possibly a modified basic frame is activated.
`G53`:	`G53` suppresses the settable zero offset and the programmable zero offset non-modally.	
`G153`:	`G153` has the same effect as `G53` and also suppresses the entire basic frame.	
`SUPA`:	`SUPA` has the same effect as `G153` and also suppresses: • Handwheel offsets (DRF) • Overlaid movements • External zero offset • PRESET offset	

Example



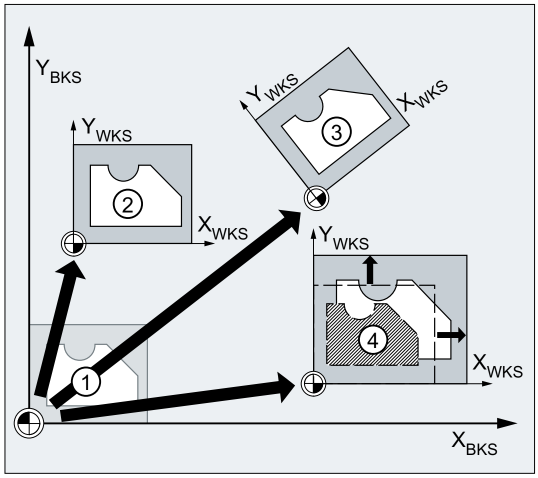
Three workpieces that are arranged on a pallet in accordance with the zero offset values `G54` to `G56` are to be machined in succession. The machining sequence is programmed in subprogram L47.

Program code	Comment
`N10 G0 G90 X10 Y10 F500 T1`	`; Approach`
`N20 G54 S1000 M3`	`; Call of the first WO, spindle clockwise`
`N30 L47`	`; Program pass as subprogram`
`N40 G55 G0 Z200`	`; Call of the second WO, Z via obstruction`
`N50 L47`	`; Program pass as subprogram`
`N60 G56`	`; Call of the third WO`
`N70 L47`	`; Program pass as subprogram`
`N80 G53 X200 Y300 M30`	`; Suppress zero offset, end of program`

Further information

A parameterized zero offset is in principle a set frame (Page 321). Consequently, the following components and frame values are also available for a parameterized zero offset:

- Offset
- Rotation
- Scaling
- Scale



① Initial position in the BCS
② Offset
③ Offset + rotation
④ Offset + scaling

Figure 9-1 Zero offsets

The frame values for the parameterized zero offsets are input from the user interface:

SINUMERIK Operate: "Parameters" > "Zero offsets" > "Details" operating area

SINUMERIK 828D

For SINUMERIK 828D, the 5th and 6th parameterized zero offsets are called with `G58` and `G59`, respectively.

The `G505` and `G506` commands are not available for SINUMERIK 828D.

Parameterized number of parameterizable frames (G505 - G599)

The number of user-specific parameterized zero offsets (G505 - G599) can be set for each specific channel via:

MD28080 $MC_MM_NUM_USER_FRAMES = <number>

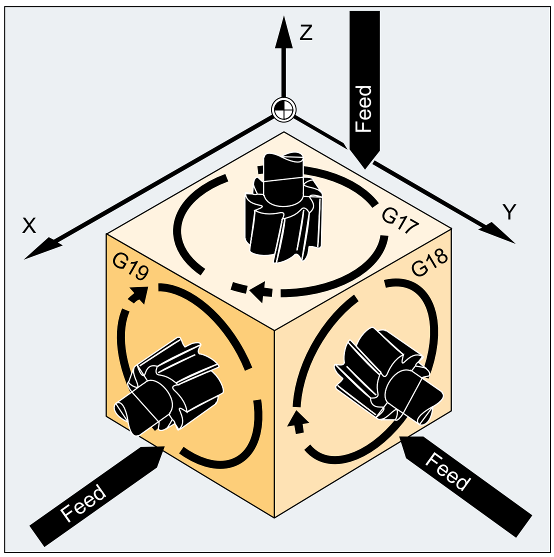
See also

Programmable work offset (G58, G59) (Page 331)

9.2 Selection of the working plane (G17/G18/G19)

The specification of the working plane, in which the desired contour is to be machined, also defines the following functions:

- The plane for tool radius compensation
- The infeed direction for tool length offset depending on the tool type
- The plane for circular interpolation



Syntax

```
G17/G18/G19, etc.
```

Meaning

G17:	Working plane X/Y Infeed direction Z, plane selection 1st - 2nd geometry axis
G18:	Working plane Z/X Infeed direction Y, plane selection 3rd - 1st geometry axis
G19:	Working plane Y/Z Infeed direction X, plane selection 2nd - 3rd geometry axis

Note

In the default setting, G17 (X/Y plane) is defined for milling and G18 (Z/X plane) is defined for turning.

When calling the tool path correction G41/G42 (see Section "Tool radius compensation (Page 263)"), the working plane must be defined so that the control can correct the tool length and radius.

Example

The "conventional" approach for milling is:

1. Define working plane (`G17` default setting for milling).
2. Select tool type (`T`) and tool offset values (`D`).
3. Switch on path correction (`G41`).
4. Program traversing movements.

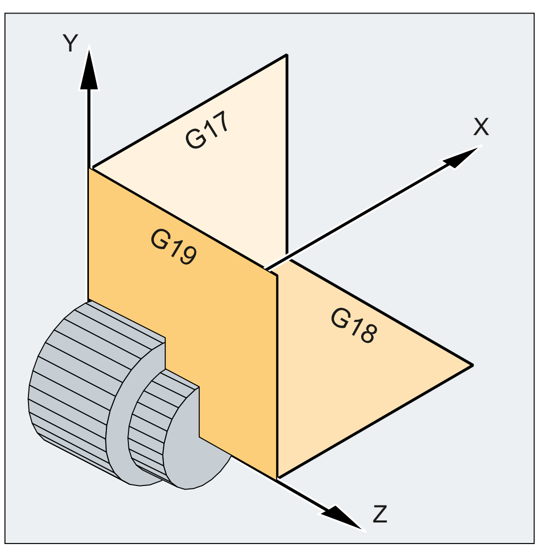
Program code	Comment
N10 G17 T5 D8	; Call of working plane X/Y, tool call. Tool length offset is performed in the Z direction.
N20 G1 G41 X10 Y30 Z-5 F500	; Radius compensation is performed in the X/Y plane.
N30 G2 X22.5 Y40 I50 J40	; Circular interpolation / tool radius compensation in the X/Y plane.

Further information

General

It is recommended that the working plane G17 to G19 be selected at the start of the program. In the default setting, the Z/X plane is preset for turning G18.

Turning:



The control requires the specification of the working plane for the calculation of the direction of rotation (see circular interpolation G2/G3).

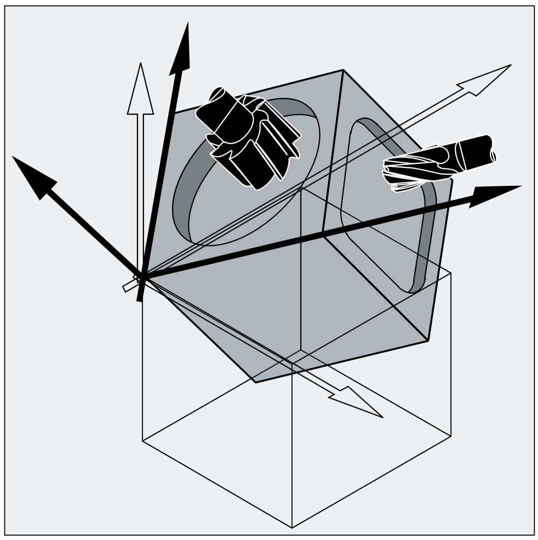
Machining on inclined planes

Rotate the coordinate system with `ROT` (see Section "Coordinate system offset") to position the coordinate axes on the inclined surface. The working planes rotate accordingly.

Tool length compensation on inclined planes

As a general rule, the tool length compensation always refers to the fixed, non-rotated working plane.

Milling:



Note

The tool length components can be calculated according to the rotated working planes with the functions for "Tool length compensation for orientable tools".

The compensation plane is selected with CUT2D, CUT2DF. For further information on this and for the description of the available calculation methods,see Chapter "Tool radius compensation (Page 263)".

The control provides convenient coordinate transformation functions for the spatial definition of the working plane. Please see Chapter "Coordinate transformations (frames) (Page 321)" for more information.

9.3 Dimensions

The basis of most NC programs is a workpiece drawing with specific dimensions.

These dimensions can be:

- In absolute dimensions or in incremental dimensions
- In millimeters or inches
- In radius or diameter (for turning)

Specific programming commands are available for the various dimension options so that the data from a dimension drawing can be transferred directly (without conversion) to the NC program.

9.3.1 Absolute dimensions (G90, AC)

With absolute dimensions, the position specifications always refer to the zero point of the currently valid coordinate system, i.e. the absolute position is programmed, on which the tool is to traverse.

Modal absolute dimensions

Modal absolute dimensions are activated with the `G90` command. Generally it applies to all axes programmed in subsequent NC blocks.

Non-modal absolute dimensions

With preset incremental dimensions (`G91`), the `AC` command can be used to set non-modal absolute dimensions for individual axes.

Note

Non-modal absolute dimensions (`AC`) are also possible for spindle positioning (`SPOS`, `SPOSA`) and interpolation parameters (`I`, `J`, `K`).

Syntax

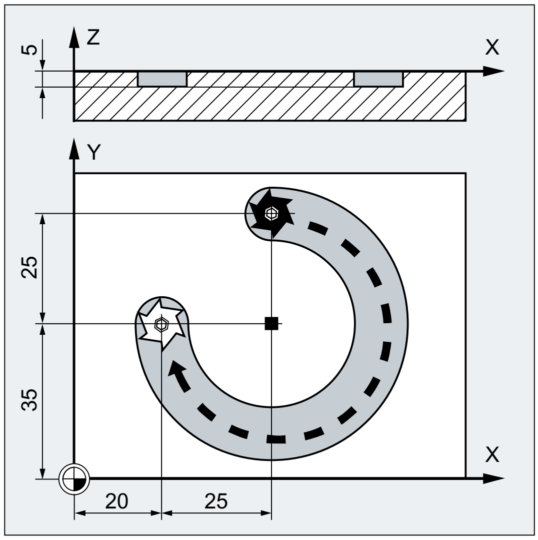
```
G90
<axis>=AC(<value>)
```

Meaning

`G90`:	Command for the activation of modal absolute dimensions
`AC`:	Command for the activation of non-modal absolute dimensions
`<axis>`:	Axis identifier of the axis to be traversed
`<value>`:	Position setpoint of the axis to be traversed in absolute dimensions

Examples

Example 1: Milling

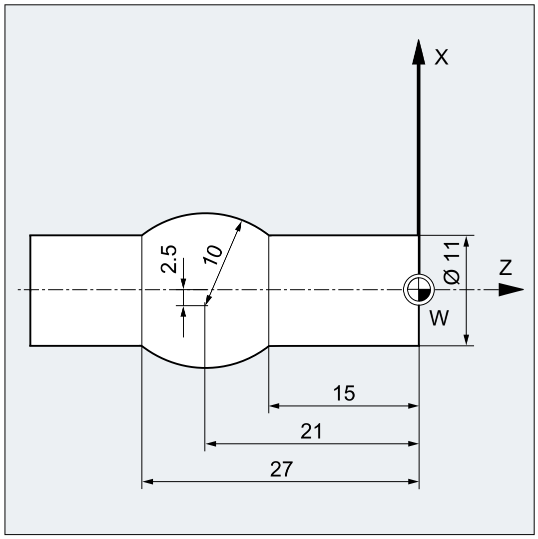


Program code	Comment
N10 G90 G0 X45 Y60 Z2 T1 S2000 M3	; Absolute dimension input, in rapid traverse to position XYZ, tool selection, spindle on with clockwise direction of rotation.
N20 G1 Z-5 F500	; Linear interpolation, feed of the tool.
N30 G2 X20 Y35 I=AC(45) J=AC(35)	; Clockwise circular interpolation, circle end point and circle center point in absolute dimensions.
N40 G0 Z2	; Traverse
N50 M30	; End of block

Note

For information on the input of the circle center point coordinates I and J, see Section "Circular interpolation".

Example 2: Turning



Program code	Comment
N5 T1 D1 S2000 M3	; Loading of tool T1, spindle on with clockwise direction of rotation.
N10 G0 G90 X11 Z1	; Absolute dimension input, in rapid traverse to position XZ.
N20 G1 Z-15 F0.2	; Linear interpolation, feed of the tool.
N30 G3 X11 Z-27 I=AC(-5) K=AC(-21)	; Counter-clockwise circular interpolation, circle end point and circle center point in absolute dimensions.
N40 G1 Z-40	; Traverse
N50 M30	; End of block

Note

For information on the input of the circle center point coordinates I and J, see Section "Circular interpolation".

See also

Absolute and incremental dimensions for turning and milling (G90/G91) (Page 161)

9.3.2 Incremental dimensions (G91, IC)

With incremental dimensions, the position specification refers to the last point approached, i.e. the programming in incremental dimensions describes by how much the tool is to be traversed.

Modal incremental dimensions

Modal incremental dimensions are activated with the `G91` command. Generally it applies to all axes programmed in subsequent NC blocks.

Non-modal incremental dimensions

With preset absolute dimensions (G90), the IC command can be used to set non-modal incremental dimensions for individual axes.

Note

Non-modal incremental dimensions (IC) are also possible for spindle positioning (SPOS, SPOSA) and interpolation parameters (I, J, K).

Syntax

```
G91
<axis>=IC(<value>)
```

Meaning

G91:	Command for the activation of modal incremental dimensions
IC:	Command for the activation of non-modal incremental dimensions
<axis>:	Axis identifier of the axis to be traversed
<value>:	Position setpoint of the axis to be traversed in incremental dimensions

G91 extension

For certain applications, such as scratching, it is necessary that only the programmed distance is traversed in incremental dimensions. The active zero offset or tool length offset is not traversed.

This behavior can be set separately for the active zero offset and tool length offset via the following setting data:

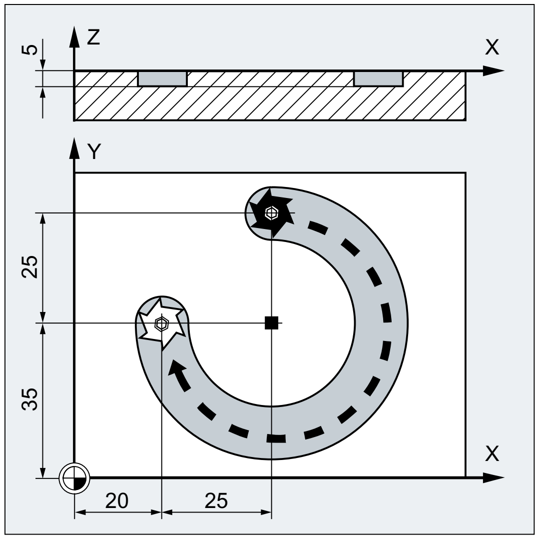
SD42440 $SC_FRAME_OFFSET_INCR_PROG (zero offsets in frames)

SD42442 $SC_TOOL_OFFSET_INCR_PROG (tool length offsets)

Value	Meaning
0	With incremental programming (incremental dimensions) of an axis, the zero offset or the tool length offset is **not** traversed.
1	With incremental programming (incremental dimensions) of an axis, the zero offset or the tool length offset is traversed.

Examples

Example 1: Milling

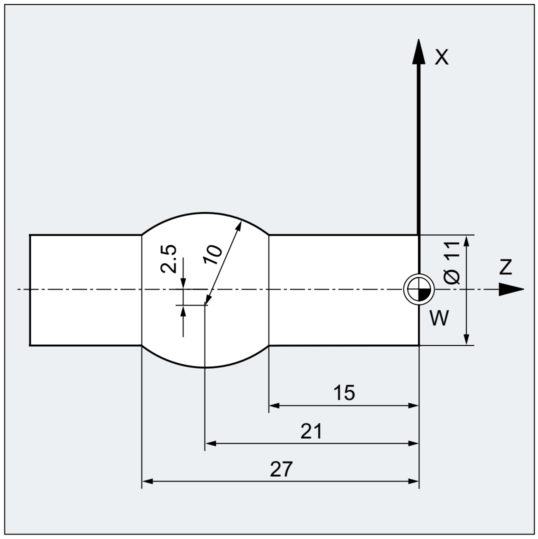


Program code	Comment
N10 G90 G0 X45 Y60 Z2 T1 S2000 M3	; Absolute dimension input, in rapid traverse to position XYZ, tool selection, spindle on with clockwise direction of rotation
N20 G1 Z-5 F500	; Linear interpolation, feed of the tool.
N30 G2 X20 Y35 I0 J-25	; Clockwise circular interpolation, circle end point in absolute dimensions, circle center point in incremental dimensions.
N40 G0 Z2	; Traverse
N50 M30	; End of block

Note

For information on the input of the circle center point coordinates I and J, see Section "Circular interpolation".

Example 2: Turning



Program code	Comment
N5 T1 D1 S2000 M3	; Loading of tool T1, spindle on with clockwise direction of rotation.
N10 G0 G90 X11 Z1	; Absolute dimension input, in rapid traverse to position XZ.
N20 G1 Z-15 F0.2	; Linear interpolation, feed of the tool.
N30 G3 X11 Z-27 I-8 K-6	; Counter-clockwise circular interpolation, circle end point in absolute dimensions, circle center point in incremental dimensions.
N40 G1 Z-40	; Traverse
N50 M30	; End of block

Note

For information on the input of the circle center point coordinates I and J, see Section "Circular interpolation".

Example 3: Incremental dimensions without traversing of the active zero offset

Settings:

- G54 contains an offset in X of 25
- SD42440 $SC_FRAME_OFFSET_INCR_PROG = 0

Program code	Comment
N10 G90 G0 G54 X100	
N20 G1 G91 X10	; Incremental dimensions active, traversing in X of 10 mm (the zero offset is not traversed).

Program code	Comment
N30 G90 X50	; Absolute dimensions active, traverse to position X75 (the zero offset is traversed).

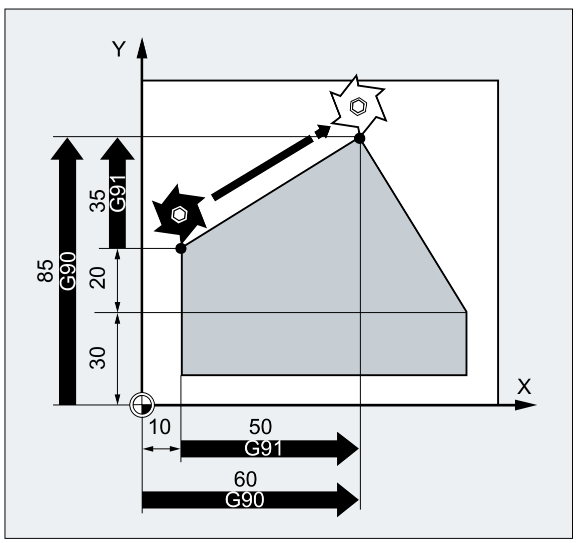
See also

Absolute and incremental dimensions for turning and milling (G90/G91) (Page 161)

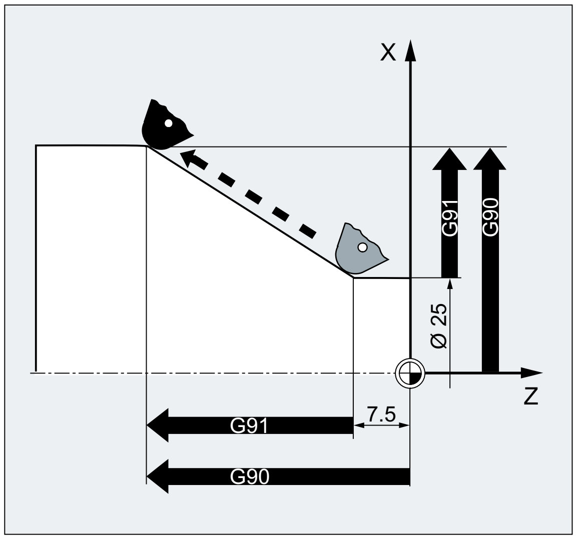
9.3.3 Absolute and incremental dimensions for turning and milling (G90/G91)

The two following figures illustrate the programming with absolute dimensions (`G90`) or incremental dimensions (`G91`) using turning and milling technology examples.

Milling:



Turning:

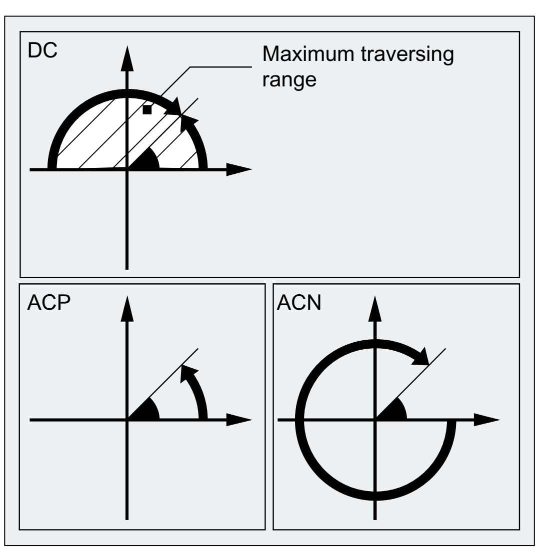


> **Note**
>
> On conventional turning machines, it is usual to consider incremental traversing blocks in the transverse axis as radius values, while diameter specifications apply for the reference dimensions. This conversion for `G90` is performed using the commands `DIAMON`, `DIAMOF` or `DIAM90`.

9.3.4 Absolute dimensions for rotary axes (DC, ACP, ACN)

The non-modal and `G90`/`G91`-independent commands `DC`, `ACP` and `ACN` are available for the positioning of rotary axes in absolute dimensions.

`DC`, `ACP` and `ACN` differ in the basic approach strategy:



Syntax

```
<rotary axis>=DC(<value>)
<rotary axis>=ACP(<value>)
<rotary axis>=ACN(<value>)
```

Meaning

`<rotary axis>`:	Identifier of the rotary axis that is to be traversed (e.g. A, B or C)	
`DC`:	Command for the **direct** approach to the position The rotary axis approaches the programmed position directly on the shortest path. The rotary axis traverses a maximum range of 180°.	
`ACP`:	Command to approach the position in a **positive** direction The rotary axis traverses to the programmed position in the positive direction of axis rotation (counter-clockwise).	
`ACN`:	Command to approach the position in a **negative** direction The rotary axis traverses to the programmed position in the negative direction of axis rotation (clockwise).	
`<value>`:	Rotary axis position to be approached in absolute dimensions	
	Range of values:	0 - 360 degrees

Note

The positive direction of rotation (clockwise or counter-clockwise) is set in the machine data.

Note

The traversing range between 0° and 360° must be set in the machine data (modulo behavior) for positioning with direction specification (`ACP`, `ACN`). `G91` or `IC` must be programmed to traverse modulo rotary axes more than 360° in a block.

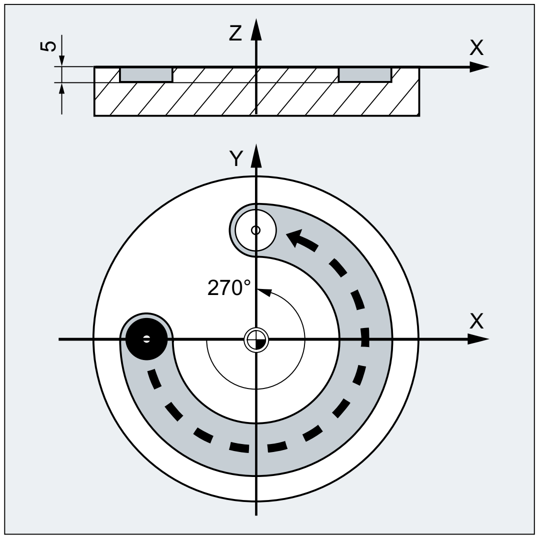
Note

The commands `DC`, `ACP` and `ACN` can also be used for spindle positioning (`SPOS`, `SPOSA`) from standstill.

Example: `SPOS=DC(45)`

Example

Milling on a rotary table



The tool is stationary, the table turns to 270° **in a clockwise direction** to produce a circular groove.

Program code	Comment
N10 SPOS=0	; Spindle in position control.
N20 G90 G0 X-20 Y0 Z2 T1	; Absolute dimensions, feed tool T1 in rapid traverse.
N30 G1 Z-5 F500	; Lower tool during feed.
N40 C=ACP(270)	; Table turns clockwise to 270 degrees (positive), the tool mills a circular groove.
N50 G0 Z2 M30	; Retraction, end of program.

References

Function Manual, Extended Functions; Rotary Axes (R2)

9.3.5 Metric/inch dimension system (G70/G71, G700/G710)

Using the commands of G group 13 (inch/metric system of units) within a part program, you can switch over between the metric and inch system of units.

Activation

In order that commands G700 and G710 are available, the extended system of units functionality must be switched on (MD10260 $MN_CONVERT_SCALING_SYSTEM = 1).

Syntax

```
G70
G71
```

```
G700
G710
```

Meaning

`G70`:	Activating the inch system of units The **inch system of units** is used to read and write **geometrical data in units of length**. **Technological data in units of length** (e.g. feedrates, tool offsets, adjustable work offsets, machine data and system variables) is read and written using the **parameterized basic system**.	
	G group:	13
	Initial setting:	Settable via MD20150 $MC_GCODE_RESET_VALUES
	Effectiveness:	Modal
`G71`:	Activating the metric system of units The **metric system of units** is used to read and write **geometrical data in units of length**. **Technological data in units of length** (e.g. feedrates, tool offsets, adjustable work offsets, machine data and system variables) is read and written using the **parameterized basic system**.	
	G group:	13
	Initial setting:	Settable via MD20150 $MC_GCODE_RESET_VALUES
	Effectiveness:	Modal
`G700`:	Activating the inch system of units All geometrical and technological data in units of length is read and written using the inch system of units.	
	G group:	13
	Initial setting:	Settable via MD20150 $MC_GCODE_RESET_VALUES
	Effectiveness:	Modal
`G710`:	Activating the metric system of units All geometrical and technological data in units of length is read and written using the metric system of units.	
	G group:	13
	Initial setting:	Settable via MD20150 $MC_GCODE_RESET_VALUES
	Effectiveness:	Modal

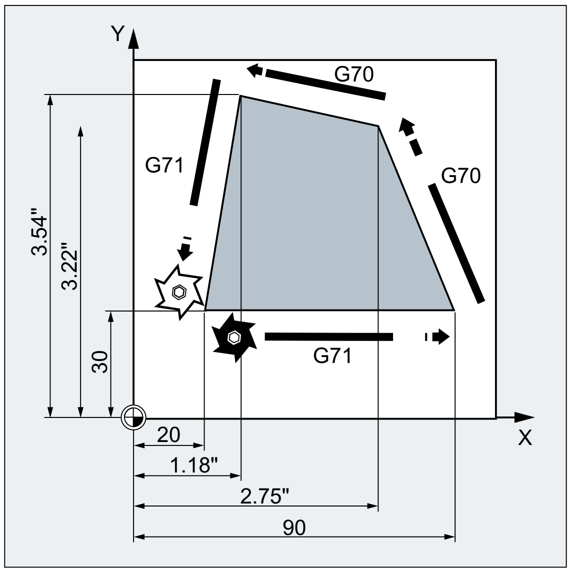
NOTICE

Axis-specific data of rotary axes

Axis-specific data of rotary axes is read and written using the parameterized basic system.

Example

The basic system is metric (MD10240 $MN_SCALING_SYSTEM_IS_METRIC = 1). However, the workpiece drawing has dimensions shown in inches. This is the reason why within the part program, the inch system of units is selected. After the inch dimensions have been processed, the metric system of units is again selected.



Program code	Comment
`N10 G0 G90 X20 Y30 Z2 S2000 M3 T1`	`; X=20 mm, Y=30 mm, Z=2 mm, F=rapid traverse 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`	`; programmed system of units: inch`
	`; X=2.75 inch, Y=3.22 inch, 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`	`; programmed system of units: Metric`
	`; X=20 mm, Y=30 mm, F=500 mm/min`
`N70 G0 Z2`	`; Z=2 mm, F=rapid traverse mm/min`
`N80 M30`	`; end of program`

Additional information

Reading and writing data in the case of G70/G71 and G700/G710

Data area	G70 / G71		G700 / G710	
	Read	Write	Read	Write
Display, decimal places (WCS)	P	P	P	P
Display, decimal places (MCS)	G	G	G	G
Feedrates	G	G	P	P
Position data X, Y, Z	P	P	P	P

Data area	G70 / G71		G700 / G710	
	Read	Write	Read	Write
Interpolation parameters I, J, K	P	P	P	P
Circle radius (CR)	P	P	P	P
Polar radius (RP)	P	P	P	P
Thread pitch	P	P	P	P
Programmable FRAME	P	P	P	P
Settable FRAMES	G	G	P	P
Basic frames	G	G	P	P
External work offsets	G	G	P	P
Axial preset offset	G	G	P	P
Working area limits (G25/G26)	G	G	P	P
Protection zones	P	P	P	P
Tool offsets	G	G	P	P
Length-related machine data	G	G	P	P
Length-related setting data	G	G	P	P
Length-related system variables	G	G	P	P
GUDs	G	G	G	G
LUDs	G	G	G	G
PUDs	G	G	G	G
R parameters	G	G	G	G
Siemens cycles	P	P	P	P
Jog/handwheel increment factor	G	G	G	G
P: Writing/reading is performed in the programmed system of units. G: Writing/reading is performed in the configured basic system				

Synchronized actions

Note

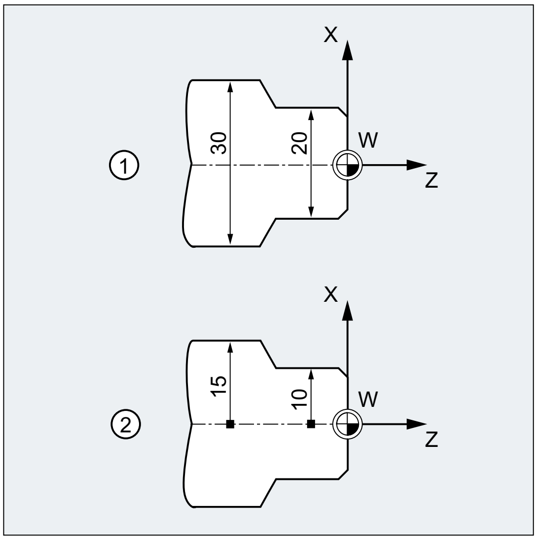
Reading position data in synchronized actions

If a system of units has not been explicitly programmed in the synchronized action (condition component and/or action component) **length-related position data** in the synchronized action will always be read in the parameterized **basic system**.

References: Function Manual, Synchronized Actions

9.3.6 Channel-specific diameter/radius programming (DIAMON, DIAM90, DIAMOF, DIAMCYCOF)

During turning, the dimensions **for the transverse axis** can be specified in the diameter (①) or in the radius (②):



So that the dimensions from a technical drawing can be transferred directly (without conversion) to the NC program, channel-specific diameter or radius programming is activated using the modal commands `DIAMON`, `DIAM90`, `DIAMOF`, and `DIAMCYCOF`.

Note

The channel-specific diameter/radius programming refers to the geometry axis defined as transverse axis via MD20100 $MC_DIAMETER_AX_DEF (→ see machine manufacturer's specifications).

Only one transverse axis per channel can be defined via MD20100.

Syntax

```
DIAMON
DIAM90
DIAMOF
```

Meaning

`DIAMON`:	Command for the activation of the **independent** channel-specific diameter programming. The effect of `DIAMON` is independent of the programmed dimensions mode (absolute dimensions `G90` or incremental dimensions `G91`):	
	• For G90:	Dimensions in the diameter
	• For G91:	Dimensions in the diameter

<table>
<tr><td rowspan="3">DIAM90:</td><td colspan="2">Command for the activation of the dependent channel-specific diameter programming.
The effect of DIAM90 depends on the programmed dimensions mode:</td></tr>
<tr><td>• For G90:</td><td>Dimensions in the diameter</td></tr>
<tr><td>• For G91:</td><td>Dimensions in the radius</td></tr>
<tr><td rowspan="3">DIAMOF:</td><td colspan="2">Command for the deactivation of the channel-specific diameter programming
Channel-specific radius programming takes effect when diameter programming is deactivated. The effect of DIAMOF is independent of the programmed dimensions mode:</td></tr>
<tr><td>• For G90:</td><td>Dimensions in the radius</td></tr>
<tr><td>• For G91:</td><td>Dimensions in the radius</td></tr>
<tr><td>DIAMCYCOF:</td><td colspan="2">Command for the deactivation of channel-specific diameter programming during cycle processing.
In this way, computations in the cycle can always be made in the radius. The last G command active in this group remains active for the position indicator and the basic block indicator.</td></tr>
</table>

Note

With `DIAMON` or `DIAM90`, the transverse-axis actual values will always be displayed as a diameter. This also applies to reading of actual values in the workpiece coordinate system with `MEAS`, `MEAW`, `$P_EP[x]` and `$AA_IW[x]`.

Example

Program code	**Comment**
`N10 G0 X0 Z0`	`; Approach starting point.`
`N20 DIAMOF`	`; Diameter programming off.`
`N30 G1 X30 S2000 M03 F0.7`	`; X axis = transverse axis, radius programming active; traverse to radius position X30.`
`N40 DIAMON`	`; The diameter programming is active for the transverse axis.`
`N50 G1 X70 Z-20`	`; Traverse to diameter position X70 and Z-20.`
`N60 Z-30`	
`N70 DIAM90`	`; Diameter programming for absolute dimensions and radius programming for incremental dimensions.`
`N80 G91 X10 Z-20`	`; Incremental dimensions active.`
`N90 G90 X10`	`; Absolute dimensions active.`
`N100 M30`	`; End of program`

Additional information

Diameter values (DIAMON/DIAM90)

The diameter values apply for the following data:

- Actual value display of the transverse axis in the workpiece coordinate system
- JOG mode: Increments for incremental dimensions and manual handwheel travel
- Programming of end positions:
  Interpolation parameters `I`, `J`, `K` for `G2`/`G3`, if these have been programmed absolutely with `AC`.
  If `I`, `J`, `K` are programmed incrementally (`IC`), the radius is always calculated.
- Reading actual values in the workpiece coordinate system for:
  `MEAS`, `MEAW`, `$P_EP[X]`, `$AA_IW[X]`

9.3.7 Axis-specific diameter/radius programming (DIAMONA, DIAM90A, DIAMOFA, DIACYCOFA, DIAMCHANA, DIAMCHAN, DAC, DIC, RAC, RIC)

In addition to channel-specific diameter programming, the axis-specific diameter programming function enables the modal or non-modal dimensions and display in the diameter for one or more axes.

Note

The axis-specific diameter programming is only possible for axes that are permitted as further transverse axes for the axis-specific diameter programming via MD30460 $MA_BASE_FUNCTION_MASK (→ see machine manufacturer's specifications).

Syntax

Modal axis-specific diameter programming for several transverse axes in the channel:

```
DIAMONA[<axis>]
DIAM90A[<axis>]
DIAMOFA[<axis>]
DIACYCOFA[<axis>]
```

Acceptance of the channel-specific diameter/radius programming:

```
DIAMCHANA[<axis>]
DIAMCHAN
```

Non-modal axis-specific diameter/radius programming:

```
<axis>=DAC(<value>)
<axis>=DIC(<value>)
<axis>=RAC(<value>)
<axis>=RIC(<value>)
```

Meaning

Modal axis-specific diameter programming	
`DIAMONA`:	Command for the activation of the **independent** axis-specific diameter programming The effect of `DIAMONA` is independent of the programmed dimensions mode (`G90`/`G91` or `AC`/`IC`): • For G90, AC: Dimensions in the diameter • For G91, IC: Dimensions in the diameter
`DIAM90A`:	Command for the activation of the **dependent** axis-specific diameter programming The effect of `DIAM90A` depends on the programmed dimensions mode: • For G90, AC: Dimensions in the diameter • For G91, IC: Dimensions in the radius
`DIAMOFA`:	Command for the deactivation of the axis-specific diameter programming Axis-specific radius programming takes effect when diameter programming is deactivated. The effect of `DIAMOFA` is independent of the programmed dimensions mode: • For G90, AC: Dimensions in the radius • For G91, IC: Dimensions in the radius
`DIACYCOFA`:	Command for the deactivation of axis-specific diameter programming during cycle processing. In this way, computations in the cycle can always be made in the radius. The last G command active in this group remains active for the position indicator and the basic block indicator.
`<axis>`:	Axis identifier of the axis for which the axis-specific diameter programming is to be activated. Permitted axis identifiers are as follows: • Geometry/channel axis name or • Machine axis name Range of values: The axis specified must be a known axis in the channel. Other conditions: • The axis must be permitted for the axis-specific diameter programming via MD30460 $MA_BASE_FUNCTION_MASK. • Rotary axes are not permitted to serve as transverse axes.
Acceptance of the channel-specific diameter/radius programming	
`DIAMCHANA`:	With the `DIAMCHANA[<axis>]` command, the **specified axis** accepts the channel status of the diameter/radius programming and is then assigned to the channel-specific diameter/radius programming.
`DIAMCHAN`:	With the `DIAMCHAN` command, **all** axes permitted for the axis-specific diameter programming accept the channel status of the diameter/radius programming and are then assigned to the channel-specific diameter/radius programming.
Non-modal axis-specific diameter/radius programming	
The non-modal axis-specific diameter/radius programming specifies the dimension type as a diameter or radius value in the part program and synchronized actions. The modal status of diameter/radius programming remains unchanged.	

DAC:	The DAC command sets the following dimensions to non-modal for the specified axis: Diameter in absolute dimensions
DIC:	The DIC command sets the following dimensions to non-modal for the specified axis: Diameter in incremental dimensions
RAC:	The RAC command sets the following dimensions to non-modal for the specified axis: Radius in absolute dimensions
RIC:	The RIC command sets the following dimensions to non-modal for the specified axis: Radius in incremental dimensions

Note

With DIAMONA[<axis>] or DIAM90A[<axis>], the transverse-axis actual values are always displayed as a diameter. This also applies to reading of actual values in the workpiece coordinate system with MEAS, MEAW, $P_EP[x] and $AA_IW[x].

Note

During the replacement of an additional transverse axis because of a GET request, the status of the diameter/radius programming in the other channel is accepted with RELEASE[<axis>].

Examples

Example 1: Modal axis-specific diameter/radius programming

X is the transverse axis in the channel, axis-specific diameter programming is permitted for Y.

Program code	Comment
N10 G0 X0 Z0 DIAMON	; Channel-specific diameter programming active for X.
N15 DIAMOF	; Channel-specific diameter programming off.
N20 DIAMONA[Y]	; Modal axis-specific diameter programming active for Y.
N25 X200 Y100	; Radius programming active for X.
N30 DIAMCHANA[Y]	; Y accepts the status of the channel-specific diameter/ radius programming and is assigned to this.
N35 X50 Y100	; Radius programming active for X and Y.
N40 DIAMON	; Channel-specific diameter programming on.
N45 X50 Y100	; Diameter programming active for X and Y.

Example 2: Non-modal axis-specific diameter/radius programming

X is the transverse axis in the channel, axis-specific diameter programming is permitted for Y.

Program code	Comment
`N10 DIAMON`	`; Channel-specific diameter programming on.`
`N15 G0 G90 X20 Y40 DIAMONA[Y]`	`; Modal axis-specific diameter programming active for Y.`
`N20 G01 X=RIC(5)`	`; Dimensions effective in this block for X: Radius in incremental dimensions.`
`N25 X=RAC(80)`	`; Dimensions effective in this block for X: Radius in absolute dimensions.`
`N30 WHEN $SAA_IM[Y]> 50 DO POS[X]=RIC(1)`	`; X is command axis. Dimensions effective in this block for X: Radius in incremental dimensions.`
`N40 WHEN $SAA_IM[Y]> 60 DO POS[X]=DAC(10)`	`; X is command axis. Dimensions effective in this block for X: Radius in absolute dimensions.`
`N50 G4 F3`	

Further information

Diameter values (DIAMONA/DIAM90A)

The diameter values apply for the following data:

- Actual value display of the transverse axis in the workpiece coordinate system
- JOG mode: Increments for incremental dimensions and manual handwheel travel
- Programming of end positions:
  Interpolation parameters `I`, `J`, `K` for `G2`/`G3`, if these have been programmed absolutely with `AC`.
  If `I`, `J`, `K` are programmed incrementally (`IC`), the radius is always calculated.
- Reading actual values in the workpiece coordinate system for:
  `MEAS`, `MEAW`, `$P_EP[X]`, `$AA_IW[X]`

Non-modal axis-specific diameter programming (DAC, DIC, RAC, RIC)

The statements `DAC`, `DIC`, `RAC`, `RIC` are permissible for any commands for which channel-specific diameter programming is relevant:

- Axis position: `X...`, `POS`, `POSA`
- Oscillation: `OSP1`, `OSP2`, `OSS`, `OSE`, `POSP`
- Interpolation parameters: `I`, `J`, `K`
- Contour definition: Straight line with angle specification

- Rapid retraction: `POLF[AX]`
- Traversing in the tool direction: `MOVT`
- Smooth approach and retraction:
  `G140` to `G143`, `G147`, G148, `G247`, `G248`, `G347`, `G348`, `G340`, `G341`

9.4 Position of workpiece for turning

Axis identifiers

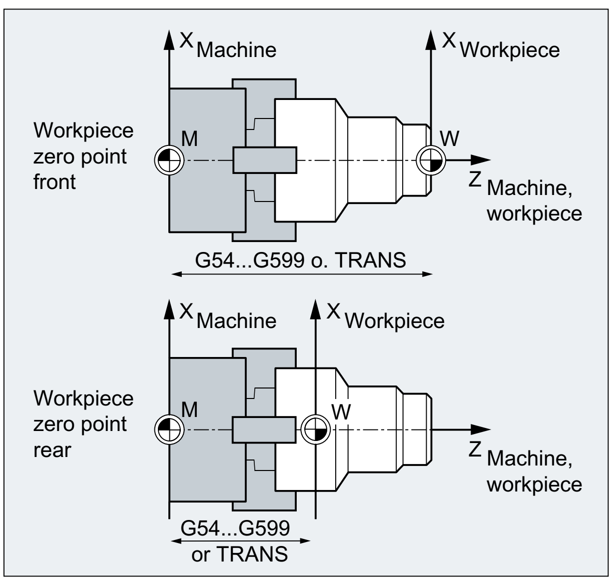
The two geometry axes perpendicular to one another are usually called:

Longitudinal axis	= Z axis (abscissa)
Transverse axis	= X axis (ordinate)

Workpiece zero

Whereas the machine zero is permanently defined, the workpiece zero can be freely selected on the longitudinal axis. Generally the workpiece zero is on the front or rear side of the workpiece.

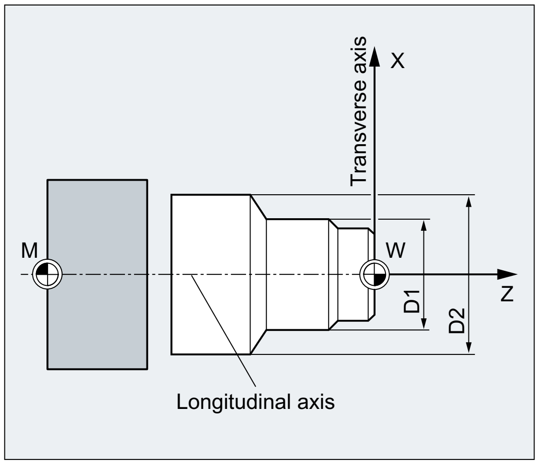
Both the machine and the workpiece zero are on the turning center. The settable offset on the X axis is therefore zero.



M	Machine zero
W	Workpiece zero
`Z`	Longitudinal axis
`X`	Transverse axis
`G54` to `G599` or `TRANS`	Call for the position of the workpiece zero

Transverse axis

Generally the dimensions for the transverse axis are diameter specifications (double path dimension compared to other axes):



The geometry axis that is to serve as transverse axis is defined in the machine data (→ machine manufacturer).

10 Motion commands

10.1 General information about the travel commands

Contour elements

The programmed workpiece contour can be made up of the following contour elements:

- Straight lines
- Circular arcs
- Helical curves (through overlaying of straight lines and circular arcs)

Travel commands

The following travel commands are available for the creation of these contour elements:

- Rapid traverse motion (`G0`)
- Linear interpolation (`G1`)
- Circular interpolation clockwise (`G2`)
- Circular interpolation counter-clockwise (`G3`)

The travel commands are modal.

Target positions

A motion block contains the target positions for the axes to be traversed (path axes, synchronized axes, positioning axes).

The target positions can be programmed in Cartesian coordinates or in polar coordinates.

Note

The axis address may only be programmed once per block.

Starting point - target point

The traversing motion is always for the last point reached to the programmed target position. This target position is then the starting position for the next travel command.

Workpiece contour

NOTICE
Tool operation undefined
Before machining, the workpiece must be positioned in such a way that the tool or workpiece cannot be damaged.

The motion blocks produce the workpiece contour when performed in succession:

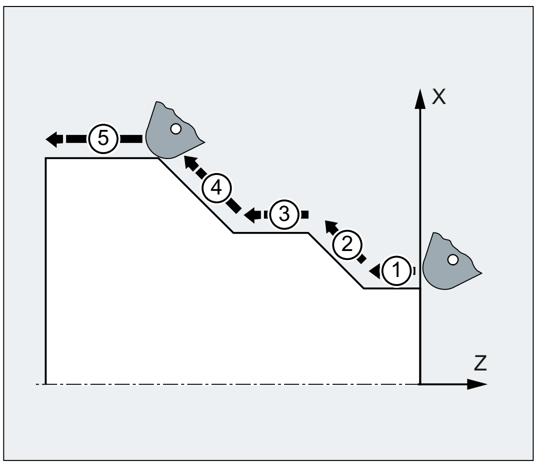


Figure 10-1 Motion blocks for turning

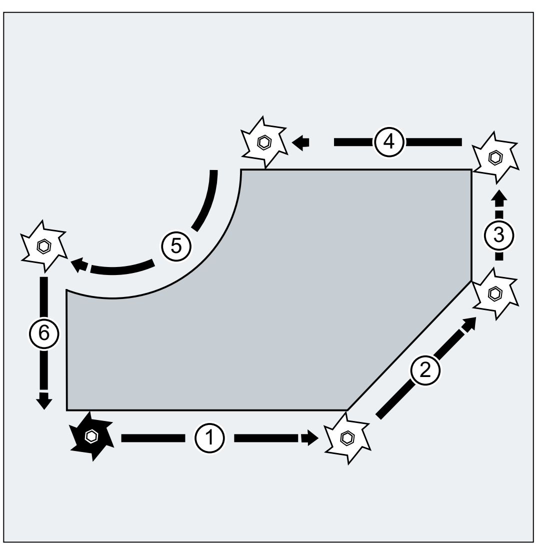


Figure 10-2 Motion blocks for milling

10.2 Travel commands with Cartesian coordinates (G0, G1, G2, G3, X..., Y..., Z...)

The position specified in the NC block with Cartesian coordinates can be approached with rapid traverse motion `G0`, linear interpolation `G1` or circular interpolation `G2` /`G3`.

Syntax

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

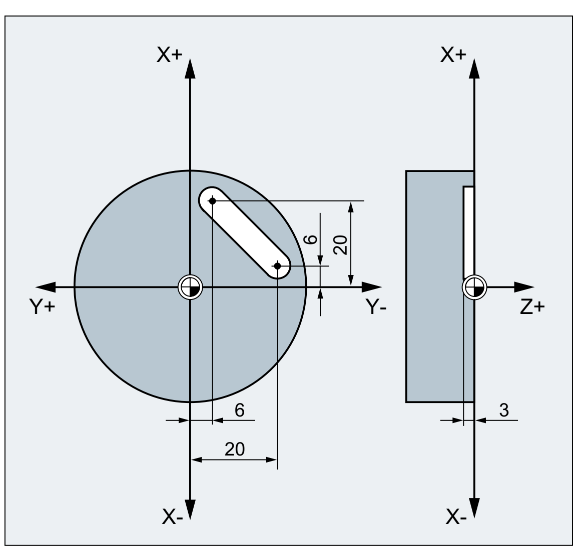
Meaning

`G0`:	Command for the activation of rapid traverse motion
`G1`:	Command for the activation of linear interpolation
`G2`:	Command for the activation of clockwise circular interpolation
`G3`:	Command for the activation of counter-clockwise circular interpolation
`X...`:	Cartesian coordinate of the target position in the X direction
`Y...`:	Cartesian coordinate of the target position in the Y direction
`Z...`:	Cartesian coordinate of the target position in the Z direction

Note

In addition to the coordinates of the target position `X...`, `Y...`, `Z...`, the circular interpolation `G2` / `G3` also requires further data (e.g. the circle center point coordinates; see "Overview (Page 195)").

Example



Program code	Comment
N10 G17 S400 M3	; Selection of the working plane, spindle clockwise
N20 G0 X40 Y-6 Z2	; Approach of the starting position specified with Cartesian coordinates in rapid traverse
N30 G1 Z-3 F40	; Activation of the linear interpolation, feed of the tool
N40 X12 Y-20	; Travel on an inclined line to an end position specified with Cartesian coordinates
N50 G0 Z100 M30	; Retraction in rapid traverse for tool change

10.3 Travel commands with polar coordinates

10.3.1 Reference point of the polar coordinates (G110, G111, G112)

The point from which the dimensioning starts is called the pole.

The pole can be specified in Cartesian or polar coordinates.

The reference point for the pole coordinates is clearly defined with the `G110` to `G112` commands. Absolute or incremental dimension inputs therefore have no effect.

Syntax

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

Meaning

<table>
<tr><td><code>G110 ...:</code></td><td colspan="3">With the command <code>G110</code>, the following pole coordinates refer to the last position reached.</td></tr>
<tr><td><code>G111 ...:</code></td><td colspan="3">With the command <code>G111</code>, the following pole coordinates refer to the zero point of the current workpiece coordinate system.</td></tr>
<tr><td><code>G112 ...:</code></td><td colspan="3">With the command <code>G112</code>, the following pole coordinates refer to the last valid pole.</td></tr>
<tr><td></td><td colspan="3">Note:
The commands <code>G110...G112</code> must be programmed in a separate NC block.</td></tr>
<tr><td><code>X… Y… Z…:</code></td><td colspan="3">Specification of the pole in Cartesian coordinates</td></tr>
<tr><td rowspan="4"><code>AP=… RP=…:</code></td><td colspan="3">Specification of the pole in polar coordinates</td></tr>
<tr><td rowspan="2"><code>AP=…:</code></td><td colspan="2">Polar angle
Angle between the polar radius and the horizontal axis of the working plane (e.g. X axis for <code>G17</code>). The positive direction of rotation runs counter-clockwise.</td></tr>
<tr><td>Range of values:</td><td>± 0…360°</td></tr>
<tr><td><code>RP=…:</code></td><td colspan="2">Polar radius
The specification is always in absolute positive values in [mm] or [inch].</td></tr>
</table>

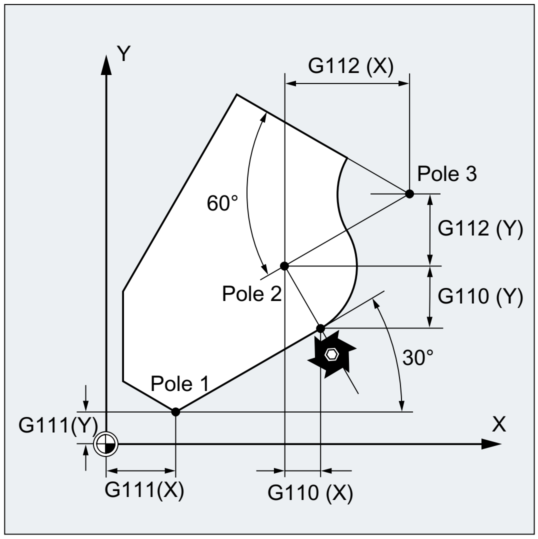
Note

It is possible to switch block-by-block in the NC program between polar and Cartesian dimensions. It is possible to return directly to the Cartesian system by using Cartesian coordinate identifiers (X..., Y..., Z...). The defined pole is moreover retained up to program end.

Note

If no pole has been specified, the zero point of the current workpiece coordinate system applies.

Example

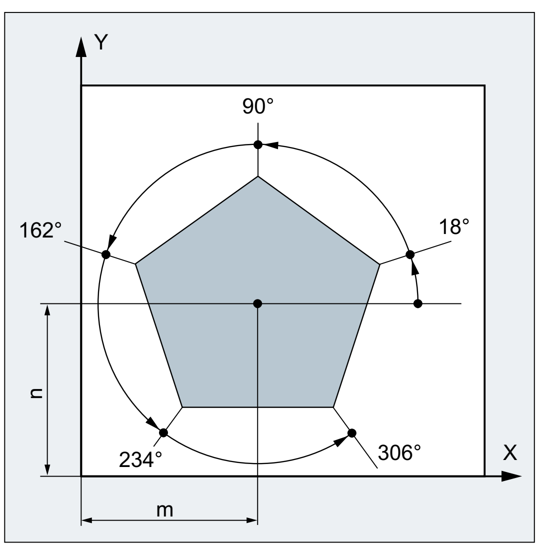


Poles 1 to 3 are defined as follows:

- Pole 1 with `G111 X… Y…`
- Pole 2 with `G110 X… Y…`
- Pole 3 with `G112 X… Y…`

10.3.2 Travel commands with polar coordinates (G0, G1, G2, G3, AP, RP)

Travel commands with polar coordinates are useful when the dimensions of a workpiece or part of the workpiece are measured from a central point and the dimensions are specified in angles and radii (e.g. for drilling patterns).



Syntax

```
G0/G1/G2/G3 AP=… RP=…
```

Meaning

`G0`:	Command for the activation of rapid traverse motion	
`G1`:	Command for the activation of linear interpolation	
`G2`:	Command for the activation of clockwise circular interpolation	
`G3`:	Command for the activation of counter-clockwise circular interpolation	
`AP`:	Polar angle Angle between the polar radius and the horizontal axis of the working plane (e.g. X axis for `G17`). The positive direction of rotation runs counter-clockwise.	
	Range of values:	± 0…360°
	The angle can be specified either incremental or absolute:	
	`AP=AC(...)`:	Absolute dimension input
	`AP=IC(...)`:	Incremental dimension input With incremental dimension input, the last programmed angle applies as reference.
	The polar angle remains stored until a new pole is defined or the working plane is changed.	
`RP`:	Polar radius The specification is **always in absolute positive values** in [mm] or [inch]. The polar radius remains stored until a new value is entered.	

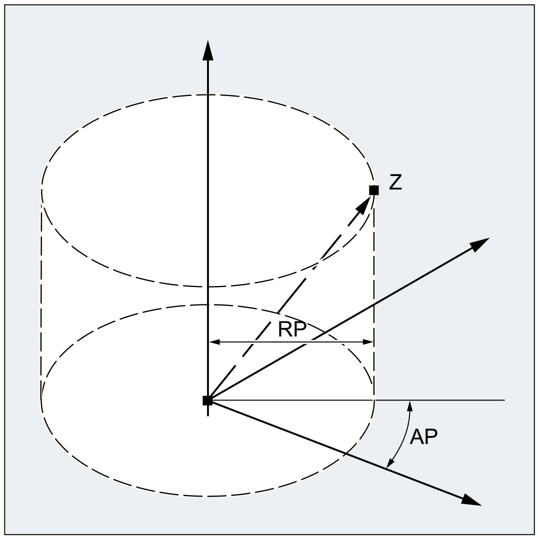
Note

The polar coordinates refer to the pole specified with `G110` ... `G112` and apply in the working plane selected with `G17` to `G19`.

Note

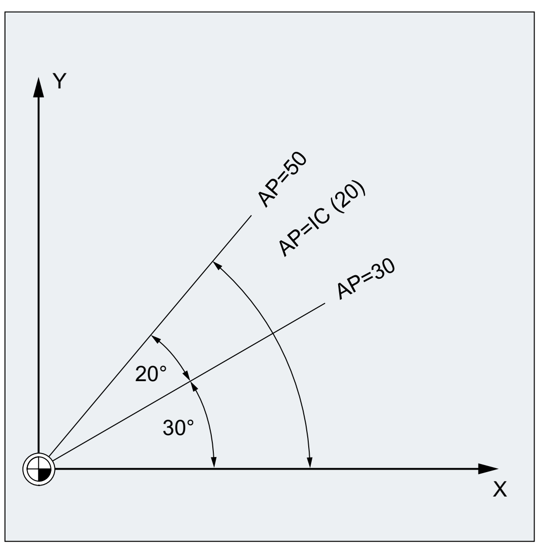
The 3rd geometry axis, which lies perpendicular to the working plane, can also be specified in Cartesian coordinates (see the following diagram). This enables spatial parameters to be programmed in cylindrical coordinates.

Example: `G17 G0 AP… RP… Z…`



Supplementary conditions

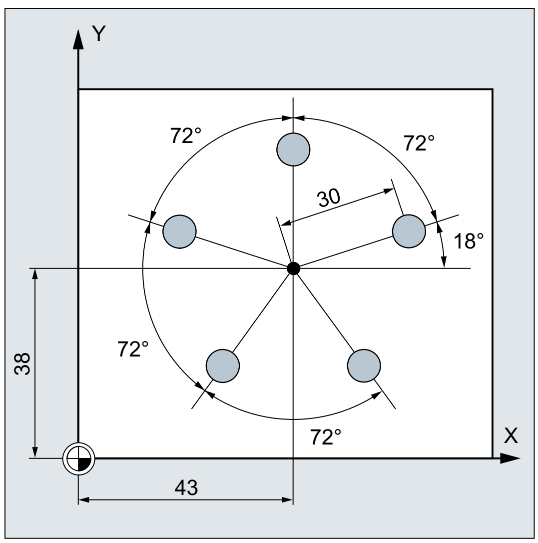
- No Cartesian coordinates such as interpolation parameters, axis addresses, etc. may be programmed for the selected working plane in NC blocks with polar end point coordinates.
- If a pole has not been defined with `G110` ... `G112`, then the zero point of the current workpiece coordinate system is automatically considered as the pole:



- Polar radius RP = 0
  The polar radius is calculated from the distance between the starting point vector in the pole plane and the active pole vector. The calculated polar radius is then saved as modal. This applies irrespective of the selected pole definition (`G110` ... `G112`). If both points have been programmed identically, this radius = 0 and alarm 14095 is generated.
- Only polar angle AP has been programmed
  If no polar radius RP has been programmed in the current block, but a polar angle AP, then when there is a difference between the current position and pole in the workpiece coordinates, this difference is used as polar radius and saved as modal. If the difference = 0, then the pole coordinates are specified again and the modal polar radius remains at zero.

Example

Creation of a drilling pattern



The positions of the holes are specified in polar coordinates.

Each hole is machined with the same production sequence:

Rough-drilling, drilling as dimensioned, reaming …

The machining sequence is stored in the subprogram.

Program code	Comment
N10 G17 G54	; Working plane X/Y, workpiece zero.
N20 G111 X43 Y38	; Specification of the pole.
N30 G0 RP=30 AP=18 Z5	; Approach starting point, specification in cylindrical coordinates.
N40 L10	; Subprogram call.
N50 G91 AP=72	; Approach next position in rapid traverse, polar angle in incremental dimensions, polar radius from block N30 remains saved and does not have to be specified.
N60 L10	; Subprogram call.
N70 AP=IC(72)	.
N80 L10	…
N90 AP=IC(72)	
N100 L10	…

Program code	Comment
N110 AP=IC(72)	
N120 L10	…
N130 G0 X300 Y200 Z100 M30	; Retract tool, end of program.

See also

Overview (Page 195)

10.4 Rapid traverse movements

10.4.1 Activating rapid traverse (G0)

The traversing of the path axes at rapid traversing velocity is activated with the G command G0.

Syntax

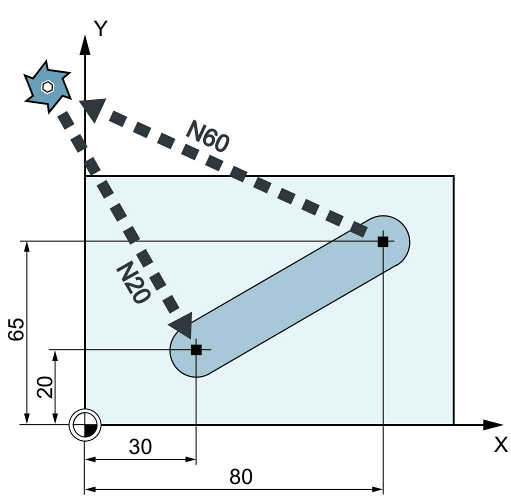
```
G0 X… Y… Z…
G0 RP=… AP=…
```

Meaning

G0:	Traversing the axis with rapid traverse velocity	
	Effective:	Modal
X... Y... Z...:	Specifying the end point in Cartesian coordinates	
RP=... AP=... :	Specifying the end point in polar coordinates	

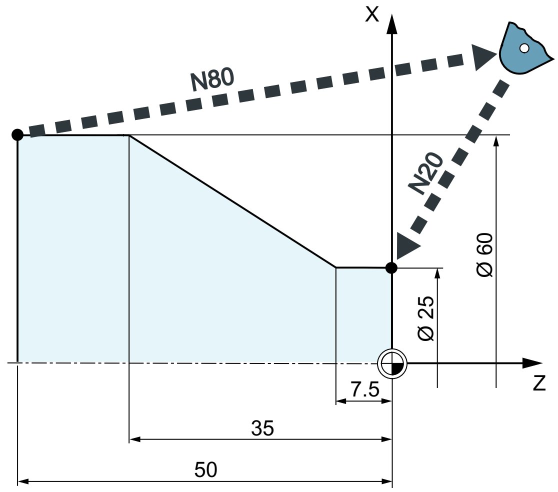
Examples

Example 1: Milling



Program code	Comment
N10 G90 S400 M3	; Absolute dimension input, spindle clockwise
N20 G0 X30 Y20 Z2	; Approach the starting position
N30 G1 Z-5 F1000	; Tool infeed
N40 X80 Y65	; Traversing along a straight line
N50 G0 Z2	
N60 G0 X-20 Y100 Z100 M30	; Retract tool, end of program

Example 2: Turning



Program code	Comment
`N10 G90 S400 M3`	`; Absolute dimension input, spindle clockwise`
`N20 G0 X25 Z5`	`; Approach the starting position`
`N30 G1 G94 Z0 F1000`	`; Tool infeed`
`N40 G95 Z-7.5 F0.2`	
`N50 X60 Z-35`	`; Traversing along a straight line`
`N60 Z-50`	
`N70 G0 X62`	
`N80 G0 X80 Z20 M30`	`; Retract tool, end of program`

10.4.2 Switch on/off linear interpolation for rapid traverse movements (RTLION, RTLIOF)

Independently of the default setting (MD20730 $MC_G0_LINEAR_MODE), the interpolation response for rapid traverse movements can also be set in the part program using the commands of the G group 55.

Syntax

```
RTLIOF
...
RTLION
```

Meaning

`RTLIOF`:	G command for switching off the linear interpolation ⇒ In the rapid traversing mode (G0), the **non-linear** interpolation is active. All of the path axes reach their end points independently of one another.	
	Effective:	Modal

`RTLION`:	G command for switching on the linear interpolation ⇒ In the rapid traversing mode (G0), the **linear** interpolation is active. All of the path axes reach their end points simultaneously.	
	Effective:	Modal

Note

Preconditions for RTLIOF

To ensure, with RTLIOF **non-linear** interpolation, the following conditions must be fulfilled:

- No transformation (TRAORI, TRANSMIT, etc.) active.
- G60 active (stop at the block end).
- No compressor active (COMPOF).
- No tool radius compensation active (G40).
- No contour handwheel selected.
- No nibbling active.

If one of these conditions is not met, linear interpolation is as with RTLION.

Example

Program code	**Comment**
	`; Linear interpolation is the default:`
	`; MD20730 $MC_GO_LINEAR_MODE == TRUE`
`...`	
`N30` **`RTLIOF`**	`; Switch off linear interpolation.`
`N40 G0 X0 Y10`	`; G0 blocks are traversed using non-linear inter-polation.`
`N50 G41 X20 Y20`	`; TRC active □ G0 blocks are traversed using lin-ear interpolation.`
`N60 G40 X30 Y30`	`; TRC not active □ G0 blocks are traversed using non-linear interpolation.`
`N70` **`RTLION`**	`; Switch on linear interpolation.`
`...`	

Further information

Reading the current interpolation behavior

The current interpolation behavior can be read via the system variables $AA_G0MODE.

10.4.3 Adapt tolerance factor for rapid traverse movements (STOLF)

The tolerance factor for rapid traverse movements (G0), which is preset using machine data (MD20560 $MC_G0_TOLERANCE_FACTOR), can be adapted in the part program by programming STOLF. In this case, the value in the machine data is not changed. After reset or end of part program, the tolerance factor set in the machine data becomes effective again.

Syntax

```
STOLF=<value>
```

Meaning

<table>
<tr><td rowspan="8"><code>STOLF</code>:</td><td colspan="4">Address to program a tolerance factor for rapid traverse movements</td></tr>
<tr><td>Applications:</td><td colspan="3">• Compressor functions COMPON, COMPCURV, COMPCAD and COMPSURF
• Smoothing functions G642 and G645
• Orientation smoothing OST
• Orientation smoothing ORISON
• Smoothing for path-relevant orientation ORIPATH</td></tr>
<tr><td>Preprocessing stop:</td><td colspan="3">No</td></tr>
<tr><td>Effective:</td><td colspan="3">Modal</td></tr>
<tr><td rowspan="4"><code><Value></code>:</td><td colspan="3">Tolerance factor
The tolerance factor can be both greater or less than 1.0. If the factor is equal to 1.0 (default value), then the same tolerances are active for rapid traverse movements as for non-rapid traverse movements. Normally, the tolerance factor is set to > 1.0.</td></tr>
<tr><td>Type:</td><td colspan="2">REAL</td></tr>
<tr><td rowspan="2">Range of values:</td><td>≥ 0:</td><td>Tolerance value</td></tr>
<tr><td>< 0:</td><td>The programmed tolerance value is deleted
⇒ The tolerance value parameterized in the machine data becomes effective again.</td></tr>
</table>

Example

Program code	Comment
COMPCAD G645 G1 F10000	; Compressor function COMPCAD
X... Y... Z...	; The machine and setting data apply here.
X... Y... Z...	
X... Y... Z...	
G0 X... Y... Z...	
G0 X... Y... Z...	; Machine data $MC_G0_TOLERANCE_FACTOR (e.g. =3) is effective here, i.e. a smoothing tolerance of: $MC_G0_TOLERANCE_FACTOR * $MA_COMPRESS_POS_TOL
CTOL=0.02	

Program code	Comment
STOLF=4	
G1 X... Y... Z...	; A contour tolerance of 0.02 mm is applied starting from here.
X... Y... Z...	
X... Y... Z...	
G0 X... Y... Z...	
X... Y... Z...	; From here, a G0 tolerance factor of 4 applies, i.e. a contour tolerance of 0.08 mm.

Additional information

Reading of the tolerance factor currently in effect

The tolerance factor for rapid traverse movements effective in the part program or in the current IPO block can be read using system variables.

- In synchronized actions or with preprocessing stop in the part program via system variable:

$AC_STOLF	Active G0 tolerance factor G0 tolerance factor, which was effective when processing the actual main run block.

- Without preprocessing stop in the part program via system variable:

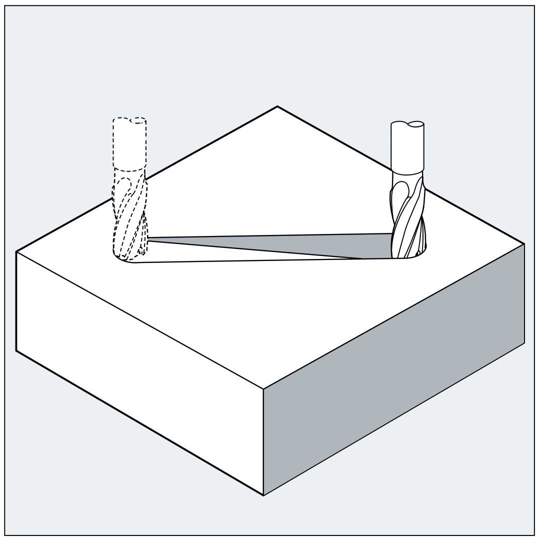
$P_STOLF	Programmed G0 tolerance factor

If no value with STOLF is programmed in the active part program, then these two system variables supply the value set using MD20560 $MC_G0_TOLERANCE_FACTOR.

If no rapid traverse (G0) is active in a block, then these system variables always supply a value of 1.

10.5 Linear interpolation (G1)

With `G1` the tool travels on paraxial, inclined or straight lines arbitrarily positioned in space. Linear interpolation permits machining of 3D surfaces, grooves, etc.



Syntax

```
G1 X… Y… Z … F…
G1 AP=… RP=… F…
```

Meaning

`G1`:	Linear interpolation with feedrate (linear interpolation)
`X... Y... Z...`:	End point in Cartesian coordinates
`AP=...`:	End point in polar coordinates, in this case polar angle
`RP=...`:	End point in polar coordinates, in this case polar radius
`F...`:	Feedrate speed in mm/min. The tool travels at feedrate F along a straight line from the current starting point to the programmed destination point. You can enter the destination point in Cartesian or polar coordinates. The workpiece is machined along this path. Example: `G1 G94 X100 Y20 Z30 A40 F100` The end point on X, Y, Z is approached at a feedrate of 100 mm/min; the rotary axis A is traversed as a synchronized axis, ensuring that all four movements are completed at the same time.

Note

`G1` is modal.

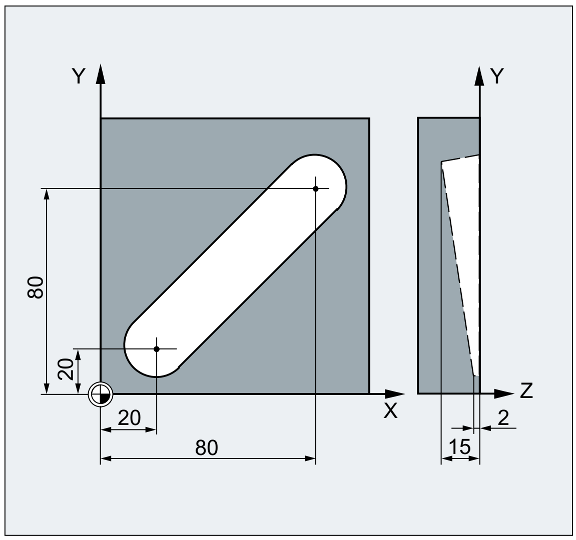
Spindle speed `S` and spindle direction `M3`/`M4` must be specified for the machining.

Axis groups, for which path feedrate `F` applies, can be defined with `FGROUP`. You will find more information in the "Path behavior" section.

Examples

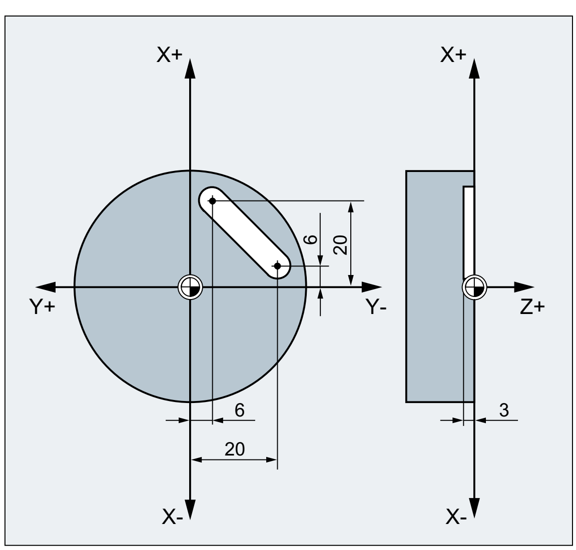
Example 1: Machining of a groove (milling)

The tool travels from the starting point to the end point in the X/Y direction. Infeed takes place simultaneously in the Z direction.



Program code	Comment
N10 G17 S400 M3	; Selection of the working plane, spindle clockwise
N20 G0 X20 Y20 Z2	; Approach the starting position
N30 G1 Z-2 F40	; Tool infeed
N40 X80 Y80 Z-15	; Travel on an inclined line
N50 G0 Z100 M30	; Retraction for tool change

Example 2: Machining of a groove (turning)



Program code	Comment
N10 G17 S400 M3	; Selection of the working plane, spindle clockwise
N20 G0 X40 Y-6 Z2	; Approach the starting position
N30 G1 Z-3 F40	; Tool infeed
N40 X12 Y-20	; Travel on an inclined line
N50 G0 Z100 M30	; Retraction for tool change

10.6 Circular interpolation

10.6.1 Overview

Circular interpolation enables the machining of full circles or arcs.

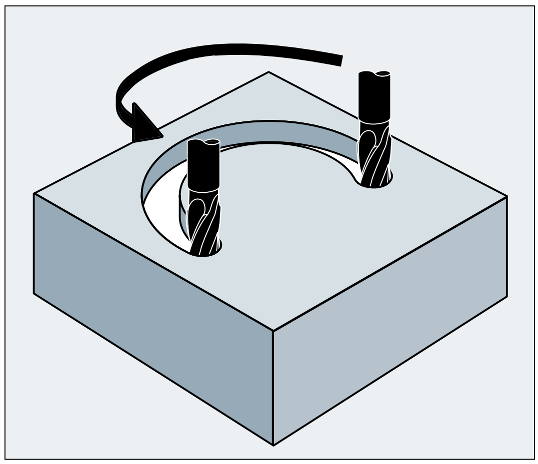


Figure 10-3 Application example: Milling a circular way

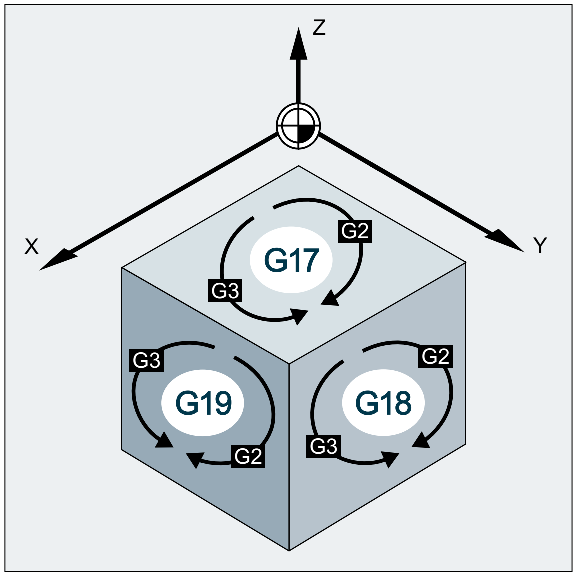
Programming options

The control system offers various options of programming circular movements. This allows the user to implement almost any type of drawing dimension directly.

- Circular interpolation with center point and end point (G2/G3, X... Y... Z..., I... J... K...) (Page 196)
- Circular interpolation with radius and end point (G2/G3, X... Y... Z..., CR) (Page 198)
- Circular interpolation with opening angle and end point / center point (G2/G3, X... Y... Z... / I... J... K..., AR) (Page 200)
- Circular interpolation with polar coordinates (G2/G3, AP, RP) (Page 202)
- Circular interpolation with intermediate point and end point (CIP, X... Y... Z..., I1... J1... K1...) (Page 204)
- Circular interpolation with tangential transition (CT, X... Y... Z...) (Page 207)

Plane for the circular interpolation

The control needs the working plane parameter (Page 152) to calculate the direction of rotation for the circle (`G2` is clockwise or `G3` is counter-clockwise).



Exception:

It is also possible to create circles outside the selected working plane (not with opening angle and helix parameters). In this case, the axis identifiers that the programmer specifies as circle end point determine the circle plane.

10.6.2 Circular interpolation with center point and end point (G2/G3, X... Y... Z..., I... J... K...)

Circular interpolation version, that uses the **center point** and **end point** of a circular contour element for the interpolation.

If the circle is programmed without an end point, the result is a full circle.

Syntax

```
G2/G3 X… Y… Z… I… J… K…
G2/G3 X… Y… Z… I=AC(…) J=AC(…) K=(AC…)
```

Meaning

`G2:`	Circular interpolation clockwise	
	Effective:	Modal
`G3:`	Circular interpolation counter-clockwise	
	Effective:	Modal

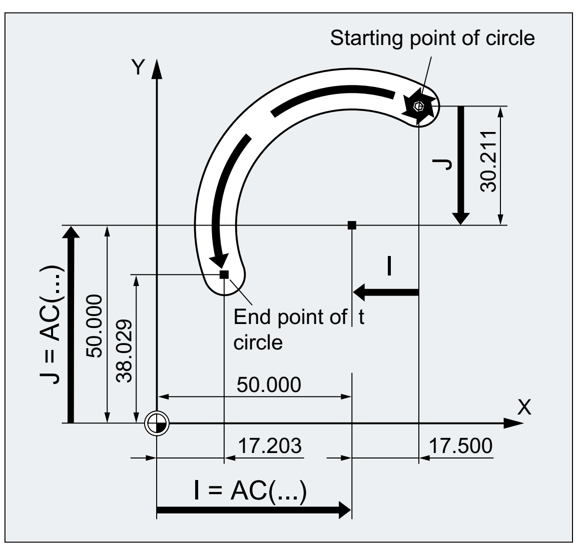
`X... Y... Z...:`	Circle end point in Cartesian coordinates. Depending on the currently valid dimensional notation setting `G90`/`G91` or `...=AC(...)` / `...=IC(...)`, the circle end point coordinates are interpreted either in the absolute dimension or in the incremental dimension.
`I... J... K...:`	Interpolation parameters to state the circle center point coordinates in the directions X, Y, Z Per default, the circle center point coordinates are stated in the incremental dimension in relation to the circle starting point. If the circle center point coordinates are stated in the absolute dimension in relation to workpiece zero, the interpolation parameters I, J, K must be programmed as follows: `I=AC(…) J=AC(…) K=AC(…)` **Note** An interpolation parameter with value 0 can be omitted, but the associated second parameter must always be specified.

Note

The default setting `G90`/`G91` absolute or incremental dimensions is only valid for the circle end point.

Examples

Example 1: Milling



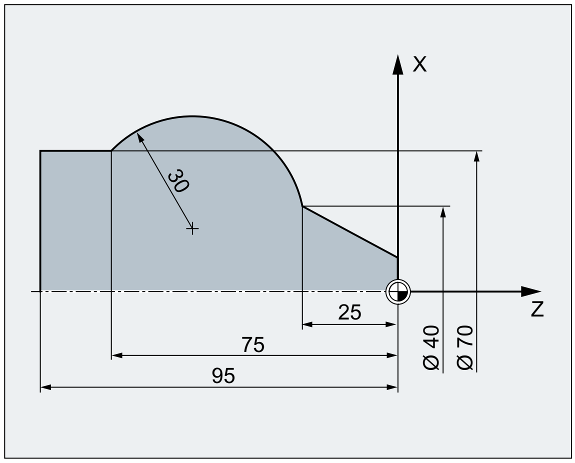
Center point data using incremental dimensions

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 I-17.5 J-30.211 F500
```

Center point data using absolute dimensions

```
N10 G0 X67.5 Y80.211
N20 G3 X17.203 Y38.029 I=AC(50) J=AC(50)
```

Example 2: Turning



Center point data using incremental dimensions

```
N120 G0 X12 Z0
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 I-3.335 K-29.25
N135 G1 Z-95
```

Center point data using absolute dimensions

```
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
```

10.6.3 Circular interpolation with radius and end point (G2/G3, X... Y... Z..., CR)

Circular interpolation version, that uses the **radius** and **end point** of a circular contour element for the interpolation.

Note

Full circles (traversing angle 360 °) can **not** be programmed with this version.

Syntax

```
G2/G3 X… Y… Z… CR=±...
```

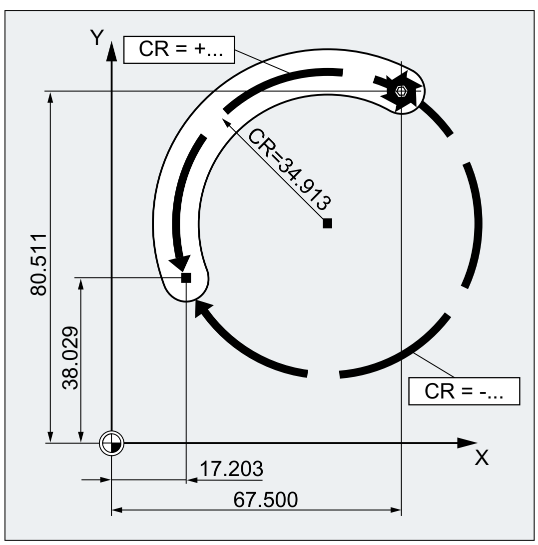
Meaning

`G2`:	Circular interpolation clockwise	
	Effective:	Modal
`G3`:	Circular interpolation counter-clockwise	
	Effective:	Modal

`X... Y... Z...:`	Circle end point in Cartesian coordinates. Depending on the currently valid dimensional notation setting `G90`/`G91` or `...=AC(...)` / `...=IC(...)`, the end point coordinates are interpreted either in the absolute dimension or in the incremental dimension.	
`CR=±...:`	Circle radius The sign indicates whether the traversing angle is to be greater than or less than 180°. A positive sign can be omitted.	
	`CR=+...:`	Traversing angle ≤ 180°
	`CR=-...:`	Traversing angle > 180°
	Note There is no practical limitation on the maximum size of the programmable radius.	

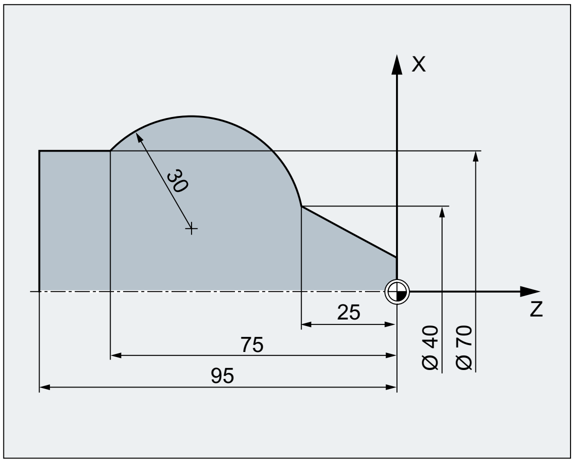
Examples

Example 1: Milling




```
Program code
N10 G0 X67.5 Y80.511
N20 G3 X17.203 Y38.029 CR=34.913 F500
...
```

Example 2: Turning



Program code
...
N125 G1 X40 Z-25 F0.2
N130 G3 X70 Z-75 CR=30
N135 G1 Z-95
...

10.6.4 Circular interpolation with opening angle and end point / center point (G2/G3, X... Y... Z... / I... J... K..., AR)

Circular interpolation version, that uses the **opening angle** and **center point** or **end point** of a circular contour element for the interpolation.

Note

Full circles (traversing angle 360 °) can **not** be programmed with this version.

Syntax

```
G2/G3 X… Y… Z… AR=...
G2/G3 I… J… K… AR=...
```

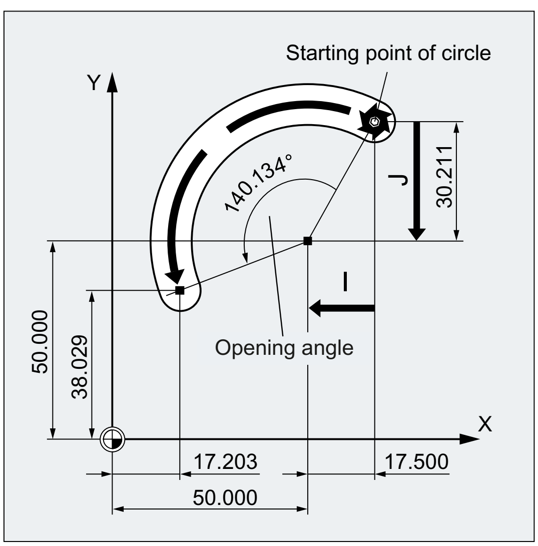
Meaning

G2:	Circular interpolation clockwise	
	Effective:	Modal
G3:	Circular interpolation counter-clockwise	
	Effective:	Modal

`X... Y... Z...:`	Circle end point in Cartesian coordinates. Depending on the currently valid dimensional notation setting `G90`/`G91` or `...=AC(...)` / `...=IC(...)`, the circle end point coordinates are interpreted either in the absolute dimension or in the incremental dimension.	
`I... J... K...:`	Interpolation parameters to state the circle center point coordinates in the directions X, Y, Z Per default, the circle center point coordinates are stated in the incremental dimension in relation to the circle starting point. If the circle center point coordinates are stated in the absolute dimension in relation to workpiece zero, the interpolation parameters I, J, K must be programmed as follows: `I=AC(…) J=AC(…) K=AC(…)` **Note** An interpolation parameter with value 0 can be omitted, but the associated second parameter must always be specified.	
`AR=...:`	Opening angle	
	Range of values:	0° ... 360°

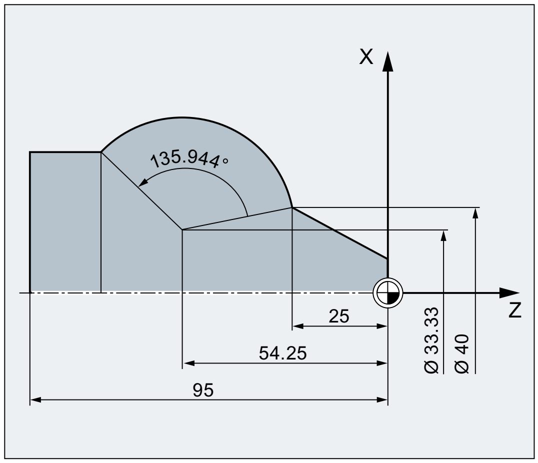
Examples

Example 1: Milling



Program code
`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`

Example 2: Turning



Program code
`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`

10.6.5 Circular interpolation with polar coordinates (G2/G3, AP, RP)

Circular interpolation version, that uses the **circle end point in polar coordinates** for the interpolation.

The following rule applies:

- The pole lies at the circle center.
- The polar radius corresponds to the circle radius.

Syntax

```
G2/G3 absolute pressure=... Recipe procedure=...
```

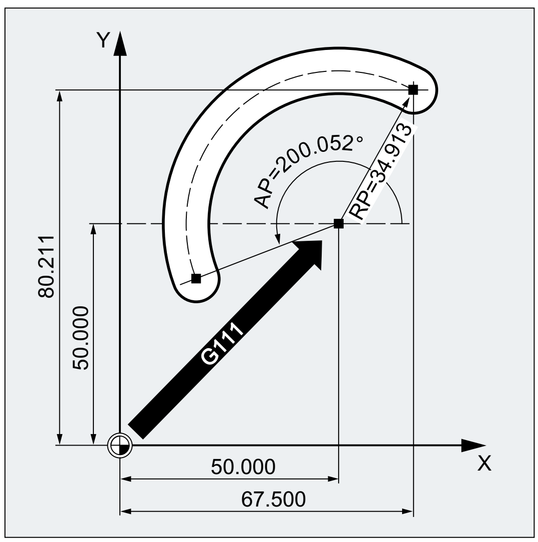
Meaning

`G2`:	Circular interpolation clockwise	
	Effective:	Modal
`G3`:	Circular interpolation counter-clockwise	
	Effective:	Modal

`Absolute pressure=... Recipe procedure=...:`	Circle end point in polar coordinates.	
	`Absolute pressure=...:`	Polar angle
	`Recipe procedure=...:`	Polar radius (≙ circle radius)

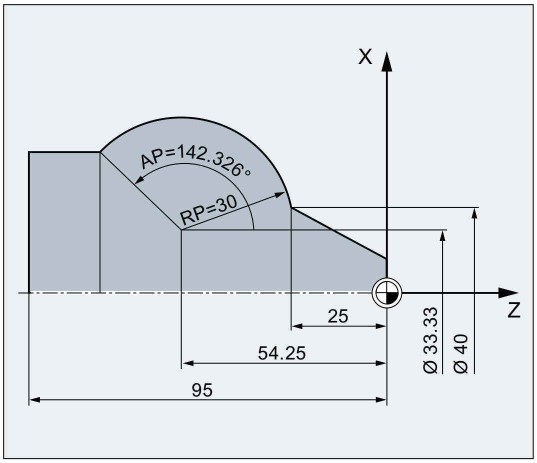
Examples

Example 1: Milling



Program code
`N10 G0 X67.5 Y80.211`
`N20 G111 X50 Y50`
`N30 G3 RP=34.913 AP=200.052 F500`

Example 2: Turning



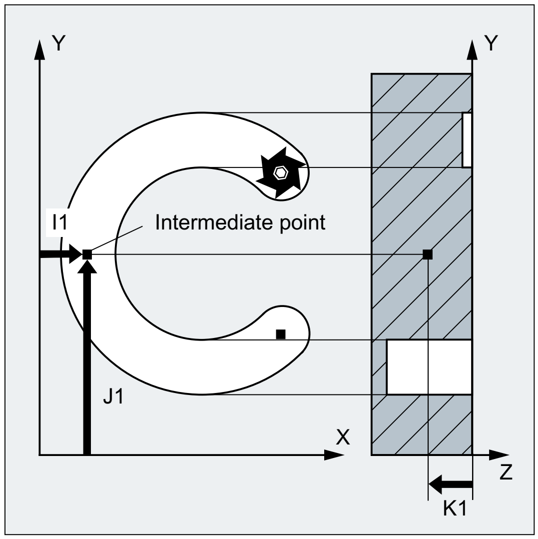
Program code
`N125 G1 X40 Z-25 F0.2`
`N130 G111 X33.33 Z-54.25`
`N135 G3 RP=30 AP=142.326`
`N140 G1 Z-95`

10.6.6 Circular interpolation with intermediate point and end point (CIP, X... Y... Z..., I1... J1... K1...)

The circular interpolation version programmed with the G command `CIP` allows the interpolation of arcs lying at an incline in the space.

The circular motion is described by the **intermediate point** and the **end point** of the circular contour.

The traversing direction is determined by the order of the starting point → intermediate point → end point.



Syntax

```
CIP X… Y… Z… I1=AC(…) J1=AC(…) K1=(AC…)
```

Meaning

`CIP`:	Circular interpolation through intermediate point	
	Effective:	Modal
`X... Y... Z...`:	Circle end point in Cartesian coordinates. Depending on the currently valid dimensional notation setting `G90`/`G91` or `...=AC(...)` / `...=IC(...)`, the circle end point coordinates are interpreted either in the absolute dimension or in the incremental dimension.	
`I1... J1... K1...`:	Interpolation parameters to state the circle intermediate point coordinates in the directions X, Y, Z Depending on the currently valid dimensional notation setting `G90`/`G91` or `...=AC(...)` / `...=IC(...)`, the circle intermediate point coordinates are interpreted either in the absolute dimension or in the incremental dimension. **Note** An interpolation parameter with value 0 can be omitted, but the associated second parameter must always be specified.	

Note

The default settings `G90`/`G91` (absolute or incremental dimensions) are only valid for the circle intermediate point and the circle end point.

With incremental dimensions `G91` or `...=IC(...)` active, the circle starting point is used as the reference for the intermediate point and the end point.

Note

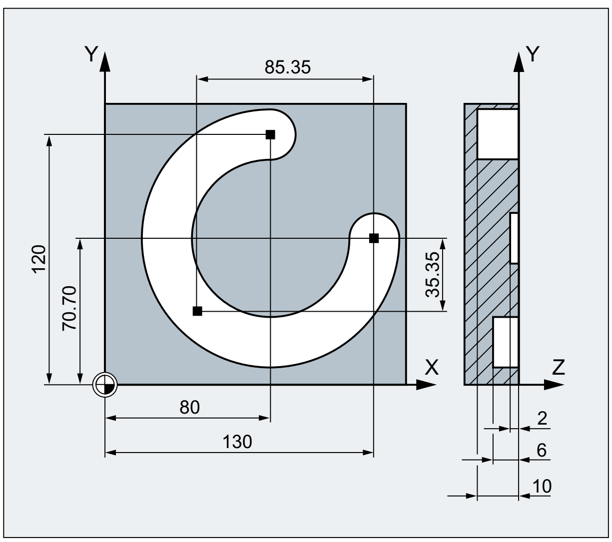
Turning technology

The diameter programming of the interpolation parameter for the transverse axis is not supported with `CIP` in the circular-path programming. The interpolation parameter for the transverse axis must therefore be programmed in the **radius**.

Examples

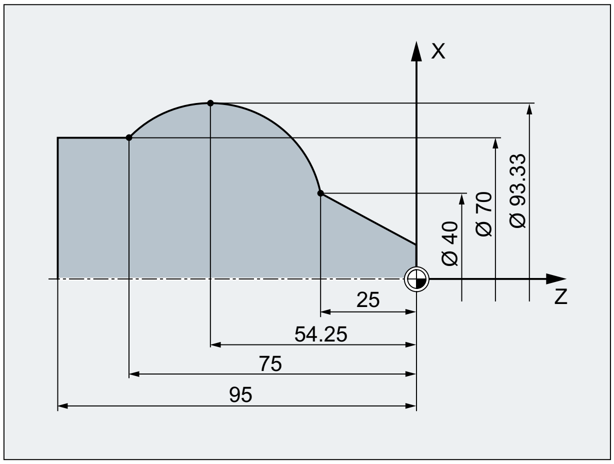
Example 1: Milling

In order to machine an inclined circular groove, a circle is described by specifying the intermediate point with three interpolation parameters, and the end point with three coordinates.



Program code	Comment
N10 G0 G90 X130 Y70.70 S800 M3	; Approach starting point.
N20 G17 G1 Z-2 F100	; Feed of the tool.
N30 CIP X80 Y120 Z-10 I1=IC(-85.35) J1=IC(-35.35) K1=-6	; Circle end point and intermediate point.
	; Coordinates for all three geometry axes.
N40 M30	; End of program

Example 2: Turning



Program code	Comment
`...`	
`N125 G1 G90 X40 Z-25 F0.2`	
`N130 CIP X70 Z-75 I1=IC(26.665) K1=IC(-29.25)`	`; Interpolation parameter I1 for transverse axis must be programmed in the radius.`
`; or`	
`; N130 CIP X70 Z-75 I1=46.665 K1=-54.25`	
`N135 G1 Z-95`	

10.6.7 Circular interpolation with tangential transition (CT, X... Y... Z...)

The circular interpolation version programmed with the G command `CT` allows the interpolation of arcs that connect tangentially to the previously programmed contour element.

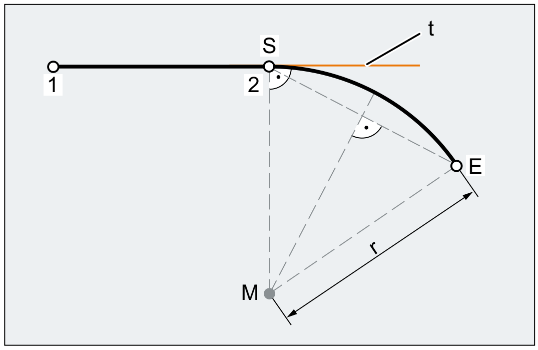
The circle is defined by the **start and end points**, and the **tangent direction at the start point**.

Note

Tangent direction at the start point.

The tangent direction in the starting point of a CT block is determined from the end tangent of the programmed contour of the last block with a traversing motion.

There can be any number of blocks without traversing information between this block and the current block.



S Start point
E End point
M Center of circle
r Circle radius
t End tangents of the programmed contour of the last block with a traversing movement.

Figure 10-4 Tangentially to the straight section 1-2 connecting circular path S-E

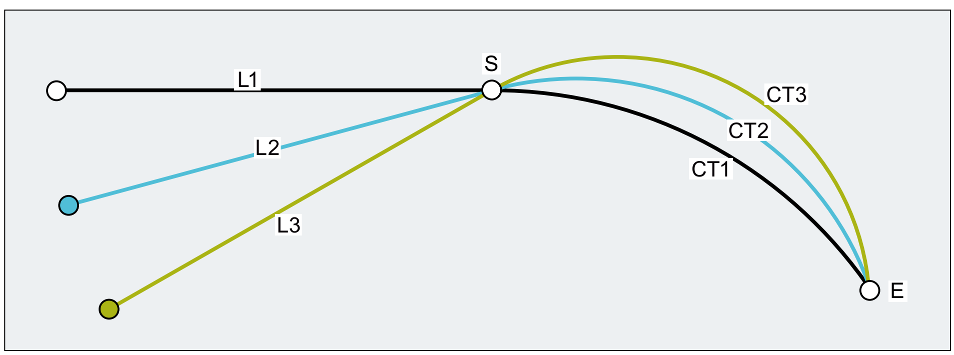


Figure 10-5 Tangentially connecting circular paths depend on the previous contour element

Syntax

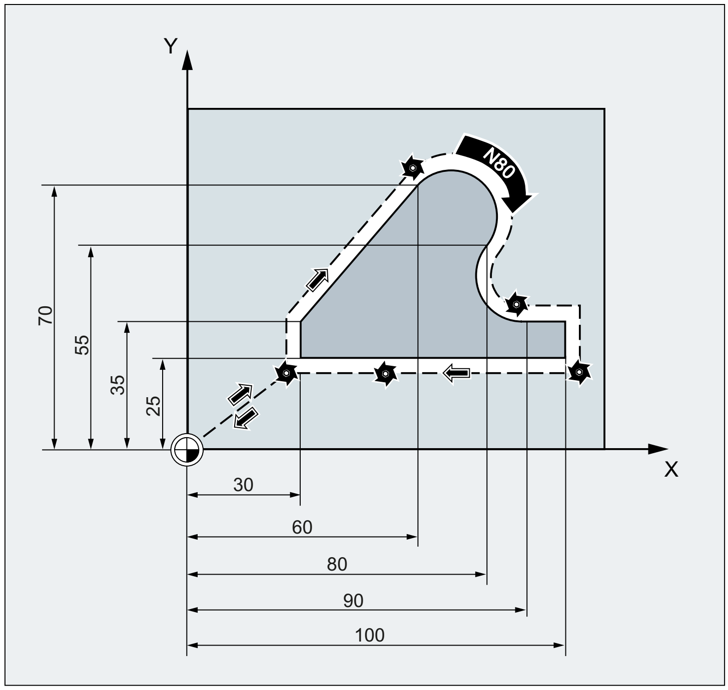
```
CT X… Y… Z…
```

Meaning

`CT`:	Circular interpolation with tangential transition	
	Effective:	Modal
`X... Y... Z...`:	Circle end point in Cartesian coordinates. Depending on the currently valid dimensional notation setting `G90`/`G91` or `...=AC(...)` / `...=IC(...)`, the circle end point coordinates are interpreted either in the absolute dimension or in the incremental dimension.	

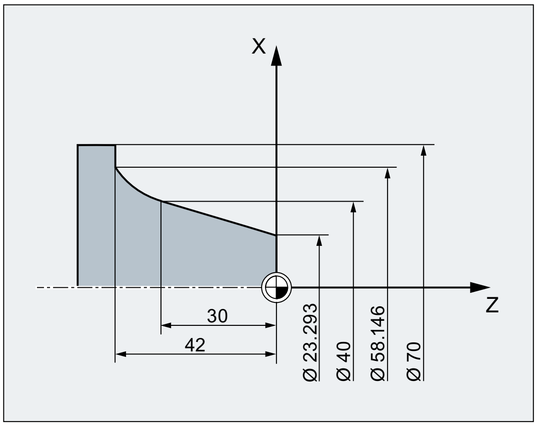
Examples

Example 1: Milling



Program code	Comment
N10 G0 Z100	
N20 G17 T1 M6	
N30 G0 X0 Y0 Z2 M3 S300 D1	
N40 Z-5 F1000	; Feed in tool.
N50 G41 X30 Y25 G1 F1000	; Switch on tool radius compensation.
N60 Y35	; Mill contour.
N70 X60 Y70	
N80 CT X80 Y55	**; Circular-path programming with tangential transition.**
N90 X90 Y35	
N100 G1 X100	
N110 Y25	
N120 X30	
N130 G0 G40 X0 Y0	; Switch off tool radius compensation.
N140 Z100	; Retract tool.
N140 M30	

Example 2: Turning



Program code	Comment
...	
N110 G1 X23.293 Z0 F10	
N115 X40 Z-30 F0.2	
N120 CT X58.146 Z-42	**; Circular-path programming with tangential transition.**
N125 G1 X70	
...	

Further information

Splines

In the case of splines, the tangential direction is defined by the straight line through the last two points. In the case of A and C splines with active ENAT or EAUTO, this direction is generally not the same as the direction at the end point of the spline.

The transition of B splines is always tangential, the tangent direction is defined as for A or C splines and active ETAN.

Frame change

If a frame change takes place between the block that defines the tangent and the CT block, the tangent is also subjected to this change.

Limit case

If the extension of the start tangent runs through the end point, a straight line is produced instead of a circle (limit case: circle with infinite radius). In this special case, TURN must either not be programmed or the value must be TURN=0.

Note

When the values tend towards this limit case, circles with an unlimited radius are produced and machining with TURN unequal to 0 is generally aborted with an alarm due to violation of the software limits.

Position of the circle plane

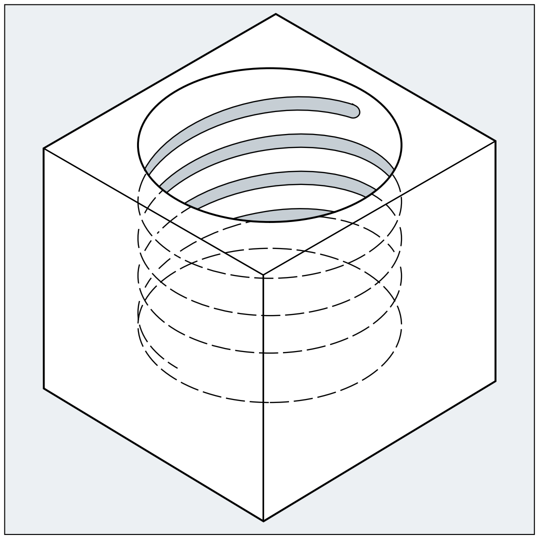
The position of the circle plane depends on the active plane (G17-G19).

If the tangent of the previous block does not lie in the active plane, its projection into the active plane is used.

If the start and end points do not have the same position components perpendicular to the active plane, a helix is produced instead of a circle.

10.7 Helical interpolation (G2/G3, TURN)

The helical interpolation enables, for example, the production of threads or oil grooves.



With helical interpolation, two motions are superimposed and executed in parallel:

- A plane circular motion on which
- A vertical linear motion is superimposed.

Syntax

```
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=
```

Meaning

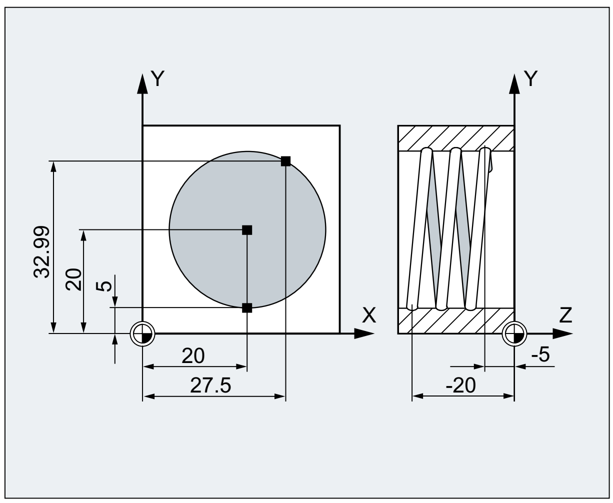
G2:	Travel on a circular path in clockwise direction
G3:	Travel on a circular path in counter-clockwise direction
X Y Z:	End point in Cartesian coordinates
I J K:	Circle center point in Cartesian coordinates
AR:	Opening angle
TURN=:	Number of additional circular passes in the range from 0 to 999
AP=:	Polar angle
RP=:	Polar radius

Note

`G2` and `G3` are modal.

The circular motion is performed in those axes that are defined by the specification of the working plane.

Example



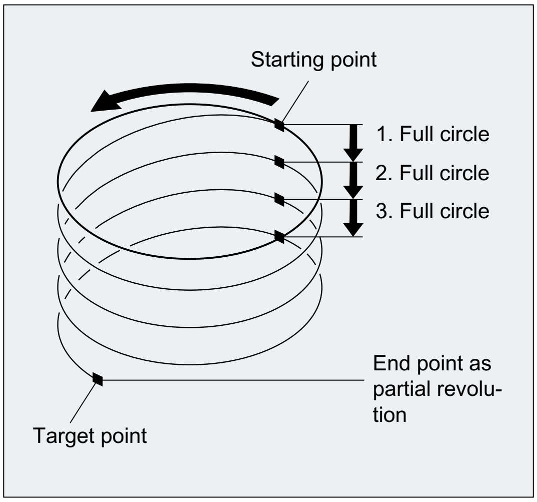
Program code	Comment
N10 G17 G0 X27.5 Y32.99 Z3	; Approach the starting position.
N20 G1 Z-5 F50	; Feed of the tool.
N30 G3 X20 Y5 Z-20 I=AC(20) J=AC(20) TURN=2	; Helix with the specifications: Execute two full circles after the starting position, then travel to end point.
N40 M30	; End of program

Additional information

Motion sequence

1. Approach starting point
2. Execute the full circles programmed with `TURN=`.
3. Approach circle end position, e.g. as part rotation.
4. Execute steps 2 and 3 across the infeed depth.

The pitch, with which the helix is to be machined is calculated from the number of full circles plus the programmed circle end position (executed across the infeed depth).



Programming the end point for helical interpolation

Please refer to circular interpolation for a detailed description of the interpolation parameters.

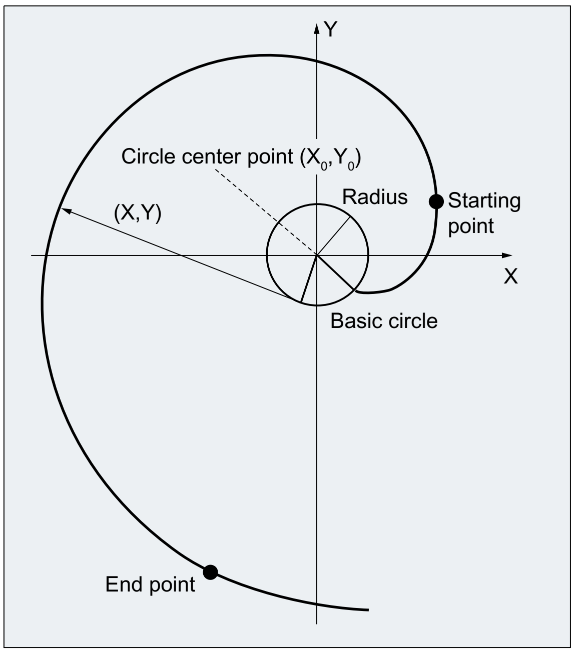
Programmed feedrate

For helical interpolation, it is advisable to specify a programmed feedrate override (`CFC`). `FGROUP` can be used to specify which axes are to be traversed with a programmed feedrate. For more information please refer to the Path behavior section.

10.8 Involute interpolation (INVCW, INVCCW)

The involute of the circle is a curve traced out from the end point on a "piece of string" unwinding from the curve.

Involute interpolation allows trajectories along an involute. It is executed in the plane in which the basic circle is defined and runs from the programmed starting point to the programmed end point.



The end point can be programmed in two ways:

1. Directly via Cartesian coordinates
2. Indirectly by specifying an opening angle (also refer to the programming of the opening angle for the circular-path programming)

If the starting point and the end point are in the plane of the basic circle, then, analogous to the helical interpolation for circles, there is a superimposition to a curve in space.

With additional specification of paths perpendicular to the active plane, an involute can be traversed in space (comparable to the helical interpolation for circles).

Syntax

```
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=...
```

Meaning

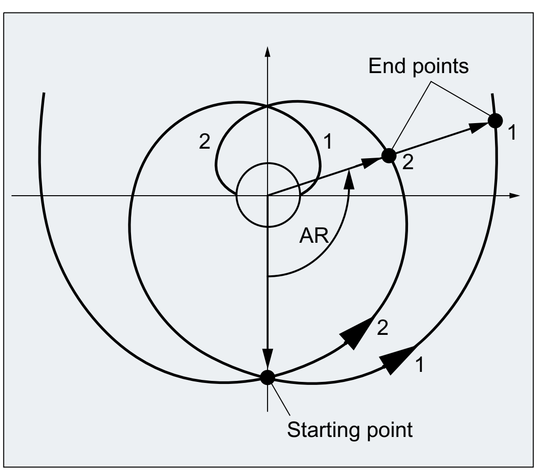
`INVCW`:	Command to travel on an involute in clockwise direction
`INVCCW`:	Command to travel on an involute in counter-clockwise direction

<table>
<tr><td><code>X... Y... Z...:</code></td><td colspan="2">Direct programming of the end point in Cartesian coordinates</td></tr>
<tr><td><code>I... J... K...:</code></td><td colspan="2">Interpolation parameters for the description of the center point of the basic circle in Cartesian coordinates
Note:
The coordinate specifications refer to the starting point of the involute.</td></tr>
<tr><td><code>CR=...:</code></td><td colspan="2">Radius of the basic circle</td></tr>
<tr><td rowspan="3"><code>AR=...:</code></td><td colspan="2">Indirect programming of the end point through specification of an opening angle (angle of rotation)
The origin of the opening angle is the line from the circle center point to the starting point.</td></tr>
<tr><td><code>AR</code> > 0:</td><td>The path of the involute moves away from the basic circle.</td></tr>
<tr><td><code>AR</code> < 0:</td><td>The path of the involute moves towards the basic circle.
For AR < 0, the maximum angle of rotation is restricted by the fact that the end point must always be outside the basic circle.</td></tr>
</table>

Indirect programming of the end point through specification of an opening angle

NOTICE
Opening angle undefined
With the indirect programming of the end point through specification of an opening angle `AR`, the sign of the angle must be taken into account, as a sign change would result in another involute and therefore another path.

This is demonstrated in the following example:

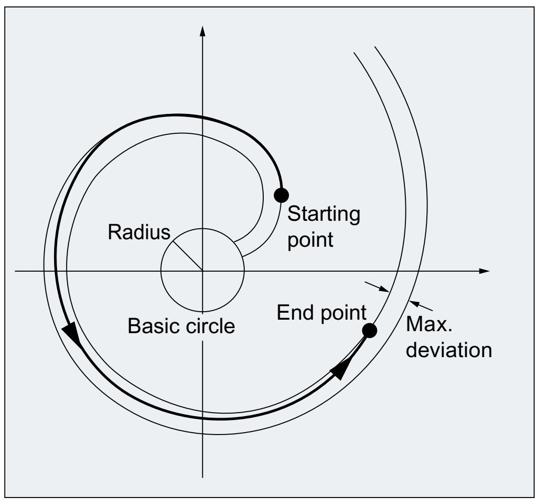


The specifications of the radius and center point of the basic circle as well as the starting point and direction of rotation (`INVCW`/`INVCCW`) are the same for involutes 1 and 2. The only difference is in the sign of the opening angle:

- With AR > 0, the path is on involute 1 and end point 1 is approached.
- With AR < 0, the path is on involute 2 and end point 2 is approached.

Supplementary conditions

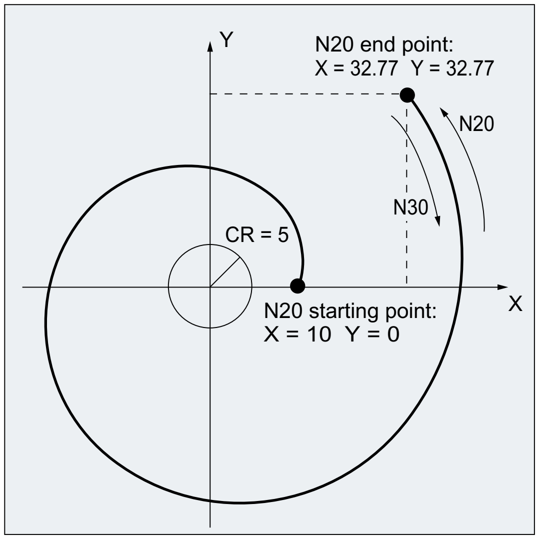
- Both the starting point and the end point must be outside the area of the basic circle of the involute (circle with radius CR around the center point specified by I, J, K). If this condition is not satisfied, an alarm is generated and the program processing is aborted.
- The two options for the programming of the end point (directly via Cartesian coordinates or indirectly via the specification of an opening angle) are mutually exclusive. Consequently, only one of the two programming options may be used in a block.
- If the programmed end point does not lie exactly on the involute defined by the starting point and basic circle, interpolation takes place between the two involutes defined by the starting and end points (see following figure).



The maximum deviation of the end point is determined by a machine data (→ machine manufacturer). If the deviation of the programmed end point in the radial direction is greater than that by the MD, then an alarm is generated and the program processing aborted.

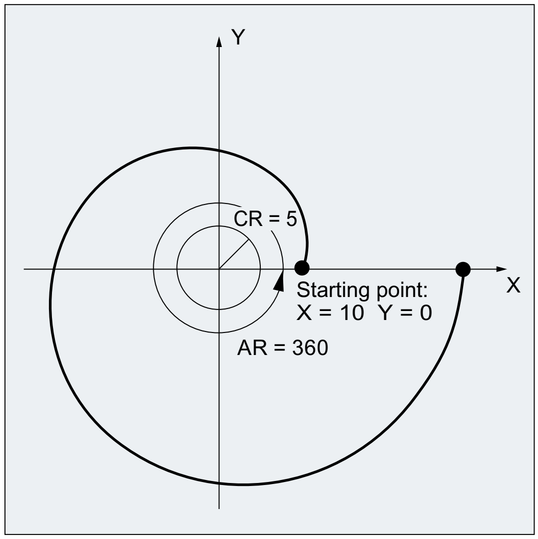
Examples

Example 1: Counter-clockwise involute from the starting point to the programmed end point and back again as clockwise involute



Program code	Comment
N10 G1 X10 Y0 F5000	; Approach the starting position.
N15 G17	; Selection of the X/Y plane as working plane.
N20 INVCCW X32.77 Y32.77 CR=5 I-10 J0	; Counter-clockwise involute, end point in Cartesian coordinates.
N30 INVCW X10 Y0 CR=5 I-32.77 J-32.77	; Clockwise involute, starting point is end point from N20, new end point is starting point from N20, new circle center point refers to a new starting point and is the same as the old circle center point.
...	

Example 2: Counter-clockwise involute with indirect programming of the end point through specification of an opening angle



Program code	Comment
N10 G1 X10 Y0 F5000	; Approach the starting position.
N15 G17	; Selection of the X/Y plane as working plane.
N20 INVCCW CR=5 I-10 J0 AR=360	; Counter-clockwise involute and away from the basic circle (as positive angle specification) with one full revolution (360 degrees).
...	

References

For more information about machine data and supplementary conditions that are relevant to involute interpolation, see:

Function Manual, Basic Functions; Various NC/PLC interface signals and functions (A2), Section: "Settings for involute interpolation"

10.9 Contour definitions

10.9.1 Contour definition programming

Function

The contour definition programming is used for the quick input of simple contours.

Programmable are contour definitions with one, two, three or more points with the transition elements chamfer or rounding, through specification of Cartesian coordinates and/or angles (ANG or ANG1 and ANG2).

Additional arbitrary NC addresses can be used, e.g. address letters for further axes (single axes or axis perpendicular to the machining plane), auxiliary function specifications, G commands, velocities, etc. in the blocks that describe contour definitions.

Note

Contour calculator

The contour definitions can be programmed easily with the aid of the contour calculator. This is a user interface tool that enables the programming and graphic display of simple and complex workpiece contours. The contours programmed via the contour calculator are transferred to the part program.

References:
Operating Manual

Parameterization

The identifiers for angle, radius and chamfer are defined via machine data:

MD10652 $MN_CONTOUR_DEF_ANGLE_NAME (name of the angle for contour definitions)

MD10654 $MN_RADIUS_NAME (name of the radius for contour definitions)

MD10656 $MN_CHAMFER_NAME (name of the chamfer for contour definitions)

Note

See machine manufacturer's specifications.

10.9.2 Contour definitions: One straight line

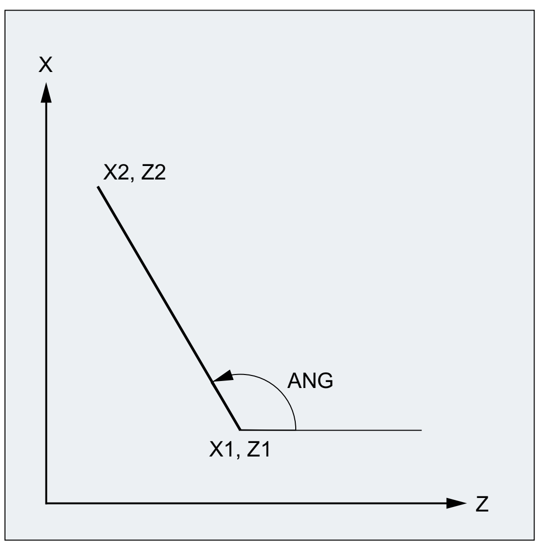
Note

In the following description it is assumed that:

- G18 is active (⇒ active working plane is the Z/X plane).
  (However, the programming of contour definitions is also possible without restrictions with G17 or G19.)
- The following identifiers have been defined for angle, radius and chamfer:
  – ANG (angle)
  – RND (radius)
  – CHR (chamfer)

The end point of the straight line is defined by the following specifications:

- Angle ANG
- **One** Cartesian end point coordinate (X2 or Z2)



ANG:	Angle of the straight line
X1, Z1:	Start coordinates
X2, Z2:	End point coordinates of the straight line

Syntax

```
X… ANG=…
Z… ANG=…
```

Meaning

`X...:`	End point coordinate in the X direction
`Z...:`	End point coordinate in the Z direction
`ANG:`	Identifier for angle programming The specified value (angle) refers to the abscissa of the active working plane (Z axis with `G18`).

Example

Program code	Comment
`N10 X5 Z70 F1000 G18`	`; Approach the starting position`
`N20 X88.8 ANG=110`	`; Straight line with angle specification`
`N30 ...`	

or

Program code	Comment
`N10 X5 Z70 F1000 G18`	`; Approach the starting position`
`N20 Z39.5 ANG=110`	`; Straight line with angle specification`
`N30 ...`	

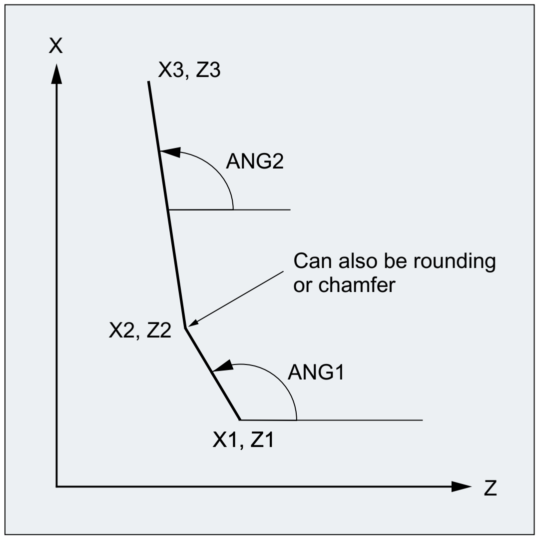
10.9.3 Contour definitions: Two straight lines

Note

In the following description it is assumed that:

- G18 is active (⇒ active working plane is the Z/X plane).
  (However, the programming of contour definitions is also possible without restrictions with G17 or G19.)
- The following identifiers have been defined for angle, radius and chamfer:
  - ANG (angle)
  - RND (radius)
  - CHR (chamfer)

The end point of the first straight line can be programmed by specifying the Cartesian coordinates or by specifying the angle of the two straight lines. The end point of the second straight line must always be programmed with Cartesian coordinates. The intersection of the two straight lines can be designed as a corner, curve or chamfer.



ANG1:	Angle of the first straight line
ANG2:	Angle of the second straight line
X1, Z1:	Start coordinates of the first straight line
X2, Z2:	End point coordinates of the first straight line or start coordinates of the second straight line
X3, Z3:	End point coordinates of the second straight line

Syntax

Programming of the end point of the first straight line by specifying the angle

- Corner as transition between the straight lines:

```
ANG=…
X… Z… ANG=…
```

- Rounding as transition between the straight lines:

```
ANG=… RND=...
X… Z… ANG=…
```

- Chamfer as transition between the straight lines:

```
ANG=… CHR=...
X… Z… ANG=…
```

Programming of the end point of the first straight line by specifying the coordinates

- Corner as transition between the straight lines:

```
X… Z…
```

```
X… Z…
```

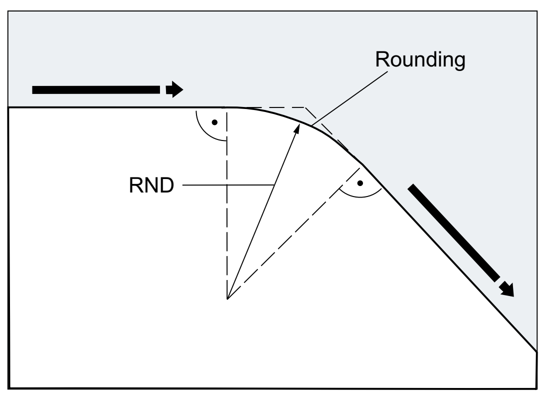
- Rounding as transition between the straight lines:

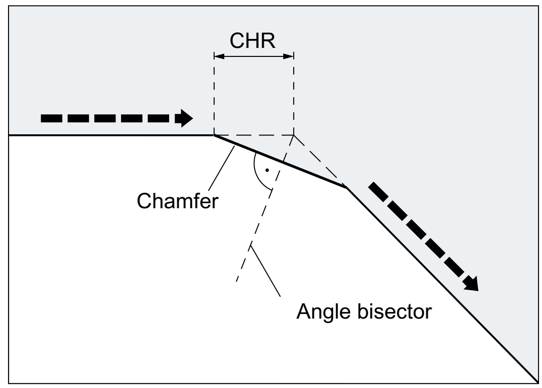
```
X… Z… RND=...
X… Z…
```

- Chamfer as transition between the straight lines:

```
X… Z… CHR=...
X… Z…
```

Meaning

`ANG=...:`	Identifier for angle programming The specified value (angle) refers to the abscissa of the active working plane (Z axis with `G18`).
`RND=...:`	Identifier for programming a rounding The specified value corresponds to the radius of the rounding: 

<table>
<tr><td><code>CHR=...:</code></td><td>Identifier for programming a chamfer
The specified value corresponds to the width of the chamfer in the direction of motion:

</td></tr>
<tr><td><code>X...:</code></td><td>Coordinates in the X direction</td></tr>
<tr><td><code>Z...:</code></td><td>Coordinates in the Z direction</td></tr>
</table>

Note

For further information on the programming of a chamfer or rounding, see "Chamfer, rounding (CHF, CHR, RND, RNDM, FRC, FRCM) (Page 256)".

Example

Program code	Comment
N10 X10 Z80 F1000 G18	; Approach the starting position.
N20 ANG=148.65 CHR=5.5	; Straight line with angle and chamfer specification.
N30 X85 Z40 ANG=100	; Straight line with angle and end point specification.
N40 ...	

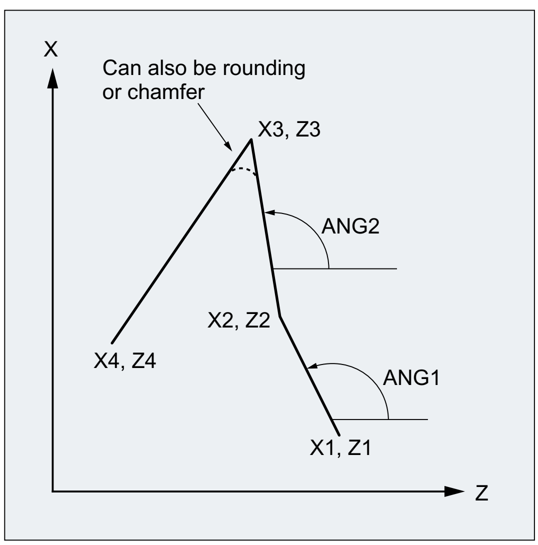
10.9.4 Contour definitions: Three straight lines

Note

In the following description it is assumed that:

- G18 is active (⇒ active working plane is the Z/X plane).
  (However, the programming of contour definitions is also possible without restrictions with G17 or G19.)
- The following identifiers have been defined for angle, radius and chamfer:
  – ANG (angle)
  – RND (radius)
  – CHR (chamfer)

The end point of the first straight line can be programmed by specifying the Cartesian coordinates or by specifying the angle of the two straight lines. The end point of the second and third straight lines must always be programmed with Cartesian coordinates. The intersection of the straight lines can be designed as a corner, a curve, or a chamfer.



ANG1:	Angle of the first straight line
ANG2:	Angle of the second straight line
X1, Z1:	Start coordinates of the first straight line
X2, Z2:	End point coordinates of the first straight line or start coordinates of the second straight line
X3, Z3:	End point coordinates of the second straight line or start coordinates of the third straight line
X4, Z4:	End point coordinates of the third straight line

Note

The programming described here for a three point contour definition can be expanded arbitrarily for contour definitions with more than three points.

Syntax

Programming of the end point of the first straight line by specifying the angle

- Corner as transition between the straight lines:

```
ANG=…
X… Z… ANG=…
X… Z…
```

- Rounding as transition between the straight lines:

```
ANG=… RND=...
X… Z… ANG=… RND=...
X… Z…
```

- Chamfer as transition between the straight lines:

```
ANG=… CHR=...
X… Z… ANG=… CHR=...
X… Z…
```

Programming of the end point of the first straight line by specifying the coordinates

- Corner as transition between the straight lines:

```
X… Z…
X… Z…
X… Z…
```

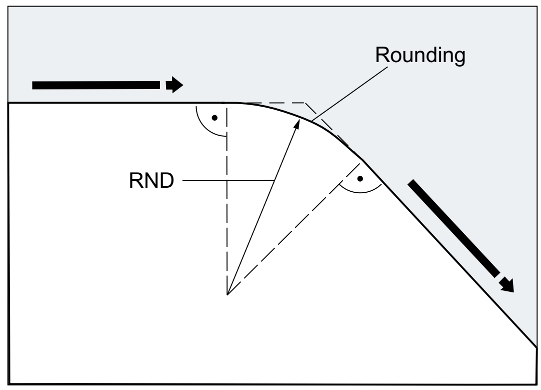
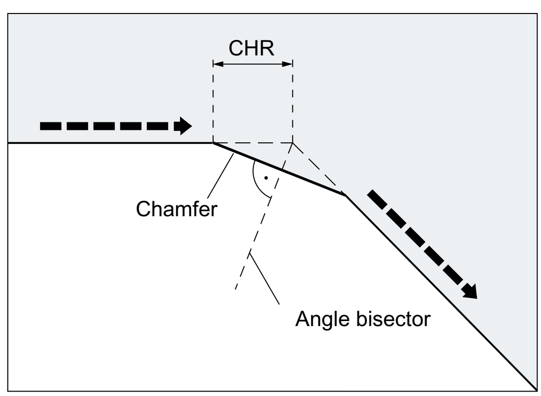
- Rounding as transition between the straight lines:

```
X… Z… RND=...
X… Z… RND=...
X… Z…
```

- Chamfer as transition between the straight lines:

```
X… Z… CHR=...
X… Z… CHR=...
X… Z…
```

Meaning

`ANG=...:`	Identifier for angle programming The specified value (angle) refers to the abscissa of the active working plane (Z axis with `G18`).
`RND=...:`	Identifier for programming a rounding The specified value corresponds to the radius of the rounding: 
`CHR=...:`	Identifier for programming a chamfer The specified value corresponds to the width of the chamfer in the direction of motion: 
`X...:`	Coordinates in the X direction
`Z...:`	Coordinates in the Z direction

Note

For further information on the programming of a chamfer or rounding, see " Chamfer, rounding (CHF, CHR, RND, RNDM, FRC, FRCM) (Page 256) ".

Example

Program code	Comment
N10 X10 Z100 F1000 G18	; Approach the starting position
N20 ANG=140 CHR=7.5	; Straight line with angle and chamfer specification.
N30 X80 Z70 ANG=95.824 RND=10	; Straight line to intermediate point with angle and chamfer specification.
N40 X70 Z50	; Straight line to end point.

10.9.5 Contour definitions: End point programming with angle

Function

If the address letter A appears in an NC block, either none, one or both of the axes in the active plane may also be programmed.

Number of programmed axes

- If **no axis** of the active plane has been programmed, then this is either the first or second block of a contour definition consisting of two blocks.
  If it is the second block of such a contour definition, then this means that the starting point and end point in the active plane are identical. The contour definition is then at best a motion perpendicular to the active plane.
- If **exactly one axis** of the active plane has been programmed, then this is either a single straight line whose end point can be clearly defined via the angle and programmed Cartesian coordinate or the second block of a contour definition consisting of two blocks. In the second case, the missing coordinate is set to the same as the last (modal) position reached.
- If **two axes** of the active plane have been programmed, then this is the second block of a contour definition consisting of two blocks. If the current block has not been preceded by a block with angle programming without programmed axes of the active plane, then this block is not permitted.

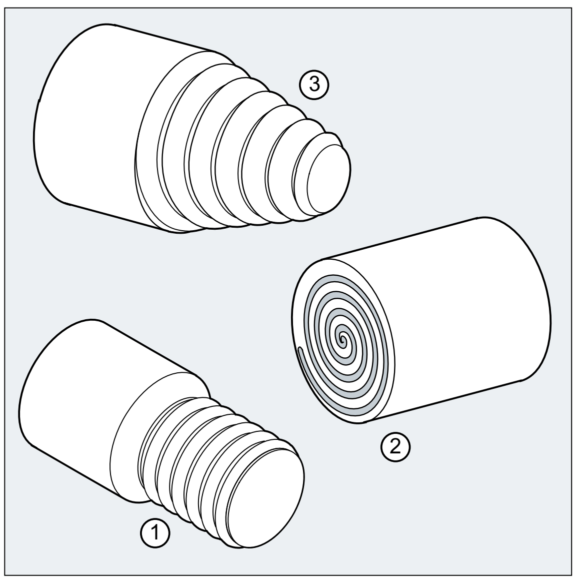
Angle A may only be programmed for linear or spline interpolation.

10.10 Thread cutting

10.10.1 Thread cutting with constant lead (G33, SF)

Threads with constant lead can be machined with `G33`:

- Cylindrical thread ①
- Face thread ②
- Taper thread ③

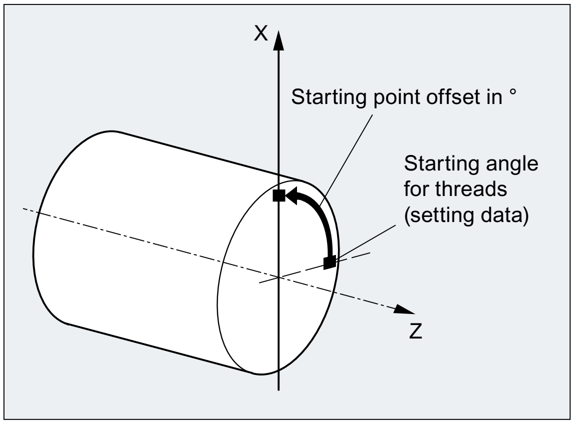


Note

Technical requirement for thread cutting with `G33` is a variable-speed spindle with position measuring system.

Multiple thread

Multiple thread (thread with offset cuts) can be machined by specifying a starting point offset. The programming is performed in the `G33` block at address `SF`.

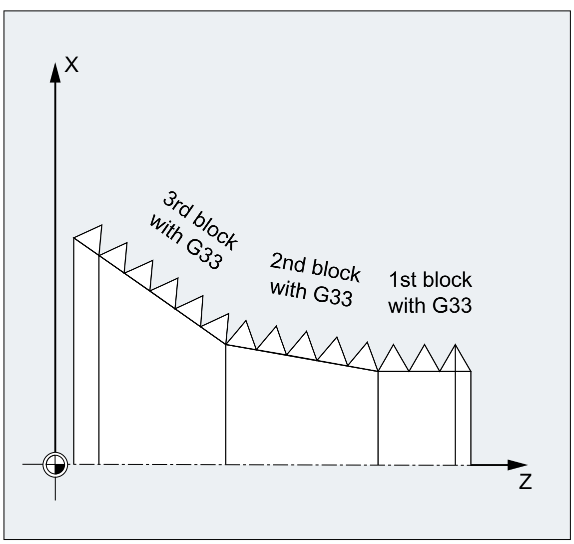


Note

If no starting point offset is specified, the "starting angle for thread" defined in the setting data is used.

Thread chain

A thread chain can be machined with several `G33` blocks programmed in succession:



Note

With continuous-path mode `G64`, the blocks are linked by the look-ahead velocity control in such a way that there are no velocity jumps.

Direction of rotation of the thread

The direction of rotation of the thread is determined by the direction of rotation of the spindle:

- Clockwise with `M3` produces a right-hand thread
- Counter-clockwise with `M4` produces a left-hand thread

Syntax

Cylinder thread:
```
G33 Z… K…
G33 Z… K… SF=…
```

Face thread:
```
G33 X… I…
G33 X… I… SF=…
```

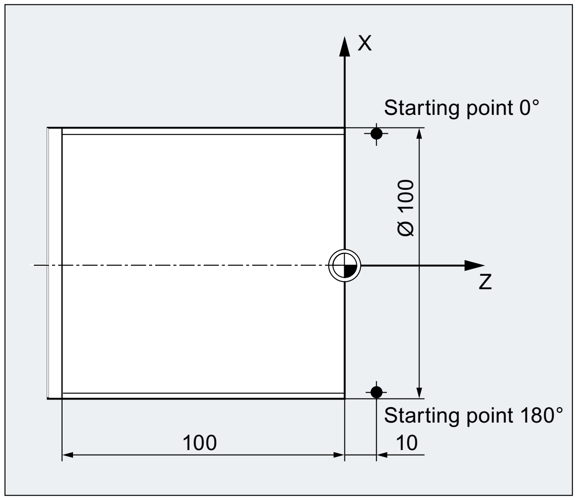
Tapered thread:
```
G33 X… Z… K…
G33 X… Z… K… SF=…
G33 X… Z… I…
G33 X… Z… I… SF=…
```

Meaning

<table>
<tr><td><code>G33</code>:</td><td colspan="2">Command for thread cutting with constant lead</td></tr>
<tr><td><code>X... Y... Z...:</code></td><td colspan="2">End point(s) in Cartesian coordinates</td></tr>
<tr><td><code>I...:</code></td><td colspan="2">Thread lead in X direction</td></tr>
<tr><td><code>J...:</code></td><td colspan="2">Thread lead in Y direction</td></tr>
<tr><td><code>K...:</code></td><td colspan="2">Thread lead in Z direction</td></tr>
<tr><td><code>Z</code>:</td><td colspan="2">Longitudinal axis</td></tr>
<tr><td><code>X</code>:</td><td colspan="2">Transverse axis</td></tr>
<tr><td><code>Z... K...:</code></td><td colspan="2">Thread length and lead for cylinder threads</td></tr>
<tr><td><code>X... I...:</code></td><td colspan="2">Thread diameter and thread lead for face threads</td></tr>
<tr><td rowspan="4"><code>I...</code> or <code>K...:</code></td><td colspan="2">Thread lead for tapered threads
The specification (<code>I...</code> or <code>K...</code>) refers to the taper angle:</td></tr>
<tr><td>< 45°:</td><td>The thread lead is specified with <code>K...</code> (thread lead in longitudinal direction).</td></tr>
<tr><td>> 45°:</td><td>The thread lead is specified with <code>I..</code> (thread lead in transverse direction).</td></tr>
<tr><td>= 45°:</td><td>The thread lead can be specified with <code>I...</code> or <code>K...</code>.</td></tr>
<tr><td rowspan="2"><code>SF=...:</code></td><td colspan="2">Starting point offset (only required for multiple threads)
The starting point offset is specified as an absolute angle position.</td></tr>
<tr><td>Range of values:</td><td>0.0000 to 359.999 degrees</td></tr>
</table>

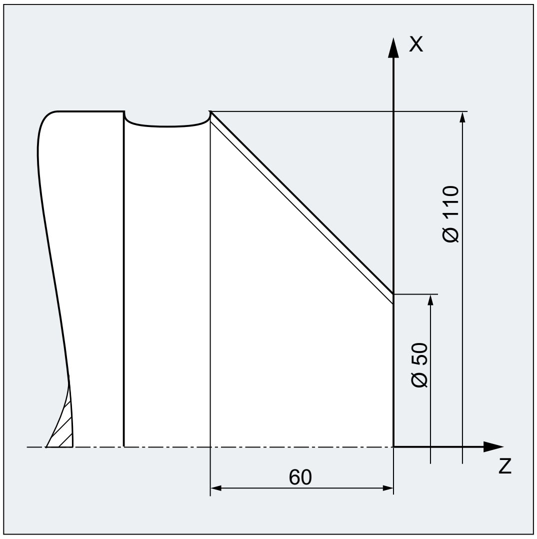
Examples

Example 1: Double cylinder thread with 180° starting point offset



Program code	Comment
N10 G1 G54 X99 Z10 S500 F100 M3	; Work offset, approach starting point, activate spindle.
N20 G33 Z-100 K4	; Cylinder thread: End point in Z.
N30 G0 X102	; Retraction to starting position.
N40 G0 Z10	
N50 G1 X99	
N60 G33 Z-100 K4 SF=180	; 2nd cut: Starting point offset 180°.
N70 G0 X110	; Retract tool.
N80 G0 Z10	
N90 M30	; End of program

Example 2: Tapered thread with angle less than 45°

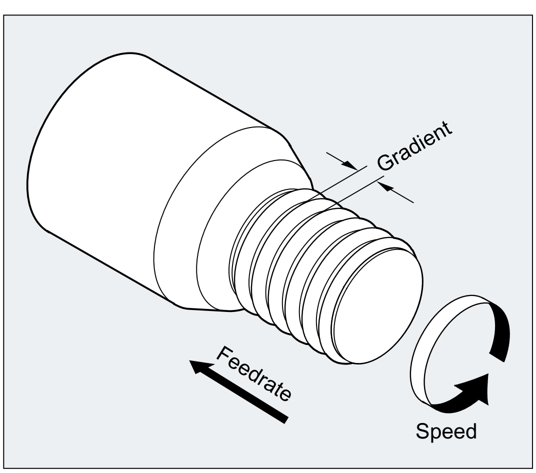


Program code	Comment
N10 G1 X50 Z0 S500 F100 M3	; Approach starting point, activate spindle.
N20 G33 X110 Z-60 K4	; Tapered thread: End point in X and Z, specification of thread lead with K... in Z direction (since angle < 45°).
N30 G0 Z0 M30	; Retraction, end of program.

Further information

Feedrate for thread cutting with G33

From the programmed spindle speed and the thread lead, the control calculates the required feedrate with which the turning tool is traversed over the thread length in the longitudinal and/or transverse direction. The feedrate `F` is not taken into account for `G33`, the limitation to maximum axis velocity (rapid traverse) is monitored by the control.



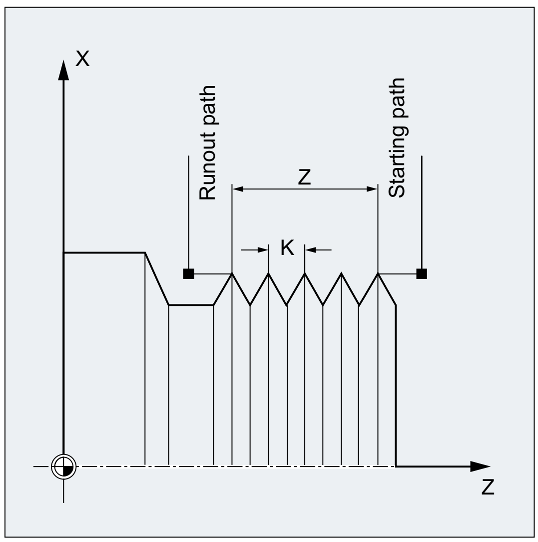
Cylinder thread

The cylinder thread is described by:

- Thread length
- Thread lead

The thread length is entered with one of the Cartesian coordinates X, Y or Z in absolute or incremental dimensions (for turning machines preferably in the Z direction). Allowance must also be made for the run-in and run-out paths, across which the feed is accelerated or decelerated.

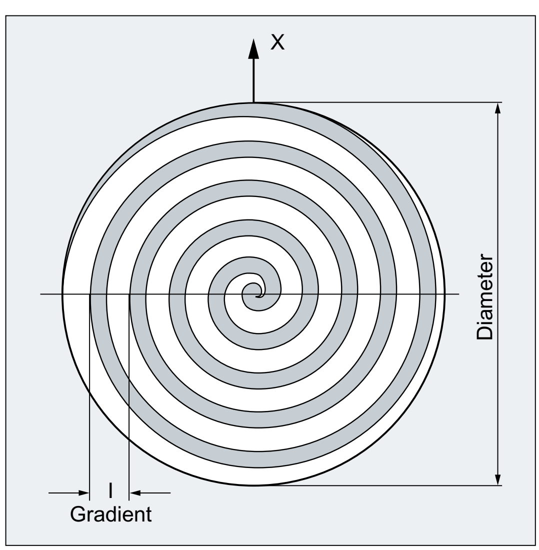
The thread lead is entered at addresses `I`, `J`, `K` (`K` is preferable for turning machines).



Face thread

The face thread is described by:

- Thread diameter (preferably in the X direction)
- Thread lead (preferably with `I`)



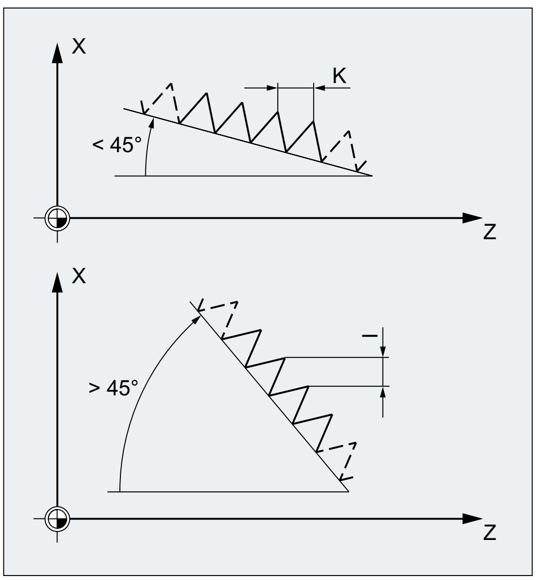
Tapered thread

The tapered thread is described by:

- End point in the longitudinal and transverse direction (taper contour)
- Thread lead

The taper contour is entered in Cartesian coordinates X, Y, Z in absolute or incremental dimensions - preferentially in the X and Z direction for machining on turning machines. Allowance must also be made for the run-in and run-out paths, across which the feed is accelerated or decelerated.

The specification of the lead depends on the taper angle (angle between the longitudinal axis and the outside of the taper):



10.10.2 Thread cutting with increasing or decreasing lead (G34, G35)

With the commands `G34` and `G35`, the `G33` functionality has been extended with the option of programming a change in the thread lead at address `F`. With `G34`, this results in a linear increase and with `G35` to a linear decrease of the thread lead. The commands `G34` and `G35` can therefore be used for the machining of self-tapping threads.

Syntax

Cylinder thread with increasing lead:
`G34 Z… K… F...`

Cylinder thread with decreasing lead:
`G35 Z… K… F...`

Face thread with increasing lead:
`G34 X… I… F...`

Face thread with decreasing lead:

```
G35 X… I… F...
```

Taper thread with increasing lead:

```
G34 X… Z… K… F...
G34 X… Z… I… F...
```

Taper thread with decreasing lead:

```
G35 X… Z… K… F...
G35 X… Z… I… F...
```

Meaning

`G34`:	Command for thread cutting with linear **in**creasing lead	
`G35`:	Command for thread cutting with linear **de**creasing lead	
`X... Y... Z...`:	End point(s) in Cartesian coordinates	
`I...`:	Thread lead in X direction	
`J...`:	Thread lead in Y direction	
`K...`:	Thread lead in Z direction	
`F...`:	Thread lead change If you already know the starting and final lead of a thread, you can calculate the thread lead change to be programmed according to the following equation: $F = \frac{k_e^2 - k_a^2}{2 * I_G}$ $[mm/rev^2]$ The meanings are as follows:	
	k_e:	Thread lead (thread lead of axis target point coordinate) [mm/rev]
	k_a:	Starting thread lead (programmed under I, J, or K) [mm/rev]
	I_G:	Thread length [mm]

Example

Program code	Comment
`N1608 M3 S10`	`; Spindle on.`
`N1609 G0 G64 Z40 X216`	`; Approach starting point.`
`N1610 G33 Z0 K100 SF=R14`	`; Thread cutting with constant lead (100 mm/rev).`
`N1611 G35 Z-200 K100 F17.045455`	`; Lead decrease: 17.0454 mm/rev2`
	`Lead at end of block: 50 mm/rev.`
`N1612 G33 Z-240 K50`	`; Traverse thread block without jerk.`
`N1613 G0 X218`	
`N1614 G0 Z40`	
`N1615 M17`	

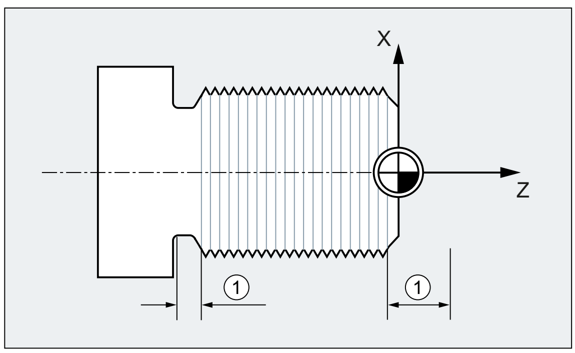
References

Function Manual, Basic Functions; Feedrates (V1), Section "Linear increasing/decreasing thread lead change with G34 and G35"

10.10.3 Programmed run-in and run-out path for G33, G34 and G35 (DITS, DITE)

The run-in and run-out path of the thread can be specified in the part program with the `DITS` and `DITE` addresses.

The thread axis is accelerated or braked along the specified path.



① Run-in/run-out path, depending on the machining direction

Short run-in path

Due to the collar on the thread runin, little room is left for the tool start ramp.
This must therefore be specified shorter via `DITS`.

Short run-out path

Because of the shoulder at the thread run-out, there is not much room for the tool braking ramp, introducing a risk of collision between the workpiece and the tool cutting edge. The deceleration ramp can be specified shorter using `DITE`. Due to the inertia of the mechanical system, however, a collision can still occur.

Remedy: Program a shorter thread, reduce the spindle speed.

Note

`DITE` acts at the end of the thread as a rounding clearance. This achieves a smooth change in the axis motion.

Effects

The programmed run-in and run-out path only increases the rate of acceleration on the path. If one of the two paths is set larger than the thread axis needs with active acceleration, the thread axis is accelerated or decelerated with maximum acceleration.

Syntax

```
DITS=<Value> DITE=<Value>
```

Meaning

`DITS`:	Define thread run-in path
`DITE`:	Define thread run-out path
`<value>`:	Only paths, and not positions, are programmed with `DITS` and `DITE`. The programmed run-in/run-out path is handled according to the current dimension setting (inches, metric).

Example

Program code	Comment
`...`	
`N40 G90 G0 Z100 X10 SOFT M3 S500`	
`N50 G33 Z50 K5 SF=180 DITS=1 DITE=3`	`; Start of smoothing with Z=53.`
`N60 G0 X20`	

Further information

SD42010 $SC_THREAD_RAMP_DISP

When a block containing `DITS` and/or `DITE` is inserted in the main run, the programmed run-in/run-out path is transferred into the setting data SD42010 $SC_THREAD_RAMP_DISP:

- SD42010 $SC_THREAD_RAMP_DISP[**0**] = programmed value of `DITS`
- SD42010 $SC_THREAD_RAMP_DISP[**1**] = programmed value of `DITE`

If no run-in/run-out path is programmed before or in the first thread block, the current value of the setting data is used.

Behavior following channel / mode group / program end reset

SD 42010 values which have been overwritten by `DITS` and/or `DITE` remain active even following a channel / mode group / program end reset.

Behavior following warm start

In case of a warm start, the setting data is reset to the values which were active before overwriting by `DITS` and/or `DITE` (standard behavior).

If, however, the values programmed with `DITS` and `DITE` shall also be active following a warm restart, the setting data SD42010 $SC_THREAD_RAMP_DISP must be listed in the machine data MD10710 $MN_PROG_SD_RESET_SAVE_TAB:

MD10710 $MN_PROG_SD_RESET_SAVE_TAB[<n>] = 42010

Behavior if the run-in and/or run-out path is very short

If the run-in and/or run-out path is very short, the acceleration of the thread axis is higher than the configured value. This causes an acceleration overload on the axis.

Alarm 22280 "Programmed run-in path too short" is then issued for the thread run-in (with the appropriate configuration in MD11411 $MN_ENABLE_ALARM_MASK). The alarm is purely for information and has no effect on part program execution.

10.10.4 Fast retraction during thread cutting (LFON, LFOF, DILF, ALF, LFTXT, LFWP, LFPOS, POLF, POLFMASK, POLFMLIN)

The "Rapid retraction during thread cutting (G33)" function can be used to interrupt thread cutting without causing irreparable damage in the following circumstances:

- NC stop via NC/PLC interface signal: DB21, ... DBX7.3 (NC stop)
- Alarms that implicitly trigger NC stop
- Switching of a rapid input
  References
  Programming Manual, Job Planning; Section "Rapid retraction from the contour"

The retraction motion can be programmed via:

- Retraction path and retraction direction (relative)
- Retraction position (absolute)

Note

NC stop signals

The following NC stop signals do not trigger a rapid retraction during thread cutting:

- DB21, ... DBX3.4 (NC stop axes plus spindles)
- DB21, ... DBX7.2 (NC stop at the block limit)

Tapping

The "Rapid retraction" function **cannot** be used with **tapping** (`G331`/`G332`).

Syntax

Enable rapid retraction, retraction motion via retraction path and retraction direction:
```
G33 ... LFON DILF=<value> LFTXT/LFWP ALF=<value>
```

Enable rapid retraction, retraction motion via retraction position:

```
POLF[<axis identifier>]=<value> LFPOS
POLFMASK/POLFMLIN(<axis 1 name>,<axis 2 name>, etc.)
G33 ... LFON
```

Disable rapid retraction during thread cutting:
```
LFOF
```

Meaning

`LFON`:	Enable rapid retraction during thread cutting (`G33`)
`LFOF`:	Disable rapid retraction during thread cutting (`G33`)
`DILF=`:	Define length of retraction path
	The value preset during MD configuration (MD21200 $MC_LIFTFAST_DIST) can be modified in the part program by programming `DILF`. **Note:** The configured MD value is always active following NC-RESET.

`LFTXT` `LFWP`:	The retraction direction is controlled in conjunction with `ALF` with G commands `LFTXT` and `LFWP`. `LFTXT`: The plane in which the retraction motion is executed is calculated from the path tangent and the tool direction (default setting). `LFWP`: The plane in which the retraction motion is executed is the active working plane.
`ALF=`:	The direction is programmed in discrete degree increments with `ALF` in the plane of the retraction motion. With `LFTXT`, retraction in the tool direction is defined for `ALF=1`. For `LFWP`, the direction in the working/machining plane has the following assignment: • `G17` (X/Y plane) `ALF=1` ; Retraction in the X direction `ALF=3` ; Retraction in the Y direction • `G18` (Z/X plane) `ALF=1` ; Retraction in the Z direction `ALF=3` ; Retraction in the X direction • `G19` (Y/Z plane) `ALF=1` ; Retraction in the Y direction `ALF=3` ; Retraction in the Z direction **References:** Programming options with `ALF` are also described in "Traverse direction for rapid retraction from the contour" in the Programming Manual, Job Planning.
`LFPOS`:	Retraction of the axis declared with `POLFMASK` or `POLFMLIN` to the absolute axis position programmed with `POLF`.
`POLFMASK`:	Release of axes `(<axis 1 name>,<axis 1 name>, etc.)` for independent retraction to absolute position.
`POLFMLIN`:	Release of axes for retraction to absolute position in linear relation **Note:** Depending on the dynamic response of all the axes involved, the linear relation cannot always be established before the lift position is reached.
`POLF[]`:	Define absolute retraction position for the geometry axis or machine axis in the index Effective: Modal `=<value>`: In the case of geometry axes, the assigned value is interpreted as a position in the workpiece coordinate system. In the case of machine axes, it is interpreted as a position in the machine coordinate system. The values assigned can also be programmed as incremental dimensions: `=IC<value>`
`<axis identifier>`:	Identifier of a geometry axis or machine axis.

Note

`LFON` or `LFOF` can always be programmed, but the evaluation is performed exclusively during thread cutting (`G33`).

Note

`POLF` with `POLFMASK`/`POLFMLIN` are not restricted to thread cutting applications.

Examples

Example 1: Enable rapid retraction during thread cutting

Program code	Comment
N55 M3 S500 G90 G18	; Active machining plane
...	; Approach the starting position
N65 MSG ("thread cutting")	; Tool infeed
MM_THREAD:	
N67 $AC_LIFTFAST=0	; Reset before starting the thread.
N68 G0 Z5	
N68 X10	
N70 G33 Z30 K5 LFON DILF=10 LFWP ALF=7	; Enable rapid retraction during thread cutting. Retraction path = 10 mm Retraction plane: Z/X (because of G18) Retraction direction: -X (with ALF=3: Retraction direction +X)
N71 G33 Z55 X15	
N72 G1	; Deselect thread cutting.
N69 IF $AC_LIFTFAST GOTOB MM_THREAD	; If thread cutting has been interrupted.
N90 MSG ("")	
...	
N70 M30	

Example 2: Switch off rapid retraction before tapping.

Program code	Comment
N55 M3 S500 G90 G0 X0 Z0	
...	
N87 MSG ("tapping")	
N88 LFOF	; Deactivate rapid retraction before tapping.
N89 CYCLE...	; Tapping cycle with G33.
N90 MSG ("")	
...	
N99 M30	

Example 3: Rapid retraction to absolute retraction position

Path interpolation of X is suppressed in the event of a stop and a motion executed to position POLF[X] at maximum velocity instead. The motion of the other axes continues to be determined by the programmed contour or the thread lead and the spindle speed.

Program code	Comment
`N10 G0 G90 X200 Z0 S200 M3`	
`N20 G0 G90 X170`	
`N22 POLF[X]=210 LFPOS`	
`N23 POLFMASK(X)`	`; Activate (enable) rapid retraction from axis X.`
`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()`	`; Disable lift for all axes.`
`M30`	

10.10.5 Convex thread (G335, G336)

The G commands G335 and G336 can be used to turn convex threads (= differing to the cylindrical form). Application is the machining of extremely large components that sag in the machine because of their self-weight. Paraxial thread would result in the thread being too small in the middle of the component. This can be compensated with convex threads.

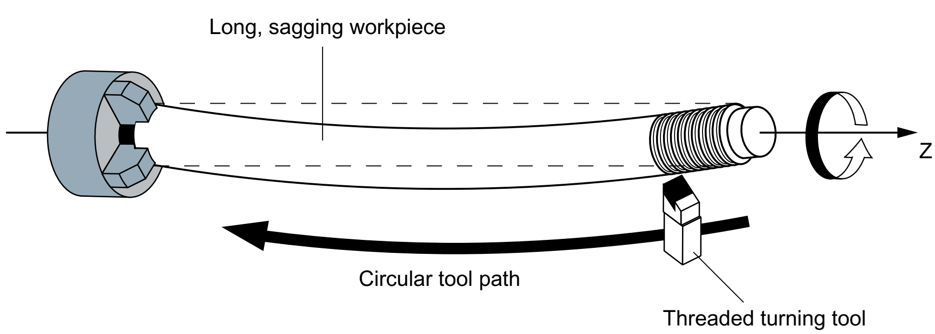


Figure 10-6 Turning a convex thread

Programming

The turning of a convex thread is programmed with `G335` or `G336`:

`G335`:	Turning of a convex thread on a circular tool path **in a clockwise direction**
`G336`:	Turning of a convex thread on a circular tool path **in a counter-clockwise direction**

The programming is performed first as for a linear thread by specifying the axial block end points and the pitch via parameters `I`, `J` and `K` (see "Thread cutting with constant lead (G33, SF) (Page 230)").

An arc is also specified. As for `G2`/`G3`, this can be programmed via the center point, radius, opening angle or intermediate point specification (see "Circular interpolation (Page 195)"). When programming the convex thread with center point programming, the following must be taken into account: Since `I`, `J` and `K` are used for the pitch in thread cutting, the circle parameters in the center point programming must be programmed with `IR=...`, `JR=...` and `KR=...`.

`IR=...:`	Cartesian coordinate for the circle center point in the X direction
`JR=...:`	Cartesian coordinate for the circle center point in the Y direction
`KR=...:`	Cartesian coordinate for the circle center point in the Z direction

Note

IR, JR and KR are the default values of the interpolation parameter names for a convex thread that can be set via machine data (MD10651 $MN_IPO_PARAM_THREAD_NAME_TAB).

Differences to the default values must be taken from the specifications of the machine manufacturer.

Optionally, a starting point offset `SF` can also be specified (see "Thread cutting with constant lead (G33, SF) (Page 230)").

Syntax

The syntax for the programming of a convex thread therefore has the following general form:

```
G335/G336 <axis target point coordinate(s)> <pitch> <arc> [<starting point offset>]
```

Examples

Example 1: Convex thread in the clockwise direction with end and center point programming

Program code	Comment
`N5 G0 G18 X50 Z50`	`; Approach starting point.`
`N10 G335 Z100 K=3.5 KR=25 IR=-20 SF=90`	`; Turn convex thread in the clockwise direction.`

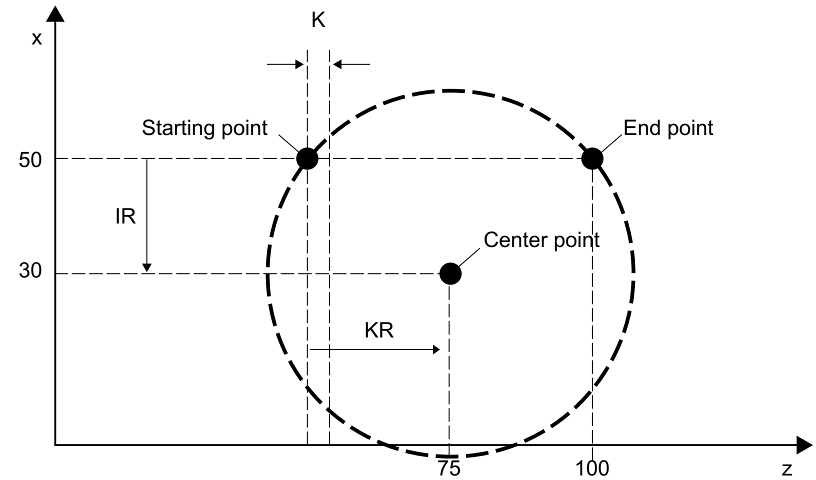


Figure 10-7 Convex thread in the clockwise direction with end and center point programming

Example 2: Convex thread in the counter-clockwise direction with end and center point programming

Program code	Comment
N5 G0 G18 X50 Z50	; Approach starting point.
N10 **G336 Z100** K=3.5 **KR=25 IR=20** SF=90	; Turn convex thread in the counter-clockwise direction.

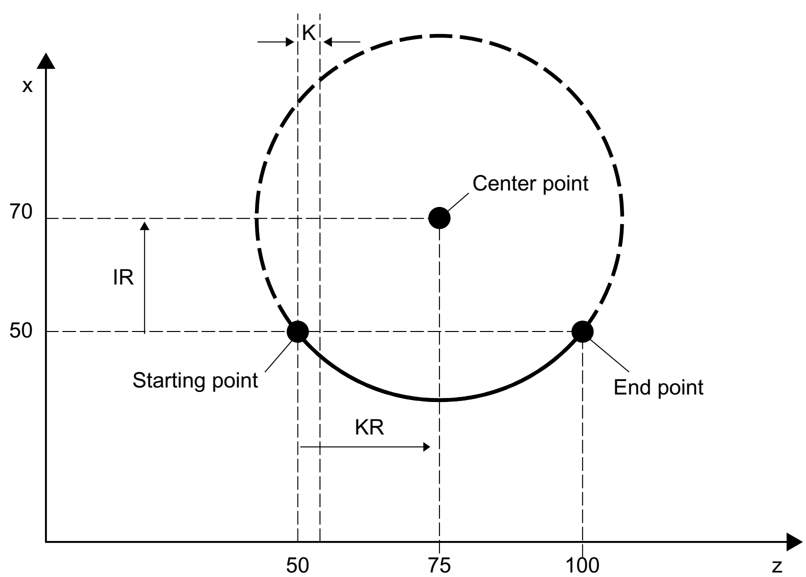


Figure 10-8 Convex thread in the counter-clockwise direction with end and center point programming

Example 3: Convex thread in the clockwise direction with end point and radius programming

Program code
N5 G0 G18 X50 Z50

Program code
N10 **G335** **Z100** K=3.5 **CR=32** SF=90

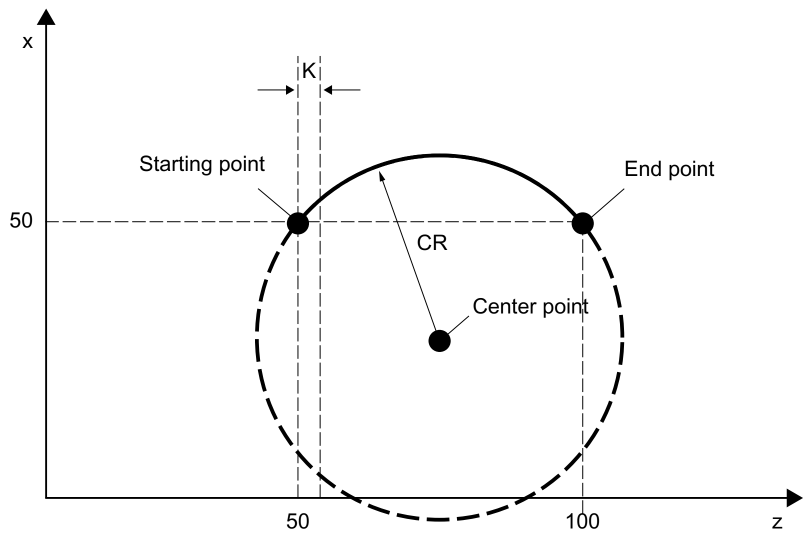


Figure 10-9 Convex thread in the clockwise direction with end point and radius programming

Example 4: Convex thread in the clockwise direction with end point and opening angle programming

Program code
N5 G0 G18 X50 Z50
N10 **G335** **Z100** K=3.5 **AR=102.75** SF=90

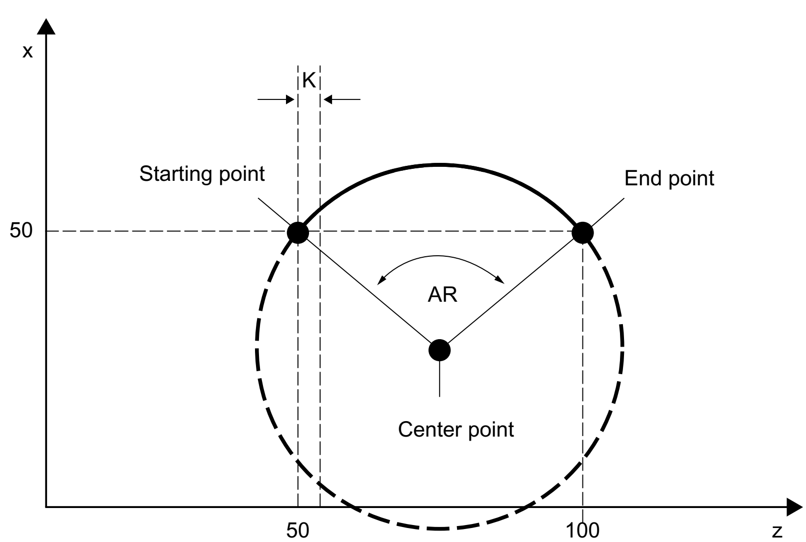


Figure 10-10 Convex thread in the clockwise direction with end point and opening angle programming

Example 5: Convex thread in the clockwise direction with center point and opening angle programming

Program code
N5 G0 G18 X50 Z50
N10 **G335** K=3.5 **KR=25 IR=-20 AR=102.75** SF=90

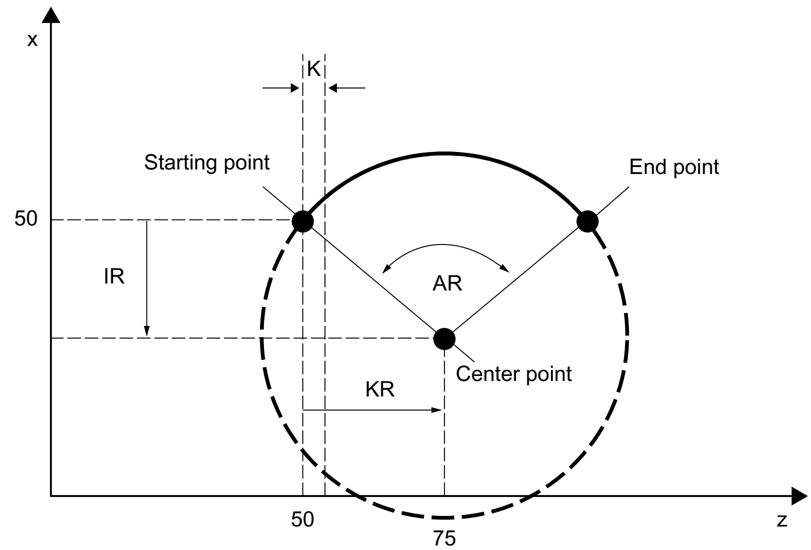


Figure 10-11 Convex thread in the clockwise direction with center point and opening angle programming

Example 6: Convex thread in the clockwise direction with end and intermediate point programming

Program code
N5 G0 G18 X50 Z50
N10 **G335 Z100** K=3.5 **I1=60 K1=64**

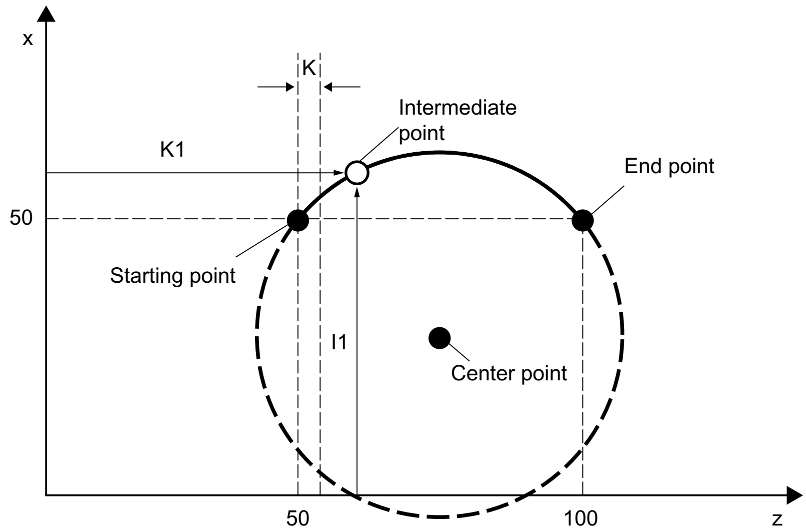
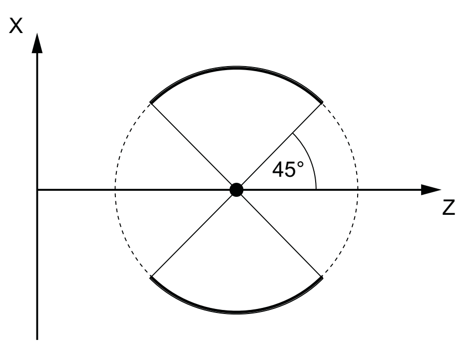


Figure 10-12 Convex thread in the clockwise direction with end and intermediate point programming

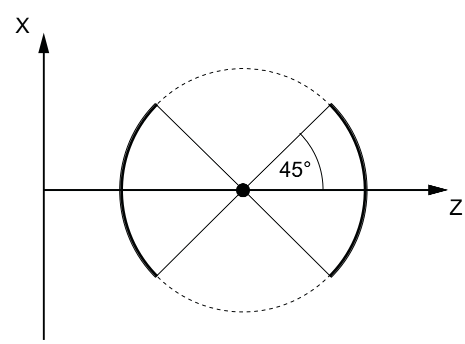
Further information

Permissible arc areas

The arc programmed at `G335`/`G336` must be in an area in which the specified thread main axis (`I`, `J` or `K`) has the main axis share on the arc over the entire arc:



Permissible areas for the **Z** axis (pitch programmed with `K`)



Permissible areas for the **X** axis (pitch programmed with `I`)

A change of the thread main axis as shown in the following figure is **not** permitted:

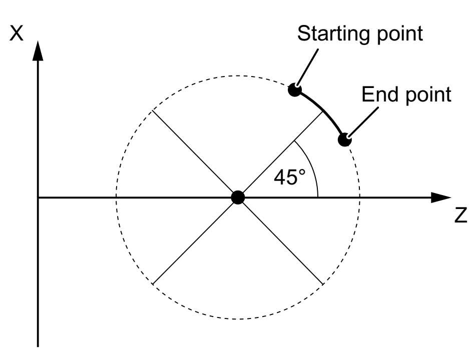


Figure 10-13 Convex thread: Area that is not permissible

Frames

G335 and G336 are also possible with active frames. However, you must ensure that the permissible arc areas are maintained in the basic coordinate system (BCS).

Supplementary conditions for the circular-path programming

The supplementary conditions described for the circular-path programming with G2/G3 apply for the circular-path programming under G335/G336 (see "Circular interpolation (Page 195)").

10.11 Tapping

10.11.1 Tapping without compensating chuck

10.11.1.1 Tapping without compensating chuck and retraction motion (G331, G332)

For tapping without compensating chuck, using the `G331` and `G332`commands, the following traversing motion is executed:

- `G331`: Tapping in the tapping direction up to the end of thread point
- `G332`: Retraction motion up to the tapping block `G331` with automatic spindle direction of rotation reversal

Syntax

```
G331 <axis> <thread pitch>
G331 <axis> <thread pitch> S...
G332 <axis> <thread pitch>
```

Meaning

`G331`:	Tapping The tapped hole is defined by the traversing motion of the axis (drilling depth) and the thread pitch.	
	Effective:	Modal
`G332`:	Retraction motion when tapping Retraction motion must have the same pitch as when tapping (`G331`). The direction of rotation of the spindle is reversed automatically.	
	Effective:	Modal
`<Axis>`:	Traversing distance/position of the geometry axis (X, Y or Z) at the end of the thread, e.g. Z50	
`<Thread pitch>`:	Thread pitch I (X), J (Y) or K (Z): • Positive pitch: Right-handed thread, e.g. K1.25 • Negative pitch: Left-handed thread, e.g. K-1.25	
	Value range:	±0.001 to ±2000.00 mm/revolution
`S...`:	Spindle speed The last active spindle speed is used if a spindle speed is not specified.	

Note

Second gear-stage data block

To achieve effective adaptation of spindle speed and motor torque and be able to accelerate faster, a second gear-stage data block for two further configurable switching thresholds (maximum speed and minimum speed) can be preset in axis-specific machine data deviating from the first gear step data block and also independent of these speed switching thresholds. The specifications of the machine manufacturer must be observed.

References:
Function Manual, Basic Functions; Spindles (S1), Section: " Configurable gear adaptation"

Examples

- Example: Tapping with G331 / G332 (Page 251)
- Example: Output the programmed drilling speed in the current gear stage (Page 252)
- Example: Application of the second gear-stage data block (Page 252)
- Example: Speed is not programmed, the gearbox stage is monitored (Page 252)
- Example: Gearbox stage cannot be changed, gearbox stage monitoring (Page 253)
- Example: Programming without SPOS (Page 253)

10.11.1.2 Example: Tapping with G331 / G332

Program code	Comment
N10 SPOS[n]=0	; Spindle: Position control mode
	; Start position 0 degrees
N20 G0 X0 Y0 Z2	; Axes: Approach starting position
N30 G331 Z-50 K-4 S200	; Tapping in Z,
	; Pitch K-4 negative =>
	; Direction of spindle rotation: CCW rotation,
	; Spindle speed 200 rpm
N40 G332 Z3 K-4	; Retraction motion in Z,
	; Pitch K-4 negative (counterclock-wise),
	; autom. direction of rotation reversal =>
	; Clockwise spindle direction of rota-tion
N50 G1 F1000 X100 Y100 Z100 S300 M3	; Spindle in spindle operation

10.11.1.3 Example: Output the programmed drilling speed in the current gear stage

Program code	Comment
N05 M40 S500	; Programmed spindle speed: 500 rpm =>
	; Gearbox stage 1 (20 to 1028 rpm)
...	
N55 SPOS=0	; Position the spindle
N60 G331 Z-10 K5 S800	; Tapping
	; Spindle speed 800 rpm => gearbox stage 1

The appropriate gear stage for the programmed spindle speed `S500` with `M40` is determined on the basis of the first gear-stage data block. The programmed drilling speed `S800` is output in the current gear stage and, if necessary, is limited to the maximum speed of the gear stage. No automatic gear-stage change is possible following an `SPOS` operation. In order for an automatic change in gear stage to be performed, the spindle must be in speed-control mode.

Note

If gearbox stage 2 is selected at a spindle speed of 800 rpm, then the switching thresholds for the maximum and minimum speed must be configured in the relevant machine data of the second gear-stage data block (see the examples below).

10.11.1.4 Example: Application of the second gear-stage data block

The switching thresholds of the second gear-stage data block for the maximum and minimum speed are evaluated for `G331`/`G332` and when programming an `S` value for the active master spindle. Automatic `M40` gear-stage change must be active. The gear stage as determined in the manner described above is compared with the active gear stage. If they are found to be different, then the gearbox stage is changed.

Program code	Comment
N05 M40 S500	; Programmed spindle speed: 500 rpm
...	
N50 G331 S800	; Master spindle: Gearbox stage 2 is selected
N55 SPOS=0	; Position the spindle
N60 G331 Z-10 K5	; Tapping
	; Spindle acceleration from second gearbox stage data block 2

10.11.1.5 Example: Speed is not programmed, the gearbox stage is monitored

If no speed is programmed when using the second gearbox stage data block with `G331`, then the last speed programmed will be used to produce the thread. The gear stage does not change. However, monitoring is performed in this case to check that the last speed programmed is within the preset speed range (defined by the maximum and minimum speed thresholds) for the active gear stage. Otherwise, alarm 16748 is output.

Program code	Comment
N05 M40 S800	; Programmed spindle speed: 800 rpm

Program code	Comment
...	
N55 SPOS=0	; Position the spindle
N60 G331 Z-10 K5	; Tapping
	; Monitoring the spindle speed, 800 rpm
	; Gearbox stage 1 is active
	; Gearbox stage 2 should be active => Alarm 16748

10.11.1.6 Example: Gearbox stage cannot be changed, gearbox stage monitoring

If the spindle speed is programmed in addition to the geometry in the `G331` block when using the second gear-stage data block, if the speed is not within the preset speed range (defined by the maximum and minimum speed thresholds) of the active gear stage, it will not be possible to change gear stages, because the path motion of the spindle and the infeed axis (axes) would not be retained.

As in the example above, the speed and gearbox stage are monitored in the `G331` block and alarm 16748 is signaled if necessary.

Program code	Comment
N05 M40 S500	; Programmed spindle speed: 500 rpm =>
	; Gearbox stage 1
...	
N55 SPOS=0	; Position the spindle
N60 G331 Z-10 K5 S800	; Tapping
	; Gearbox stage cannot be changed,
	; Monitoring the spindle speed, 800 rpm
	; with gearbox stage data set 1: Gearbox stage 2
	; should be active => Alarm 16748

10.11.1.7 Example: Programming without SPOS

Program code	Comment
N05 M40 S500	; Programmed spindle speed: 500 rpm =>
	; Gearbox stage 1 (20 to 1028 rpm)
...	
N50 G331 S800	; Master spindle: Gearbox stage 2 is selected
N60 G331 Z-10 K5	; Tapping
	; Spindle acceleration from second gearbox stage data block 2

Thread interpolation for the spindle starts from the current position, which is determined by the previously processed section of the part program, e.g. if the gear stage was changed. Therefore, it might not be possible to remachine the thread.

Note

Please note that when machining with multiple spindles, the drill spindle also has to be the master spindle. `SETMS(<spindle number>)` can be programmed to set the drill spindle as the master spindle.

10.11.2 Tapping with compensating check and retraction motion (G63)

For tapping with compensating chuck, using the `G63` command, the following traversing motion is executed:

- `G63`: Tapping in the tapping direction up to the end of thread point
- `G63`: Retraction motion with programmed spindle direction of rotation reversal

---

Note

After a `G63`block, the last effective interpolation type `G0`, `G1`, `G2` is active.

Syntax

```
G63 <axis> <direction of rotation> <speed <feedrate>
```

Meaning

`G63`:	Tapping with compensating chuck	
	Effective:	Non-modal
`<Axis>`:	Traversing distance/position of the geometry axis (X, Y or Z) at the end of the thread, e.g. Z50	
`<Direction of rotation>`:	Direction of spindle rotation: • `M3`: Clockwise rotation, right-hand thread • `M4`: Counterclockwise rotation, left-hand thread	
`<Speed>`:	Maximum permissible spindle speed while tapping, e.g. S100	
`<Feedrate>`:	Feedrate of the tapping axis F, with F = spindle speed * thread pitch	

Example

Tapping an M5 thread:

- Spindle pitch according to the standard: 0.8 mm/rev
- Spindle speed S: 200 rpm
- Feedrate F = 200 rpm * 0.8 mm/rev = 160 mm/min.

Program code	Comment
N10 G1 X0 Y0 Z2 F1000 S200 M3	; Approach starting point
	; Spindle clockwise direction of rotation, 200 rpm
N20 G63 Z-50 F160	; Tapping with compensating chuck
	; Drilling depth: absolute Z=50mm
	; Feedrate: 160 mm/min
N30 G63 Z3 M4	; Retraction movement: absolute Z=3mm
	; Direction of rotation reversal
	; Spindle with counterclockwise direction of rotation, 200 rpm

10.12 Chamfer, rounding (CHF, CHR, RND, RNDM, FRC, FRCM)

Contour corners within the active working plane can be executed as roundings or chamfers.

For optimum surface quality, a separate feedrate can be programmed for chamfer/rounding. If a feedrate is not programmed, the standard path feedrate F will be applied.

The "Modal rounding" function can be used to round multiple contour corners in the same way one after the other.

Syntax

Chamfer the contour corner:

```
G... X... Z... CHR/CHF=<value> FRC/FRCM=<value>
G... X... Z...
```

Round the contour corner:

```
G... X... Z... RND=<value> FRC=<value>
G... X... Z...
```

Modal rounding:

```
G... X... Z... RNDM=<value> FRCM=<value>
...
RNDM=0
```

Note

The technology (feedrate, feedrate type, M commands, etc.) for chamfer/rounding is derived from either the previous or the next block dependent on the setting of bit 0 in machine data MD20201 $MC_CHFRND_MODE_MASK (chamfer/rounding behavior). The recommended setting is the derivation from the previous block (bit 0 = 1).

Meaning

`CHF=...` :	Chamfer the contour corner	
	`<value>`:	Length of the chamfer (unit corresponding to G70/G71)
`CHR=...` :	Chamfer the contour corner	
	`<value>`:	Width of the chamfer in the original direction of motion (unit corresponding to G70/G71)
`RND=...` :	Round the contour corner	
	`<value>`:	Radius of the rounding (unit corresponding to G70/G71)
`RNDM=...` :	Modal rounding (rounding multiple contour corners in the same way one after the other)	
	`<value>`:	Radius of the roundings (unit corresponding to G70/G71)
		Modal rounding is deactivated with `RNDM=0`.
`FRC=...` :	Non-modal feedrate for chamfer/rounding	
	`<value>`:	Feedrate in mm/min (with active G94) or mm/rev (with active G95)

`FRCM=...` :	Modal feedrate for chamfer/rounding	
	`<value>`:	Feedrate in mm/min (with active G94) or mm/rev (with active G95)
		`FRCM=0` deactivates modal feedrate for chamfer/rounding and activates the feedrate programmed under F.

Note

Chamfer/rounding too high

If the values programmed for chamfer (CHF/CHR) or rounding (RND/RNDM) are too high for the contour elements involved, chamfer or rounding will automatically be adapted:

1. If MD11411 $MN_ENABLE_ALARM_MASK bit 4 is set, alarm 10833 "Chamfer or rounding must be reduces" is output (cancel alarm).
2. The chamfer/rounding is reduced until it fits in the contour corner. This results in at least one block without motion. At this block, the required motion is stopped.

Note

Chamfer/rounding not possible

No chamfer/rounding is performed if:

- No straight or circular contour is available in the plane
- A movement takes place outside the plane
- The plane is changed
- The number of blocks specified in the machine data that are not to contain any information about traversing (e.g. only command outputs) is exceeded

Note

FRC/FRCM

FRC/FRCM has no effect if a chamfer is traversed with G0; the command can be programmed according to the F value without error message.

FRC is only effective if a chamfer/rounding is programmed in the block or if RNDM has been activated.

FRC overwrites the F or FRCM value in the current block.

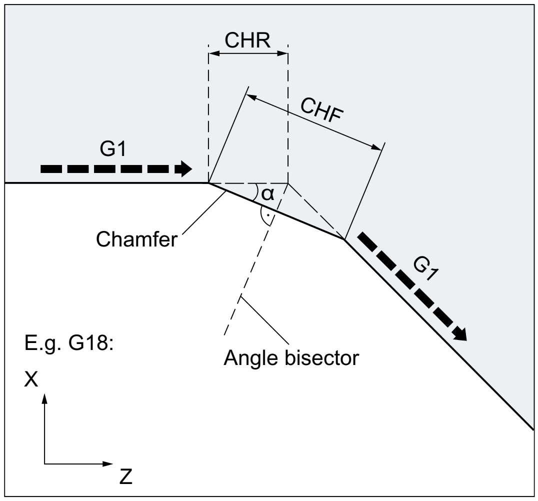
The feedrate programmed under FRC must be greater than zero.

FRCM=0 activates the feedrate programmed under F for chamfer/rounding.

If FRCM is programmed, the FRCM value will need to be reprogrammed like F on change G94 ↔ G95, etc. If only F is reprogrammed and if the feedrate type FRCM > 0 before the change, an error message will be output.

Examples

Example 1: Chamfer between two straight lines



- MD20201 Bit 0 = 1 (derived from previous block).
- G71 is active.
- The width of the chamfer in the direction of motion (CHR) should be 2 mm and the feedrate for chamfer 100 mm/min.

Programming can be performed in two ways:

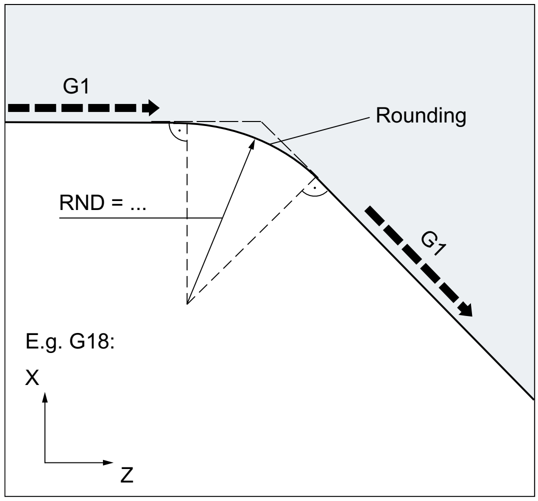
- Programming with CHR

```
Program code
...
N30 G1 Z… CHR=2 FRC=100
N40 G1 X…
...
```

- Programming with CHF

```
Program code
...
N30 G1 Z… CHF=2(cosα*2) FRC=100
N40 G1 X…
...
```

Example 2: Rounding between two straight lines

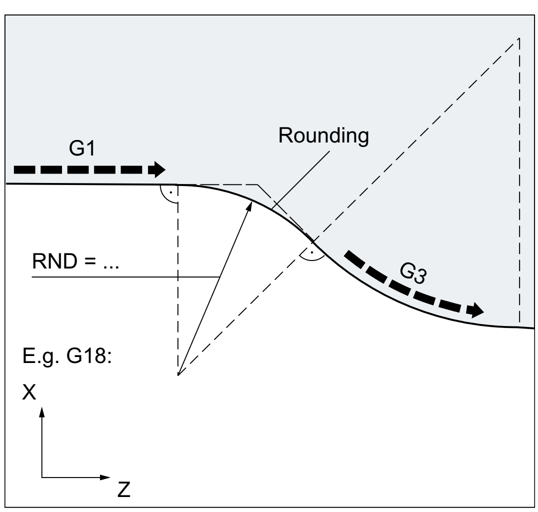


- MD20201 Bit 0 = 1 (derived from previous block).
- G71 is active.
- The radius of the rounding should be 2 mm and the feedrate for rounding 50 mm/min.

Program code
...
N30 G1 Z… RND=2 FRC=50
N40 G1 X…
...

Example 3: Rounding between straight line and circle

The RND function can be used to insert a circle contour element with tangential connection between the linear and circle contours in any combination.



- MD20201 Bit 0 = 1 (derived from previous block).
- G71 is active.
- The radius of the rounding should be 2 mm and the feedrate for rounding 50 mm/min.

Program code
...

Program code
N30 G1 Z… RND=2 FRC=50
N40 G3 X… Z… I… K…
...

Example 4: Modal rounding to deburr sharp workpiece edges

Program code	Comment
...	
N30 G1 X… Z… RNDM=2 FRCM=50	; Activate modal rounding.
	Radius of rounding: 2 mm
	Feedrate for rounding: 50 mm/min
N40...	
N120 RNDM=0	; Deactivate modal rounding.
...	

Example 5: Apply technology from following block or previous block

- MD20201 Bit 0 = 0: Derived from following block (default setting!)

Program code	Comment
N10 G0 X0 Y0 G17 F100 G94	
N20 G1 X10 CHF=2	; Chamfer N20-N30 with F=100 mm/min
N30 Y10 CHF=4	; Chamfer N30-N40 with FRC=200 mm/min
N40 X20 CHF=3 FRC=200	; Chamfer N40-N60 with FRCM=50 mm/min
N50 RNDM=2 FRCM=50	
N60 Y20	; Modal rounding N60-N70 with FRCM=50 mm/min
N70 X30	; Modal rounding N70-N80 with FRCM=50 mm/min
N80 Y30 CHF=3 FRC=100	; Chamfer N80-N90 with FRC=100 mm/min
N90 X40	; Modal rounding N90-N100 with F=100 mm/min (deselection of FRCM)
N100 Y40 FRCM=0	; Modal rounding N100-N120 with G95 FRC=1 mm/rev
N110 S1000 M3	
N120 X50 G95 F3 FRC=1	
...	
M02	

- MD20201 Bit 0 = 1: Derived from previous block (recommended setting!)

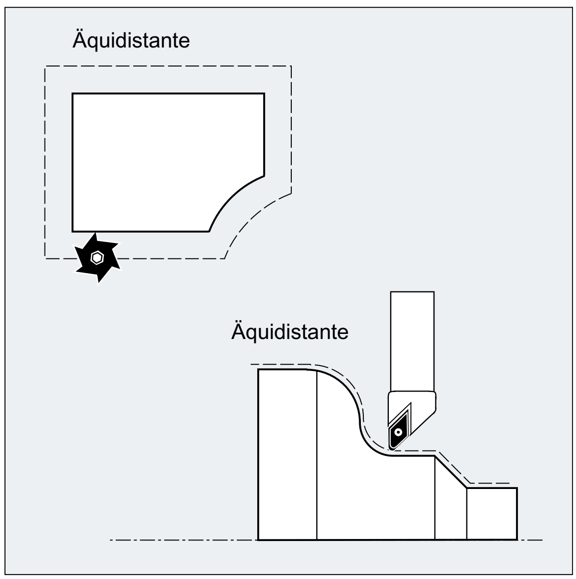
Program code	Comment
N10 G0 X0 Y0 G17 F100 G94	
N20 G1 X10 CHF=2	; Chamfer N20-N30 with F=100 mm/min
N30 Y10 CHF=4 FRC=120	; Chamfer N30-N40 with FRC=120 mm/min
N40 X20 CHF=3 FRC=200	; Chamfer N40-N60 with FRC=200 mm/min

Program code	Comment
N50 RNDM=2 FRCM=50	
N60 Y20	; Modal rounding N60-N70 with FRCM=50 mm/min
N70 X30	; Modal rounding N70-N80 with FRCM=50 mm/min
N80 Y30 CHF=3 FRC=100	; Chamfer N80-N90 with FRC=100 mm/min
N90 X40	; Modal rounding N90-N100 with FRCM=50 mm/min
N100 Y40 FRCM=0	; Modal rounding N100-N120 with F=100 mm/min
N110 S1000 M3	
N120 X50 CHF=4 G95 F3 FRC=1	; Chamfer N120-N130 with G95 FRC=1 mm/rev
N130 Y50	; Modal rounding N130-N140 with F=3 mm/rev
N140 X60	
...	
M02	

Tool radius compensation 11

11.1 Tool radius compensation (G40, G41, G42, OFFN)

When tool radius compensation (TRC) is active, the control automatically calculates the equidistant tool paths for various tools.



Syntax

G0/G1 X... Y… Z... **G41**/**G42** [**OFFN=<value>**]	
...	
G40 X... Y… Z...	

Meaning

G41:	Activate TRC with machining direction **left** of the contour.
G42:	Activate TRC with machining direction **right** of the contour.
OFFN=<value>:	Allowance on the programmed contour (normal contour offset) (optional), e.g. to generate equidistant paths for rough finishing.
G40:	Deactivate TRC.

Note

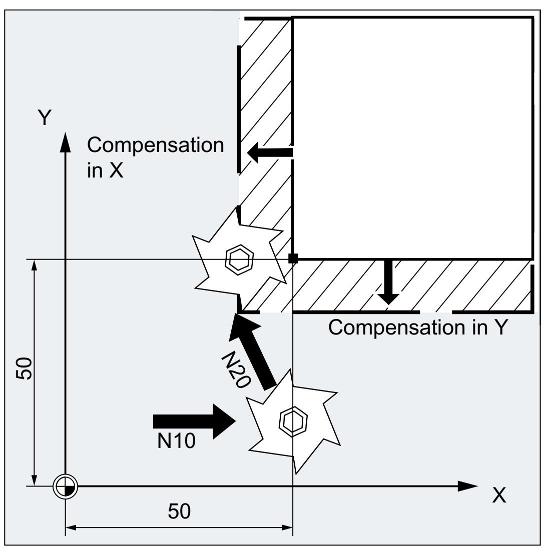
In the NC block with `G40`/`G41`/`G42`, `G0` or `G1` has to be active and at least one axis has to be specified on the selected working plane.

If only one axis is specified on activation, the last position on the second axis is added automatically and traversed with **both** axes.

The two axes must be active as geometry axes in the channel. This can be achieved by means of `GEOAX` programming.

Examples

Example 1: Milling

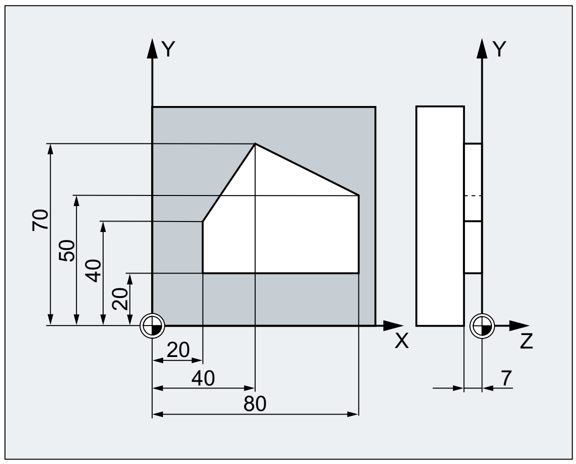


Program code	Comment
N10 G0 X50 T1 D1	; Only tool length compensation is activated. X50 is approached without compensation.
N20 G1 G41 Y50 F200	; Radius compensation is activated, point X50/Y50 is approached with compensation.
N30 Y100	
…	

Example 2: "Conventional" procedure using milling as an example

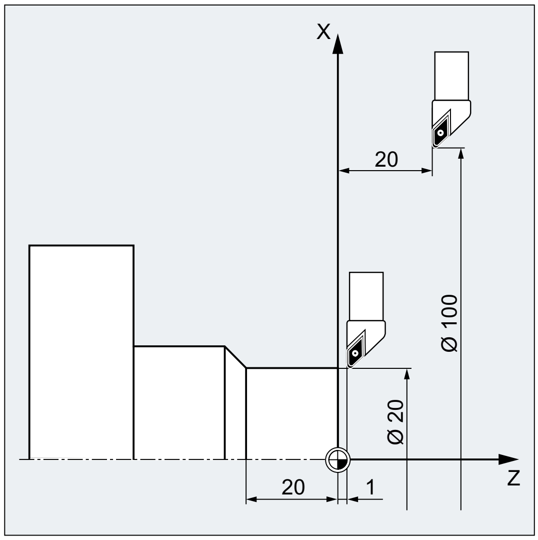
"Conventional" procedure:

1. Tool call.
2. Change tool.
3. Activate working plane and tool radius compensation.



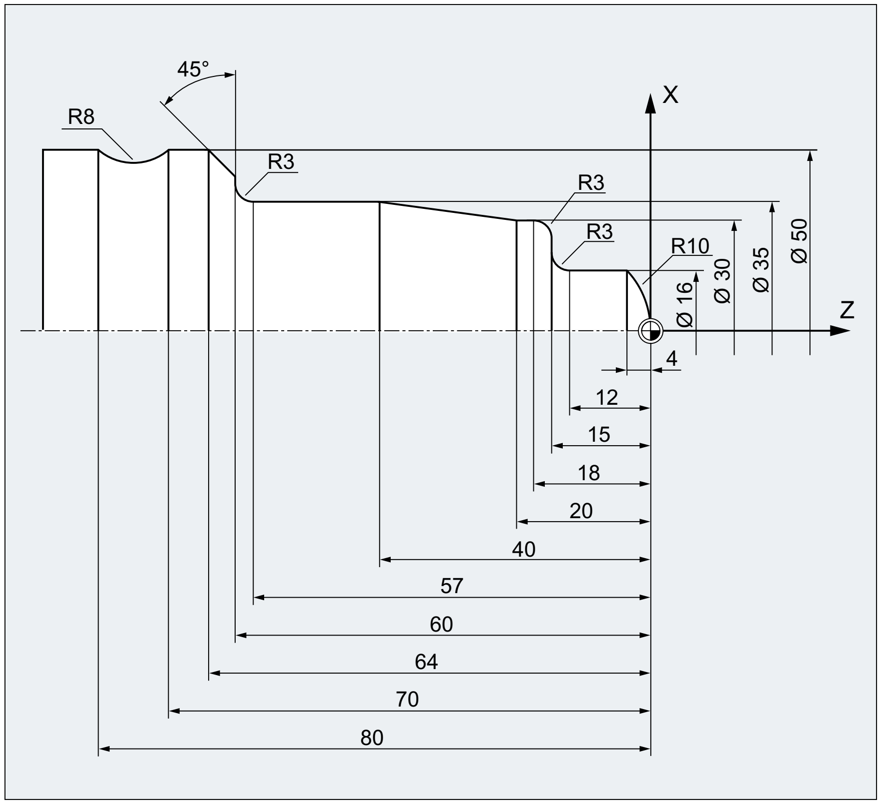
Program code	Comment
N10 G0 Z100	; Retraction for tool change.
N20 G17 T1 M6	; Tool change
N30 G0 X0 Y0 Z1 M3 S300 D1	; Call tool offset values, select length compensation.
N40 Z-7 F500	; Feed in tool.
N50 G41 X20 Y20	; Activate tool radius compensation, tool machines to the left of the contour.
N60 Y40	; Mill contour.
N70 X40 Y70	
N80 X80 Y50	
N90 Y20	
N100 X20	
N110 G40 G0 Z100 M30	; Retract tool, end of program.

Example 3: Turning



Program code	Comment
…	
N20 T1 D1	; Only tool length compensation is activated.
N30 G0 X100 Z20	; X100 Z20 is approached without compensation.
N40 G42 X20 Z1	; Radius compensation is activated, point X20/Z1 is approached with compensation.
N50 G1 Z-20 F0.2	
…	

Example 4: Turning



Program code	Comment
N5 G0 G53 X280 Z380 D0	; Starting point.
N10 TRANS X0 Z250	; Work offset.
N15 LIMS=4000	; Speed limitation (G96).
N20 G96 S250 M3	; Select constant feedrate
N25 G90 T1 D1 M8	; Select tool selection and offset.
N30 G0 G42 X-1.5 Z1	; Set tool with tool radius compensation.
N35 G1 X0 Z0 F0.25	
N40 G3 X16 Z-4 I0 K-10	; Turn radius 10.
N45 G1 Z-12	
N50 G2 X22 Z-15 CR=3	; Turn radius 3.
N55 G1 X24	
N60 G3 X30 Z-18 I0 K-3	; Turn radius 3.
N65 G1 Z-20	
N70 X35 Z-40	
N75 Z-57	
N80 G2 X41 Z-60 CR=3	; Turn radius 3.
N85 G1 X46	
N90 X52 Z-63	
N95 G0 G40 G97 X100 Z50 M9	; Deselect tool radius compensation and approach tool change location.

Program code	Comment
N100 T2 D2	; Call tool and select offset.
N105 G96 S210 M3	; Select constant cutting rate.
N110 G0 G42 X50 Z-60 M8	; Set tool with tool radius compensation.
N115 G1 Z-70 F0.12	; Turn diameter 50.
N120 G2 X50 Z-80 I6.245 K-5	; Turn radius 8.
N125 G0 G40 X100 Z50 M9	; Retract tool and deselect tool radius compensation.
N130 G0 G53 X280 Z380 D0 M5	; Approach tool change location.
N135 M30	; End of program.

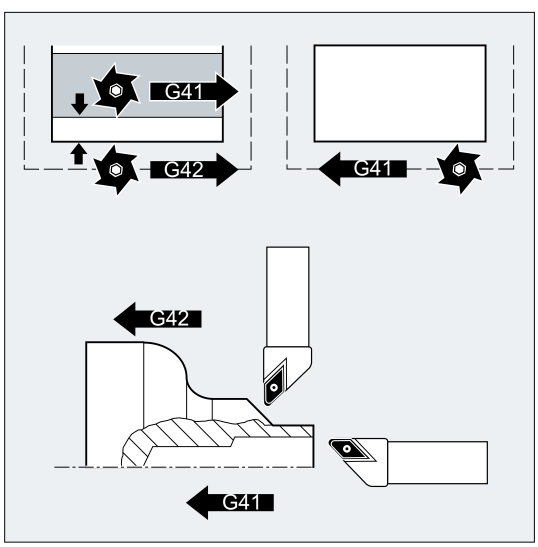
Further information

The control requires the following information in order to calculate the tool paths:

- Tool no. (T...), cutting edge no. (D...)
- Machining direction (G41/G42)
- Working plane (G17/G18/G19)

Tool no. (T...), cutting edge no. (D...)

The distance between tool path and workpiece contour is calculated from the milling cutter radii or cutting edge radii and the tool point direction parameters.



With a flat D number structure, only the D number has to be programmed.

Machining direction (G41/G42)

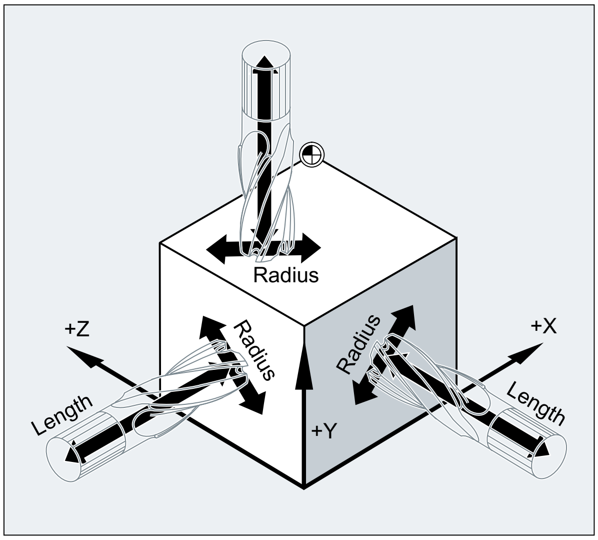
From this information, the control detects the direction in which the tool path is to be displaced.

Note

A negative offset value has the same significance as a change of offset side (`G41` ↔ `G42`).

Working plane (G17/G18/G19)

From this information, the control detects the plane and therefore the axis directions for compensation.



Example: Milling tool

Program code	Comment
...	
N10 G17 G41 …	; The tool radius compensation is performed in the X/Y plane, the tool length offset is performed in the Z direction.
...	

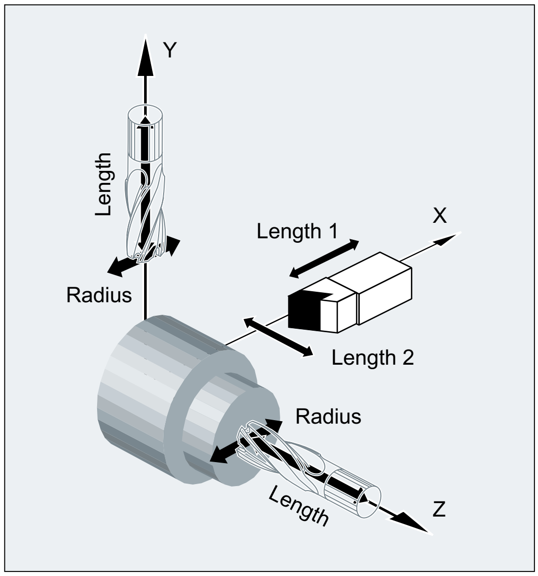
Note

On 2-axis machines, tool radius compensation is only possible in "real" planes, usually with `G18`.

Tool length offset

The wear parameter assigned to the diameter axis on tool selection can be defined as the diameter value using an MD. This assignment is not automatically altered when the plane is subsequently changed. To do this, the tool must be selected again after the plane has been changed.

Turning:



`NORM` and `KONT` can be used to define the tool path on activation and deactivation of compensation mode (see "Approaching and leaving contour (NORM, KONT, KONTC, KONTT) (Page 273)").

Point of intersection

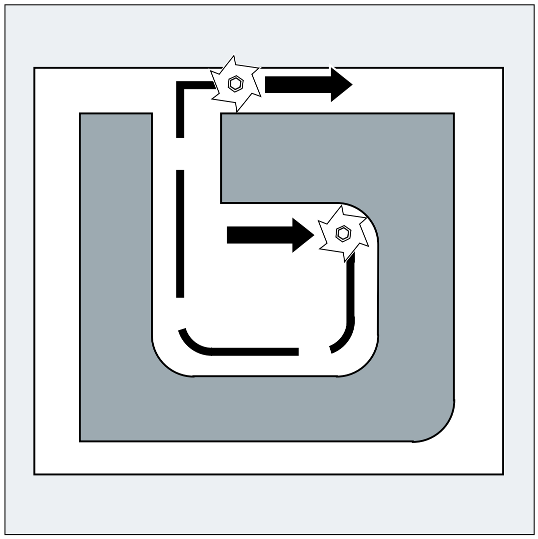
The intersection point is selected in the setting data:

SD42496 $SC_CUTCOM_CLSD_CONT (response of tool radius compensation with closed contour)

Value	Meaning
FALSE	If two intersections appear on the inside when offsetting a (virtually) closed contour, which consists of two circle blocks following on from one another, or from one circle block and one linear block, the intersection positioned closer to the end of block on the first partial contour is selected, in accordance with the standard procedure. A contour is deemed to be (virtually) closed if the distance between the starting point of the first block and the end point of the second block is less than 10% of the effective compensation radius, but not more than 1000 path increments (corresponds to 1 mm with 3 decimal places).
TRUE	In the same situation as described above, the intersection positioned on the first partial contour closer to the block start is selected.

Change in compensation direction (G41 ↔ G42)

A change in compensation direction (`G41` ↔ `G42`) can be programmed without an intermediate `G40`.



Change in the working plane

The working plane (`G17`/`G18`/`G19`) **cannot** be changed if `G41`/`G42` is active.

Change in tool offset data set (D…)

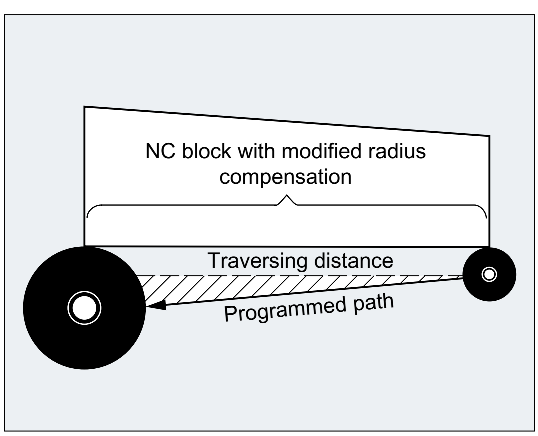
The tool offset data set can be changed in compensation mode.

A modified tool radius is active with effect from the block in which the new D number is programmed.

Note

The radius change or compensation movement is performed across the entire block and only reaches the new equidistance at the programmed end point.

In the case of linear movements, the tool travels along an inclined path between the starting point and end point:



Circular interpolation produces spiral movements.

Changing the tool radius

The change can be made, e.g. using system variables. The sequence is the same as when changing the tool offset data set (`D…`).

Note

The modified values only take effect the next time `T` or `D` is programmed. The change only applies with effect from the next block.

Compensation mode

Compensation mode may only be interrupted by a certain number of consecutive blocks or M functions which do not contain drive commands or positional data in the compensation plane.

Note

The number of consecutive blocks or M commands can be set in a machine data item (see machine manufacturer's specifications).

Note

A block with a path distance of zero also counts as an interruption!

11.2 Approaching and leaving contour (NORM, KONT, KONTC, KONTT)

Requirement

The KONTC and KONTT commands will only be available if the "Polynomial interpolation" option has been enabled in the control.

Function

If tool radius compensation is active (G41/G42), the NORM, KONT, KONTC or KONTT command can be used to adapt the tool's approach and retract paths to the required contour profile or blank form.

KONTC or KONTT ensure observance of the continuity conditions in all three axes. It is, therefore, permissible to program a path component perpendicular to the offset plane simultaneously.

Syntax

G41/G42 **NORM**/**KONT**/**KONTC**/**KONTT** X... Y... Z...	
...	
G40 X... Y... Z...	

Meaning

NORM:	Activate direct approach/retraction to/from a straight line. The tool is oriented perpendicular to the contour point.
KONT:	Activate approach/retraction with travel around the starting/end point according to the programmed corner behavior G450 or G451.
KONTC:	Activate approach/retraction with constant curvature.
KONTT:	Activate approach/retraction with constant tangent.

Note

Only G1 blocks are permissible as original approach/retraction blocks for KONTC and KONTT. The control replaces these with polynomials for the appropriate approach/retract path.

Supplementary conditions

KONTT and KONTC are not available in 3D variants of tool radius compensation (CUT3DC, CUT3DCC, CUT3DF). If they are programmed, the control switches internally to NORM without an error message.

Example

KONTC

The full circle is approached beginning at the circle center point. The direction and curvature radius at the block end point of the approach block are identical to the values of the next circle. Infeed takes place in the Z direction in both approach/retraction blocks simultaneously. The figure below shows the perpendicular projection of the tool path.

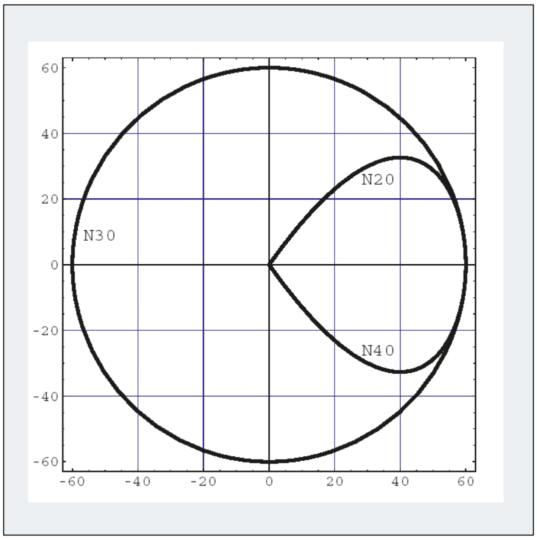


Figure 11-1 Perpendicular projection

The associated NC program segment is as follows:

Program code	Comment
`$TC_DP1[1,1]=121`	; Milling tool
`$TC_DP6[1,1]=10`	; Radius 10 mm
`N10 G1 X0 Y0 Z60 G64 T1 D1 F10000`	
`N20 G41 KONTC X70 Y0 Z0`	; Approach
`N30 G2 I-70`	; Full circle
`N40 G40 G1 X0 Y0 Z60`	; Retract
`N50 M30`	

At the same time as the curvature is being adapted to the circular path of the full circle, traversing is performed from Z60 to the plane of the circle Z0:

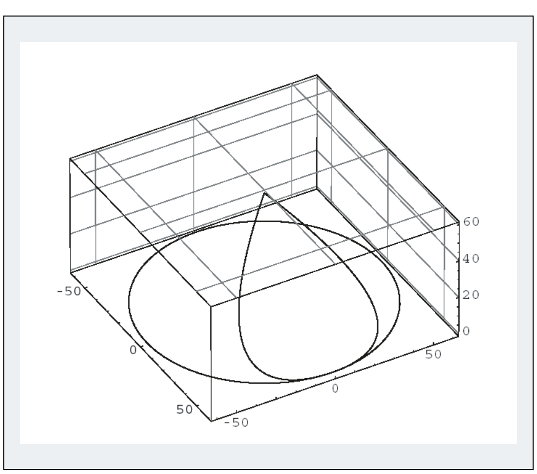
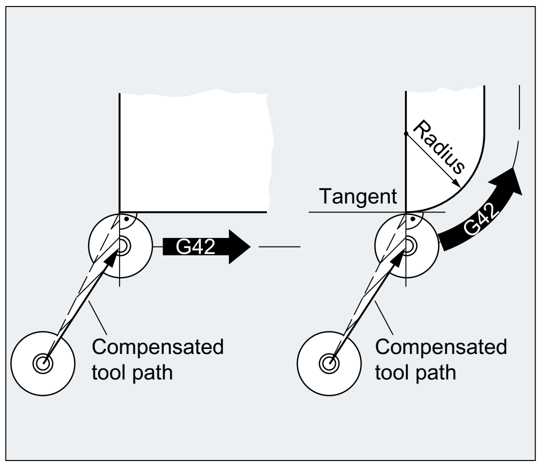


Figure 11-2 3D representation.

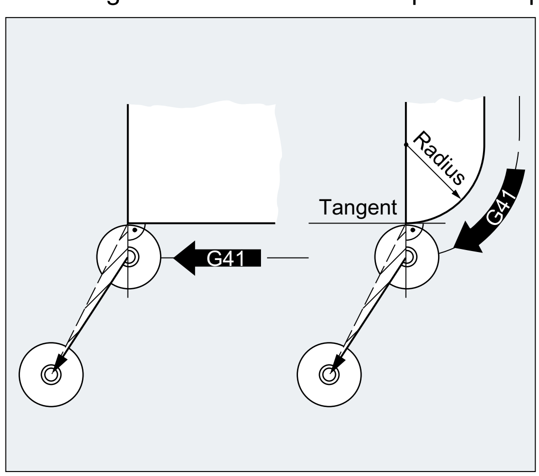
Further information

Approach/retraction with NORM

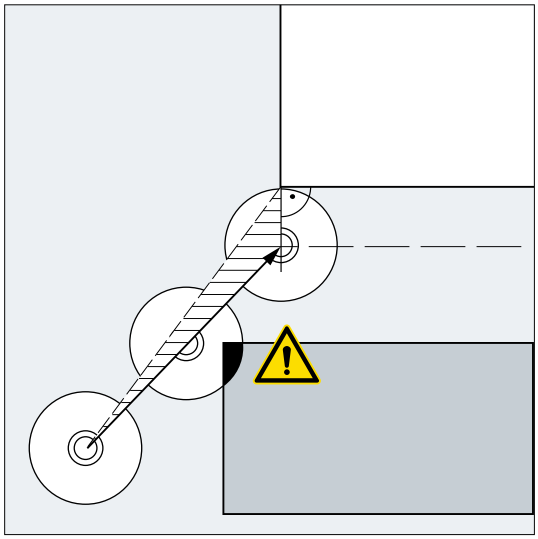
1. Approach:
   If `NORM` is activated, the tool will move directly to the compensated start position along a straight line (irrespective of the preset approach angle programmed for the travel movement) and is positioned perpendicular to the path tangent at the starting point.



2. Retraction:
   The tool is perpendicular to the last compensated path end point and then moves (irrespective of the preset approach angle programmed for the travel movement) directly in a straight line to the next uncompensated position, e.g. to the tool change point.



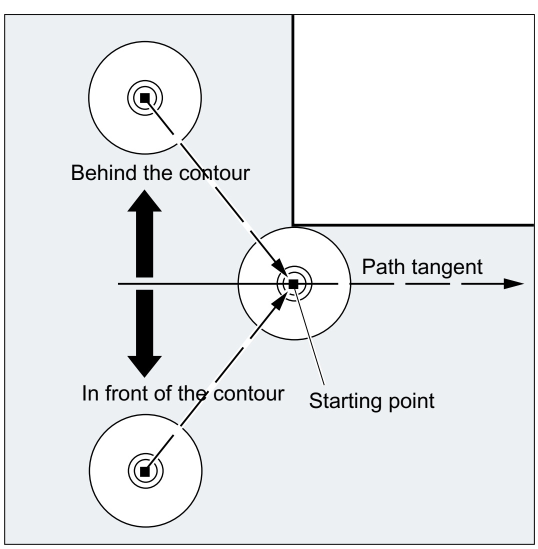
Modifying approach/retract angles introduces a collision risk:



NOTICE
Risk of collision
Modified approach/retract angles must be taken into account during programming in order that potential collisions can be avoided.

Approach/retraction with KONT

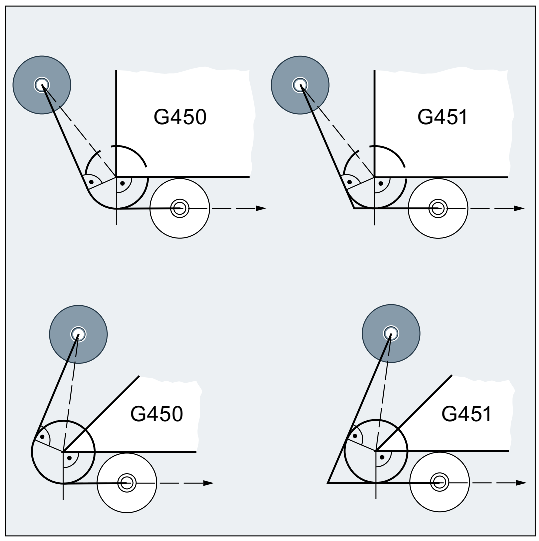
Prior to the approach, the tool can be located **in front of** or **behind** the contour. The path tangent at the starting point serves as a separation line:



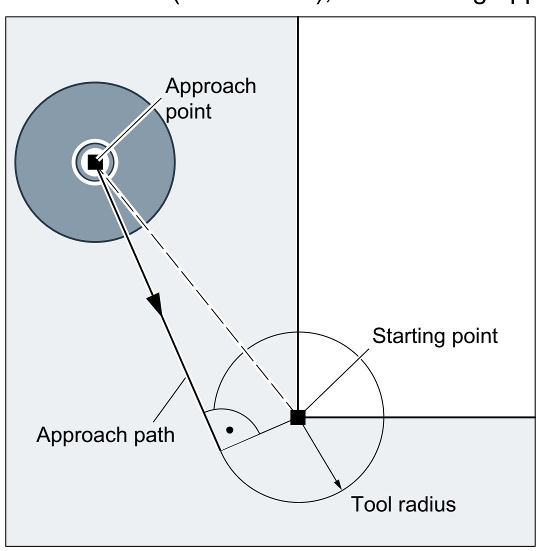
Accordingly, two scenarios need to be distinguished where approach/retraction with `KONT` is concerned:

1. The tool is located in front of the contour.
   → The approach/retract strategy is the same as with NORM.
2. The tool is located behind the contour.

– Approach:
  The tool travels around the starting point either along a circular path or over the intersection of the equidistant paths depending on the programmed corner behavior (`G450`/`G451`).
  The commands `G450`/`G451` apply to the transition from the current block to the next block:



In both cases (`G450`/`G451`), the following approach path is generated:



A straight line is drawn from the uncompensated approach point. This line is a tangent to a circle with circle radius = tool radius. The center point of the circle is on the starting point.

– Retraction:
  The same applies to retraction as to approach, but in the reverse order.

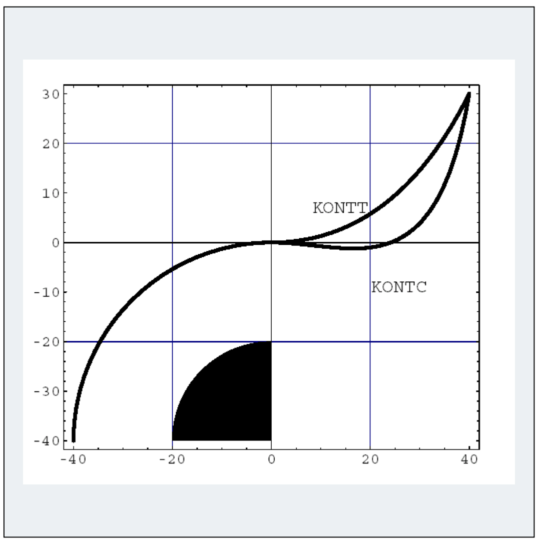
Approach/retraction with KONTC

The contour point is approached/exited with constant curvature. There is no jump in acceleration at the contour point. The path from the start point to the contour point is interpolated as a polynomial.

Approach/retraction with KONTT

The contour point is approached/exited with constant tangent. A jump in the acceleration can occur at the contour point. The path from the start point to the contour point is interpolated as a polynomial.

Difference between KONTC and KONTT

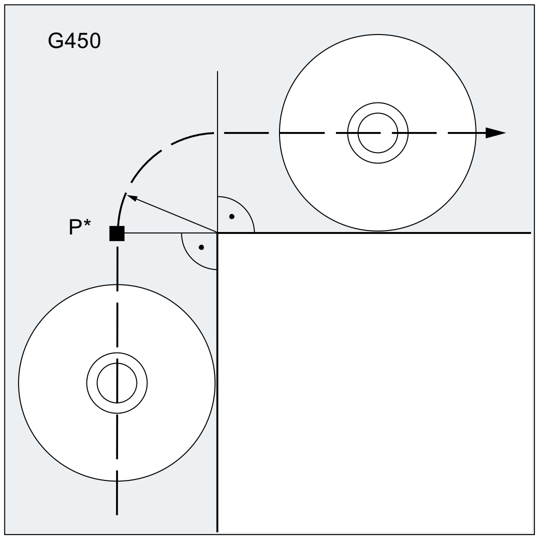


The figure below shows the differences in approach/retraction behavior between KONTT and KONTC. A circle with a radius of 20 mm about the center point at X0 Y-40 is compensated with a tool with an external radius of 20 mm. The tool center point therefore moves along a circular path with radius 40 mm. The end point of the approach blocks is at X40 Y30. The transition between the circular block and the retraction block is at the zero point. Due to the extended continuity of curvature associated with KONTC, the retraction block first executes a movement with a negative Y component. This will often be undesired. This response does not occur with the KONTT retraction block. However, with this block, an acceleration step change occurs at the block transition.

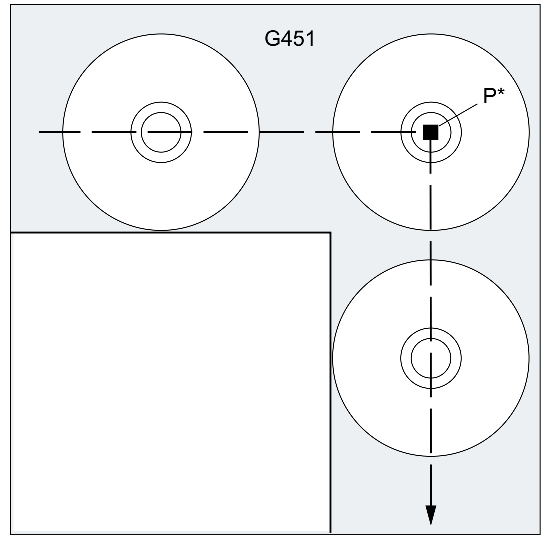
If the KONTT or KONTC block is the approach block rather than the retraction block, the contour is exactly the same, but it is machined in the opposite direction.

11.3 Compensation at the outside corners (G450, G451, DISC)

With tool radius compensation activated (`G41`/`G42`), command `G450` or `G451` can be used to define the course of the compensated tool path when traveling around outside corners:



With `G450`, the tool center point travels around the workpiece corner across an arc with tool radius.



With `G451`, the tool center point approaches the point of intersection of the two equidistants, which are located at a distance equivalent to the tool radius from the programmed contour. `G451` applies only to circles and straight lines.

Note

`G450`/`G451` is also used to define the approach path with `KONT` active and approach point behind the contour (see "Approaching and leaving contour (NORM, KONT, KONTC, KONTT) (Page 273)").

The `DISC` command can be used to distort the transition circles with `G450`, thereby producing sharper contour corners.

Syntax

```
G450 [DISC=<value>]
G451
```

Meaning

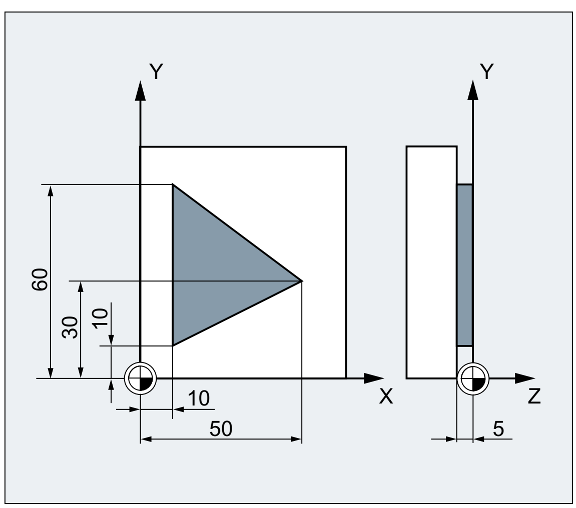
<table>
<tr><td>G450:</td><td colspan="4">G450 is used to travel around workpiece corners on a circular path.</td></tr>
<tr><td rowspan="5">DISC:</td><td colspan="4">Flexible programming of the circular path with G450 (optional)</td></tr>
<tr><td rowspan="4"><value>:</td><td>Type:</td><td colspan="2">INT</td></tr>
<tr><td>Range of values:</td><td colspan="2">0, 1, 2, ... 100</td></tr>
<tr><td rowspan="2">Meaning:</td><td>0</td><td>Transition circle</td></tr>
<tr><td>100</td><td>Intersection of the equidistant paths (theoretical value)</td></tr>
<tr><td>G451:</td><td colspan="4">G451 is used to approach the intersection point of the two equidistant paths in the case of workpiece corners. The tool backs off from the workpiece corner.</td></tr>
</table>

Note

DISC only applies with call of G450, but can be programmed in a previous block without G450. Both commands are modal.

Example

In the following example, a transition radius is programmed for all outside corners (corresponding to the programming of the corner behavior in block N30). This prevents the tool stopping and backing off at the change of direction.



Program code	**Comment**
`N10 G17 T1 G0 X35 Y0 Z0 F500`	`; Starting conditions.`
`N20 G1 Z-5`	`; Feed in tool.`
`N30 G41 KONT` **`G450`** `X10 Y10`	`;` **`Activate`** `TRC with KONT approach/retract mode and` **`corner behavior G450.`**
`N40 Y60`	`; Mill the contour.`
`N50 X50 Y30`	
`N60 X10 Y10`	

Program code	Comment
`N80 G40 X-20 Y50`	`; Deactivate compensation mode, retraction on transition circle.`
`N90 G0 Y100`	
`N100 X200 M30`	

Further information

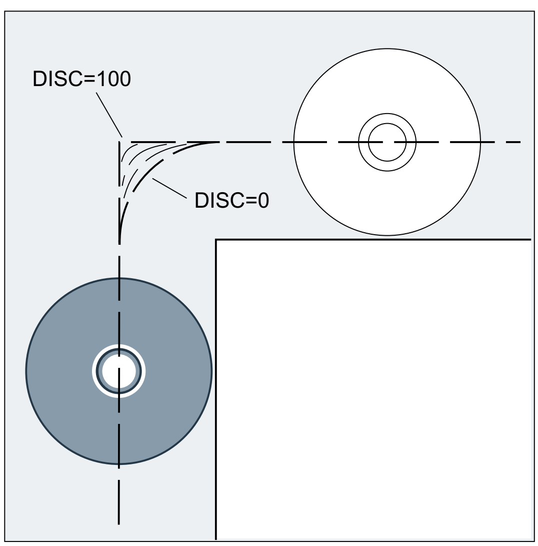
G450/G451

At intermediate point P*, the control executes operations such as infeed movements or switching functions. These operations are programmed in blocks inserted between the two blocks forming the corner.

With `G450` the transition circle belongs to the next travel command with respect to the data.

DISC

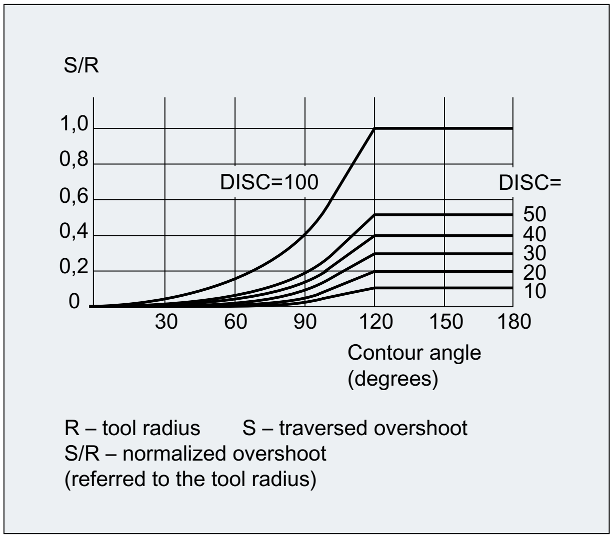
When `DISC` values greater than 0 are specified, intermediate circles are shown with a magnified height – the result is transition ellipses or parabolas or hyperbolas:



An upper limit can be defined in machine data – generally DISC=50.

Traversing behavior

When `G450` is activated and with acute contour angles and high `DISC` values, the tool is lifted off the contour at the corners. In the case of contour angles equal to or greater than 120°, there is uniform travel around the contour:

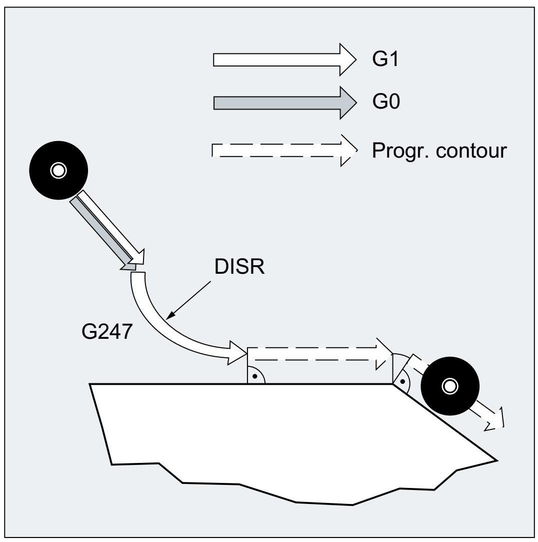


When G451 is activated and with acute contour angles, superfluous non-cutting tool paths can result from lift-off movements. A parameter can be used in the machine data to define automatic switchover to transition circle in such cases.

11.4 Smooth approach and retraction

11.4.1 Approach and retraction (G140 to G143, G147, G148, G247, G248, G347, G348, G340, G341, DISR, DISCL, DISRP, FAD, PM, PR)

The SAR (Smooth Approach and Retraction) function is used to achieve a tangential approach to the start point of a contour, regardless of the position of the start point.



This function is used preferably in conjunction with the tool radius compensation.

When the function is activated, the control calculates the intermediate points in such a way that the transition to the following block (or the transition from previous block during retraction) is performed in accordance with the specified parameters.

The approach movement consists of a maximum of four sub-movements. The starting point of the movement is called P_0, the end point P_4 in the following. Up to three intermediate points P_1, P_2 and P_3 can be between these points. Points P_0, P_3 and P_4 are always defined. Intermediate points P_1 and P_2 can be omitted, according to the parameters defined and the geometrical conditions. On retraction, the points are traversed in the reverse direction, i.e. starting at P_4 and ending at P_0.

Syntax

Smooth approach:

- With a straight line:
  `G147 G340/G341 ... DISR=..., DISCL=..., DISRP=... FAD=...`
- With a quadrant/semicircle:
  `G247/G347 G340/G341 G140/G141/G142/G143 ... DISR=... DISCL=... DISRP=... FAD=...`

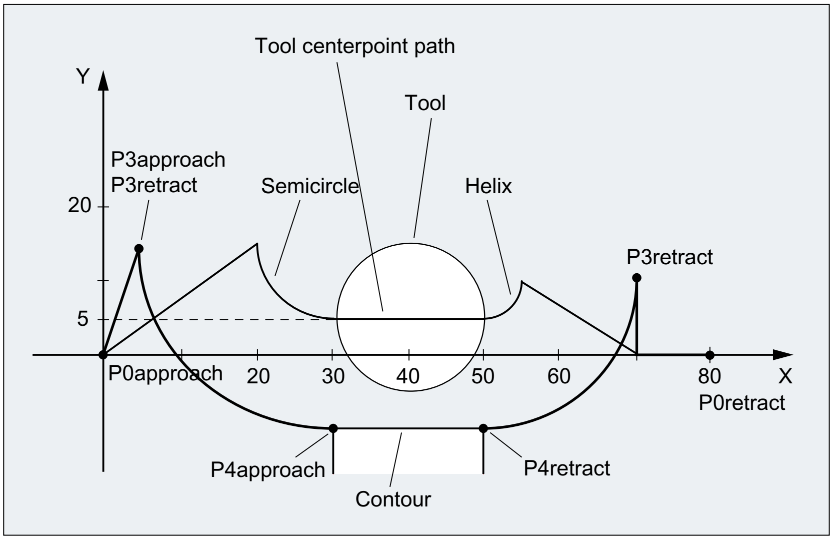
Smooth retraction:

- With a straight line:
  `G148 G340/G341 ... DISR=..., DISCL=..., DISRP=... FAD=...`
- With a quadrant/semicircle:
  `G248/G348 G340/G341 G140/G141/G142/G143 ... DISR=... DISCL=... DISRP=... FAD=...`

Meaning

`G147`:	Approach with a straight line
`G148`:	Retraction with a straight line
`G247`:	Approach with a quadrant
`G248`:	Retraction with a quadrant
`G347`:	Approach with a semicircle
`G348`:	Retraction with a semicircle
`G340`:	Approach and retraction in space (default setting)
`G341`:	Approach and retraction in the plane
`G140`:	Approach and retraction direction dependent on the current compensation side (default setting)
`G141`:	Approach from the left or retraction to the left
`G142`:	Approach from the right or retraction to the right
`G143`:	Approach and retraction direction dependent on the relative position of the start or end point to the tangent direction
`DISR=...`:	1. For approach and retraction with straight lines (G147/G148): Distance of the cutter edge from the starting point of the contour 2. For approach and retraction with circles (G247, G347/G248, G348): Radius of the tool center point path **Notice:** For REPOS with a semicircle, DISR is the circle diameter
`DISCL=...`:	Distance of the end point for the fast infeed motion from the machining plane DISCL=AC(...) Specification of the absolute position of the end point for the fast infeed motion
`DISCL=AC(...)`:	Specification of the absolute position of the end point for the fast infeed motion
`DISRP`:	Distance of point P1 (retraction plane) from the machining plane
`DISRP=AC(...)`:	Specification of the absolute position of point P1
`FAD=...`:	Speed of the slow feed movement The programmed value acts in accordance with the active feedrate type (G group 15).
`FAD=PM(...)`:	The programmed value is interpreted as linear feedrate (like G94) irrespective of the active feedrate type.
`FAD=PR(...)`:	The programmed value is interpreted as revolutional feedrate (like G95) irrespective of the active feedrate type.

Example



- Smooth approach (block N20 activated)
- Approach with quadrant (G247)
- Approach direction not programmed, G140 applies, i.e. TRC is active (G41)
- Contour offset OFFN=5 (N10)
- Current tool radius=10, and so the effective compensation radius for TRC=15, the radius of the SAR contour =25, with the result that the radius of the tool center path is equal to DISR=10
- The end point of the circle is obtained from N30, since only the Z position is programmed in N20
- Infeed movement
  - From Z20 to Z7 (DISCL=AC(7)) with rapid traverse.
  - Then to Z0 with FAD=200.
  - Approach circle in X-Y-plane and following blocks with F1500 (for this velocity to take effect in the following blocks, the active G0 in N30 must be overwritten with G1, otherwise the contour would be machined further with G0).
- Smooth retraction (block N60 activated)
- Retraction with quadrant (G248) and helix (G340)
- FAD not programmed, since irrelevant for G340
- Z=2 in the starting point; Z=8 in the end point, since DISCL=6
- When DISR=5, the radius of the SAR contour=20, the radius of the tool center point path=5

Retraction movements from Z8 to Z20 and the movement parallel to the X-Y plane to X70 Y0.

Program code	Comment
`$TC_DP1[1,1]=120`	`;Tool definition T1/D1`
`$TC_DP6[1,1]=10`	`; Radius`
`N10 G0 X0 Y0 Z20 G64 D1 T1 OFFN=5`	`; (P0 app)`
`N20 G41 G247 G341 Z0 DISCL=AC(7) DISR=10 F1500 FAD=200`	`; Approach (P3 app)`
`N30 G1 X30 Y-10`	`; (P4 app)`
`N40 X40 Z2`	
`N50 X50`	`; (P4 ret)`
`N60 G248 G340 X70 Y0 Z20 DISCL=6 DISR=5 G40 F10000`	`; Retraction (P3 ret)`
`N70 X80 Y0`	`; (P0 ret)`
`N80 M30`	

Further information

Selecting the approach and retraction contour

The approach and retraction contour are selected with the appropriate G command from the 2nd G group:

G147:	Approach with a straight line
G247:	Approach with a quadrant
G347:	Approach with a semicircle
G148:	Retraction with a straight line
G248:	Retraction with a quadrant
G348:	Retraction with a semicircle

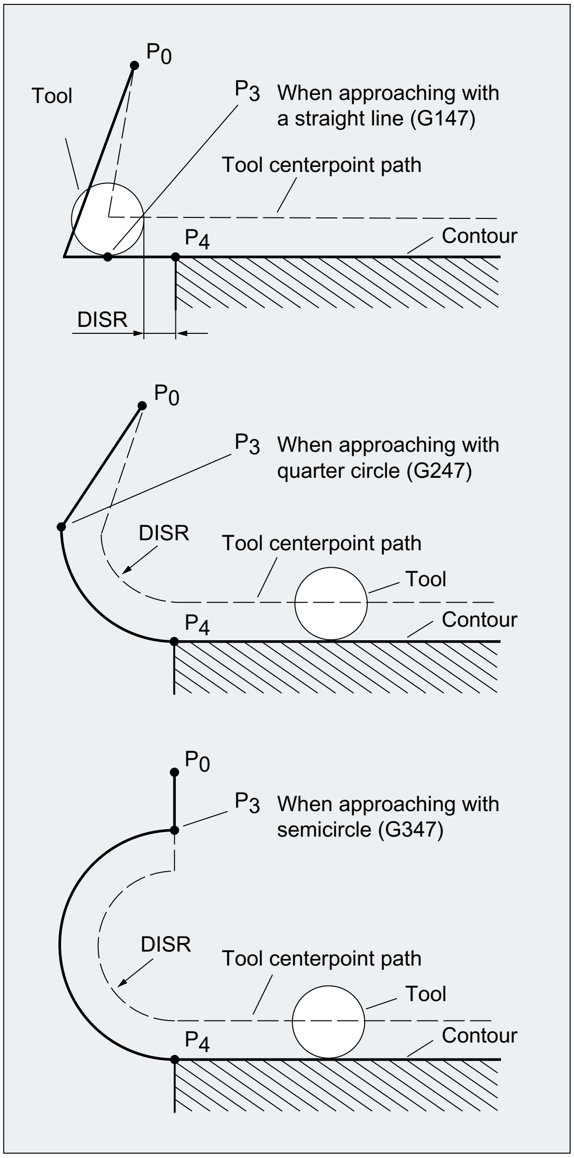


Figure 11-3 Approach movements with simultaneous activation of the tool radius compensation

Selecting the approach and retraction direction

Use the tool radius compensation (G140, default setting) to determine the approach and retraction direction with positive tool radius:

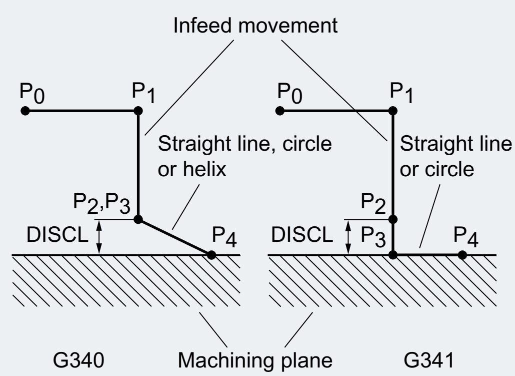
- G41 active → approach from left
- G42 active → approach from right

G141, G142 and G143 provide further approach options.

The G codes are only significant when the approach contour is a quadrant or a semicircle.

Motion steps between start point and end point (G340 and G341).

In all cases, the movements are made up of one or more straight lines and, depending on the G command for determining the approach contour, an additional straight line or a quadrant or semicircle. The two variants of the path segmentation are shown in the following figure:



Approach movement dependent on G340/G341

G340:	Approach with a straight line from point P_0 to point P_1. This straight line is parallel to the machining plane, if parameter DISRP has not been programmed. Infeed perpendicular to the machining plane from point P_1 to point P_3 to the safety clearance to the machining plane defined by the DISCL parameter. Approach end point P4 with the curve determined by the G command of the second group (straight line, circle, helix). If G247 or G347 is active (quadrant or semicircle) and start point P_3 is outside the machining plane defined by the end point P_4, a helix is inserted instead of a circle. Point P_2 is not defined or coincides with P_3. The circle plane or the helix axis is determined by the plane, which is active in the SAR block (G17/G18/G19), i.e. the projection of the start tangent is used by the following block, instead of the tangent itself, to define the circle. The movement from point P_0 to point P_3 takes place along two straight lines at the velocity valid before the SAR block.
G341:	Approach with a straight line from point P_0 to point P_1. This straight line is parallel to the machining plane, if parameter DISRP has not been programmed. Infeed perpendicular to the machining plane from point P_1 up to the safety clearance to the machining plane defined by the DISCL parameter in point P_2. Infeed perpendicular to the machining plane from point P_2 to point P_3. Approach end point with the curve determined by the G command of the second group. P_3 and P_4 are located within the machining plane, with the result that a circle is always inserted instead of a helix with G247 or G347.

In all cases that include the position of the active plane G17/G18/G19 (circular plane, helical axis, infeed motion perpendicular to the active plane), any active rotating frame is taken into account.

Length of the approach straight line or radius for approach circles (DISR)

- Approach/retract with straight lines

  DISR specifies the distance of the cutter edge from the starting point of the contour, i.e. the length of the straight line when TRC is active is the sum of the tool radius and the programmed value of DISR. The tool radius is only taken into account when it is positive. The resulting straight line length must be positive, i.e. negative values for DISR are allowed provided that the absolute value of DISR is less than the tool radius.
- Approach/retract with circles
  DISR specifies the radius of the tool center point path. If TRC is activated, a circle is produced with a radius that results in the tool center point path with the programmed radius.

Distance of point P2 from the machining plane (DISCL)

If the position of point P_2 is to be specified by an absolute reference on the axis perpendicular to the circle plane, the value must be programmed in the form `DISCL=AC(...)`.

The following applies for DISCL=0:

- With G340: The whole of the approach motion now only consists of two blocks (P_1, P_2 and P_3 are combined). The approach contour is formed by P_1 to P_4.
- With G341: The whole approach contour consists of three blocks (P_2 and P_3 are combined). If P_0 and P_4 are on the same plane, only two blocks result (infeed movement from P_1 to P_3 is omitted).
- The point defined by DISCL is monitored to ensure that it is located between P_1 and P_3, i.e. the sign must be identical for the component perpendicular to the machining plane in all motions that possess such a component.
- On detection of a reversal of direction, a tolerance defined by the machine data MD20204 $MC_SAR_CLEARANCE_TOLERANCE is permitted.

Distance of point P1 (retraction plane) from the machining plane (DISRP)

If the position of point P_1 is to be specified by an absolute reference on the axis perpendicular to the machining plane, the value must be programmed in the form `DISRP=AC(...)`.

If this parameter is not programmed, point P_1 has the same distance to the machining plane as point P_0, i.e. the approach straight line $P_0 \rightarrow P_1$ is parallel to the machining plane.

The system checks that the point defined by DISRP lies between P_0 and P_2, i.e. in all movements that have a component perpendicular to the machining plane (e.g. infeed movements, approach movements from P_3 to P_4), this component must have the same leading sign. It is not permitted to change direction. An alarm is output if this condition is violated.

On detection of a reversal of direction, a tolerance defined by the machine data MD20204 $MC_SAR_CLEARANCE_TOLERANCE is permitted. However, if P_1 is outside the range defined by P_0 and P_2, but the deviation is less than or equal to this tolerance, it is assumed that P_1 is in the plane defined by P_0 or P_2.

Programming of the end point

The end point is generally programmed with X... Y... Z...

The programming of the contour end point when approaching differs greatly from that for retraction. Both cases are therefore treated separately here.

Programming of end point P4 for approach

End point P_4 can be programmed in the actual SAR block. Alternatively, P_4 can be determined by the end point of the next traversing block. More blocks can be inserted between an SAR block and the next traversing block without moving the geometry axes.

Example:

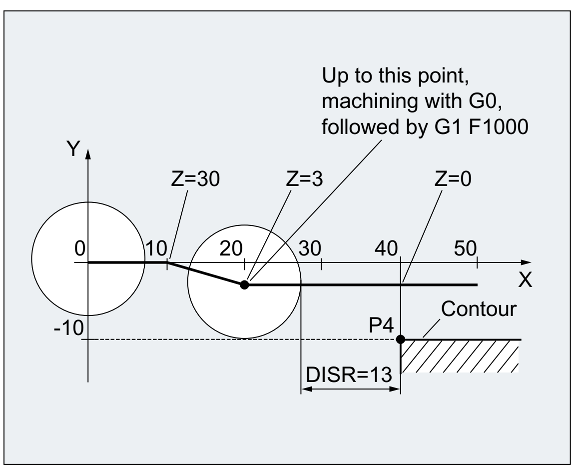
Program code	Comment
`$TC_DP1[1,1]=120`	`;Milling tool T1/D1`
`$TC_DP6[1,1]=7`	`;Tool with 7 mm radius`
`N10 G90 G0 X0 Y0 Z30 D1 T1`	
`N20 X10`	
`N30 G41 G147 DISCL=3 DISR=13 Z=0 F1000`	
`N40 G1 X40 Y-10`	
`N50 G1 X50`	
`...`	

`N30/N40` can be replaced by:

```
N30 G41 G147 DISCL=3 DISR=13 X40 Y-10 Z0 F1000
```

or

```
N30 G41 G147 DISCL=3 DISR=13 F1000
N40 G1 X40 Y-10 Z0
```



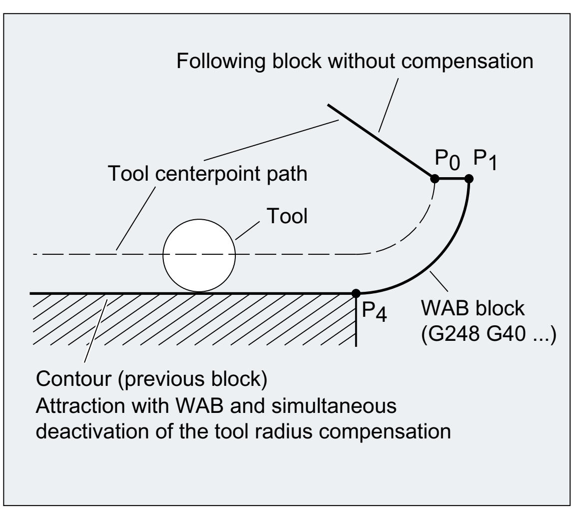
Programming of end point P0 for retraction

For retraction, the end point of the SAR contour cannot be programmed in a following block, i.e. the end position is always taken from the SAR block, irrespective of how many axes have been programmed. When determining the end point, a distinction is made between the following three cases:

1. No geometry axis is programmed in the SAR block. In this case, the contour ends at point P_1 (if DISRP has been programmed), at point P_2 (if DISCL, but not DISRP has been programmed) or point P_3 (if neither DICLS nor DISRP has been programmed).
   The position in the axes, which describe the machining plane, is determined by the retraction contour (end point of the straight line or arc). The axis component perpendicular to this is defined by DISCL or DISPR. If in this case both DISCL=0 and DISRP=0, the motion is completely in the plane, i.e. points P_0 to P_3 coincide.
2. Only the axis perpendicular to the machining plane is programmed in the SAR block. In this case, the contour ends at point P_0. If DISRP has been programmed (i.e. points P_0 and P_1 do not coincide), the straight line $P_1 \rightarrow P_0$ is perpendicular to the machining plane. The positions of the two other axes are determined in the same way as in 1.
3. At least one axis of the machining plane is programmed. The second axis of the machining plane can be determined modally from its last position in the preceding block.

The position of the axis perpendicular to the machining plane is generated as described in 1. or 2., depending on whether this axis is programmed or not. The position generated in this way defines the end point P_0. If the SAR retraction block is also used to deactivate the tool radius compensation, in the first two cases, an additional path component is inserted in the machining plane from P_1 to P_0 so that no movement is produced when the tool radius compensation is deactivated at the end of the retraction contour, i.e. this point defines the tool center point and not a position on a contour to be corrected. In case 3, no special measures are required for deselection of the tool radius compensation, because the programmed point P_0 already directly defines the position of the tool center point at the end of the complete contour.

The behavior in cases 1 and 2, i.e. when an end point is not explicitly programmed in the machining plane with simultaneous deselection of the tool radius compensation, is shown in the following figure:

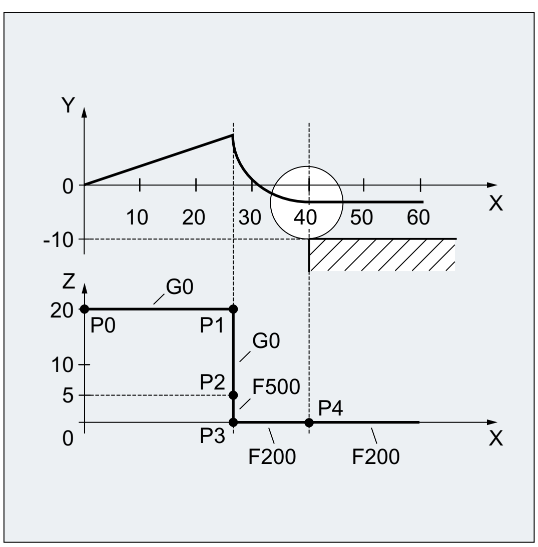


Approach and retraction velocities

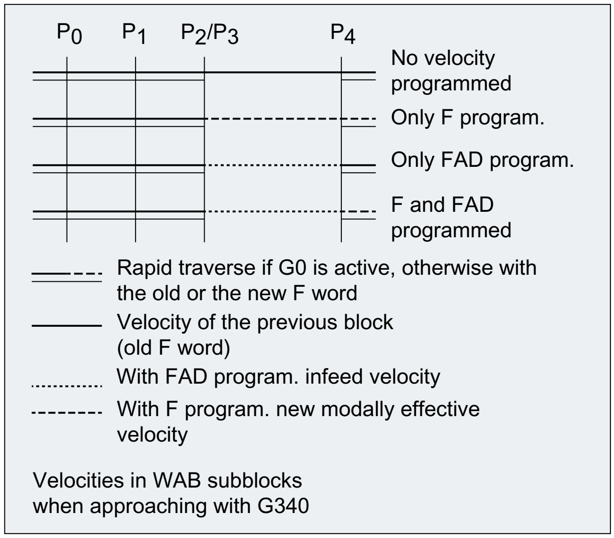
- Velocity of the previous block (G0)
  All motions from P_0 up to P_2 are executed at this velocity, i.e. the motion parallel to the machining plane and the part of the infeed motion up to the safety clearance.
- Programming with FAD
  Specification of the feedrate for
  - G341: Infeed movement perpendicular to the machining plane from P_2 to P_3
  - G340: From point P_2 or P_3 to P_4.
    If FAD is not programmed, this part of the contour is traversed at the speed which is active modally from the preceding block, in the event that no F command defining the speed is programmed in the SAR block.
- Programmed feedrate F
  This feedrate value is effective as of P_3 or P_2 if FAD is not programmed. If no F word is programmed in the SAR block, the speed of the previous block is active.

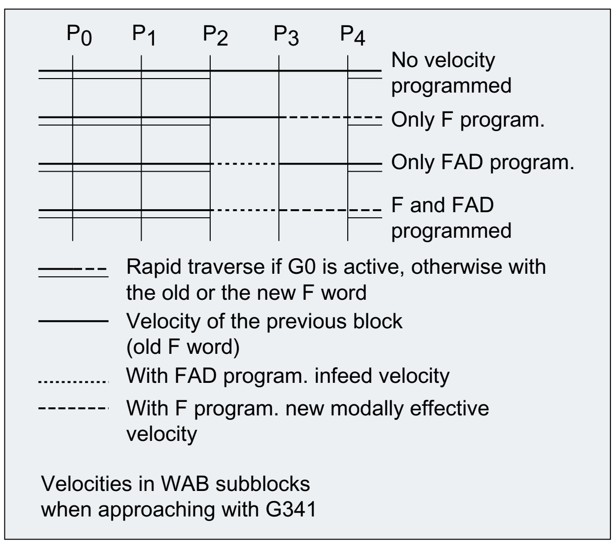
Example:

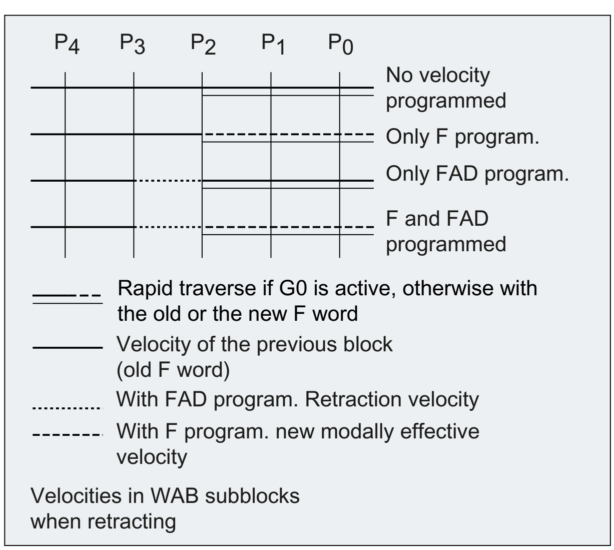
Program code	Comment
`$TC_DP1[1,1]=120`	`;Milling tool T1/D1`
`$TC_DP6[1,1]=7`	`;Tool with 7 mm radius`
`N10 G90 G0 X0 Y0 Z20 D1 T1`	
`N20 G41 G341 G247 DISCL=AC(5) DISR=13 FAD 500 X40 Y-10 Z=0 F200`	
`N30 X50`	
`N40 X60`	
`...`	



During retraction, the roles of the modally active feedrate from the previous block and the programmed feedrate value in the SAR block are reversed, i.e. the actual retraction contour is traversed with the old feedrate and a new speed programmed with the F word applies from P_2 up to P_0.


P0
P1
P2/P3
P4
No velocity programmed
Only F program.
Only FAD program.
F and FAD programmed
Rapid traverse if G0 is active, otherwise with the old or the new F word
Velocity of the previous block (old F word)
With FAD program. infeed velocity
With F program. new modally effective velocity
Velocities in WAB subblocks when approaching with G340


P0
P1
P2
P3
P4
No velocity programmed
Only F program.
Only FAD program.
F and FAD programmed
Rapid traverse if G0 is active, otherwise with the old or the new F word
Velocity of the previous block (old F word)
With FAD program. infeed velocity
With F program. new modally effective velocity
Velocities in WAB subblocks when approaching with G341


P4
P3
P2
P1
P0
No velocity programmed
Only F program.
Only FAD program.
F and FAD programmed
Rapid traverse if G0 is active, otherwise with the old or the new F word
Velocity of the previous block (old F word)
With FAD program. Retraction velocity
With F program. new modally effective velocity
Velocities in WAB subblocks when retracting

Reading positions

Points P_3 and P_4 can be read in the WCS as a system variable during approach.

- \$P_APR: reading P
- $_3$ (initial point)
- \$P_AEP: reading P
- $_4$ (contour starting point)
- \$P_APDV: read whether \$P_APR and \$P_AEP contain valid data

11.4.2 Approach and retraction with extended retraction strategies (G460, G461, G462)

In certain special geometrical situations, special extended approach and retraction strategies, compared with the previous implementation with activated collision detection for the approach and retraction block, are required in order to activate or deactivate tool radius compensation. A collision detection can result, for example, in a section of the contour not being completely machined, see following figure:

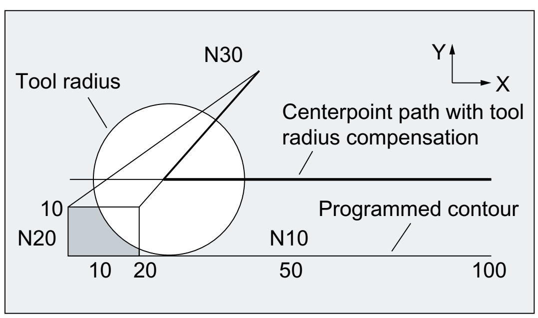


Figure 11-4 Retraction behavior with G460

Syntax

```
G460
G461
G462
```

Meaning

`G460`:	As previously (activation of the collision detection for the approach and retraction block).
`G461`:	Insertion of a circle in the TRC block, if it is not possible to have an intersection whose center point is in the end point of the uncorrected block, and whose radius is the same as the tool radius. Up to the intersection, machining is performed with an **auxiliary circle** around the contour end point (i.e. up to the end of the contour).
`G462`:	Insertion of a circle in the TRC block, if it is not possible to have an intersection; the block is extended by its end tangent (default setting). Machining is performed up to the **extension** of the last contour element (i.e. until shortly before the end of the contour).

Note

The approach behavior is symmetrical to the retraction behavior.

The approach/retraction behavior is determined by the state of the G command in the approach/ retraction block. The approach behavior can therefore be set independently of the retraction behavior.

Examples

Example 1: Retraction behavior with G460

The following example describes only the situation for deactivation of tool radius compensation: The behavior for approach is exactly the same.

Program code	Comment
`G42 D1 T1`	`; Tool radius 20 mm`
`...`	
`G1 X110 Y0`	
`N10 X0`	
`N20 Y10`	
`N30 G40 X50 Y50`	

Example 2: Approach with G461

Program code	Comment
`N10 $TC_DP1[1,1]=120`	`; Milling tool type`
`N20 $TC_DP6[1,1]=10`	`;Tool radius`
`N30 X0 Y0 F10000 T1 D1`	
`N40 Y20`	
`N50 G42 X50 Y5 G461`	
`N60 Y0 F600`	
`N70 X30`	
`N80 X20 Y-5`	

Program code	Comment
N90 X0 Y0 G40	
N100 M30	

Further information

G461

If no intersection is possible between the last TRC block and a preceding block, the offset curve of this block is extended with a circle whose center point lies at the end point of the uncorrected block and whose radius is equal to the tool radius.

The control attempts to cut this circle with one of the preceding blocks.

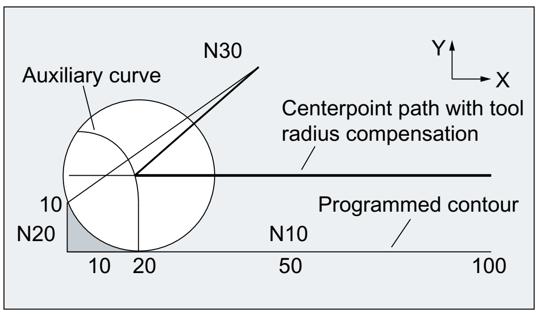


Figure 11-5 Retraction behavior with G461

Collision monitoring CDON, CDOF

If CDOF is active (see section Collision monitoring, CDON, CDOF), the search is aborted when an intersection is found, i.e., the system does not check whether further intersections with previous blocks exist.

If CDON is active, the search continues for further intersections after the first intersection is found.

An intersection point, which is found in this way, is the new end point of a preceding block and the start point of the deactivation block. The inserted circle is used exclusively to calculate the intersection and does not produce a traversing movement.

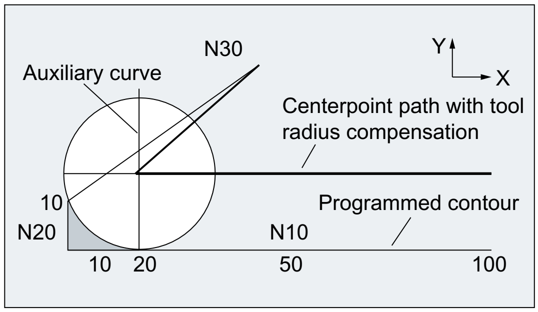
Note

If no intersection is found, alarm 10751 (collision danger) is output.

G462

If no intersection is possible between the last TRC block and a preceding block, a straight line is inserted, on retraction with G462 (initial setting), at the end point of the last block with tool radius compensation (the block is extended by its end tangent).

The search for the intersection is then identical to the procedure for G461.



Retraction behavior with G462 (see example)

With G462, the corner generated by N10 and N20 in the example program is not machined to the full extent actually possible with the tool used. However, this behavior may be necessary if the part contour (as distinct from the programmed contour), to the left of N20 in the example, is not permitted to be violated even with y values greater than 10 mm.

Corner behavior with KONT

If KONT is active (travel round contour at start or end point), the behavior differs according to whether the end point is in front of or behind the contour.

- **End point in front of contour**
  If the end point is in front of the contour, the retraction behavior is the same as with NORM. This property does not change even if the last contour block for G451 is extended with a straight line or a circle. Additional circumnavigation strategies to avoid a contour violation in the vicinity of the contour end point are therefore not required.
- **End point behind contour**
  If the end point is behind the contour, a circle or straight line is always inserted depending on G450/G451. In this case, G460-462 has no effect. If the last traversing block in this situation has no intersection with a preceding block, an intersection with the inserted contour element or with the straight line of the end point of the bypass circle to the programmed endpoint can result.
  If the inserted contour element is a circle (G450), and this forms an interface with the preceding block, this is equal to the interface that would occur with NORM and G461. In general, however, a remaining section of the circle still has to be traversed. For the linear part of the retraction block, no further calculation of intersection is required.
  In the second case, if no interface of the inserted contour element with the preceding blocks is found, the intersection between the retraction straight line and a preceding block is traversed.
  Therefore, a behavior that deviates from G460 can only occur with active G461 or G462 either if NORM is active or the behavior with KONT is geometrically identical to that with NORM.

11.5 Activation/deactivation of collision detection ("bottleneck detection") (CDON, CDOF, CDOF2)

The collision detection ("bottleneck detection") with active TRC is activated or deactivated in the NC program with the commands of G group 23.

Syntax

```
G41/G42 CDON
...
CDOF/CDOF2
```

Meaning

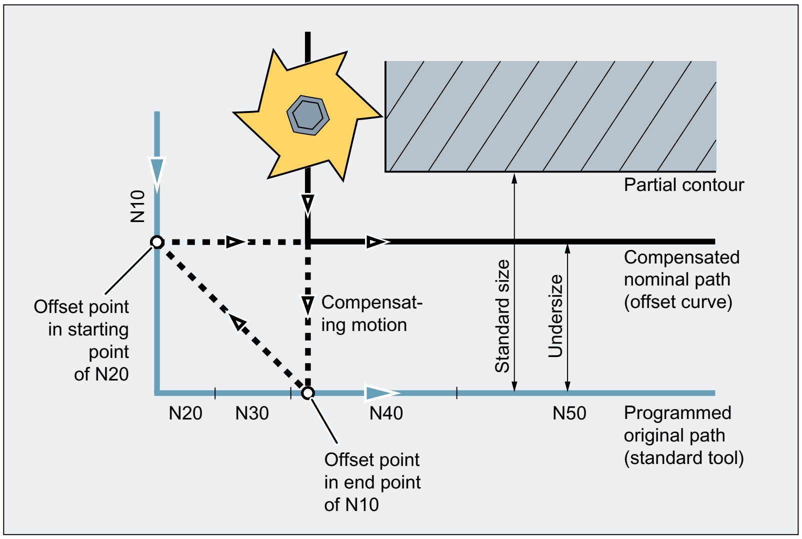
`CDON`:	Activating collision detection ("bottleneck detection") CDON performs a check over an adjustable (MD20240) number of blocks as to whether the tool paths of **non-adjacent** blocks intersect. This look-ahead function allows possible collisions to be detected in advance and permits the control to actively avoid them.
`CDOF`:	Deactivating collision detection ("bottleneck detection") With CDOF, a search is made in the **previous** traversing block (at inside corners) for a common intersection for the current block; if necessary the search is extended to even earlier blocks. If an intersection is found, no further blocks are examined. With outside corners, an intersection is always found between two consecutive blocks. **Note:** CDOF can be used to avoid the faulty detection of bottlenecks which may occur due to a lack of information in the NC program, for example.
`CDOF2`:	Deactivating collision detection for 3D circumferential milling The tool offset direction is determined from adjacent block parts with CDOF2. CDOF2 is only effective for 3D circumferential milling and has the same meaning as CDOF for all other types of machining (e.g. 3D face milling).

Effect of collision detection using an example

The NC program describes the center point path of a standard tool. The contour for a tool that is actually used results in undersize, which is shown unrealistically large to demonstrate the geometric relationships in the following figure.

The control also only has an overview of three blocks in the example:

MD20240 $MC_CUTCOM_MAXNUM_CHECK_BLOCKS = 3



Since an intersection exists only between the offset curves of the two blocks N10 and N40, the two blocks N20 and N30 would have to be omitted. In the example, the control does not know in block N40 if N10 has to be completely processed. Only a single block can therefore be omitted.

With active CDOF2, the compensation motion shown in the figure is executed and not stopped. In this situation, an active CDOF or CDON would result in an alarm.

11.6 2 1/2 D tool offset (CUT2D, CUT2DD, CUT2DF, CUT2DFD)

The 2½ D tool radius compensation should be used if, when machining inclined surfaces, the **workpiece** is to be rotated, and not the tool alignment. This function is activated using commands `CUT2D`, `CUT2DD`, `CUT2DF` oder `CUT2DFD`.

Tool length offset

The tool length compensation is always taken into account referred to the machining plane that is not rotated and is fixed in space.

2½ D tool radius compensation for contour tools

2½ D tool radius compensation for contour tools is activated, if, together with `CUT2D`, `CUT2DD`, `CUT2DF` or `CUT2DFD`, one of the two commands `G41` (tool radius compensation left of the contour) or `G42` (tool radius compensation right of the contour) is programmed. It is used for automatic cutting-edge selection in the case of non-axially symmetrical tools that can be used for piece-by-piece machining of individual contour segments.

Note

If 2½ D tool radius compensation is not activated, a contour tool behaves like a standard tool, which only has the first cutting edge.

2½ tool radius compensation referred to a differential tool

2½ D tool radius compensation, referred to a differential tool, is activated using the`CUT2DD` or `CUT2DFD` commands. It should be applied if the programmed contour refers to the center point path of a differential tool, and a tool other than a differential tool is used for machining. When calculating the 2½ D tool radius compensation, only the wear of the radius of the active tool ($TC_DP_15) and the possibly programmed tool offsetOFFN (Page 263) and TOFFR (Page 84) are taken into account. The basic radius ($TC_DP6) of the active tool is **not** taken into account.

Syntax

```
CUT2D
CUT2DD
CUT2DF
CUT2DFD
```

Meaning

`CUT2D`:	Activating the 2½ D radius compensation
`CUT2DD`:	Activating the 2½ D radius compensation referred to a differential tool
`CUT2DF`:	Activating 2½ D radius compensation, tool radius compensation relative to the current frame and/or inclined plane
`CUT2DFD`:	Activating 2½ D radius compensation, tool radius compensation relative to the current frame and/or inclined plane

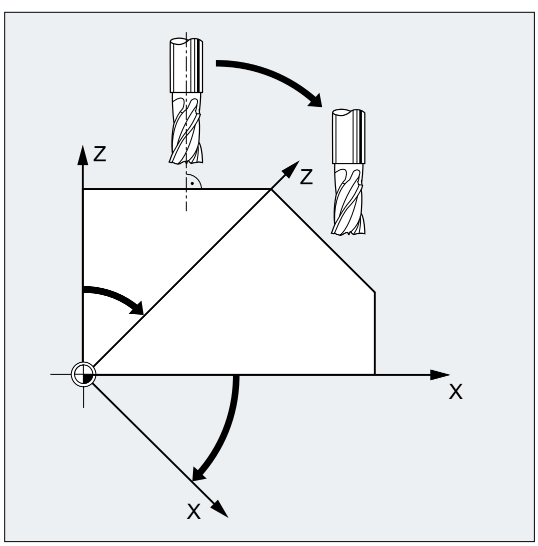
Further information

Contour tools

- Enabling
  Tool radius compensation for contour tools is enabled on a channel-specific basis using:
  MD28290 $MC_MM_SHAPED_TOOLS_ENABLE
- Tool type
  Contour tool types are defined on a channel-specific basis using:
  MD20370 $MC_SHAPED_TOOL_TYPE_NO
- Cutting edges
  A number of cutting edges (D numbers) can be assigned to each contour tool in any sequence. The maximum number of cutting edges per tool is parameterized using:
  MD18106 $MN_MM_MAX_CUTTING_EDGE_PERTOOL
  The first cutting edge of a contour tool is the cutting edge, which is selected when activating the tool. If, e.g. in a program, using the commands `T3 D5`, the fifth cutting edge (D5) of the third tool (T3) is activated, then D5 and the following cutting edges define with one part, or altogether, the contour tool. The cutting edges located before D5 are ignored.

2½ D tool radius compensation without rotating the correction plane (CUT2D, CUT2DD)

If a frame that contains a rotation is programmed, then for `CUT2D` or `CUT2DD`, the plane in which the tool radius compensation (correction plane) takes place **is not rotated at the same time**. The tool radius compensation is taken into account, referred to the **non rotated** machining plane (`G17`, `G18`, `G19`). The tool length compensation acts relative to the correction plane.

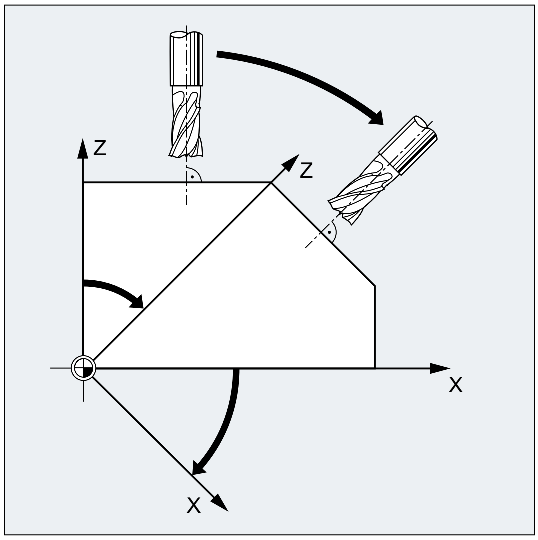


For machining inclined surfaces, the tool offsets must be appropriately defined or calculated based on the functions for "Tool length compensation for tools that can be orientated".

2½ D tool radius compensation with rotation of the compensation plane (CUT2DF, CUT2DFD)

If a frame is programmed that contains a rotation, then for `CUT2D` or `CUT2DD`, the plane in which the tool radius compensation takes place (correction plane)**is also rotated**. The tool radius compensation is taken into account, referred to the **rotated** machining plane (`G17`, `G18`, `G19`). However, the tool length compensation still acts relative to the **non-rotated** machining plane.

Precondition: At the machine, the tool orientation must be able to be adjusted perpendicular to the rotated machining plane, and set for machining.



Note

The tool length compensation continues to be active relative to the non-rotated working plane.

Reference

Function Manual, Basic Functions; Tool Offset (W1)

11.7 Keep tool radius compensation constant (CUTCONON, CUTCONOF)

The "Keep tool radius compensation constant" function is used to suppress tool radius compensation for a number of blocks, whereby a difference between the programmed and the actual tool center path traveled set up by tool radius compensation in the previous blocks is retained as the compensation. It can be an advantage to use this method when several traversing blocks are required during line milling in the reversal points, but the contours produced by the tool radius compensation (follow strategies) are not wanted. It can be used independently of the type of tool radius compensation ($2^1/_2$D, 3D face milling, 3D circumferential milling).

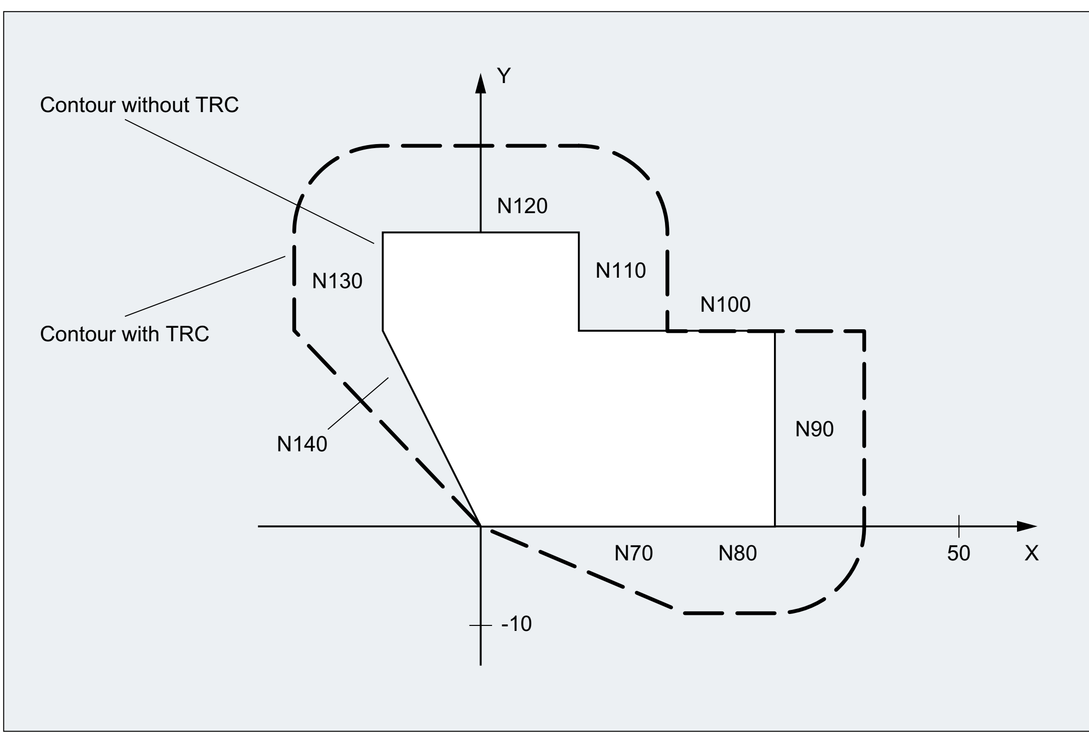
Syntax

```
CUTCONON
CUTCONOF
```

Meaning

`CUTCONON`:	Command to activate the "Keep tool radius compensation constant" function
`CUTCONOF`:	Command to deactivate the "Keep tool radius compensation constant" function

Example



Program code	Comment
N10	; Definition of tool d1.

Program code	**Comment**
N20 $TC_DP1[1,1] = 110	; Type
N30 $TC_DP6[1,1]= 10.	; Radius
N40	
N50 X0 Y0 Z0 G1 G17 T1 D1 F10000	
N60	
N70 X20 G42 NORM	
N80 X30	
N90 Y20	
N100 X10 CUTCONON	; Activation of the compensation suppression.
N110 Y30 KONT	; If required, insert bypass circle when deactivating the compensation suppression.
N120 X-10 CUTCONOF	
N130 Y20 NORM	; No bypass circle when deactivating the TRC.
N140 X0 Y0 G40	
N150 M30	

Further information

Tool radius compensation is normally active before the compensation suppression and is still active when the compensation suppression is deactivated again. In the last traversing block before CUTCONON, the offset point in the block end point is approached. All following blocks in which offset suppression is active are traversed without offset. However, they are offset by the vector from the end point of the last offset block to its offset point. These blocks can have any type of interpolation (linear, circular, polynomial).

The deactivation block of the compensation suppression, i.e. the block that contains CUTCONOF, is compensated normally. It starts in the offset point of the starting point. One linear block is inserted between the end point of the previous block, i.e. the last programmed traversing block with active CUTCONON, and this point.

Circular blocks, for which the circle plane is perpendicular to the compensation plane (vertical circles), are treated as though they had CUTCONON programmed. This implicit activation of the offset suppression is automatically canceled in the first traversing block that contains a traversing motion in the offset plane and is not such a circle. Vertical circle in this sense can only occur during circumferential milling.

11.8 Tools with a relevant cutting edge position

In the case of tools with a relevant tool point direction (turning and grinding tools - tool types 400-599; see Section "Sign evaluation wear"), a change from G40 to G41/G42 or vice-versa is treated as a tool change. If a transformation is active (e.g., TRANSMIT), this leads to a preprocessing stop (decoding stop) and hence possibly to deviations from the intended part contour.

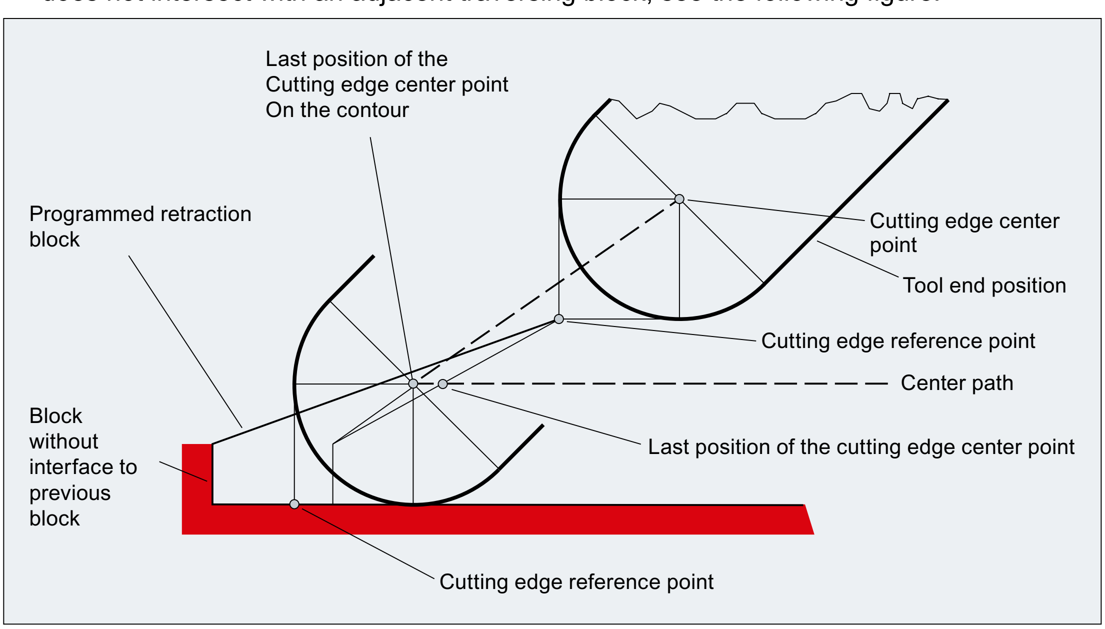
This original functionality changes with regard to:

1. Preprocessing stop on TRANSMIT
2. Calculation of intersection points at approach and retraction with KONT
3. Tool change with active tool radius compensation
4. Tool radius compensation with variable tool orientation at transformation

Further information

The original functionality has been modified as follows:

- A change from G40 to G41/G42 and vice-versa is no longer treated as a tool change. Therefore, a preprocessing stop no longer occurs with TRANSMIT.
- The straight line between the tool edge center points at the block start and block end is used to calculate intersection points with the approach and retraction block. The difference between the tool edge reference point and the tool edge center point is superimposed on this movement.
  On approach and retraction with KONT (tool circumnavigates the contour point, see above subsection "Contour approach and retraction"), superimposition takes place in the linear part block of the approach or retraction motion. The geometric conditions are therefore identical for tools with and without a relevant tool point direction. Deviations from the previous behavior occur only in relatively rare cases where the approach or retraction block does not intersect with an adjacent traversing block, see the following figure:



- In circle blocks and in motion blocks containing rational polynomials with a denominator degree > 4, it is not permitted to change a tool with active tool radius compensation in cases where the distance between the tool edge center point and the tool edge reference point changes. With other types of interpolation, it is now possible to change when a transformation is active (e.g., TRANSMIT).
- For tool radius compensation with variable tool orientation, the transformation from the tool edge reference point to the tool edge center point can no longer be performed by means of a simple zero offset. Tools with a relevant tool point direction are therefore not permitted for 3D peripheral milling (an alarm is output).

Note

The subject is irrelevant with respect to face milling as only defined tool types without relevant tool point direction are permitted for this operation anyway. (A tool with a type, which has not been explicitly approved, is treated as a ball end mill with the specified radius. A tool point direction parameter is ignored).

Path action 12

12.1 Exact stop (G60, G9, G601, G602, G603)

In exact stop traversing mode, all path axes and special axes involved in the traversing motion that are not traversed modally, are decelerated at the end of each block until they come to a standstill.

Exact stop is used when sharp outside corners have to be machined or inside corners finished to exact dimensions.

The exact stop specifies how exactly the corner point has to be approached and when the transition is made to the next block:

- "Exact stop fine"
  The block change is performed as soon as the axis-specific tolerance limits for "Exact stop fine" are reached for all axes involved in the traversing motion.
  "Exact stop fine" is set via: MD36010 $MA_STOP_LIMIT_FINE[<Axis>]
- "Exact stop coarse"
  The block change is performed as soon as the axis-specific tolerance limits for "Exact stop coarse" are reached for all axes involved in the traversing motion.
  "Exact stop coarse" is set via: MD36000 $MA_STOP_LIMIT_COARSE[<Axis>]
- "Interpolator end"
  The block change is performed as soon as the control has calculated a set velocity of zero for all axes involved in the traversing motion. The actual position or the following error of the axes involved are not taken into account

Syntax

```
G60 ...
G9 ...
G601/G602/G603, etc.
```

Meaning

`G60`:	Command for activation of the **modal** exact stop
`G9`:	Command for activation of the **non-modal** exact stop
`G601`:	Command for activation of the exact stop criterion "**Exact stop fine**"
`G602`:	Command for activation of the exact stop criterion "**Exact stop coarse**"
`G603`:	Command for activation of the exact stop criterion "**Interpolator end**"

Note

The commands for activating the exact stop criteria (`G601`/`G602`/`G603`) are only effective if `G60` or `G9` is active.

Example

Program code	Comment
`N5 G602`	`; Criterion "Exact stop coarse" selected.`
`N10 G0 G60 Z...`	`; Exact stop modal active.`
`N20 X... Z...`	`; G60 continues to act.`
`...`	
`N50 G1 G601`	`; Criterion "Exact stop fine" selected.`
`N80 G64 Z...`	`; Switchover to continuous-path mode.`
`...`	
`N100 G0 G9`	`; Exact stop acts only in this block.`
`N110 ...`	`; Continuous-path mode active again.`

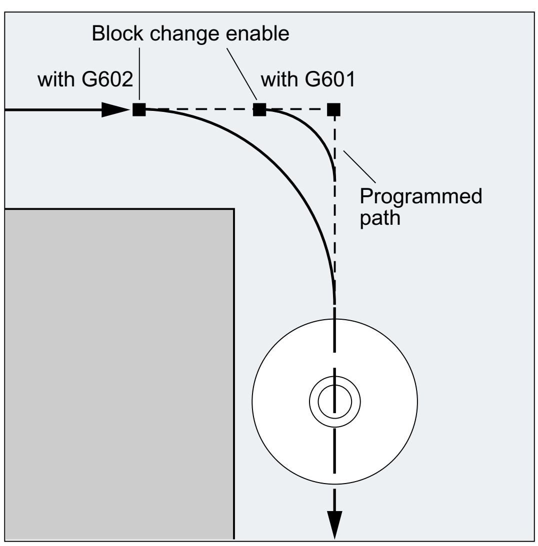
Further information

G60, G9

`G9` generates the exact stop in the current block, `G60` in the current block and in all following blocks.

Continuous-path-mode commands `G64` or `G641` - `G645` are used to deactivate `G60`.

G601, G602



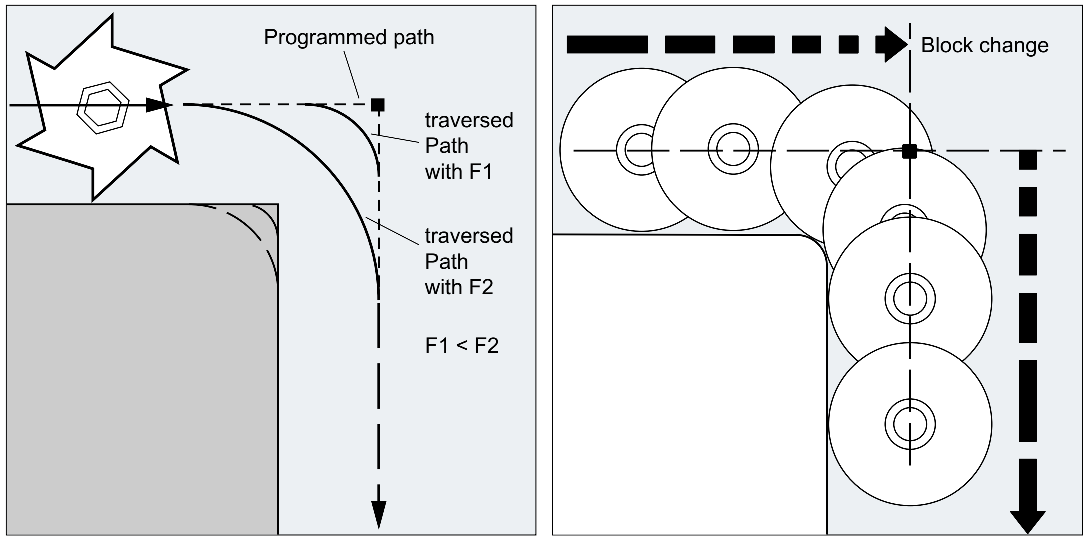
The movement is decelerated and stopped briefly at the corner point.

Note

Do not set the limits for the exact stop criteria any tighter than necessary. The tighter the limits, the longer it takes to position and approach the target position.

G603

The block change is initiated when the control has calculated a set velocity of zero for the axes involved. At this point, the actual value lags behind by a proportionate factor depending on the dynamic response of the axes and the path velocity. The workpiece corners can now be rounded.



Configured exact stop criterion

For `G0` and the other commands of the 1st G group, for specific channels it can be set that contrary to the programmed exact stop criterion a preset criterion should be used automatically (see machine manufacturer's specifications).

References

Function Manual, Basic Functions, Continuouspath Mode, Exact Stop, Look Ahead (B1)

12.2 Continuous-path mode (G64, G641, G642, G643, G644, G645, ADIS, ADISPOS)

In continuous-path mode, the path velocity at the end of the block (for the block change) is not decelerated to a level which would permit the fulfillment of an exact stop criterion. The objective of this mode is, in fact, to avoid rapid deceleration of the path axes at the block-change point so that the axis velocity remains as constant as possible when the program moves to the next block. To achieve this objective, the "Look-head" function is also activated when continuous-path mode is selected.

Continuous-path mode with smoothing facilitates the tangential shaping and/or smoothing of angular block transitions caused by local changes in the programmed contour.

Continuous path mode:

- Rounds the contour
- Reduces machining times by eliminating braking and acceleration processes that are required to fulfill the exact-stop criterion
- Improves cutting conditions because of the more constant velocity

Continuous-path mode is suitable if:

- A contour needs to be traversed as quickly as possible (e.g. with rapid traverse)
- The exact contour may deviate from the programmed contour within a specific tolerance for the purpose of obtaining a continuous contour

Continuous-path mode is not suitable if:

- A contour needs to be traversed precisely
- An absolutely constant velocity is required

Note

Continuous-path mode is interrupted by blocks which trigger a preprocessing stop implicitly, e.g. due to:

- Access to specific machine status data ($A...)
- Auxiliary function outputs

Syntax

```
G64 ...
G641 ADIS=…
G641 ADISPOS=…
G642 ...
G643 ...
G644 ...
G645 ...
```

Meaning

`G64`:	Continuous-path mode with reduced velocity as per the overload factor
`G641`:	Continuous-path mode with smoothing as per distance criterion
`ADIS=...`:	Distance criterion with `G641` for path functions `G1`, `G2`, `G3`, etc.
`ADISPOS=...`:	Distance criterion with `G641` for rapid traverse `G0`
	The distance criterion (= rounding clearance) `ADIS` or `ADISPOS` describes the maximum distance the rounding block may cover before the end of the block, or the distance after the end of block within which the rounding block must be terminated respectively. **Note:** If `ADIS/ADISPOS` is not programmed, a value of "zero" applies and the traversing behavior therefore corresponds to `G64`. The rounding clearance is automatically reduced (by up to 36%) for short traversing distances.
`G642`:	Continuous-path mode with smoothing within the defined tolerances In this mode, under normal circumstances smoothing takes place within the maximum permissible path deviation. However, instead of these axis-specific tolerances, observation of the maximum contour deviation (contour tolerance) or the maximum angular deviation of the tool orientation (orientation tolerance) can be configured. **Note:** Expansion to include contour and orientation tolerance is only supported on systems featuring the "Polynomial interpolation" option.
`G643`:	Continuous-path mode with smoothing within the defined tolerances (block-internal) `G643` differs from `G642` in that is not used to generate a separate rounding block; instead, axis-specific block-internal rounding movements are inserted. The rounding clearance can be different for each axis.
`G644`:	Continuous-path mode with smoothing with maximum possible dynamic response **Note:** `G644` is not available with an active kinematic transformation. The system switches internally to `G642`.
`G645`:	Continuous-path mode with smoothing and tangential block transitions within the defined tolerances `G645` has the same effect on corners as `G642`. With `G645`, rounding blocks are also only generated on tangential block transitions if the curvature of the original contour exhibits a jump in at least one axis.

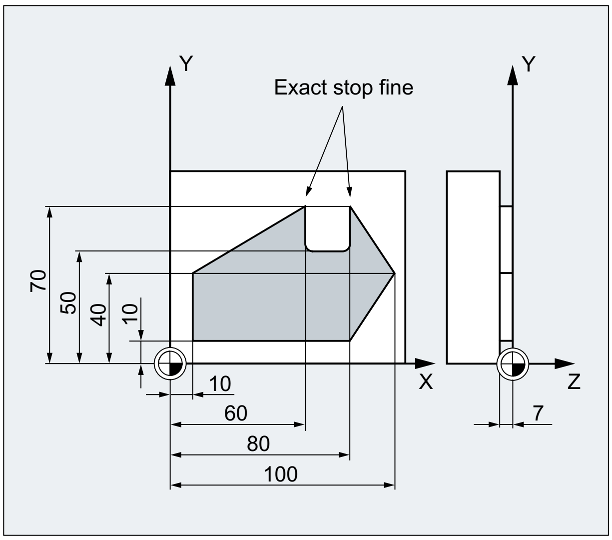
Note

Rounding cannot be used as a substitute for smoothing (`RND`). The user should not make any assumptions with respect to the appearance of the contour within the rounding area. The type of rounding can depend on dynamic conditions, e.g. on the tool path velocity. Rounding on the contour is therefore only practical with small `ADIS` values. `RND` must be used if a defined contour is to be traversed at the corner.

Note

If a rounding movement initiated by `G641`, `G642`, `G643`, `G644` or `G645` is interrupted, the starting or end point of the original traversing block (as appropriate for REPOS mode) will be used for subsequent repositioning (REPOS), rather than the interruption point.

Example



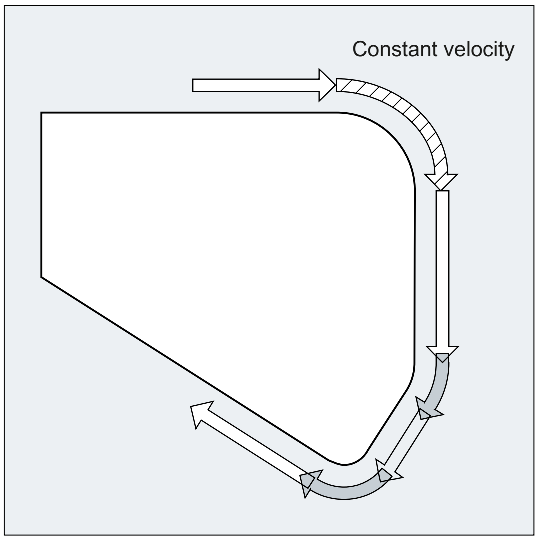
The two outside corners on the groove are to be approached exactly. Otherwise machining should be performed in continuous-path mode.

Program code	Comment
N05DIAMOF	; Radius as dimension
N10 G17 T1 G41 G0 X10 Y10 Z2 S300 M3	; Approach starting position, acti-vate spindle, path compensation.
N20 G1 Z-7 F8000	; Feed in tool.
N30 G641 ADIS=0.5	; Contour transitions are smoothed.
N40 Y40	
N50 X60 Y70 G60 G601	; Approach position exactly with ex-act stop fine.
N60 Y50	
N70 X80	
N80 Y70	
N90 G641 ADIS=0.5 X100 Y40	; Contour transitions are smoothed.
N100 X80 Y10	
N110 X10	
N120 G40 G0 X-20	; Deactivate path compensation
N130 Z10 M30	; Retract tool, end of program.

Further information

Continuous-path mode G64

In continuous-path mode, the tool travels across tangential contour transitions with as constant a path velocity as possible (no deceleration at block boundaries). LookAhead deceleration is applied before corners and blocks with exact stop.



Corners are also traversed at a constant velocity. In order to minimize the contour error, the velocity is reduced according to an acceleration limit and an overload factor.

Note

The extent of smoothing the contour transitions depends on the feedrate and the overload factor. The overload factor can be set in MD32310 $MA_MAX_ACCEL_OVL_FACTOR.

Setting MD20490 $MC_IGNORE_OVL_FACTOR_FOR_ADIS means that block transitions will always be rounded irrespective of the set overload factor.

The following points should be noted in order to prevent an undesired stop in path motion (relief cutting):

- Auxiliary functions, which are enabled after the end of the motion or before the next motion, interrupt the continuous path mode (exception: fast auxiliary functions).
- Positioning axes always traverse according to the exact stop principle, positioning window fine (as for `G601`). If an NC block has to wait for positioning axes, continuous-path mode is interrupted on the path axes.

However, intermediate blocks containing only comments, calculation blocks or subprogram calls do not affect continuous-path mode.

Note

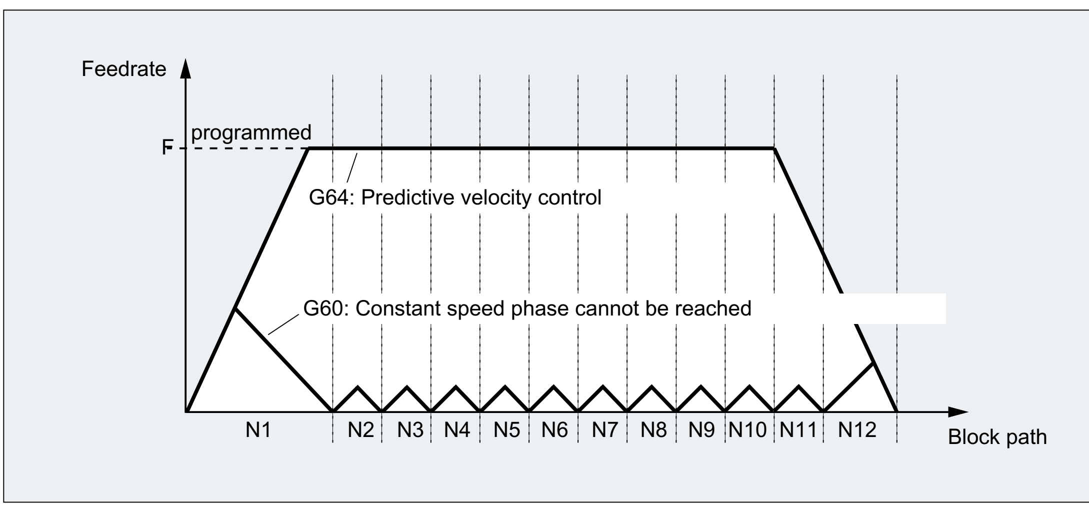
If `FGROUP` does not contain all the path axes, there is often a step change in the velocity at block transitions for those axes excluded from FGROUP; the control limits this change in velocity to the permissible values set in MD32300 $MA_MAX_AX_ACCEL and MD32310 $MA_MAX_ACCEL_OVL_FACTOR. This braking operation can be avoided through the application of a rounding function, which "smoothes" the specific positional interrelationship between the path axes.

LookAhead predictive velocity control

In continuous-path mode, the control automatically determines the velocity control for several NC blocks in advance. This enables acceleration and deceleration across multiple blocks with almost tangential transitions.

Look Ahead is particularly suitable for the machining of motion sequences comprising short traverse paths with high path feedrates.

The number of NC blocks included in the Look Ahead calculation can be defined in machine data.



Continuous-path mode with smoothing as per distance criterion (G641)

With `G641`, the control inserts transition elements at contour transitions. The rounding clearance `ADIS` (or `ADISPOS` for `G0`) specifies the maximum extent to which the corners can be rounded. Within this rounding clearance, the control is free to ignore the path construct and replace it with a dynamically optimized distance.

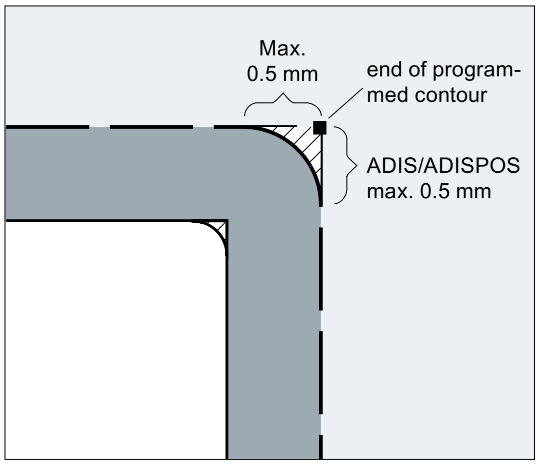
Disadvantage: Only one `ADIS` value is available for all axes.

The effect of `G641` is similar to `RNDM`; however, it is not restricted to the axes of the working plane.

Like `G64`, `G641` works with LookAhead predictive velocity control. Corner rounding blocks with a high degree of curvature are approached at reduced velocity.

Example:

Program code	Comment
`N10 G641 ADIS=0.5 G1 X... Y...`	`; The rounding block must begin no more than 0.5 mm before the programmed end of the block and must finish 0.5 mm after the end of the block. This setting remains modal.`



Note

Smoothing cannot and should not replace the functions for defined smoothing (`RND`, `RNDM`, `ASPLINE`, `BSPLINE`, `CSPLINE`).

Smoothing with axial precision with G642

With G642, smoothing does not take place within a defined `ADIS` range, but the axial tolerances defined with MD33100 $MA_COMPRESS_POS_TOL are complied with. The rounding clearance is determined based on the shortest rounding clearance of all axes. This value is taken into account when generating a rounding block.

Block-internal smoothing with G643

The maximum deviations from the precise contour in the case of smoothing with `G643` are defined for each axis using machine data MD33100 $MA_COMPRESS_POS_TOL.

`G643` is not used to generate a separate rounding block, but axis-specific block-internal rounding motions are inserted. In the case of `G643`, the rounding clearance of each axis can be different.

Smoothing with contour and orientation tolerance with G642/G643

MD20480 $MC_SMOOTHING_MODE can be used to configure rounding with `G642` and `G643` so that instead of the axis-specific tolerances, a contour tolerance and an orientation tolerance can be applied.

The contour tolerance and orientation tolerance are set in the channel-specific setting data:

SD42465 $SC_SMOOTH_CONTUR_TOL (maximum contour deviation)

SD42466 $SC_SMOOTH_ORI_TOL (maximum angular deviation of the tool orientation)

The setting data can be programmed in the NC program; this means that it can be specified differently for each block transition. Very different specifications for the contour tolerance and the tolerance of the tool orientation can only take effect with `G643`.

Note

Expansion to include contour and orientation tolerance is only supported on systems featuring the "Polynomial interpolation" option.

Note

An orientation transformation must be active for smoothing within the orientation tolerance.

Corner rounding with greatest possible dynamic response in G644

Smoothing with maximum possible dynamic response is configured in the thousands place with MD20480 $MC_SMOOTHING_MODE.

Value	Meaning
0	Specification of maximum axial deviations with: MD33100 $MA_COMPRESS_POS_TOL
1	Specification of maximum rounding clearance by programming: `ADIS=...` or `ADISPOS=...`
2	Specification of the maximum possible frequencies of each axis occurring in the rounding area with: MD32440 $MA_LOOKAH_FREQUENCY The rounding area is defined such that no frequencies in excess of the specified maximum can occur while the rounding motion is in progress.
3	When rounding with `G644`, neither the tolerance nor the rounding distance are monitored. Each axis traverses around a corner with the maximum possible dynamic response. With `SOFT`, both the maximum acceleration and the maximum jerk of each axis is maintained. With the `BRISK` command, the jerk is not limited; instead, each axis travels at the maximum possible acceleration.

Smoothing of tangential block transitions with G645

With `G645`, the smoothing motion is defined so that the acceleration of all axes involved remains smooth (no jumps) and the parameterized maximum deviations from the original contour (MD33120 $MA_PATH_TRANS_POS_TOL) are not exceeded.

In the case of angular non-tangential block transitions, the smoothing behavior is the same as with `G642`.

No intermediate rounding blocks

An intermediate rounding block is not inserted in the following cases:

- The axis stops between the two blocks.
  This occurs when:
  - The following block contains an auxiliary function output before the motion.
  - The following block does not contain a path motion.
  - An axis is traversed for the first time as a path axis for the following block when it was previously a positioning axis.
  - An axis is traversed for the first time as a positioning axis for the following block when it was previously a path axis.
  - The previous block traverses geometry axes and the following block does not.
  - The following block traverses geometry axes and the previous block does not.
  - Before tapping, the following block uses `G33` as preparatory function and the previous block does not.
  - A change is made between `BRISK` and `SOFT`.
  - Axes involved in the transformation are not completely assigned to the path motion (e.g. for oscillation, positioning axes).
- The rounding block would slow down the part program execution.
  This occurs:
  - Between very short blocks.
    Since each block requires at least one interpolator clock cycle, the added intermediate block would double the machining time.
  - If a block transition `G64` (continuous-path mode without smoothing) can be traversed without a reduction in velocity.
    Rounding would increase the machining time. This means that the value of the permitted overload factor (MD32310 $MA_MAX_ACCEL_OVL_FACTOR) affects whether a block transition is rounded or not. The overload factor is only taken into account for corner rounding with `G641`/`G642`. The overload factor has no effect in the case of smoothing with `G643` (this behavior can also be set for `G641` and `G642` by setting MD20490 $MC_IGNORE_OVL_FACTOR_FOR_ADIS to TRUE).

- Rounding is not parameterized.
  This occurs when:
  - For `G641` in `G0` blocks `ADISPOS = 0` (default!)
  - For `G641` in non-`G0` blocks `ADIS = 0` (default!)
  - For `G641` on transition from `G0` and non-`G0` or non-`G0` and `G0` respectively, the smaller value from `ADISPOS` and `ADIS` applies.
  - For `G642/G643`, all axis-specific tolerances are zero.
- The block does not contain traversing motion (zero block).
  This occurs when:
  - Synchronized actions are active.
    Normally, the Interpreter eliminates zero blocks. However, if synchronous actions are active, this zero block is included and also executed. In so doing, an exact stop is initiated corresponding to the active programming. This allows the synchronous action to also switch.
  - Zero blocks are generated by program jumps.

Continuous-path mode in rapid traverse G0

One of the specified functions `G60`/`G9` or `G64`, or `G641` - `G645`, also has to be specified for rapid traverse motion. Otherwise, the default in the machine data is used.

References

For further information about the continuous-path mode see:
Function Manual, Basic Functions; Continuous-Path Mode, Exact Stop, LookAhead (B1).

Coordinate transformations (frames) 13

13.1 Frames

Frame

The frame is a self-contained arithmetic rule that transforms one Cartesian coordinate system into another Cartesian coordinate system.

Basic frame (basic offset)

The basic frame describes coordinate transformation from the basic coordinate system (BCS) to the basic zero system (BZS) and has the same effect as settable frames.

See Basic coordinate system (BCS) (Page 30).

Settable frames

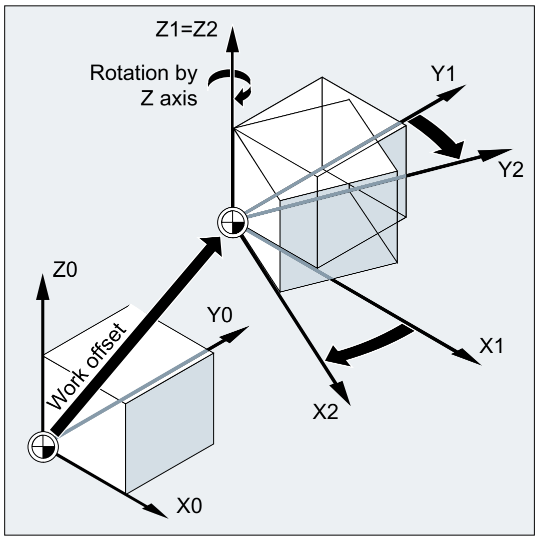
Settable frames are the configurable zero offsets which can be called from within any NC program with the `G54` to `G57` and `G505` to `G599` commands. The offset values are predefined by the user and stored in the zero offset memory on the controller . They are used to define the settable zero system (SZS).

See:

- Settable zero system (SZS) (Page 33)
- Settable zero offset (G54 to G57, G505 to G599, G53, G500, SUPA, G153) (Page 149)

Programmable frames

Sometimes it is useful or necessary within an NC program, to move the originally selected workpiece coordinate system (or the "settable zero system") to another position and, if required, to rotate it, mirror it and/or scale it. This can be achieved using programmable frames.



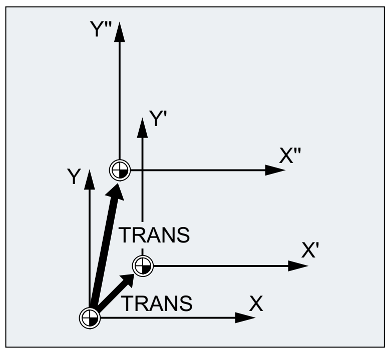
See Frame instructions (Page 323).

13.2 Frame instructions

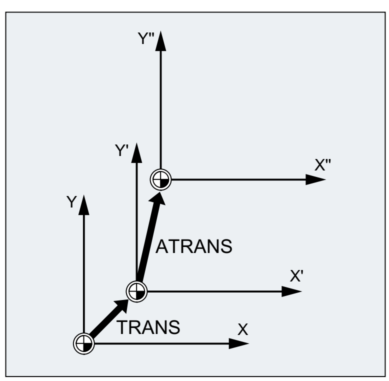
Function

The statements for programmable frames apply in the current NC program. They function as either additive or substitute elements:

- Substitute statement
  Deletes all previously programmed frame statements. The reference is provided by the last settable zero offset called (G54 to G57, G505 to G599).

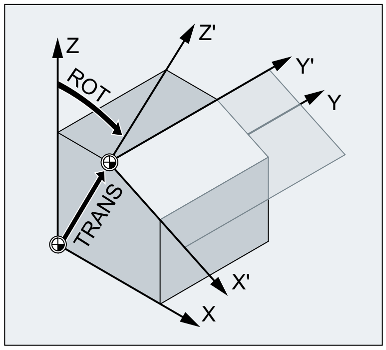


- Additive statement
  Appended to existing frames. The reference is provided by the currently set workpiece zero or the last workpiece zero programmed with a frame statement.



Application example

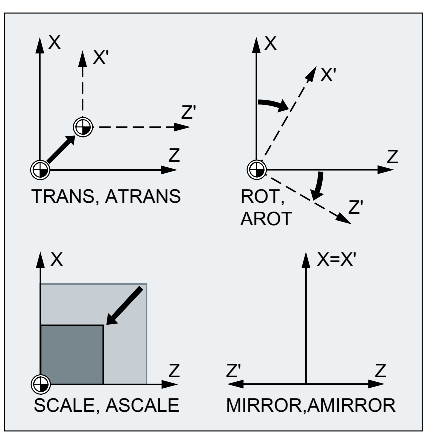
1. Move the zero point of the workpiece coordinate system (WCS).
2. Rotate the workpiece coordinate system (WCS) to orientate a plane parallel to the desired work plane.

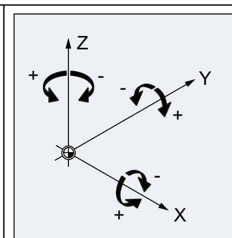


Syntax

Substituting statements	Additive statements
`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`

Meaning



<table>
<tr><td><code>TRANS/ATRANS</code>:</td><td colspan="3">Workpiece coordinate system offset in the direction of the specified geometry axis or axes</td></tr>
<tr><td rowspan="11"><code>ROT/AROT</code>:</td><td colspan="3">Workpiece coordinate system rotation:
• By linking individual rotations around the specified geometry axis or axes
or
• Around the angle <code>RPL=...</code> in the current working plane (<code>G17</code>/<code>G18</code>/<code>G19</code>)</td></tr>
<tr><td>Direction of rotation:</td><td colspan="2">
</td></tr>
<tr><td rowspan="2">Rotation sequence:</td><td>With RPY notation:</td><td>Z, Y', X''</td></tr>
<tr><td>With Euler angle:</td><td>Z, X', Z''</td></tr>
<tr><td rowspan="7">Range of values:</td><td colspan="2">The angles of rotation are only defined unambiguously in the following ranges:</td></tr>
<tr><td rowspan="3">With RPY notation:</td><td>-180 ≤ x ≤ 180</td></tr>
<tr><td>-90 < y < 90</td></tr>
<tr><td>-180 ≤ z ≤ 180</td></tr>
<tr><td rowspan="3">With Euler angle:</td><td>0 ≤ x < 180</td></tr>
<tr><td>-180 ≤ y ≤ 180</td></tr>
<tr><td>-180 ≤ z ≤ 180</td></tr>
<tr><td><code>ROTS/AROTS</code>:</td><td colspan="3">Workpiece coordinate system rotation by means of the specification of solid angles
The orientation of a plane in space is defined unambiguously by specifying two solid angles. Therefore, up to two solid angles may be programmed:
<code>ROTS/AROTS X... Y... / Z... X... / Y... Z...</code></td></tr>
<tr><td><code>CROTS</code>:</td><td colspan="3"><code>CROTS</code> works in the same way as <code>ROTS</code> but refers to the valid frame in the database.</td></tr>
<tr><td><code>SCALE/ASCALE</code>:</td><td colspan="3">Scaling in the direction of the specified geometry axis or axes to increase/reduce the size of a contour</td></tr>
<tr><td rowspan="2"><code>MIRROR/AMIRROR</code>:</td><td colspan="3">Workpiece coordinate system mirroring by means of mirroring (direction change) the specified geometry axis</td></tr>
<tr><td>Value:</td><td colspan="2">Freely selectable (in this case: "0")</td></tr>
</table>

Supplementary conditions

- Frame statements must be programmed in a separate NC block.
- Frame statements can be used individually or combined as required.

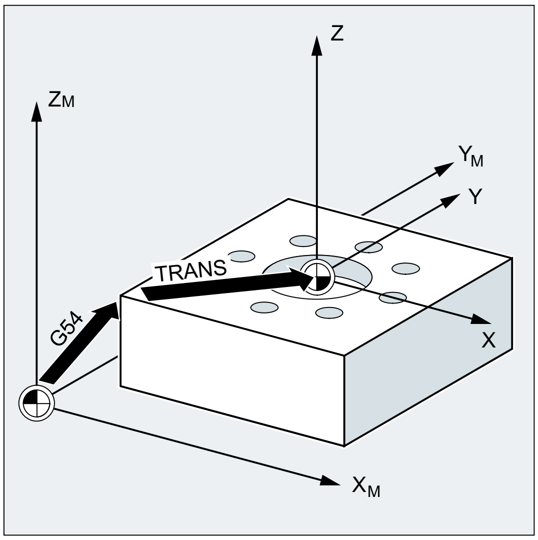
- Frame statements are executed in the programmed sequence.
- Additive statements are frequently used in subprograms. The basic statements defined in the main program are not lost after the end of the subprogram if the subprogram has been programmed with the SAVE attribute.

13.3 Programmable work offset (TRANS, ATRANS)

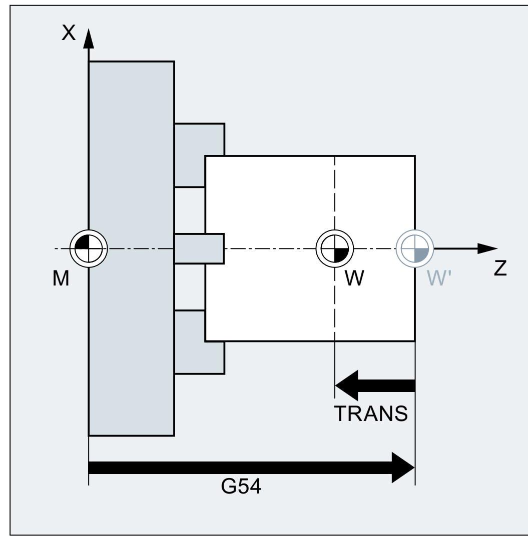
The `TRANS` command moves the WCS absolutely based on the SZS created with a settable work offset (G54 ... G57, G505 ... G599).

The `ATRANS` command moves additively the WCS created with `TRANS`.

Milling:



Turning:



Syntax

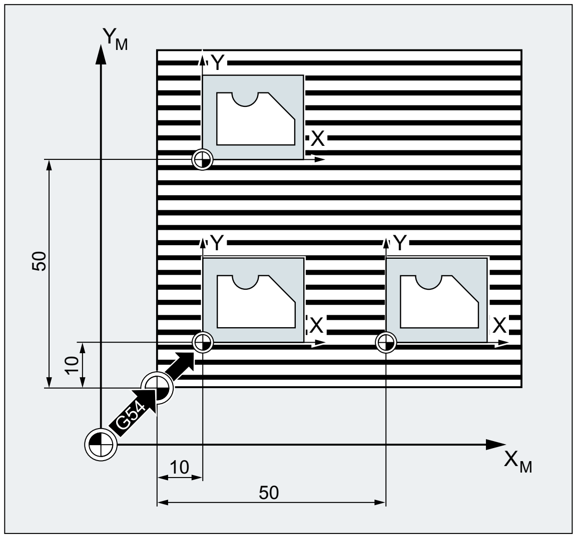
```
TRANS X… Y… Z…
ATRANS X… Y… Z…
```

Meaning

`TRANS`:	Absolute offset of the WCS with reference to the workpiece zero (SZS) set with a settable work offset (G54 ... G57, G505 ... G599).	
	Alone in the block:	yes
`ATRANS`:	Additive zero offset of the WCS with reference to the parameterized workpiece zero set with `TRANS`	
	Alone in the block:	yes
`X... Y... Z...`:	Offset values in the direction of the specified geometry axes	

Examples

Example 1: Milling



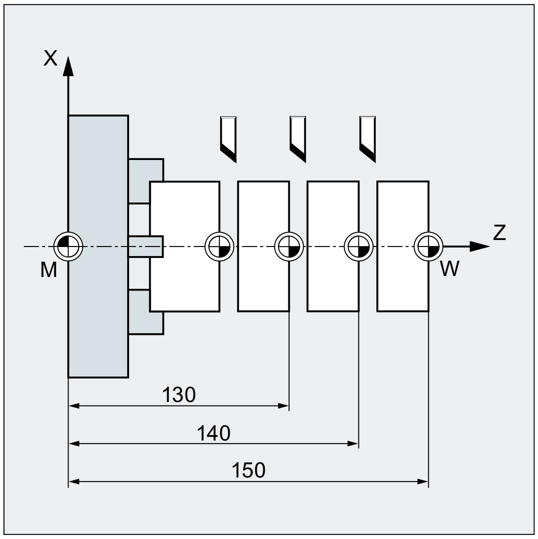
With this workpiece, the shapes shown recur in a program.

The machining sequence for this shape is stored in a subprogram.

Zero offset is used to set the workpiece zeros required in each case and then call the subprogram.

Program code	Comment
N10 G1 G54	; Working plane X/Y, workpiece zero
N20 G0 X0 Y0 Z2	; Approach starting point
N30 TRANS X10 Y10	; Absolute offset
N40 L10	; Subprogram call
N50 TRANS X50 Y10	; Absolute offset
N60 L10	; Subprogram call
N70 M30	; End of program

Example 2: Turning



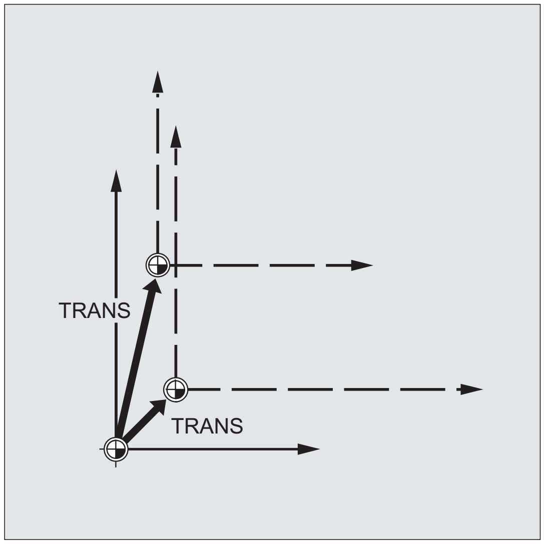

```
Program code                             Comment
...
N10 TRANS X0 Z150                        ; Absolute offset
N15 L20                                  ; Subprogram call
N20 TRANS X0 Z140 (or ATRANS Z-10)       ; Absolute offset
N25 L20                                  ; Subprogram call
N30 TRANS X0 Z130 (or ATRANS Z-10)       ; Absolute offset
N35 L20                                  ; Subprogram call
...
```

Further information

TRANS X... Y... Z...

Translation through the offset values programmed in the specified axis directions (path, synchronized axes and positioning axes). The reference is provided by the last settable work offset called (G54 to G57, G505 to G599).

NOTICE
No original frame
The `TRANS` command resets all frame components of the previously activated programmable frame.

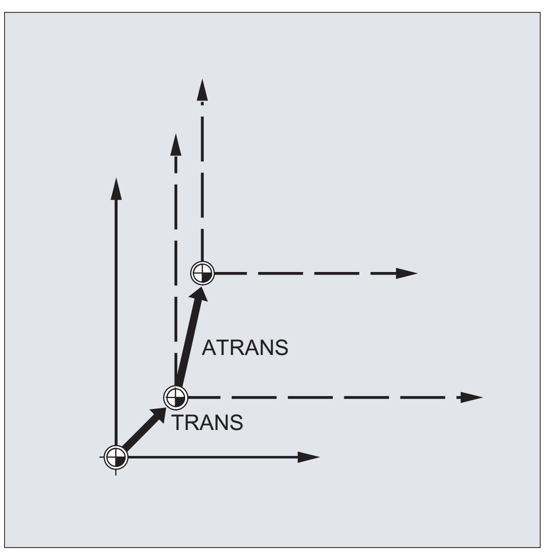


Note

`ATRANS` can be used to program an offset to be added to existing frames.

ATRANS X... Y... Z...

Translation through the offset values programmed in the specified axis directions. The currently set or last programmed zero point is used as the reference.



13.4 Programmable work offset (G58, G59)

Note

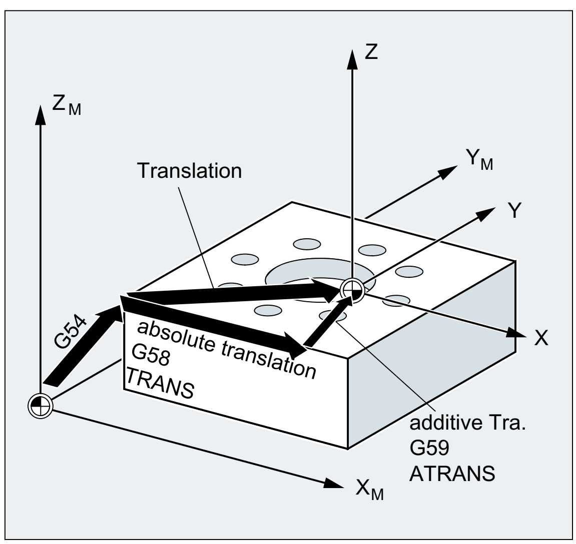
For SINUMERIK 828D the statements `G58`/`G59` have a different function than for SINUMERIK 840D sl:

- `G58`: Calls the 5th settable zero offset (this corresponds to statement `G505` for SINUMERIK 840D sl)
- `G59`: Calls the 6th settable zero offset (this corresponds to statement `G506` for SINUMERIK 840D sl)

Therefore, the following description of `G58`/`G59` is only valid for SINUMERIK 840D sl.

The G58 and G59 functions can be used to substitute translation components of the programmable zero offset (TRANS/ATRANS) (Page 327) with specific axes:

- G58: Absolute translation component (coarse offset)
- G59: Additive translation component (fine offset)



Requirements

The G58 and G59 functions can only be used if fine offset has been configured (MD24000 $MC_FRAME_ADD_COMPONENTS = 1).

Syntax

```
G58 <axis_1><value_1> ... <axis_3><value_3>
G59 <axis_1><value_1> ... <axis_3><value_3>
```

Meaning

G58:	G58 replaces the absolute translation component of the programmable zero offset for the specified axis, but the programmed additive offset remains valid. The reference is provided by the last settable zero offset called (G54 ... G57, G505 ... G599).	
	Alone in the block: Yes	
G59:	G59 replaces the additive translation component of the programmable zero offset for the specified axis, but the programmed absolute offset remains valid.	
	Alone in the block: Yes	
<axis_n>:	Geometry axis in channel	
<value_n>:	Offset values in the direction of the specified geometry axis	

Example

Program code	Comment
...	
N50 TRANS X10 Y10 Z10	; Absolute translation component X10 Y10 Z10
N60 ATRANS X5 Y5	; Additive translation component X5 Y5
	→ total offset: X15 Y15 Z10
N70 G58 X20	; Absolute translation component X20
	→ total offset X25 Y15 Z10
N80 G59 X10 Y10	; Additive translation component X10 Y10
	→ total offset X30 Y20 Z10
...	

Further information

The absolute translation component (**coarse offset**) is modified by the following statements:

- TRANS
- G58
- CTRANS
- CFINE
- $P_PFRAME[X,TR]

The additive translation component (**fine offset**) is modified by the following statements:

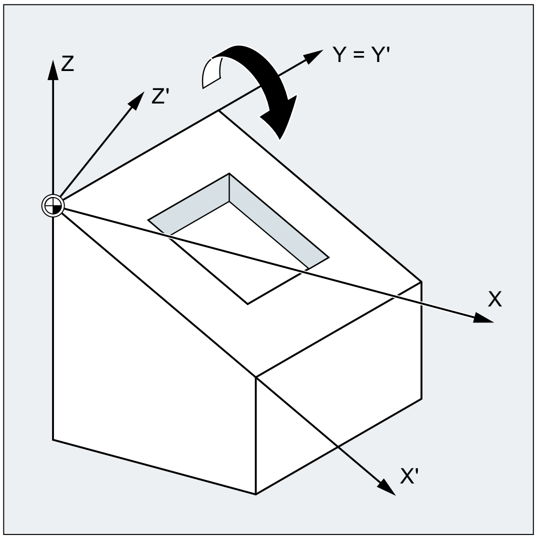
- ATRANS
- G59
- CTRANS
- CFINE
- $P_PFRAME[X,FI]

Examples

Statement	Coarse offset V_C	Fine offset V_F
`TRANS X10`	$V_C = 10$	unchanged
`G58 X10`	$V_C = 10$	unchanged
`$P_PFRAME[X,TR]=10`	$V_C = 10$	unchanged
`ATRANS X10`	unchanged	$V_F = V_F + 10$
`G59 X10`	unchanged	$V_F = 10$
`$P_PFRAME[X,FI]=10`	unchanged	$V_F = 10$
`CTRANS(X,10)`	$V_C = 10$	$V_F = 0$
`CTRANS()`	$V_C = 0$	$V_F = 0$
`CFINE(X,10)`	$V_C = 0$	$V_F = 10$

13.5 Programmable rotation (ROT, AROT, RPL)

The workpiece coordinate system can be rotated in space with the ROT/AROT statements. The statements refer exclusively to the programmable frame $P_PFRAME.



Syntax

```
ROT <1st GeoAx><angle> <2nd GeoAx><angle> <3rd GeoAx><angle>
ROT RPL=<angle>
AROT <1st GeoAx><angle> <2nd GeoAx><angle> <3rd GeoAx><angle>
AROT RPL=<angle>
```

Note

Euler angle

The rotations of the workpiece coordinate system are performed via Euler angles. A detailed description can be found in:

References
Function Manual, Basic Functions; Section "Axes, coordinate systems, frames (K2)" > "Frames" > "Frame components" > "Rotation ..."

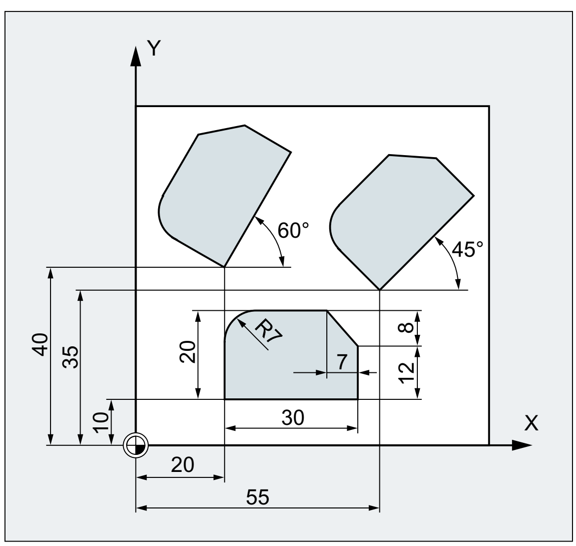
Meaning

ROT:	Absolute rotation	
	Reference frame:	Programmable frame $P_PFRAME
	Reference point:	Zero point of the current workpiece coordinate system set with G54 ... G57, G505 ... G599

AROT:	Additive rotation	
	Reference frame:	Programmable frame $P_PFRAME
	Reference point:	Zero point of the current workpiece coordinate system set with G54 ... G57, G505 ... G599
<nth GeoAx>:	Identifier of the nth geometry axis around which rotation is to be performed with the specified angle. The value 0° is implicitly set as angle of rotation for a geometry axis that has not been programmed.	
RPL:	Rotation around the geometry axis perpendicular to the active plane (G17, G18, G19) by the specified angle	
	Reference frame:	Programmable frame $P_PFRAME
	Reference point:	Zero point of the current workpiece coordinate system set with G54 ... G57, G505 ... G599
<Angle>	Angle specification in degrees.	
	Range of values:	-360° ≤ angle ≤ 360°

Examples

Example 1: Rotation in the G17 plane

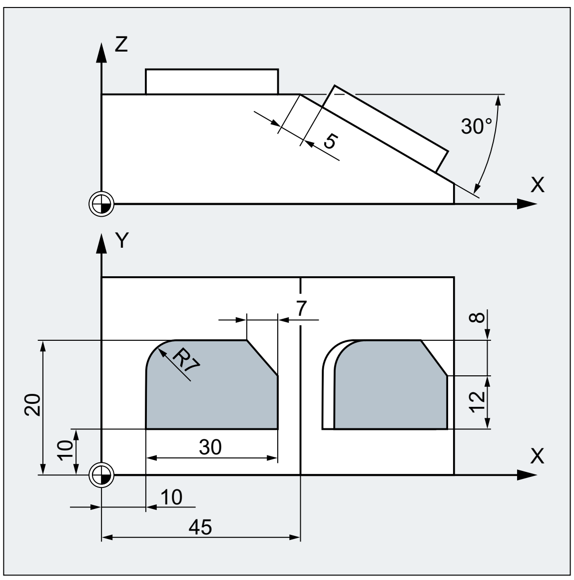


With this workpiece, the shapes shown recur in a program. In addition to the zero offset, rotations have to be performed, as the shapes are not arranged paraxially.

Program code	Comment
N10 G17 G54	; Working plane X/Y, workpiece zero
N20 TRANS X20 Y10	; Absolute offset
N30 L10	; Subprogram call
N40 TRANS X55 Y35	; Absolute offset
N50 AROT RPL=45	; Additive rotation around the Z axis perpendicular; to the G17 plane through 45°
N60 L10	; Subprogram call

Program code	Comment
N70 TRANS X20 Y40	; Absolute offset (resets all previous offsets)
N80 AROT RPL=60	; Additive rotation around the Z axis perpendicular
	; to the G17 plane through 60°
N90 L10	; Subprogram call
N100 G0 X100 Y100	; Retraction
N110 M30	; End of program

Example 2: Spatial rotation around the Y axis



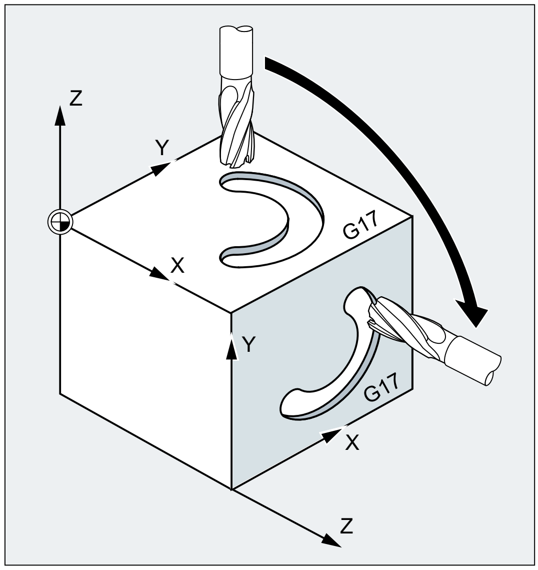
In this example, paraxial and inclined workpiece surfaces are to be machined in a clamping.

Condition:

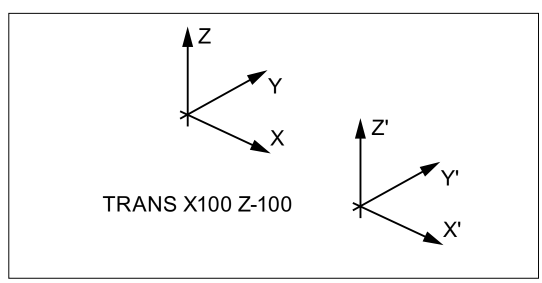
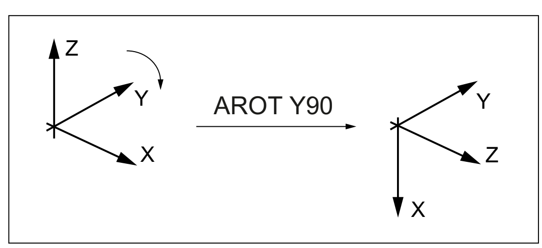
The tool must be aligned perpendicular to the inclined surface in the rotated Z direction.

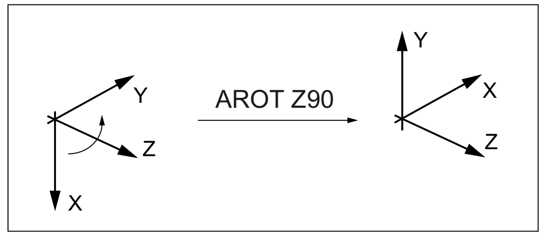
Program code	Comment
N10 G17 G54	; Working plane X/Y, workpiece zero
N20 TRANS X10 Y10	; Absolute offset
N30 L10	; Subprogram call
N40 ATRANS X35	; Additive offset
N50 AROT Y30	; Additive rotation around the Y axis
N60 ATRANS X5	; Additive offset
N70 L10	; Subprogram call
N80 G0 X300 Y100 M30	; Retraction, end of program

Example 3: Multi-face machining



In this example, identical shapes are machined in two workpiece surfaces perpendicular to one another via subprograms. In the new coordinate system on the right-hand workpiece surface, infeed direction, working plane and the zero point have been set up as on the top surface. Therefore, the conditions required for the subprogram execution still apply: Working plane `G17`, coordinate plane X/Y, infeed direction Z.

Program code	Comment
`N10 G17 G54`	`; Working plane X/Y, workpiece zero`
`N20 L10`	`; Subprogram call`
`N30 TRANS X100 Z-100`	`; Absolute offset of the WCS`
	
`N40 AROT Y90`	`; Additive rotation of the WCS around Y through 90°`
	

Program code	Comment
`N50 AROT Z90`	`; Additive rotation of the WCS around Z through 90°`
	
`N60 L10`	`; Subprogram call`
`N70 G0 X300 Y100 M30`	`; Retraction, end of program`

Further information

Rotation in the active plane

When programming using `RPL=...`, the WCS is rotated around the axis perpendicular to the active plane.

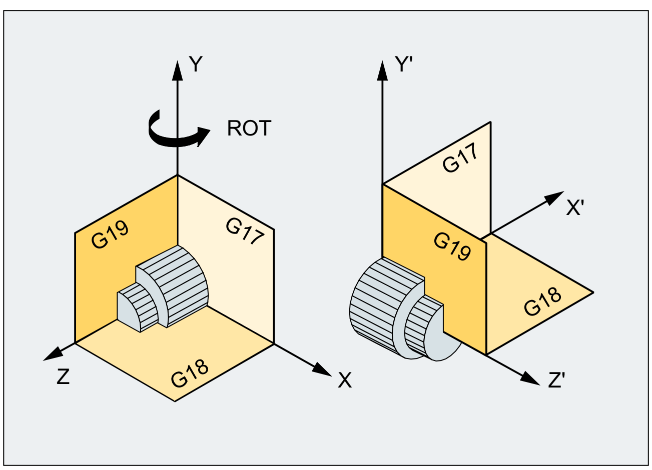
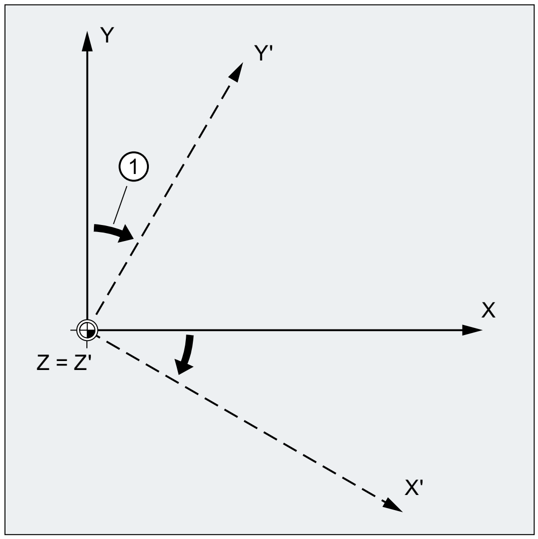


Figure 13-1 Rotation around the Y axis or in the G18 plane

> **⚠ WARNING**
>
> **Plane change**
>
> If a plane change (`G17`, `G18`, `G19`) is programmed after a rotation, the current angles of rotation of the respective axes are retained and are also effective in the new plane. It is therefore strongly recommended that the current angles of rotation be reset to 0 before a plane change:
>
> - `N100 ROT X0 Y0 Z0 ; explicit angle programming`
> - `N100 ROT ; implicit angle programming`

Absolute rotation with ROT X... Y... Z...

The WCS is rotated around the specified axes to the programmed angles of rotation.

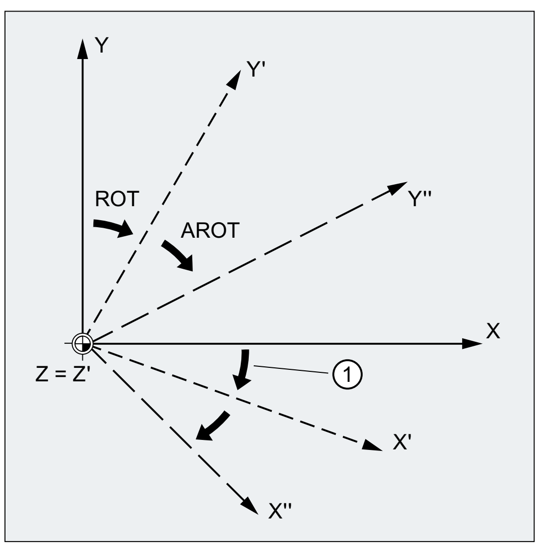


① Angle of rotation

Figure 13-2 Absolute rotation around the Z axis

Additive rotation with AROT X... Y... Z...

The WCS is rotated further around the specified axes through the programmed angles of rotation.



① Angle of rotation

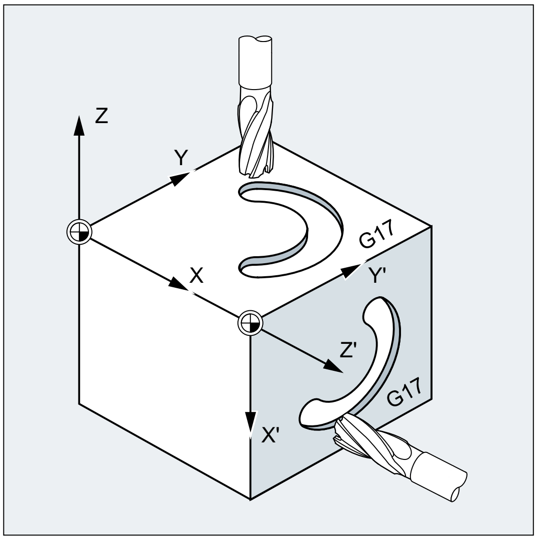
Figure 13-3 Absolute and additive rotation around the Z axis

Rotation of the working plane

During a rotation using `ROT`/`AROT`, the working plane (`G17`, `G18`, `G19`) also rotates.

Example: Working plane G17
The WCS is positioned on the top surface of the workpiece. Using offset and rotation, the coordinate system is moved to one of the side faces. Working plane G17 also rotates. In this way, traversing motions can still be programmed in the G17 plane via X and Y and infeeds via Z.

Requirement:
The tool must be perpendicular to the working plane and the positive direction of the infeed axis points in the direction of the tool base.
Specifying CUT2DF activates the tool radius compensation in the rotated plane.



13.6 Programmable frame rotations with solid angles (ROTS, AROTS, CROTS)

Rotations of the workpiece coordinate system can be specified in solid angles with the `ROTS`, `AROTS` and `CROTS` statements. Solid angles are the angles formed by the intersections of the plane rotated in space with the main planes of the not yet rotated WCS.

Note

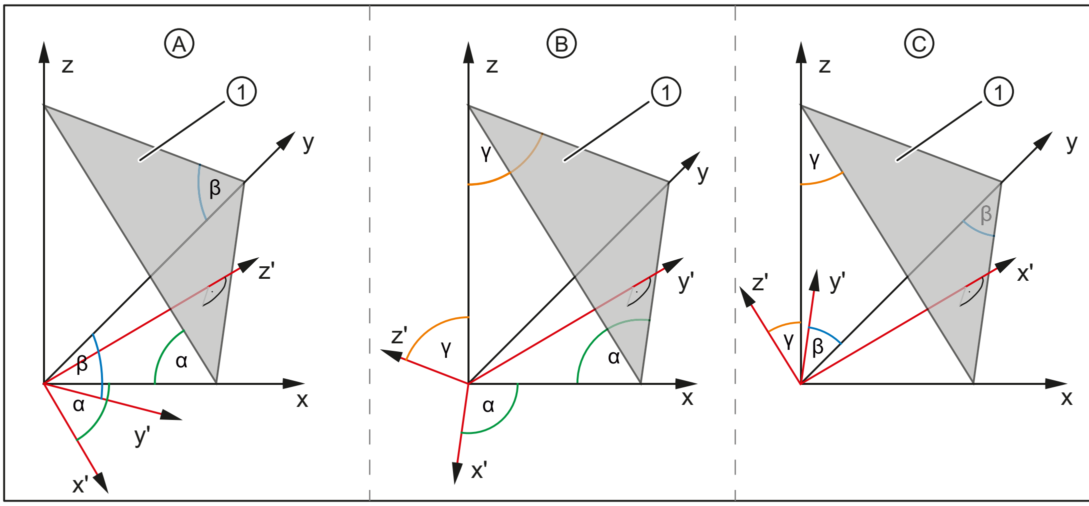
Geometry axis identifiers

The following definition is made as an example for the further description:

- 1st geometry axis: X
- 2nd geometry axis: Y
- 3rd geometry axis: Z

As shown in the following figure, the programming of `ROTS Xα Yβ` results in an alignment of the G17 plane of the WCS parallel to the displayed inclined plane. The position of the zero point of the WCS remains unchanged.

The orientation of the rotated WCS is defined so that the first rotated axis lies in the plane formed by this and the 3rd axis of the original coordinate system. In the example: X' is in the original X/Z plane.



①	Inclined plane
α, β, γ	Solid angle
A	Alignment of the G17 plane parallel to the inclined plane:

- 1st rotation
  Rotation of x around y through angle α ⇒
  x'-axis parallel to the inclined plane
- 2nd rotation
  Rotation of y' around x' through β ⇒
  y'-axis parallel to the inclined plane
  ⇒ z'-axis parallel to the inclined plane
  ⇒ G17 parallel to the inclined plane

B Alignment of the G18 plane parallel to the inclined plane:

- 1st rotation
  Rotation of z around x through the angle γ ⇒
  z'-axis parallel to the inclined plane
- 2nd rotation
  Rotation of x' around z' through angle α ⇒
  x'-axis parallel to the inclined plane
  ⇒ y'-axis parallel to the inclined plane
  ⇒ G18 parallel to the inclined plane

C Alignment of the G19 plane parallel to the inclined plane:

- 1st rotation
  Rotation of y around z through the angle β ⇒
  y'-axis parallel to the inclined plane
- 2nd rotation
  Rotation of z' around y' through angle γ ⇒
  z'-axis parallel to the inclined plane
  ⇒ x'-axis parallel to the inclined plane
  ⇒ G19 parallel to the inclined plane

Syntax

Requirements

The position of a plane in space is clearly defined by two solid angles. The plane would be "over-defined" by the specification of a third solid angle. It is therefore not permitted.

If only one solid angle is programmed, the rotation of the WCS is identical to `ROT`, `AROT` (see Section "Programmable rotation (ROT, AROT, RPL) (Page 334)").

Through the two programmed axes, a plane is specified according to the plane definitions for `G17`, `G18`, `G19`. This defines the sequence of the coordinate axes (1st axis / 2nd axis of the plane) or the sequence of the rotations through the solid angles:

Plane	1st axis	2nd axis
G17	X	Y
G18	Z	X
G19	Y	Z

Alignment of the G17 plane ⇒ solid angle for X and Y

- 1st rotation: X around Y through the angle α
- 2nd rotation: Y around X' through the angle β
- Orientation: X' is in the original Z/X plane.

```
ROTS  X<α> Y<β>
AROTS X<α> Y<β>
CROTS X<α> Y<β>
```

Alignment of the G18 plane ⇒ solid angle for Z and X

- 1st rotation: Z around X through the angle γ
- 2nd rotation: X around Z' through the angle α
- Orientation: Z' is in the original Y/Z plane

```
ROTS  Z<γ> X<α>
AROTS Z<γ> X<α>
CROTS Z<γ> X<α>
```

Alignment of the G19 plane ⇒ solid angle for Y and Z

- 1st rotation: Y around Z through the angle β
- 2nd rotation: Z around Y' through the angle γ
- Orientation: Y' is in the original X/Z plane.

```
ROTS  Y<β> Z<γ>
AROTS Y<β> Z<γ>
CROTS Y<β> Z<γ>
```

Meaning

`ROTS`:	Absolute frame rotations with solid angles, reference frame: Programmable frame $P_PFRAME
`AROTS`:	Additive frame rotations with solid angles, reference frame: Programmable frame $P_PFRAME
`CROTS`:	Absolute frame rotations with solid angles, reference frame: Programmed frame $P_ ...
`X`, `Y`, `Z`:	Geometry axis identifiers (see note above: Geometry axis identifiers)
A, β, γ:	Solid angle in relation to the appropriate geometry axis: • α → X • β → Y • γ → Z

13.7 Programmable scaling factor (SCALE, ASCALE)

`SCALE`/`ASCALE` can be used to program up or down scale factors for all path, synchronized, and positioning axes in the direction of the axes specified in each case. This makes it possible, therefore, to take geometrically similar shapes or different shrinkage allowances into account in the programming.

Syntax

```
SCALE X… Y… Z…
ASCALE X… Y… Z…
```

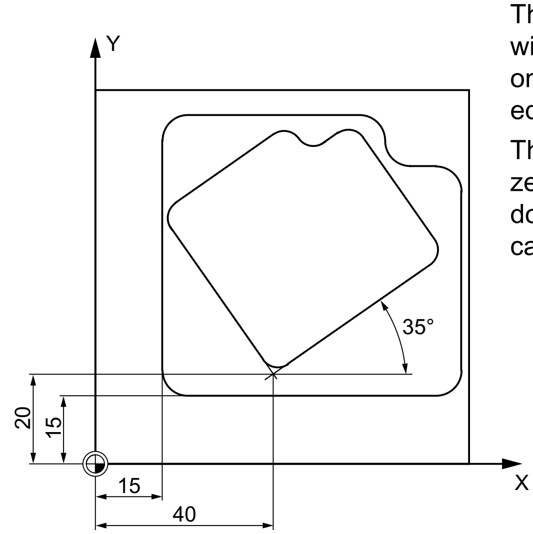
Note

Each frame operation is programmed in a separate NC block.

Meaning

`SCALE`:	Scale up/down absolute in relation to the currently valid coordinate system set with G54 to G57, G505 to G599.
`ASCALE`:	Scale up/down additive in relation to the currently valid set or programmed coordinate system.
`X… Y… Z…`:	Scale factors in the direction of the specified geometry axes.

Example



The pocket occurs twice on this workpiece, but with different sizes and rotated in relation to one another. The machining sequence is stored in the subprogram.

The required workpiece zeroes are set with zero offset and rotation, the contour is scaled down with scaling and the subprogram is then called again.

Program code	Comment
`N10 G17 G54`	`; Working plane X/Y, workpiece zero`
`N20 TRANS X15 Y15`	`; Absolute offset`

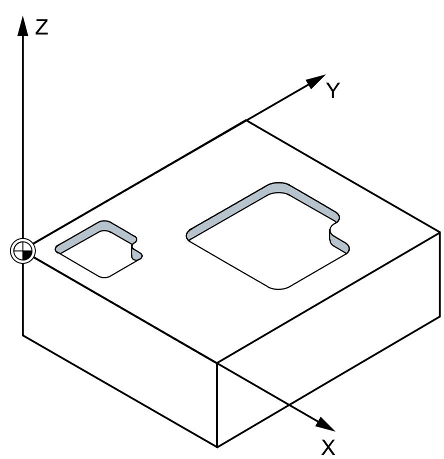
Program code	Comment
N30 L10	; Machine large pocket
N40 TRANS X40 Y20	; Absolute offset
N50 AROT RPL=35	; Rotation in the plane through 35°
N60 ASCALE X0.7 Y0.7	; Scaling factor for the small pocket
N70 L10	; Machine small pocket
N80G0 X300 Y100 M30	; Retraction, end of program

Further information

SCALE X... Y... Z...

You can specify an individual scale factor for each axis, by which the shape is to be reduced or enlarged. The scale refers to the workpiece coordinate system set with G54 to G57, G505 to G599.

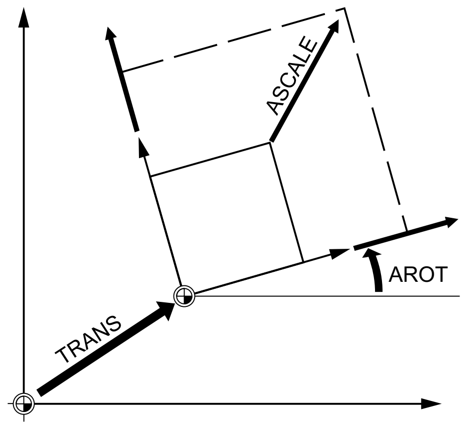
NOTICE
No original frame
The `SCALE` command resets all frame components of the previously activated programmable frame.



ASCALE X... Y... Z...

The `ASCALE` command is used to program scale changes to be added to existing frames. In this case, the last valid scale factor is multiplied by the new one.

The currently set or last programmed coordinate system is used as the reference for the scale change.



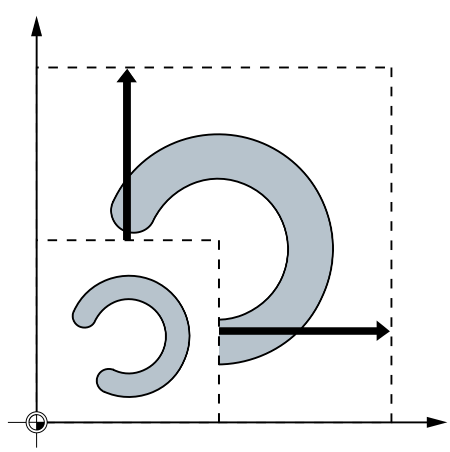
Scaling and offset

Note

If an offset is programmed with `ATRANS` after `SCALE`, the offset values will also be scaled.

Different scale factors

NOTICE
Risk of collision
Please take great care when using different scale factors! Circular interpolations can, for example, only be scaled using identical factors.



Note

However, different scale factors can be used specifically to program distorted circles.

13.8 Programmable mirroring (MIRROR, AMIRROR)

MIRROR/AMIRROR can be used to mirror workpiece shapes on coordinate axes. All traversing movements programmed after the mirror call (e.g. in the subprogram) are executed with mirroring.

Syntax

```
MIRROR X... Y... Z...
AMIRROR X... Y... Z...
```

Note

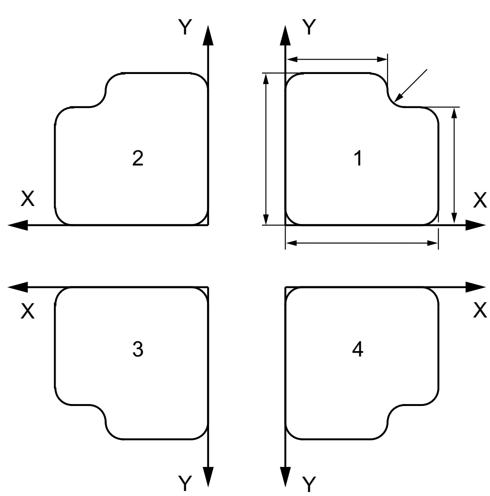
Each frame operation is programmed in a separate NC block.

Meaning

MIRROR:	Mirror absolute in relation to the currently valid coordinate system set with G54 to G57, G505 to G599.
AMIRROR:	Additive mirror image with reference to the currently valid set or programmed coordinate system.
X... Y... Z...:	Geometry axis whose direction is to be changed. The value specified here can be chosen freely, e.g. X0 Y0 Z0.

Examples

Example 1: Milling

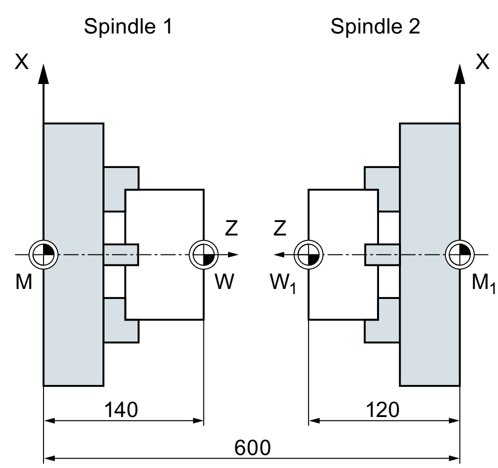


The contour shown here is programmed once as a subprogram. The three other contours are generated using mirroring. The workpiece zero is located at the center of the contours.

Program code	Comment
N10 G17 G54	; Working plane X/Y, workpiece zero

Program code	Comment
N20 L10	; Machine first contour at top right
N30 MIRROR X0	; Mirror X axis (the direction is changed in X)
N40 L10	; Machine second contour at top left
N50 AMIRROR Y0	; Mirror Y axis (the direction is changed in Y)
N60 L10	; Machine third contour at bottom left
N70 MIRROR Y0	; MIRROR resets previous frames. Mirror Y axis (the direction is changed in Y)
N80 L10	; Machine fourth contour at bottom right
N90 MIRROR	;Deactivate mirroring
N100 G0 X300 Y100 M30	; Retraction, end of program

Example 2: Turning



The actual machining is stored as a subprogram and execution at the respective spindle is implemented by means of mirroring and offsets.

Program code	Comment
N10 TRANS X0 Z140	; Zero offset to W
...	; Machining of the first side with spindle 1
N30 TRANS X0 Z600	; Zero offset to spindle 2
N40 AMIRROR Z0	; Mirroring of the Z axis
N50 ATRANS Z120	; Zero offset to W1
...	; Machining of the second side with spindle 2

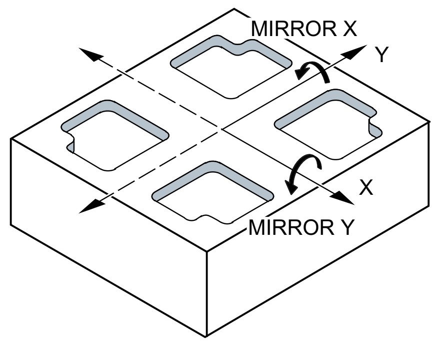
Further information

MIRROR X... Y... Z...

The mirror is programmed by means of an axial change of direction in the selected working plane.

Example: Working plane G17 X/Y

The mirror (on the Y axis) requires a direction change in X and, accordingly, is programmed with `MIRROR X0`. The contour is then mirrored on the opposite side of the mirror axis Y.

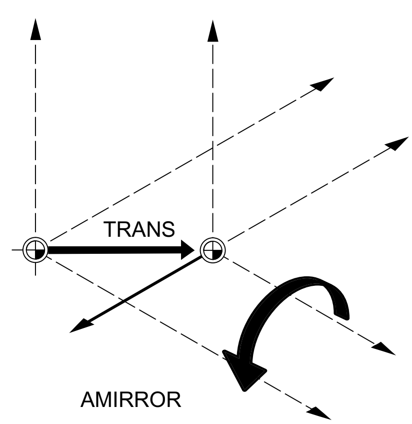


Mirroring is implemented in relation to the currently valid coordinate system set with G54 to G57, G505 to G599.

NOTICE
No original frame The `MIRROR` command resets all frame components of the previously activated programmable frame.

AMIRROR X... Y... Z...

A mirror image, which is to be added to an existing transformation, is programmed with `AMIRROR`. The currently set or last programmed coordinate system is used as the reference.



Deactivate mirroring

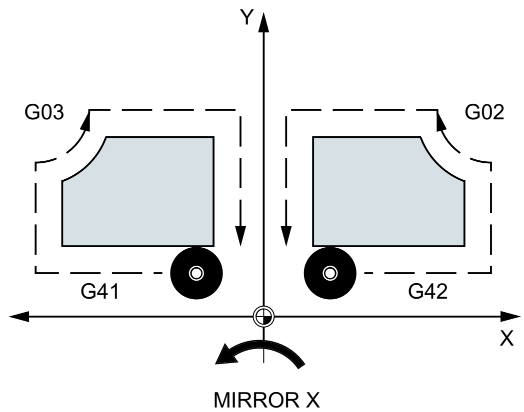
For all axes: `MIRROR` (without axis parameter)

All frame components of the previously programmed frame are reset.

Tool radius compensation

Note

The mirror command causes the control to automatically change the path compensation commands (`G41`/`G42` or `G42`/`G41`) according to the new machining direction.



The same applies to the direction of circle rotation (G2/G3 or G3/G2).

Note

If you program an additive rotation with `AROT` after `MIRROR`, you may have to work with reversed directions of rotation (positive/negative or negative/positive). Mirrors on the geometry axes are converted automatically by the control into rotations and, where appropriate, mirrors on the mirror axis specified in the machine data. This also applies to settable zero offsets.

Mirror axis

The axis to be mirrored can be set in machine data:

MD10610 $MN_MIRROR_REF_AX = <value>

Value	Meaning
0	Mirroring is performed around the programmed axis (negation of values).
1	The reference axis is the X axis.
2	The reference axis is the Y axis.
3	The reference axis is the Z axis.

Interpreting the programmed values

Machine data is used to specify how the programmed values are to be interpreted:

MD10612 $MN_MIRROR_TOGGLE = <value>

Value	Meaning
0	Programmed axis values are not evaluated.
1	Programmed axis values are evaluated: • For programmed axis values ≠ 0, the axis is mirrored if it has not yet been mirrored. • For a programmed axis value = 0, mirroring is deactivated.

13.9 Frame generation according to tool orientation (TOFRAME, TOROT, PAROT):

`TOFRAME` generates a rectangular frame whose Z axis coincides with the current tool orientation. This means that the user can retract the tool in the Z direction without risk of collision (e.g. after a tool break in a 5-axis program).

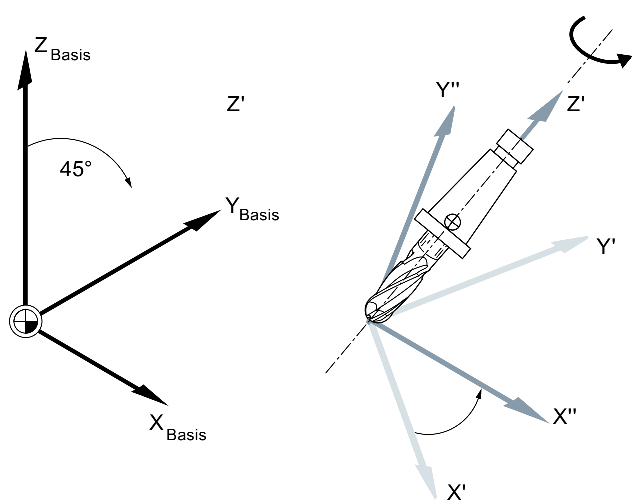
The position of the X and Y axes is determined by the setting in machine data MD21110 $MC_X_AXES_IN_OLD_X_Z_PLANE (coordinate system with automatic frame definition). The new coordinate system is either left as generated from the machine kinematics or is turned around the new Z axis additionally so that the new X axis lies in the old Z/X plane (see machine manufacturer's specifications).

The resulting frame describing the orientation is written in the system variable for the programmable frame ($P_PFRAME).

`TOROT` only overwrites the rotation component in the programmed frame. All other components remain unchanged.

`TOFRAME` and `TOROT` are designed for milling operations in which `G17` (working plane X/Y) is typically active. In the case of turning operations or generally when `G18` or `G19` is active, however, frames are needed where the X or Y axis matches the orientation of the tool. These frames are programmed with the `TOFRAMEX`/`TOROTX` or `TOFRAMEY`/`TOROTY` statements.

`PAROT` aligns the workpiece coordinate system on the workpiece.



Syntax

```
TOFRAME/TOFRAMEZ/TOFRAMEY/TOFRAMEX
...
TOROTOF


TOROT/TOROTZ/TOROTY/TOROTX
...
```

```
TOROTOF


PAROT
...
PAROTOF
```

Meaning

`TOFRAME`:	Align Z axis of the WCS by rotating the frame parallel to the tool orientation
`TOFRAMEZ`:	As `TOFRAME`
`TOFRAMEY`:	Align Y axis of the WCS by rotating the frame parallel to the tool orientation
`TOFRAMEX`:	Align X axis of the WCS by rotating the frame parallel to the tool orientation
`TOROT`:	Align Z axis of the WCS by rotating the frame parallel to the tool orientation The rotation defined with `TOROT` is the same as that defined with `TOFRAME`.
`TOROTZ`:	As `TOROT`
`TOROTY`:	Align Y axis of the WCS by rotating the frame parallel to the tool orientation
`TOROTX`:	Align X axis of the WCS by rotating the frame parallel to the tool orientation
`TOROTOF`:	Deactivate orientation parallel to tool orientation
`PAROT`:	Rotate frame to align workpiece coordinate system on workpiece Translations, scaling and mirroring in the active frame remain valid
`PAROTOF`:	The workpiece-specific frame rotation activated with `PAROT` is deactivated with `PAROTOF`.

Note

The `TOROT` statement ensures consistent programming with active orientable toolholders for each kinematic type.

Just as in the situation for rotatable toolholders, `PAROT` can be used to activate a rotation of the work table. This defines a frame which changes the position of the workpiece coordinate system in such a way that no compensatory movement is performed on the machine. Language command `PAROT` is not rejected if no toolholder with orientation capability is active.

Example

Program code	Comment
`N100 G0 G53 X100 Z100 D0`	
`N120 TOFRAME`	
`N140 G91 Z20`	`; TOFRAME is included in the calculation, all programmed geometry axis movements refer to the new coordinate system.`
`N160 X50`	
`...`	

Further information

Assigning axis direction

If one of the TOFRAMEX, TOFRAMEY, TOROTX, TOROTY statements is programmed instead of TOFRAME/TOFRAMEZ or TOROT/TOROTZ, the axis direction statements listed in this table will apply:

Statement	Tool direction (applicate)	Secondary axis (abscissa)	Secondary axis (ordinate)
TOFRAME/TOFRAMEZ / TOROT/TOROTZ	Z	X	Y
TOFRAMEY/TOROTY	Y	Z	X
TOFRAMEX/TOROTX	X	Y	Z

Separate system frame for TOFRAME or TOROT

The frames resulting from TOFRAME or TOROT can be written in a separate system frame $P_TOOLFRAME. For this purpose, bit 3 must be enabled in machine data MD28082 $MC_MM_SYSTEM_FRAME_MASK. The programmable frame remains unchanged. Differences occur when the programmable frame is processed further elsewhere.

References

For further information about machines with orientable toolholder, see:

- Programming Manual, Job Planning; Chapter: "Tool orientation"
- Function Manual, Basic Functions; Tool Offset (W1),
  Chapter: "Toolholder with orientation capability"

13.10 Deselect frame (G53, G153, SUPA, G500)

When executing certain processes, such as approaching the tool change point, various frame components have to be defined and suppressed at different times.

Settable frames can either be deactivated modally or suppressed non-modally.

Programmable frames can be suppressed or deleted non-modally.

Syntax

```
G53
G153
SUPA
G500
TRANS
ROT
SCALE
MIRROR
```

Meaning

`G53`:	Non-modal suppression of all programmable and settable frames
`G153`:	`G153` has the same effect as `G53` and also suppresses the entire basic frame ($P_ACTBFRAME).
`SUPA`:	`SUPA` has the same effect as `G153` and also suppresses: • Handwheel offsets (DRF) • Overlaid movements • External zero offset • PRESET offset
`G500`:	Modal deactivation of all settable frames (G54 to G57, G505 to G599) if G500 does not contain a value.
`TRANS ROT SCALE MIRROR`:	Without axis details, a deletion of the programmable frames acts.

13.11 Programming: Deselecting overlays axis-specifically (CORROF)

The following axis-specific overlays are deleted with the CORROF procedure:

- Additive work offsets (DRF offsets) set via handwheel traversal
- Position offsets programmed via the $AA_OFF system variable
- Overlays of the tool orientation programmed via the $AC_OFF_... system variable

A preprocessing stop is initiated through the deletion of an overlay value and the position component of the deselected overlaid movement is transferred to the position in the basic coordinate system. Whereby, no axis is traversed.

The position value that can be read via the $AA_IM system variable (current **MCS** setpoint of the axis) **does not change** in the **machine** coordinate system.

The position value that can be read via the $AA_IW system variable (current **WCS** setpoint of the axis) **changes** in the **workpiece**coordinate system because it now contains the deselected component of the overlaid movement.

Note

`CORROF` can be programmed in an **NC program**.

`CORROF` must **not** be programmed in a **synchronized action**.

Syntax

```
CORROF(<Axis>,"<String>"[,<Axis>,"<String>"])
```

Meaning

<table>
<tr><td rowspan="2"><code>CORROF</code>:</td><td colspan="2">Procedure for the deselection of the following offsets and overlays of an axis:
• DRF offset
• Position offsets ($AA_OFF)
• Overlay of the tool orientation ($AC_OFF_...)</td></tr>
<tr><td>Effective-
ness:</td><td>Modal</td></tr>
<tr><td rowspan="2"><code><Axis></code>:</td><td colspan="2">Axis identifier (channel, geometry or machine axis identifier)</td></tr>
<tr><td>Data type:</td><td>AXIS</td></tr>
<tr><td rowspan="6"><code><String></code>:</td><td colspan="2">Character string for the definition of the overlay type</td></tr>
<tr><td>Data type:</td><td>BOOL</td></tr>
<tr><td>Value</td><td>Meaning</td></tr>
<tr><td><code>DRF</code></td><td>DRF offset</td></tr>
<tr><td><code>AA_OFF</code></td><td>Position offset ($AA_OFF)</td></tr>
<tr><td><code>OFF_ORI</code></td><td>Overlay of the tool orientation ($AC_OFF_...)
Note
The deselection of the overlay of the tool orientation is performed by deleting the axis-specific offsets of the orientation axes. An arbitrary channel axis can be specified as <Axis> parameter.</td></tr>
</table>

Examples

Example 1: Axis-specific deselection of a DRF offset (1)

A DRF offset is generated in the X axis by DRF handwheel traversal. No DRF offsets are operative for any other axes in the channel.

Program code	Comment
N10 CORROF(X,"DRF")	; CORROF has the same effect as DRFOF here.
...	

Example 2: Axis-specific deselection of a DRF offset (2)

A DRF offset is generated in the X and Y axes by DRF handwheel traversal. No DRF offsets are operative for any other axes in the channel.

Program code	Comment
; Only the DRF offset of the X axis is deselected; the DRF offset of the Y axis is retained.	
; With DRFOF, both offsets would have been deselected.	
N10 CORROF(X,"DRF")	
...	

Example 3: Axis-specific deselection of a $AA_OFF position offset

Program code	Comment
; A position offset == 10 is interpolated for the X axis.	
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5	
...	
; The position offset of the X axis is deselected: $AA_OFF[X]=0	
; The X axis is not traversed.	
; The position offset is added to the current position of the X axis.	
N80 CORROF(X,"AA_OFF")	
...	

Example 4: Axis-specific deselection of a DRF offset and a $AA_OFF position offset (1)

A DRF offset is generated in the X axis by DRF handwheel traversal. No DRF offsets are operative for any other axes in the channel.

Program code	Comment
; A position offset of 10 is interpolated for the X axis.	
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5	
...	
; Only the DRF offset and the position offset of the X axis are deselected.	
; The DRF offset of the Y axis is retained.	
N70 CORROF(X,"DRF",X,"AA_OFF")	
...	

Example 5: Axis-specific deselection of a DRF offset and a $AA_OFF position offset (2)

A DRF offset is generated in the X and Y axes by DRF handwheel traversal. No DRF offsets are operative for any other axes in the channel.

Program code	Comment
`; A position offset == 10 is interpolated for the X axis.`	
`N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5`	
`...`	
`; The DRF offset of the Y axis and the position offset of the X axis are deselected.`	
`; The DRF offset of the X axis is retained.`	
`N70 CORROF(Y,"DRF",X,"AA_OFF")`	
`...`	

Further information

$AA_OFF_VAL

Once the position offset has been deselected by means of $AA_OFF, system variable $AA_OFF_VAL (integrated distance of axis overlay) for the corresponding axis will equal zero.

$AA_OFF in JOG mode

Also in JOG mode, if $AA_OFF changes, the position offset will be interpolated as an overlaid movement if this function has been enabled via machine data MD 36750 $MA_AA_OFF_MODE.

$AA_OFF in synchronized action

If a synchronized action which immediately resets $AA_OFF (`DO $AA_OFF[<axis>]=<value>`) is active when the position offset is deselected using the `CORROF(<axis>,"AA_OFF")`, then $AA_OFF will be deselected and not reset, and alarm 21660 will be displayed. However, if the synchronized action becomes active later, e.g. in the block after `CORROF`, $AA_OFF will remain set and a position offset will be interpolated.

Automatic channel axis exchange

If an axis that is active in another channel has been programmed for a `CORROF`, it will be fetched into the channel with an axis exchange (requirement: MD30552 $MA_AUTO_GET_TYPE > 0) and the position offset and/or the DRF offset deselected.

13.12 Deselecting additive work offsets (DRFROF)

The additive work offsets (DRF offsets) set via handwheel traversal are deselected via the `DRFOF` procedure.

A preprocessing stop is initiated through the deselection and the position component of the deselected DRF offset is transferred to the position in the basic coordinate system whereby no axis is traversed. The value of the $AA_IM system variable (current MCS setpoint of an axis) does not change; the value of the $AA_IW system variable (current WCS setpoint of an axis) changes because it now contains the deselected component from the overlaid movement.

Syntax

```
DRFOF
```

Meaning

`DRFOF`:	Procedure for deselection of the DRF offsets for all active axes in the channel	
	Effectiveness:	Modal

Examples

Example 1: Axis-specific deselection of a DRF offset (1)

A DRF offset is generated in the X axis by DRF handwheel traversal. No DRF offsets are operative for any other axes in the channel.

Program code	Comment
`N10 CORROF(X,"DRF")`	`; CORROF has the same effect as DRFOF here.`
`...`	

Example 2: Axis-specific deselection of a DRF offset (2)

A DRF offset is generated in the X and Y axes by DRF handwheel traversal. No DRF offsets are operative for any other axes in the channel.

Program code	Comment
`N10 CORROF(X,"DRF")`	`; Only the DRF offset of the X axis is deselected; the DRF offset of the Y axis is retained (in the case of DRFOF both offsets would have been deselected).`
`...`	

Example 3: Axis-specific deselection of a DRF offset and a $AA_OFF position offset (1)

A DRF offset is generated in the X axis by DRF handwheel traversal. No DRF offsets are operative for any other axes in the channel.

Program code	Comment
`N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5`	`; A position offset == 10 is interpolated for the X axis.`
`...`	

Program code	Comment
N70 CORROF(X,"DRF",X,"AA_OFF")	; Only the DRF offset and the position offset of the X axis are deselected; the DRF offset of the Y axis is retained.
...	

Example 4: Axis-specific deselection of a DRF offset and a $AA_OFF position offset (2)

A DRF offset is generated in the X and Y axes by DRF handwheel traversal. No DRF offsets are operative for any other axes in the channel.

Program code	Comment
N10 WHEN TRUE DO $AA_OFF[X]=10 G4 F5	; A position offset == 10 is interpolated for the X axis.
...	
N70 CORROF(Y,"DRF",X,"AA_OFF")	; The DRF offset of the Y axis and the position offset of the X axis are deselected; the DRF offset of the X axis is retained.
...	

13.13 Grinding-specific work offsets (GFRAME0, GFRAME1 ... GFRAME100)

Command for activating a grinding frame in the channel

The programming of the GFRAME<n> command makes the associated grinding frame of the $P_GFR[<n>] data management active in the channel. This sets the active $P_GFRAME grinding frame identical to the $P_GFR[<n>] grinding frame of the data management:

GFRAME<n> ⇒ $P_GFRAME = $P_GFR[<n>]

Command	Grinding frame activated in the channel
GFRAME0	$P_GFR[0] (**null frame**)
GFRAME1	$P_GFR[1]
...	...
GFRAME100	$P_GFR[100]

Syntax

```
GFRAME<n>
```

Meaning

GFRAME<n>:	Activation of the grinding frame **<n>** of the data management	
	G group:	64
	Basic position:	MD20150 $MC_GCODE_RESET_VALUES[**63**]
	Effective:	Modal
<n>:	Number of the grinding frame	
	Value range:	0, 1, 2, ... 100

14 Auxiliary function outputs

Function

The auxiliary function output sends information to the PLC indicating when the NC program needs the PLC to perform specific switching operations on the machine tool. The auxiliary functions are output, together with their parameters, to the PLC interface. The transferred values and signals must be processed by the PLC user program.

Auxiliary functions

The following auxiliary functions can be transferred to the PLC:

Auxiliary Function	Address
Tool selection	T
Tool offset	D, DL
Feedrate	F/FA
Spindle speed	S
M functions	M
H functions	H

For each function group or single function, machine data is used to define whether the output is triggered **before**, **with** or **after** the traversing motion.

The PLC can be programmed to acknowledge auxiliary function outputs in various ways.

Properties

Important properties of the auxiliary function are shown in the following overview table:

Function	Address extension		Value			Explanations	Maximum number per block
	Meaning	Range	Range	Type	Meaning		
M	-	0 (implicit)	0 ... 99	INT	Function	The address extension is 0 for the range between 0 and 99. Mandatory without address extension: M0, M1, M2, M17, M30	5
	Spindle no.	1 - 12	1 ... 99	INT	Function	M3, M4, M5, M19, M70 with address extension spindle no. (e.g. `M2=5`; spindle stop for spindle 2). Without spindle number, the function applies for the master spindle.	
	Any	0 - 99	100 ... 2147483647	INT	Function	User M function*	
S	Spindle no.	1 - 12	0 ... ± $1.8*10^{308}$	REAL	Speed	Without spindle number, the function applies for the master spindle.	3
H	Any	0 - 99	0 ... ± 2147483647 ± $1.8*10^{308}$	 INT REAL	Any	Functions have no effect in the NC; only to be implemented on the PLC.*	3
T	Spindle no. (for active tool management)	1 - 12	0 - 32000 (or tool names with active tool management)	INT	Tool selection	Tool names are not passed to the PLC interface.	1
D	-	-	0 - 12	INT	Tool offset selection	D0: Deselection Default setting: D1	1
DL	Location-dependent offset	1 - 6	0 ... ± $1.8*10^{308}$	REAL	Tool fine offset selection	Refers to previously selected D number.	1
F	-	-	0.001 - 999 999.999	REAL	Path feedrate		6
FA	Axis No.	1 - 31	0.001 - 999 999.999	REAL	Axial feedrate		
* The meaning of the functions is defined by the machine manufacturer (**see machine manufacturer's specifications**).							

Further information

Number of function outputs per NC block

Up to 10 function outputs can be programmed in one NC block. Auxiliary functions can also be output from the action component of **synchronized actions**.

References:
Function Manual, Synchronized Actions

Grouping

The functions described can be grouped together. Group assignment is predefined for some M commands. The acknowledgment behavior can be defined by the grouping.

High-speed function outputs (QU)

Functions, which have not been programmed as high-speed outputs, can be defined as high-speed outputs for individual outputs with the keyword `QU`. Program execution continues without waiting for the acknowledgment of the miscellaneous function (the program waits for the transport acknowledgment). This helps avoid unnecessary hold points and interruptions to traversing movements.

Note

The appropriate machine data must be set for the "High-speed function outputs" function (→ **machine manufacturer**).

Function outputs for travel commands

The transfer of information as well as waiting for the appropriate response takes time and therefore influences the traversing movements.

High-speed acknowledgment without block change delay

Block change behavior can be influenced by machine data. When the "without block change delay" setting is selected, the system response with respect to high-speed auxiliary functions is as follows:

Auxiliary function output	Response
Before the movement	The block transition between blocks with high-speed auxiliary functions occurs **without** interruption and **without** a reduction in velocity. The auxiliary function output takes place in the first interpolator clock cycle of the block. The following block is executed with no acknowledgment delay.
During the movement	The block transition between blocks with high-speed auxiliary functions occurs **without** interruption and **without** a reduction in velocity. The auxiliary function output takes place during the block. The following block is executed with no acknowledgment delay.
After the movement	The movement stops at the end of the block. The auxiliary function output takes place at the end of the block. The following block is executed with no acknowledgment delay.

CAUTION

Function outputs in continuous-path mode

Function outputs **before** the traversing movements interrupt the continuous-path mode (`G64`/ `G641`) and generate an exact stop for the previous block.

Function outputs **after** the traversing movements interrupt the continuous-path mode (`G64`/ `G641`) and generate an exact stop for the current block.

Important: A wait for an outstanding acknowledgment signal from the PLC can also interrupt the continuous-path mode, e.g. for M command sequences in blocks with extremely short path lengths.

14.1 M functions

The M functions initiate switching operations, such as "Coolant ON/OFF" and other functions on the machine.

Syntax

```
M<value>
M[<address extension>] = <value>
```

Meaning

`M`:	Address for the programming of the M functions.	
`<address extension>`:	The extended address notation applies for some M functions (e.g. specification of the spindle number for spindle functions).	
`<value>`:	Assignment is made to a certain machine function through the value assignment (M function number).	
	Type:	INT
	Range of values:	0 ... 2147483647 (max. INT value)

Predefined M functions

Certain important M functions for program execution are supplied as standard with the control:

M function	Meaning
M0*	Programmed stop
M1*	Optional stop
M2*	End of program, main program (as M30)
M3	Spindle clockwise
M4	Spindle counter-clockwise
M5	Spindle stop
M6	Tool change (default setting)
M17*	End of subprogram
M19	Spindle positioning
M30*	End of program, main program (as M2)
M40	Automatic gear change
M41	Gear stage 1
M42	Gear stage 2
M43	Gear stage 3
M44	Gear stage 4
M45	Gear stage 5
M70	Spindle is switched to axis mode

Note

Extended address notation cannot be used for the functions marked with *.

The functions M0, M1, M2, M17 and M30 are always triggered **after** the traversing movement.

M functions defined by the machine manufacturer

All free M function numbers can be used by the machine manufacturer, e.g. for switching functions to control the clamping devices or for the activation/deactivation of further machine functions.

Note

The functions assigned to the free M function numbers are machine-specific. A certain M function can therefore have a different functionality on another machine.

Refer to the machine manufacturer's specifications for the M functions available on a machine and their functions.

Examples

Example 1: Maximum number of M functions in the block

Program code	Comment
`N10 S...`	
`N20 X... M3`	`; M function in the block with axis movement,`
	`; spindle accelerates prior to X axis movement.`
`N180 M789 M1767 M100 M102 M376`	`; Maximum of five M functions in the block.`

Example 2: M function as high-speed output

Program code	Comment
`N10 H=QU(735)`	`; Fast output for H735.`
`N10 G1 F300 X10 Y20 G64`	
`N20 X8 Y90 M=QU(7)`	`; Fast output for M7.`

M7 has been programmed as fast output so that the continuous-path mode (G64) is not interrupted.

Note

Only use this function in special cases as, for example, the chronological alignment is changed in combination with other function outputs.

Further information about the predefined M commands

Programmed stop: M0

The machining is stopped in the NC block with `M0`. You can now remove chips, remeasure, etc.

Programmed stop 1 - optional stop: M1

M1 can be set via:

- HMI / dialog box "Program Control"
  or
- NC/PLC interface

The program execution of the NC is stopped by the programmed blocks.

Programmed stop 2 - an auxiliary function associated with M1 with stop in the program execution

Programmed stop 2 can be set via the HMI / dialog box "Program Control" and allows the technological sequences to be interrupted at any time at the end of the part to be machined. In this way, the operator can interrupt the production, e.g. to remove chip flows.

End of program: M2, M17, M30

A program is ended with `M2`, `M17` or `M30`. If the main program is called from another program (as subprogram), `M2`/`M30` has the same effect as `M17` and vice versa, i.e. `M17` has the same effect in the main program as `M2`/`M30`.

Spindle functions: M3, M4, M5, M19, M70

The extended address notation with specification of the spindle number applies for all spindles.

Example:

Program code	Comment
`M2=3`	`; Clockwise spindle rotation for the second spindle`

If an address extension has not been programmed, the function applies for the master spindle.

15 Supplementary commands

15.1 Output messages (MSG)

Using the `MSG()` statement, any character string from the part program can be output as message to the operator.

Syntax

```
MSG("<Message text>"[,<Execution>])
...
MSG ()
```

Meaning

<table>
<tr><td>MSG:</td><td colspan="3">Predefined subprogram call for output of a message</td></tr>
<tr><td rowspan="4"><message text>:</td><td colspan="3">Any character string to be displayed as message</td></tr>
<tr><td>Type:</td><td colspan="2">STRING</td></tr>
<tr><td>Maximum length:</td><td colspan="2">124 characters; the display takes up two lines (2*62 characters)</td></tr>
<tr><td colspan="3">By using the link operator "<<", variables can also be output in the message text.</td></tr>
<tr><td rowspan="4"><Execution>:</td><td colspan="3">Parameter to define the time when the message is written (optional)</td></tr>
<tr><td>Type:</td><td colspan="2">INT</td></tr>
<tr><td rowspan="2">Value:</td><td>0 (basic setting)</td><td>To write the message, a dedicated main run block is not generated. This is realized in the next NC block that can be executed. Active continuous-path mode is not interrupted.</td></tr>
<tr><td>1</td><td>To write the message, a dedicated main run block is generated. Active continuous-path mode is interrupted.</td></tr>
<tr><td>MSG():</td><td colspan="3">The actual message can be deleted by programming MSG() without message text. If not deleted, the display remains until the next message is present.</td></tr>
</table>

Note

If the message is to be output in the language active at the user interface, then the user requires information about the language that is currently set at the HMI. This information can be interrogated in the part program and in the synchronized actions using system variable $AN_LANGUAGE_ON_HMI (see "Currently set language in the HMI (Page 571)").

Examples

Example 1: Output/delete message

Program code	Comment
N10 G91 G64 F100	; Continuous path mode
N20 X1 Y1	
N... X... Y...	
N20 MSG ("Machining part 1")	; The message is first output with N30.
	; continuous-path mode is retained.
N30 X... Y...	
N... X... Y...	
N400 X1 Y1	
N410 MSG ("Machining part 2",1)	; The message is output with N410.
	; continuous-path mode is interrupted.
N420 X1 Y1	
N... X... Y...	
N900 MSG ()	; Delete message.

Example 2: Message text with variable

Program code	Comment
N10 R12=$AA_IW [X]	; Actual position of the X axis in R12.
N20 MSG ("Check position of X axis"<<R12<<)	; Output message with variable R12.
...	
N90 MSG ()	; Clear message from N20.

15.2 Writing string in OPI variable (WRTPR)

Using the `WRTPR()` function, it is possible to write any character string from the part program into the OPI variable progProtText.

Syntax

```
WRTPR(<String>[,<ExecTime>])
```

Meaning

`WRTPR`:	Function call for outputting a character string.		
`<String>`:	Any character string, which is written to the OPI variable progProtText.		
	Type:	STRING	
	Maximum length:	128 characters	
`<ExecTime>`:	Optional parameters to define the instant in time when the string is written.		
	Type:	INT	
	Range of values:	0, 1	
		0 (default)	No dedicated main run block is not generated to write the character string. This is realized in the next NC block that can be executed. Active continuous-path mode is not interrupted.
		1	A dedicated main run block is generated to write the character string. Active continuous-path mode is interrupted.

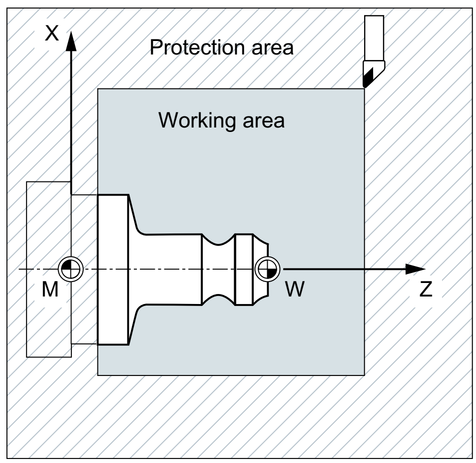
Examples

Program code	Comment
`N10 G91 G64 F100`	`; continuous path mode`
`N20 X1 Y1`	
`N30 WRTPR("N30")`	`; The character string "N30" is first written to N40.`
	`; Continuous-path mode is kept.`
`N40 X1 Y1`	
`N50 WRTPR("N50",1)`	`; The character string "N50" is written to N50.`
	`; Continuous-path mode is interrupted.`
`N60 X1 Y1`	

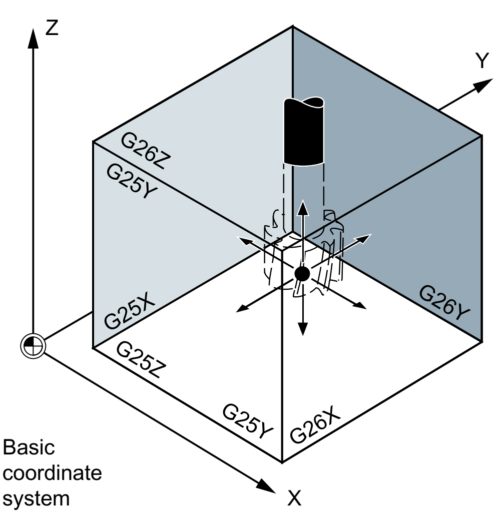
15.3 Working area limitation

15.3.1 Working area limitation in BCS (G25/G26, WALIMON, WALIMOF)

`G25`/`G26` limits the working area (working field, working space) in which the tool can traverse. The areas outside the working area limitations defined with `G25`/`G26` are inhibited for any tool motion.



The coordinates for the individual axes apply in the basic coordinate system:



The working area limitation for all validated axes must be programmed with the `WALIMON` command. The `WALIMOF` command deactivates the working area limitation. `WALIMON` is the default setting. Therefore, it only has to be programmed if the working area limitation has been disabled beforehand.

Syntax

```
G25 X…Y…Z…
G26 X…Y…Z…
WALIMON
...
WALIMOF
```

Meaning

`G25`:	**Lower** working area limitation Assignment of values in channel axes in the basic coordinate system
`G26`:	**Upper** working area limitation Assignment of values in channel axes in the basic coordinate system
`X… Y… Z…`:	Lower or upper working area limits for individual channel axes The limits specified refer to the basic coordinate system.
`WALIMON`:	Switch working area limitation **on** for all axes
`WALIMOF`:	Switch working area limitation **off** for all axes

In addition to programming values using `G25`/`G26`, values can also be entered using axis-specific setting data:

SD43420 $SA_WORKAREA_LIMIT_PLUS (Working area limitation plus)

SD43430 $SA_WORKAREA_LIMIT_MINUS (Working area limitation minus)

Activating and deactivating the working area limitation, parameterized using SD43420 and SD43430, are carried out for a specific direction using the axis-specific setting data that becomes immediately effective:

SD43400 $SA_WORKAREA_PLUS_ENABLE (Working area limitation active in the positive direction)

SD43410 $SA_WORKAREA_MINUS_ENABLE (Working area limitation active in the negative direction)

Using the direction-specific activation/deactivation, it is possible to limit the working range for an axis in just one direction.

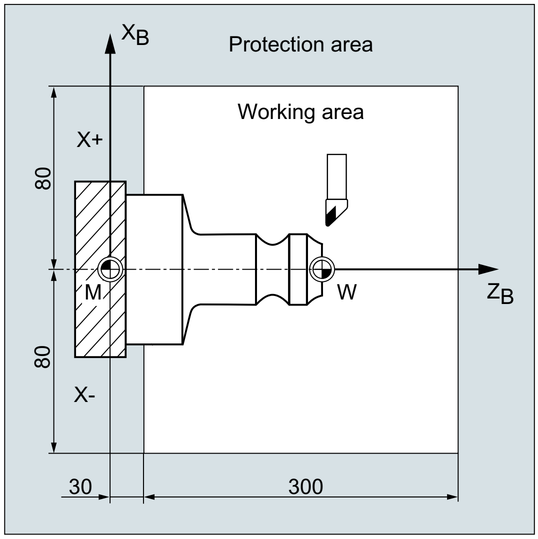
Note

The programmed working area limitation, programmed with `G25`/`G26`, has priority and overwrites the values entered in SD43420 and SD43430.

Note

`G25`/`G26` can also be used to program limits for spindle speeds at the address `S`. For more information see " Programmable spindle speed limitation (G25, G26) (Page 106) ".

Example



Using the working area limitation `G25/26`, the working area of a lathe is limited so that the surrounding devices and equipment - such as revolver, measuring station, etc. are protected against damage.

Default setting: `WALIMON`

Program code	Comment
`N10 G0 G90 F0.5 T1`	
`N20 G25 X-80 Z30`	`; Define the lower limit for the individual coordinate axes`
`N30 G26 X80 Z330`	`;Define the upper limit`
`N40 L22`	`;Cutting program`
`N50 G0 G90 Z102 T2`	`;To tool change location`
`N60 X0`	
`N70 WALIMOF`	`;Deactivate working area limitation`
`N80 G1 Z-2 F0.5`	`;Drill`
`N90 G0 Z200`	`;Back`
`N100 WALIMON`	`; Switch on working area limitation`
`N110 X70 M30`	`; End of program`

Further information

Reference point at the tool

When tool length offset is active, the tip of the tool is monitored as reference point, otherwise it is the toolholder reference point.

Consideration of the tool radius must be activated separately. This is done using channel-specific machine data:

MD21020 $MC_WORKAREA_WITH_TOOL_RADIUS

If the tool reference point lies outside the working area defined by the working area limitation or if this area is left, the program sequence is stopped.

Note

If transformations are active, the tool data taken into consideration (tool length and tool radius) can deviate from the described behavior.

References:
Function Manual, Basic Functions; Axis Monitoring, Protection Zones (A3),
Section: "Monitoring the working area limitation"

Programmable working area limitation, G25/G26

An upper (`G26`) and a lower (`G25`) working area limitation can be defined for each axis. These values are effective immediately and remain effective for the corresponding MD setting (→ MD10710 $MN_PROG_SD_RESET_SAVE_TAB) after RESET and after being powered-up again.

Note

The CALCPOSI subprogram is described in the Job Planning Programming Manual Using this subprogram before any traversing motion is made, it can be checked as to whether the predicted path is moved through taking into account the working area limits and/or the protection zones.

15.3.2 Working area limitation in WCS/SZS (WALCS0 ... WALCS10)

The "working area limitation in WCS/SZS" enables a flexible workpiece-specific limitation of the traversing range of the channel axes in the workpiece coordinate system (WCS) or settable zero system (SZS). It is intended mainly for use in conventional lathes.

Requirement

The channel axes must be homed.

Working area limitation group

In order that the axis-specific working area limits do not have to be rewritten for all channel axes when switching axis assignments, e.g. when switching transformations or the active frame on/off, working area limitation groups are available.

A working area limitation group comprises the following data:

- Working area limits for all channel axes
- Reference system of the working area limitation

Syntax

```
...
$P_WORKAREA_CS_COORD_SYSTEM[<WALimNo>]=<Value>
$P_WORKAREA_CS_PLUS_ENABLE[<WALimNo>,<Ax>]=<Value>
$P_WORKAREA_CS_LIMIT_PLUS[<WALimNo>,<Ax>]=<Value>
$P_WORKAREA_CS_MINUS_ENABLE[<WALimNo>,<Ax>]=<Value>
$P_WORKAREA_CS_LIMIT_MINUS[<WALimNo>,<Ax>]=<Value>
...
WALCS<n>
...
WALCS0
```

Meaning

<table>
<tr><td colspan="4">$P_WORKAREA_CS_COORD_SYSTEM[<WALimNo>]=<Value></td></tr>
<tr><td colspan="4">The coordinate system to which the working area limitation group refers</td></tr>
<tr><td rowspan="3"><WALimNo>:</td><td colspan="3">Working area limitation group</td></tr>
<tr><td colspan="2">Type:</td><td>INT</td></tr>
<tr><td colspan="2">Range of values:</td><td>0 (group 1) ... 9 (group 10)</td></tr>
<tr><td rowspan="3"><Value>:</td><td colspan="3">Value of the type INT</td></tr>
<tr><td>1</td><td colspan="2">Workpiece coordinate system (WCS)</td></tr>
<tr><td>3</td><td colspan="2">Settable zero system (SZS)</td></tr>
</table>

<table>
<tr><td colspan="3">$$P_WORKAREA_CS_PLUS_ENABLE[<WALimNo>,<Ax>]=<Value></td></tr>
<tr><td colspan="3">Enable the working area limitation in the positive axis direction for the specified channel axis</td></tr>
<tr><td rowspan="3"><WALimNo>:</td><td colspan="2">Working area limitation group</td></tr>
<tr><td>Type:</td><td>INT</td></tr>
<tr><td>Range of values:</td><td>0 (group 1) ... 9 (group 10)</td></tr>
<tr><td><Ax>:</td><td colspan="2">Channel axis name</td></tr>
<tr><td rowspan="3"><Value>:</td><td colspan="2">Value of the type BOOL</td></tr>
<tr><td>0 (FALSE)</td><td>No release</td></tr>
<tr><td>1 (TRUE)</td><td>Release</td></tr>
</table>

<table>
<tr><td colspan="3">$P_WORKAREA_CS_MINUS_ENABLE[<WALimNo>,<Ax>]=<Value></td></tr>
<tr><td colspan="3">Enable the working area limitation in the negative axis direction for the specified channel axis</td></tr>
<tr><td rowspan="3"><WALimNo>:</td><td colspan="2">Working area limitation group</td></tr>
<tr><td>Type:</td><td>INT</td></tr>
<tr><td>Range of values:</td><td>0 (group 1) ... 9 (group 10)</td></tr>
<tr><td><Ax>:</td><td colspan="2">Channel axis name</td></tr>
<tr><td rowspan="3"><Value>:</td><td colspan="2">Value of the type BOOL</td></tr>
<tr><td>0 (FALSE)</td><td>This has not been released</td></tr>
<tr><td>1 (TRUE)</td><td>Enable</td></tr>
</table>

$P_WORKAREA_CS_LIMIT_PLUS[<WALimNo>,<Ax>]=<Value>		
Working area limitation in the **positive** direction of the specified channel axis		
<WALimNo>:	Working area limitation group	
	Type:	INT
	Range of values:	0 (group 1) ... 9 (group 10)
<Ax>:	Channel axis name	
<Value>:	Value of the type REAL	

$P_WORKAREA_CS_LIMIT_MINUS[<WALimNo>,<Ax>]=<Value>		
Working area limitation in the **negative** direction of the specified channel axis		
<WALimNo>:	Working area limitation group	
	Type:	INT
	Range of values:	0 (group 1) ... 9 (group 10)
<Ax>:	Channel axis name	
<Value>:	Value of the type REAL	

WALCS<n>:	Activation of the working area limitations of a working area limitation group	
<n>:	Number of the working area limitation group	
	Range of values:	1 ... 10

WALCS0:	Deactivation of the working area limits active in the channel

Note

The actual available number of working area limitation groups depends on the configuration (→ see details of the machine manufacturer).

Example

Three axes are defined in the channel: X, Y and Z

A working area limitation group No. 2 is to be defined and then activated in which the axes are to be limited in the WCS according to the following specifications:

- X axis in the plus direction: 10 mm
- X axis in the minus direction: No limitation
- Y axis in the plus direction: 34 mm
- Y axis in the minus direction: -25 mm

- Z axis in the plus direction: No limitation
- Z axis in the minus direction: -600 mm

Program code	Comment
...	
N51 $P_WORKAREA_CS_COORD_SYSTEM[1]=1	; The working area limitation of working area limitation group 2 applies in the WCS.
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	
N82 $P_WORKAREA_CS_MINUS_ENABLE[1,Z]=TRUE	
N83 $P_WORKAREA_CS_LIMIT_PLUS[1,Z]=-600	
...	
N90 WALCS2	; Activate working area limitation group 2.
...	

Further information

Effectivity

The working area limitation with `WALCS1` - `WALCS10` acts independently of the working area limitation with `WALIMON`. If both functions are active, that limit becomes effective which the axis motion first reaches.

Reference point at the tool

Taking into account the tool data (tool length and tool radius) and therefore the reference point at the tool when monitoring the working area limitation corresponds to the behavior for the working area limitation with `WALIMON`.

15.4 Reference point approach (G74)

When the machine has been powered up (where incremental position measuring systems are used), all of the axis slides must approach their reference mark. Only then can traversing movements be programmed.

The reference point can be approached in the NC program with `G74`.

Syntax

`G74 X1=0 Y1=0 Z1=0 A1=0 …` ; Programmed in a separate NC block

Meaning

`G74`:	G command call reference point approach
`X1=0 Y1=0 Z1=0 …`:	The specified machine axis address `X1`, `Y1`, `Z1` … for **linear axes** is approached as the reference point.
`A1=0 B1=0 C1=0 …`:	The specified machine axis address `A1`, `B1`, `C1` … for **rotary axes** is approached as the reference point.

Note

A transformation must not be programmed for an axis which is to approach the reference point with `G74`.

The transformation is deactivated with command `TRAFOOF`.

Example

When the measuring system is changed, the reference point is approached and the workpiece zero point is set up.

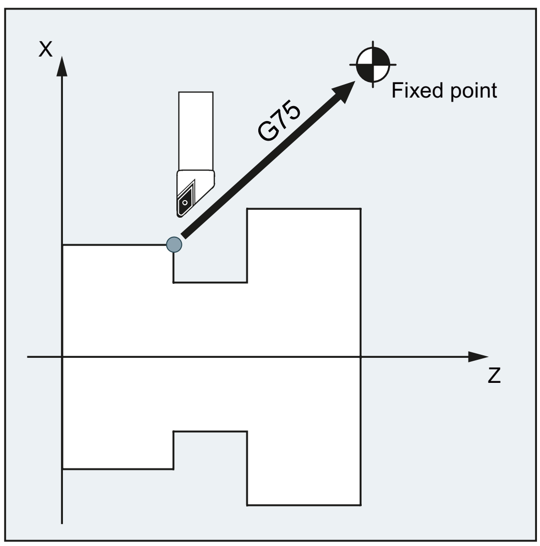
Program code	Comment
`N10 SPOS=0`	`;Spindle in position control`
`N20 G74 X1=0 Y1=0 Z1=0 C1=0`	`;Reference point approach for linear axes and rotary axes`
`N30 G54`	`; Zero offset`
`N40 L47`	`;Cutting program`
`N50 M30`	`; End of program`

15.5 Approaching a fixed point (G75)

The non-modal command `G75` can be used to move axes individually and independently of one another to fixed points in the machine space, e.g. to tool change points, loading points, pallet change points, etc.

The fixed points are positions in the machine coordinate system which are stored in the machine data (MD30600 $MA_FIX_POINT_POS[n]). A maximum of four fixed points can be defined for each axis.

The fixed points can be approached from every NC program irrespective of the current tool or workpiece positions. An internal preprocessing stop is executed prior to moving the axes.



Requirements

The following requirements must be satisfied to approach fixed points with `G75`:

- The fixed-point coordinates must have been calculated exactly and written to machine data.
- The fixed points must be located within the valid traversing range (→ note the software limit switch limits!)
- The axes to be traversed must be referenced.
- No tool radius compensation must be active.
- A kinematic transformation may not be active.
- None of the axes to be traversed must be involved in active transformation.
- None of the axes to be traversed must be a following axis in an active coupling.
- None of the axes to be traversed must be an axis in a gantry grouping.
- Compile cycles must not activate motion components.

Syntax

```
G75 <axis name><axis position> ... FP=<n>
```

Meaning

`G75`:	Fixed point approach		
`<axis name>`:	Name of the machine axis to be traversed to the fixed point All axis identifiers are permitted.		
`<axis position>`:	The position value has no significance. A value of "0" is, therefore, usually specified.		
`FP=`:	Fixed point that is to be approached		
	`<n>`:	Fixed point number	
		Range of values:	1, 2, 3, 4
	Note: In the absence of `FP=<n>` or a fixed point number, or if `FP=0` has been programmed, this is interpreted as `FP=1` and fixed point 1 is approached.		

Note

Multiple axes can be programmed in one `G75` block. The axes are then traversed simultaneously to the specified fixed point.

Note

The value of the address `FP` must not be greater than the number of fixed points specified for each programmed axis (MD30610 $MA_NUM_FIX_POINT_POS).

Example

For a tool change, axes X (= AX1) and Z (= AX3) need to move to the fixed machine axis position 1 where X = 151.6 and Z = -17.3.

Machine data:

- MD30600 $MA_FIX_POINT_POS[AX1,0] = 151.6
- MD30600 $MA_FIX_POINT[AX3,0] = 17.3

NC program:

Program code	Comment
…	
N100 G55	; Activate settable zero offset.
N110 X10 Y30 Z40	; Approach positions in the WCS.

Program code	Comment
N120 G75 X0 Z0 FP=1 M0	; The X axis moves to 151.6
	; and the Z axis moves to 17.3 (in the MCS).
	; Each axis travels at its maximum velocity.
	; No additional movements are permitted to be active in this block.
	; A stop is inserted here so that after reaching
	; the end positions,
	; no additional motion takes place.
N130 X10 Y30 Z40	; The position of N110 is approached again.
	; The zero offset is reactivated.
…	

Note

If the "Tool management with magazines" function is active, the auxiliary function `T…` or `M...` (typically `M6`) will not be sufficient to trigger a block change inhibit at the end of `G75` motion.

Reason: If "Tool management with magazines" is active, auxiliary functions for tool change are not output to the PLC.

Further information

G75

The axes are traversed as machine axes in rapid traverse. The motion is mapped internally using the "SUPA" (suppress all frames) and "G0 RTLIOF" (rapid traverse motion with single-axis interpolation) functions.

If the conditions for "RTLIOF" (single-axis interpolation) are not met, the fixed point is approached as a path.

When the fixed point is reached, the axes come to a standstill within the "Exact stop fine" tolerance window.

Parameterizable dynamic response for G75

The required dynamic response mode can be set via the following machine data for positioning movements to fixed-point positions (`G75`):

MD18960 $MN_POS_DYN_MODE (type of positioning axis dynamic response)

References
Function Manual, Basic Functions, Chapter "Acceleration (B2)" > "Functions" > "Jerk limiting for single axis interpolation (SOFTA) (axis-specific)"

Additional axis movements

The following additional axis movements are taken into account at the instant in time at which the G75 block is interpolated:

- External zero offset
- DRF
- Synchronization offset ($AA_OFF)

After this, the additional axis movements are not permitted to change until the end of traversing is reached by the G75 block.

Additional movements following interpretation of the G75 block will offset the approach to the fixed point accordingly.

The following additional movements are not taken into account, irrespective of the point at which interpolation takes place, and will offset the target position accordingly:

- Online tool offset
- Additional movements from compile cycles in the BCS and machine coordinate system

Active frames

All active frames are ignored. Traversing is performed in the machine coordinate system.

Working area limitation in the workpiece coordinate system/SZS

Coordinate-system-specific working area limitation (WALCS0 ... WALCS10) is not effective in the block with G75. The destination point is monitored as the starting point of the following block.

Axis/Spindle movements with POSA/SPOSA

If programmed axes/spindles were previously traversed with POSA or SPOSA, these movements will be completed first before the fixed point is approached.

Spindle functions in the G75 block

If the spindle is excluded from "fixed-point approach", then additional spindle functions (e.g. positioning with SPOS/SPOSA) can be programmed in the G75.

Modulo axes

In the case of modulo axes, the fixed point is approached along the shortest distance.

References

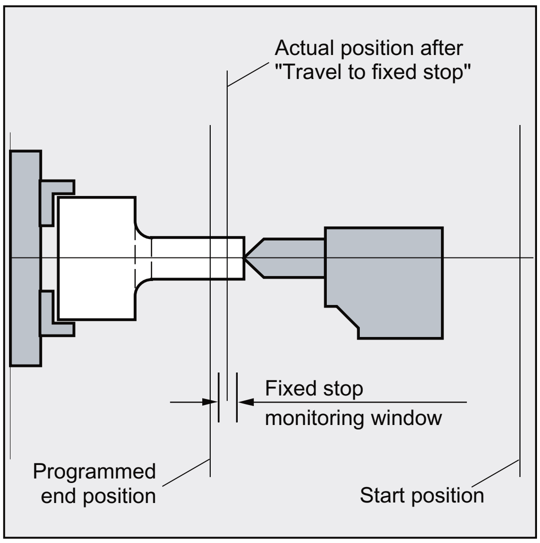
For further information about "Fixed-point approach", see:

Function Manual Extended Functions; manual traversing and manual handwheel travel (H1), Chapter: "Fixed-point approach in JOG"

15.6 Travel to fixed stop (FXS, FXST, FXSW)

Function

The "Travel to fixed stop" function can be used to establish defined forces for clamping workpieces, such as those required for tailstocks, quills and grippers. The function can also be used for the approach of mechanical reference points.



With sufficiently reduced torque, it is also possible to perform simple measurement operations without connecting a probe. The "travel to fixed stop" function can be implemented for axes as well as for spindles with axis-traversing capability.

Syntax

```
FXS[<axis>]=…
FXST[<axis>]=…
FXSW[<axis>]=…
FXS[<axis>]=… FXST[<axis>]=…
FXS[<axis>]=… FXST[<axis>]=… FXSW[<axis>]=…
```

Meaning

FXS:	Command for activation and deactivation of the "Travel to fixed stop" function	
	FXS[<axis>]=1:	Activate function
	FXS=[<axis>]=0:	Deactivate function
FXST:	Optional command for setting the clamping torque Specified as % of the maximum drive torque	
FXSW:	Optional command for setting the window width for the fixed stop monitoring Specified in mm, inches or degrees	
<axis>:	Machine axis name Machine axes (X1, Y1, Z1, etc.) are programmed	

Note

The commands `FXS`, `FXST` and `FXSW` are modal.

The programming of `FXST` and `FXSW` is optional: If no parameter is specified, the last programmed value or the value set in the relevant machine data applies.

Activate travel to fixed stop: FXS[<axis>] = 1

The movement to the destination point can be described as a path or positioning axis movement. With positioning axes, the function can be performed across block boundaries.

Travel to fixed stop can be performed simultaneously for several axes and parallel to the movement of other axes. The fixed stop must be located between the start and end positions.

NOTICE
Risk of collision
It is not permissible to program a new position for an axis if the "Travel to fixed stop" function has already been activated for an axis/spindle.
Spindles must be switched to position-controlled mode before the function is selected.

Example:

Program code	Comment
X250 Y100 F100 FXS[X1]=1 FXST[X1]=12.3 FXSW[X1]=2	; Axis X1 travels with feedrate F100 (specification optional) to target position X=250 mm. The clamping torque is 12.3% of the maximum drive torque, monitoring is performed in a 2 mm wide window.
...	

Deactivate travel to fixed stop: FXS[<axis>] = 0

Deselection of the function triggers a preprocessing stop.

The block with `FXS[<axis>]=0` may and should contain traversing movements.

NOTICE
Risk of collision
The traversing movement to the retraction position must move away from the fixed stop, otherwise damage to the stop or to the machine may result.
The block change takes place when the retraction position has been reached. If no retraction position is specified, the block change takes place immediately after the torque limit has been deactivated.

Example:

Program code	Comment
`X200 Y400 G01 G94 F2000 FXS[X1]=0`	`; Axis X1 is retracted from the fixed stop to position X = 200 mm. All other parameters are optional.`
`...`	

Clamping torque (FXST) and monitoring window (FXSW)

Any programmed torque limiting `FXST` is effective from the block start, i.e. the fixed stop is also approached at a reduced torque. `FXST` and `FXSW` can be programmed and changed in the part program at any time. The changes take effect before traversing movements in the same block.

NOTICE
Risk of collision
Programming of a new fixed stop monitoring window causes a change not only in the window width, but also in the reference point for the center of the window if the axis has moved prior to reprogramming. The actual position of the machine axis when the window is changed is the new window center point.
The window must be selected such that only a breakaway from the fixed stop causes the fixed stop monitoring to respond.

Further information

Rise ramp

A rate of rise ramp for the new torque limit can be defined in MD to prevent any abrupt changes to the torque limit setting (e.g. insertion of a quill).

Alarm suppression

The fixed stop alarm can be suppressed for applications by the part program by masking the alarm in a machine data item and activating the new MD setting with NEW_CONF.

Activation

The commands for travel to fixed stop can be called from synchronized actions or technology cycles. They can be activated without initiation of a motion, the torque is limited instantaneously. As soon as the axis is moved via a setpoint, the limit stop monitor is activated.

Activation from synchronized actions

Example:

If the expected event ($R1) occurs and travel to fixed stop is not yet running, `FXS` should be activated for axis Y. The torque must correspond to 10% of the rated torque value. The width of the monitoring window is set to the default.

Program code

```
N10 IDS=1 WHENEVER (($R1=1) AND ($AA_FXS[Y]==0)) DO $R1=0 FXS[Y]=1
FXST[Y]=10
```

The normal part program must ensure that $R1 is set at the desired point in time.

Deactivation from synchronized actions

Example:

If an anticipated event ($R3) has occurred and the status "Limit stop contacted" (system variable $AA_FXS) is reached, then FXS must be deselected.

Program code

```
IDS=4 WHENEVER (($R3==1) AND ($AA_FXS[Y]==1)) DO FXS[Y]=0
FA[Y]=1000 POS[Y]=0
```

Fixed stop reached

When the fixed stop has been reached:

- The distance-to-go is deleted and the set position is tracked.
- The drive torque increases up to the programmed limit value `FXSW`, and then remains constant.
- Fixed stop monitoring is activated within the specified window width.

Supplementary conditions

- Measurement with delete distance-to-go
  "Measurement with delete distance-to-go" (`MEAS` command) and "Travel to fixed stop" cannot be programmed at the same time in one block.
  Exception: One function acts on a path axis and the other on a positioning axis or both act on positioning axes.
- Contour monitoring
  Contour monitoring is not performed while "Travel to fixed stop" is active.
- Positioning axes
  For "Travel to fixed stop" with positioning axes, the block change is performed irrespective of the fixed stop movement.

- Link and container axes
  Travel to fixed stop is also permitted for link and container axes.
  The status of the assigned machine axis is maintained beyond the container rotation. This also applies for modal torque limiting with FOCON.
  References:
  - Function Manual, Extended Functions; Several Control Panels on Multiple NCUs, Distributed Systems (B3)
  - Programming Manual, Job Planning; Subject: "Travel to fixed stop (FXS and FOCON/FOCOF)"
- Travel to fixed stop is **not** possible:
  - With gantry axes
  - For competing positioning axes that are controlled exclusively from the PLC (`FXS` must be selected from the NC program).
- If the torque limit is reduced too far, the axis will not be able to follow the specified setpoint; the position controller then goes to the limit and the contour deviation increases. In this operating state, an increase in the torque limit may result in sudden, jerky movements. To ensure that the axis can follow the setpoint, check the contour deviation to make sure it is not greater than the deviation with an unlimited torque.

15.7 Dwell time (G4)

With the command `G4`, a time (dwell time) is programmed in a block that expires as soon as the block is executed in the main run. The block change to the following block is performed as soon as the time has completely expired.

Note

`G4` interrupts continuous-path mode.

Syntax

```
G4 F<Time>
G4 S<NumSpi>
G4 S<n> = <NumSpi>
```

Meaning

`G4`:	Activate dwell time	
	Alone in the block:	Yes
`F<Time>`:	The dwell time `<Time>` in seconds is specified at address `F`.	
`S<NumSpi>`:	The dwell time is programmed at address S in spindle revolutions `<NumSpi>` with reference to the current main spindle.	
`S<n>=NumSpi>`:	The dwell time is programmed at address S in spindle revolutions `<NumSpi>` with reference to the spindle addressed with the address extension `<n>`.	

Note

The addresses `F` and `S` used for the time specified in the dwell block `G4` do not influence the feedrates `F...` and the spindle speeds `S...` of the program.

Supplementary conditions

Synchronized actions

Two synchronized actions are programmed in one program in such a way that the following block with the dwell time becomes the action block in which the synchronized actions are performed. One synchronized action is a modal synchronized action. The other synchronized action is a non-modal synchronized action. If the non-modal synchronized action is intended to influence the model synchronized action, e.g. release it for execution with UNLOCK, **at least two interpolation cycles** e.g. `G4 F<interpolator_cycle * 2>` must be provided as the effective dwell time.

The effective dwell time depends on the setting in the machine data MD10280 $MN_PROG_FUNCTION_MASK, Bit 4 = <value>

Value	Meaning
0	The effective dwell time is equal to the programmed dwell time
1	The effective dwell time is equal to the programmed dwell time rounded to the next largest multiple of the interpolator cycle (MD10071 $MN_IPO_CYCLE_TIME)

Program example:

- MD10071 $MN_IPO_CYCLE_TIME == 8 ms
- MD10280 $MN_PROG_FUNCTION_MASK, Bit 4 = 1

Program code	Comment
N10 WHEN TRUE DO LOCK(1)	; Non-modal SynAct: LOCK of the
	; modal SynAct. ID=1
N20 G4 F2	; Action block for SynAct from N10
N30 WHEN TRUE DO UNLOCK(1)	; Non-modal SynAct: UNLOCK
	; of the modal SynAct. ID=1
N40 ID=1 WHENEVER TRUE DO $R0=1 RDISABLE	; Modal SynAct ID=1
	; R parameter R0=1
	; Set the read-in disable
N50 G4 F0.012	; Action block for SynAct from N40 and N50
	; See paragraph "Description" below
N60 G4 F10	

Description

The desired behavior is that the modal synchronized action from N30 cancels the active lock (LOCK) of the modal synchronized action with ID=1 from N40, causing the R parameter to be written in N50 and the read-in disable to become active. This behavior is only achieved if the active dwell time is at least two interpolation cycles long.

The active dwell time results from the programmed dwell time, the interpolation cycle, and the setting in MD10280 $MN_PROG_FUNCTION_MASK, Bit 4. To ensure that the active dwell time is at least two interpolation cycles long, the following dwell time must be programmed:

- Bit 4 == 0: Programmed dwell time ≥ 2 * interpolator cycle
- Bit 4 == 1: Programmed dwell time ≥ 1.5 * interpolator cycle

If the active dwell time is shorter than two interpolation cycles, writing the R parameter and read-in disable will not be executed until block N60.

Example

Program code	Comment
N10 G1 F200 Z-5 S300 M3	;Feed F; spindle speed S
N20 G4 F3	; Dwell time: 3 s
N30 X40 Y10	

Program code	Comment
N40 G4 S30	; Dwelling 30 revolutions of the spindle (at S=300 rpm and 100% speed override, corresponds to: t = 0.1 min).
N50 X...	; The feedrate and spindle speed programmed in N10 continue to apply.

15.8 Internal preprocessing stop

Function

The control generates an internal preprocessing stop on access to machine status data ($A...). The following block is not executed until all preprocessed and saved blocks have been executed in full. The previous block is stopped in exact stop (as G9).

Example

Program code	Comments
...	
N40 POSA[X]=100	
N50 IF $AA_IM[X]==R100 GOTOF MARKE1	; Access to machine status data ($A...), the control generates an internal pre-processing stop.
N60 G0 Y100	
N70 WAITP(X)	
N80 LABEL1:	
...	

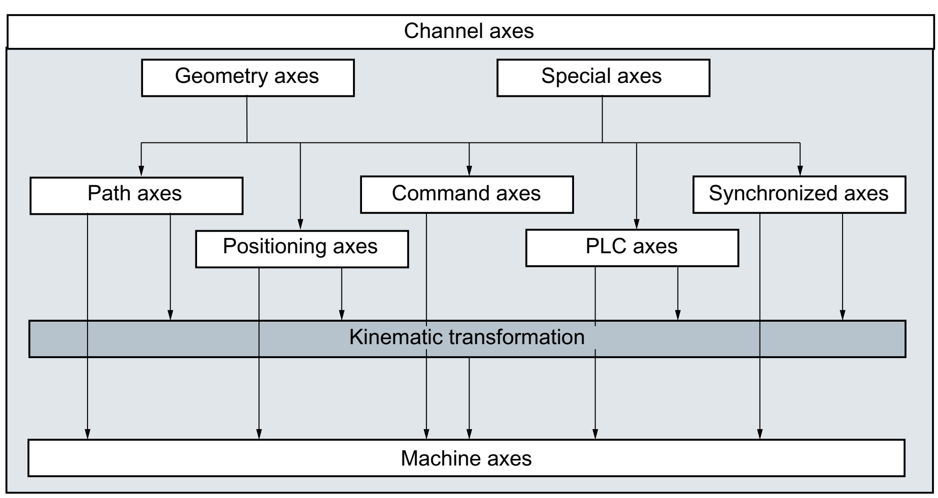
16 Other information

16.1 Axes

Axis types

A distinction is made between the following types of axis types when programming:

- Machine axes
- Geometry axes
- Special axes
- Path axes
- Synchronized axes
- Positioning axes
- Command axes
- PLC axes / competing positioning axes
- Link axes (function NCU link)
- Lead link axes (function NCU link)



16.1.1 Main axes/Geometry axes

The main axes define a right-angled, right-handed coordinate system. Tool movements are programmed in this coordinate system.

In NC technology, the main axes are called geometry axes. This term is also used in this Programming Guide.

Replaceable geometry axes

The "Replaceable geometry axes" function (see Function Manual, Job Planning) can be used to alter the geometry axes grouping configured using machine data from the part program. Here any geometry axis can be replaced by a channel axis defined as a synchronous special axis.

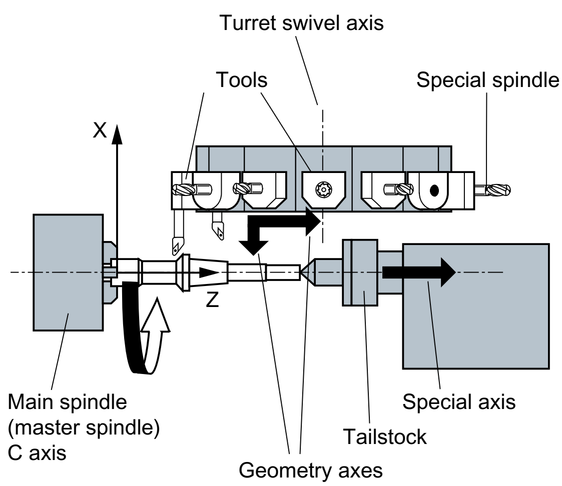
Axis identifier

The name/identifier of a geometry axis can be defined using the following machine data:

MD20060 $MC_AXCONF_GEOAX_NAME_TAB (name of the geometry axis in the channel)

Standard identifier for turning machines:

1. Geometry axis: X
2. Geometry axis: Z



Standard identifier for milling machines:

1. Geometry axis: X
2. Geometry axis: Y
3. Geometry axis: Z

Further information

A maximum of three geometry axes are used for programming frames and the workpiece geometry (contour).

The identifiers for geometry and channel axes may be the same, provided a reference is possible.

Geometry and channel axis names must be the same in all channels. This means that a program can be executed in any channel.

16.1.2 Special axes

In contrast to the geometry axes, no geometrical relationship is defined between the special axes.

Typical special axes are:

- Tool revolver axes
- Swivel table axes
- Swivel head axes
- Loader axes

Axis identifier

On a turning machine with circular magazine, for example:

- Revolver position U
- Tailstock V

Programming example

Program code	Comment
N10 G1 X100 Y20 Z30 A40 F300	; Path axis movements
N20 POS[U]=10POS[X]=20 FA[U]=200 FA[X]=350	; Positioning axis movements.
N30 G1 X500 Y80 POS[U]=150FA[U]=300 F550	; Path and positioning axis.
N40 G74 X1=0 Z1=0	; Approach reference point.

16.1.3 Main spindle, master spindle

The machine kinematics determine, which spindle is the main spindle. This spindle is usually declared as the master spindle in the machine data.

This assignment can be changed with the `SETMS(<spindle number>)` program command. `SETMS` can be used without specifying a spindle number to switch back to the master spindle defined in the machine data.

Special functions such as thread cutting are supported by the master spindle.

Spindle identifier

S or S0

16.1.4 Machine axes

Machine axes are the axes physically existing on a machine.

The programmed motion of a path or additional axis can act on several machine axes due to transformation (TRANSMIT, TRACYL or TRAORI) active in the channel.

Machine axes are only directly addressed in the program in special circumstances (e.g. for reference point or fixed point approach).

Axis identifier

The name/identifier of a machine axis can be defined using the following NC-specific machine data:

MD10000 $MN_AXCONF_MACHAX_NAME_TAB (machine axis name)

Default setting: X1, Y1, Z1, A1, B1, C1, U1, V1

Further, machine axes have fixed axis identifiers, which can always be used, independent of the names set in the machine data:

AX1, AX2, …, AX<n>

16.1.5 Channel axes

All geometry, special and machine axes, which are assigned to a channel, are called channel axes.

Axis identifier

The channel-specific name/identifier of a geometry and special axis can be defined using the following machine data:

MD20080 $MC_AXCONF_CHANAX_NAME_TAB (channel axis name)

Default setting: X, Y, Z, A, B, C, U, V

The assignment regarding on which machine axis a geometry or special axis is emulated in the channel, is defined in the following machine data:

MD20070 $MC_AXCONF_MACHAX_USED (machine axes used)

16.1.6 Path axes

Path axes define the path and therefore the movement of the tool in space.

The programmed feed is active for this path. The axes involved in this path reach their position at the same time. As a rule, these are the geometry axes.

However, default settings define, which axes are the path axes, and therefore determine the velocity.

Path axes can be specified in the NC program with `FGROUP`.

For more information about `FGROUP`, see "Feedrate (G93, G94, G95, F, FGROUP, FL, FGREF) (Page 107)".

16.1.7 Positioning axes

Positioning axes are interpolated separately; in other words, each positioning axis has its own axis interpolator and its own feedrate. Positioning axes do not interpolate with the path axes.

Positioning axes are traversed by the NC program or the PLC. If an axis is to be traversed simultaneously by the NC program and the PLC, an error message appears.

Typical positioning axes are:

- Loaders for moving workpieces to the machine
- Loaders for moving workpieces away from the machine
- Tool magazine/turret

Types

A distinction is made between positioning axes with synchronization at the block end or over several blocks.

POS axes

Block change occurs at the end of the block when all the path and positioning axes programmed in this block have reached their programmed end point.

POSA axes

The movement of these positioning axes can extend over several blocks.

POSP axes

The movement of these positioning axes for approaching the end position takes place in sections.

Note

Positioning axes become synchronized axes if they are traversed without the special POS/POSA identifier.

Continuous-path mode (G64) for path axes is only possible if the positioning axes (POS) reach their final position before the path axes.

Path axes programmed with `POS`/`POSA` are removed from the path axis grouping for the duration of this block.

For more information about `POS`, `POSA`, and `POSP`, see "Traverse positioning axes (POS, POSA, POSP, FA, WAITP, WAITMC) (Page 116)".

16.1.8 Synchronized axes

Synchronized axes traverse synchronously to the path from the start position to the programmed end position.

The feedrate programmed in `F` applies to all the path axes programmed in the block, but does not apply to synchronized axes. Synchronized axes take the same time as the path axes to traverse.

A synchronized axis can be a rotary axis, which is traversed synchronously to the path interpolation.

16.1.9 Command axes

Command axes are started from synchronized actions in response to an event (command). They can be positioned, started, and stopped fully asynchronous to the parts program. An axis cannot be moved from the part program and from synchronized actions simultaneously.

Command axes are interpolated separately; in other words, each command axis has its own axis interpolator and its own feedrate.

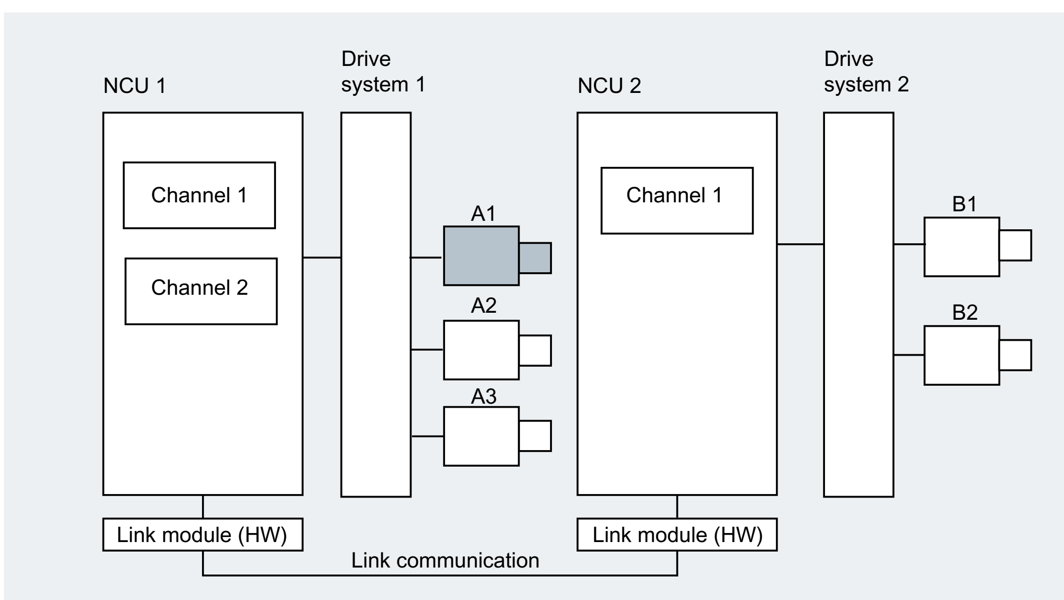
References:
Function Manual, Synchronized Actions

16.1.10 PLC axes

PLC axes are traversed by the PLC via special function blocks in the basic program; their movements can be asynchronous to all other axes. Traversing movements take place independently of path and synchronized movements.

16.1.11 Link axes

Link axes are axes, which are physically connected to another NCU and whose position is controlled from this NCU. Link axes can be assigned dynamically to channels of **another** NCU. Link axes are non-local axes from the perspective of a specific NCU.



The **axis container** concept is used for the dynamic modification of the assignment to an NCU. Axis exchange with `GET` and `RELEASE` from the part program is **not** available for link axes.

Further information

Requirements

- The participating NCUs, NCU1 and NCU2, must be connected by means of high-speed communication via the link module.
  References:
  Configuration Manual, NCU
- The axis must be configured appropriately by machine data.
- The "Link axis" option must be installed.

Description

The position control is implemented on the NCU on which the axis is physically connected to the drive. This NCU also contains the associated axis VDI interface. The position setpoints for link axes are generated on another NCU and communicated via the NCU link.

The link communication must provide the means of interaction between the interpolators and the position controller or PLC interface. The setpoints calculated by the interpolators must be transported to the position control loop on the home NCU and, vice versa, the actual values must be returned from there back to the interpolators.

References:
For more detailed information about link axes see:
Function Manual, Advanced Functions; Multiple Operator Panels and NCUs (B3)

Axis container

An axis container is a circular buffer data structure, in which local axes and/or link axes are assigned to channels. The entries in the circular buffer can be **shifted cyclically**.

In addition to the direct reference to local axes or link axes, the link axis configuration in the logical machine axis image also allows references to axis containers. This type of reference consists of:

- A container number **and**
- A slot (circular buffer location within the container)

The entry in a circular buffer location contains:

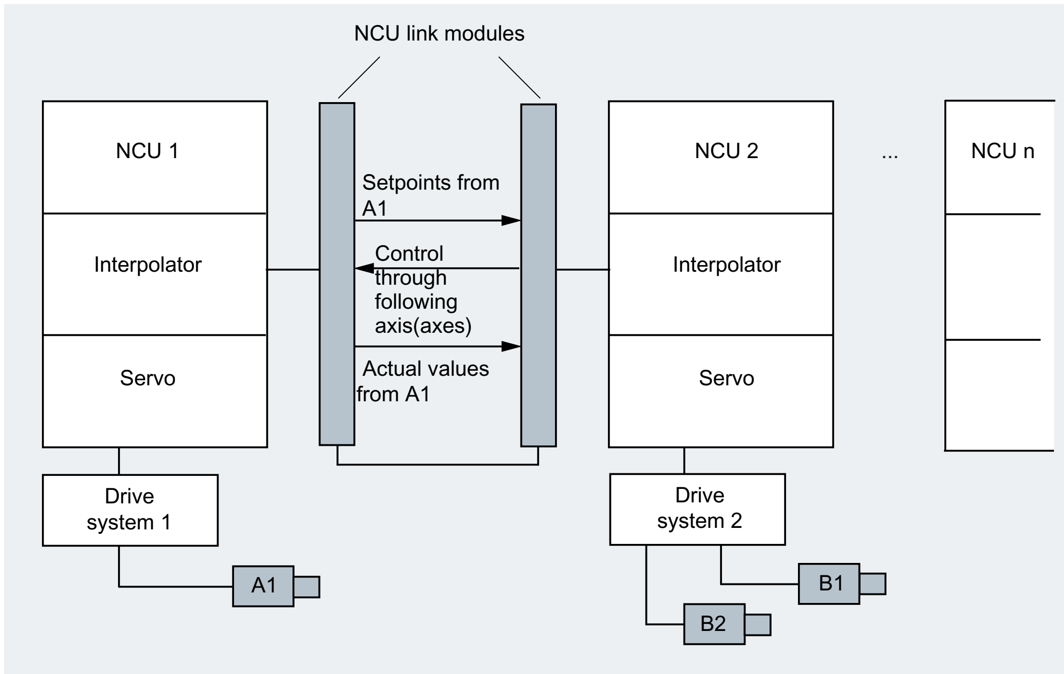
- A local axis **or**
- A link axis

Axis container entries contain local machine axes or link axes from the perspective of an individual NCU. The entries in the logical machine axis image (MD10002 $MN_AXCONF_LOGIC_MACHAX_TAB) of an individual NCU are fixed.

References:
The axis container function is described in:
Function Manual, Advanced Functions; Multiple Operator Panels and NCUs (B3)

16.1.12 Lead link axes

A leading link axis is one that is interpolated by one NCU and utilized by one or several other NCUs as the master axis for controlling slave axes.



An axial position controller alarm is sent to all other NCUs, which are connected to the affected axis via a leading link axis.

NCUs that are dependent on the leading link axis can utilize the following coupling relationships with it:

- Master value (setpoint, actual master value, simulated master value)
- Coupled motion
- Tangential correction
- Electronic gear (ELG)
- Synchronous spindle

Programming

Master NCU:

Only the NCU, which is physically assigned to the master value axis can program travel motions for this axis. The travel program must not contain any special functions or operations.

NCUs of slave axes:

The travel program on the NCUs of the slave axes must not contain any travel commands for the leading link axis (master value axis). Any violation of this rule triggers an alarm.

The leading link axis is addressed in the usual way via channel axis identifiers. The states of the leading link axis can be accessed by means of selected system variables.

Further information

Conditions

- The NCUs involved, i.e., NCU1 to NCU<n> (<n> equals max. of 8), must be interconnected via the link module for high-speed communication.
  Reference:
  Device Manual Configuration NCU
- The axis must be configured appropriately via machine data.
- The "Link axis" option must be installed.
- The same interpolator clock cycle must be configured for all NCUs connected to the leading link axis.

Restrictions

- A master axis which is a leading link axis cannot be a link axis, i.e. it cannot be traversed by NCUs other than its home NCU.
- A master axis which is a leading link axis cannot be a container axis, i.e. it cannot be addressed alternately by different NCUs.
- A leading link axis cannot be the programmed leading axis in a gantry grouping.
- Couplings with leading link axes cannot be cascaded.
- Axis replacement can only be implemented within the home NCU of the leading link axis.

System variables

The following system variables can be used in conjunction with the channel axis identifier of the leading link axis:

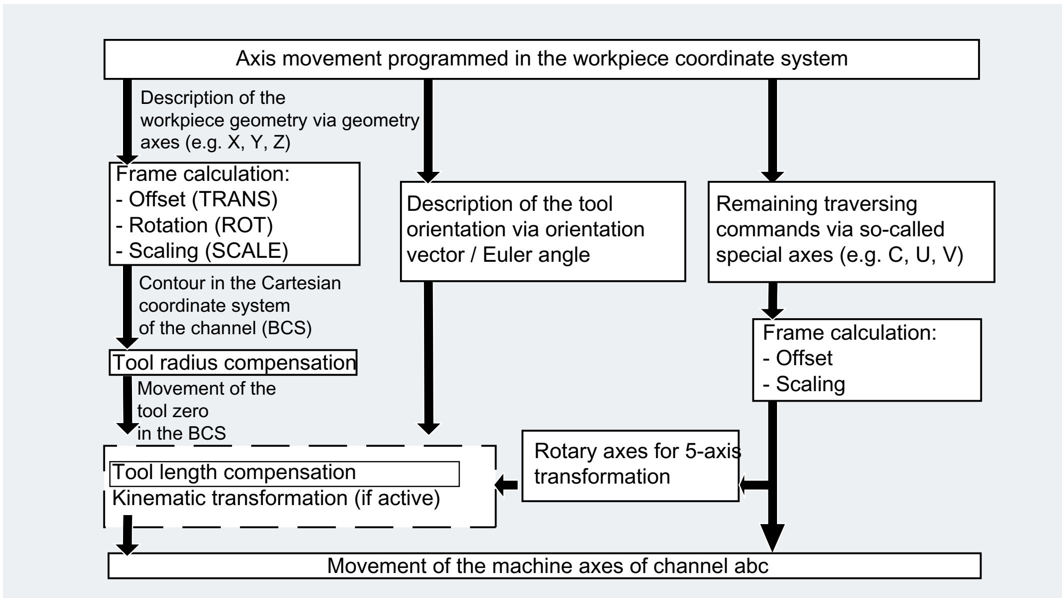
System variables	Meaning
$AA_LEAD_SP	Simulated master value - position
$AA_LEAD_SV	Simulated master value - velocity

If these system variables are updated by the home NCU of the master axis, the new values are also transferred to any other NCUs, which wish to control slave axes as a function of this master axis.

References:
Function Manual, Extended Functions; Multiple Operator Panels and NCUs (B3)

16.2 From travel command to machine movement

The relationship between the programmed axis movements (travel commands) and the resulting machine movements is illustrated in the following figure:

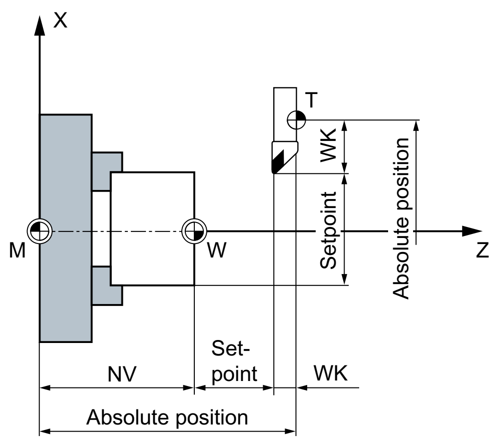


16.3 Path calculation

The path calculation determines the distance to be traversed in a block, taking into account all offsets and compensations.

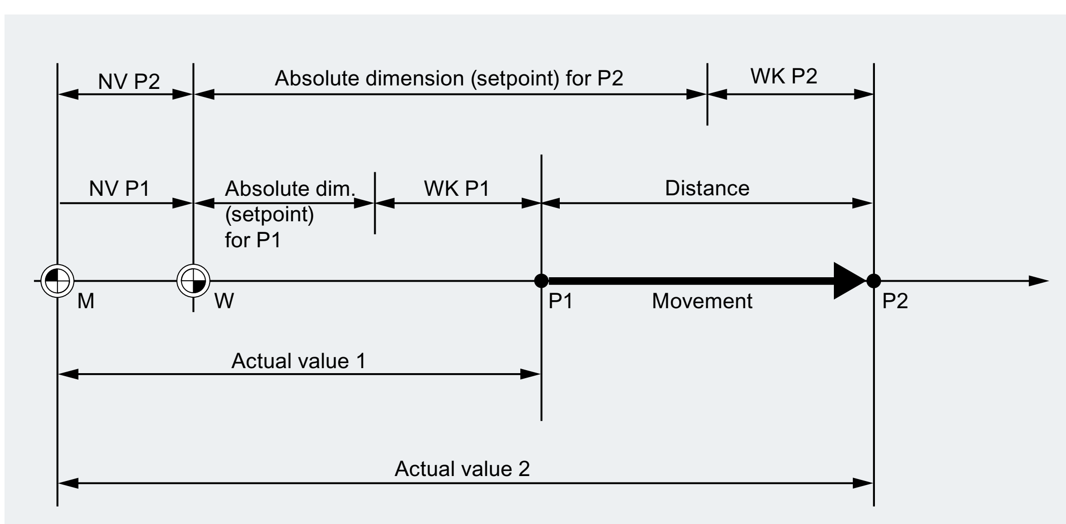
In general:

Path =
setpoint - actual value + zero offset (ZO) + tool offset (TO)



If a new zero offset and a new tool offset are programmed in a new program block, the following applies:

- With absolute dimensioning:
  Path = (absolute dimension P2 - absolute dimension P1) + (WO P2 - WO P1) +
  (TO P2 - TO P1).
- With incremental dimensioning:
  Path = incremental dimension + (WO P2 - WO P1) +
  (TO P2 - TO P1).



16.4 Addresses

Fixed addresses

These addresses are permanently set, i.e. the address characters cannot be changed.

A list can be found in Table "Fixed addresses (Page 507)".

Settable addresses

The machine manufacturer may assign another name to these addresses via machine data.

Note

Settable addresses must be unique within the control, i.e. the same address name must not be used for different address types (axis values and end points, tool orientation, interpolation parameters, etc.).

A list can be found in Table "Settable addresses (Page 511)".

Modal/non-modal addresses

Modal addresses remain valid with the programmed value (in all subsequent blocks) until a new value is programmed at the same address.

Non-modal addresses only apply in the block, in which they were programmed.

Example:

Program code	Comment
N10 G01 F500 X10	
N20 X10	; Feedrate F from N10 remains active until a new feedrate is entered.

Addresses with axial extension

In addresses with axial extension, an axis name is inserted in square brackets after the address. The axis name assigns the axis.

Example:

Program code	Comment
FA[U]=400	; Axis-specific feedrate for U axis.

See also Table "Fixed addresses (Page 507)".

Extended address notation

Extended address notation enables a larger number of axes and spindles to be organized in a system.

An extended address consists of a numeric extension and an arithmetic expression assigned with an "=" character. The numeric extension has one or two digits and is always positive.

The extended address notation is only permitted for the following direct addresses:

Address	Meaning
X, Y, Z, …	Axis addresses
I, J, K	Interpolation parameters
S	Spindle speed
SPOS, SPOSA	Spindle position
M	Special functions
H	Auxiliary functions
T	Tool number
F	Feedrate

Examples:

Program code	Comment
`X7`	`; No "=" required, 7 is a value, but the "=" character can also be used here`
`X4=20`	`; Axis X4; "=" is required`
`CR=7.3`	`; Two letters; "=" are required`
`S1=470`	`; Speed for 1st spindle: 470 rpm`
`M3=5`	`; Spindle stop for 3rd spindle`

The numeric extension can be replaced by a variable for addresses M, H, S and for SPOS and SPOSA. The variable identifier is enclosed in square brackets.

Examples:

Program code	Comment
`S[SPINU]=470`	`; Speed for the spindle whose number is stored in the SPINU variable.`
`M[SPINU]=3`	`; Clockwise rotation for the spindle whose number is stored in the SPINU variable.`
`T[SPINU]=7`	`; Selection of the tool for the spindle whose number is stored in the SPINU variable.`

16.5 Names

The commands according to DIN 66025 are supplemented with named objects, etc. by the NC high-level language.

Examples of named objects:

- System variables
- User-defined variables
- Axes/spindles
- Subprograms
- Keywords
- Jump markers
- Macros

Note

Identifiers must be unique. It is **not** permissible to use the same identifier for different objects.

Naming rules

A name can be chosen freely providing the following rules are observed:

- Permissible characters:
  - Letters: A ... Z, a ... z
  - Numbers: 0 ... 9
  - Underscore: _
- The first two characters should be letters or underscores.
- Maximum length:
  - Program names (Page 38): 24 characters
  - Axis names: 8 characters
  - Variable names: 31 characters

Note

Reserved keywords must not be used as identifiers.

Cycles

To prevent name conflicts, we recommend that the following specification for the assignment of names for user cycles is observed:

Character string	Reserved for names for
• CYCLE • CUST_ • GROUP_ • _ • S_ • E_ • F_	SIEMENS cycles
• CCS_	SIEMENS compile cycles
• CC_	User compile cycles

User cycles

We recommend that the names of user cycles begin with U_.

Variables

A detailed description of the name assignment for variables appears in:

Programming Manual Job Planning

- **System variables**
  "Flexible NC programming" > "Variables" > "System variable" section
- **User variables**
  "Flexible NC programming" > "Variables" > "Definition of user variables (DEF)" section

16.6 Constants

Constant (general)

A constant is a data element whose value does not change during the execution of a program, e.g. a value assignment to an address.

Decimal constant

The numeric value of a decimal constant is displayed in the decimal system.

INTEGER constant

An INTEGER constant is an integer value, i.e. a sequence of digits without decimal point, with or without sign.

Examples:

`X10`	Assignment of the value +10 to address X
`X-35`	Assignment of the value -35 to address X
`X0`	Assignment of the value 0 to address X **Note:** X0 cannot be replaced by X.

REAL constant

A REAL constant is a sequence of digits with decimal point, with or without sign and with or without exponent.

Examples:

`X10.25`	Assignment of the value +10.25 to address X
`X-10.25`	Assignment of the value -10.25 to address X
`X0.25`	Assignment of the value +0.25 to address X
`X.25`	Assignment of the value +0.25 to address X without leading "0"
`X=-.1EX-3`	Assignment of the value $-0.1*10^{-3}$ to address X

Note

If, in an address, which permits decimal point input, more decimal places are specified than actually provided for the address, then they are rounded to fit the number of places provided.

Hexadecimal constant

Constants can also be interpreted as hexadecimal format, i.e. based on 16. The letters A to F are hexadecimal digits with the decimal values 10 to 15.

Hexadecimal constants are enclosed in single quotation marks and start with the letter "H", followed by the value in hexadecimal notation. Separators are permitted between the letters and digits.

Example:

Program code	Comment
`$MC_TOOL_MANAGEMENT_MASK='H7F'`	; By assigning the hexadecimal constant, bits 0 to 7 are set in the machine data.

Note

The maximum number of characters is limited by the value range of the integer data type.

Binary constant

Constants can also be interpreted in binary format. In this case, only the digits "0" and "1" are used.

Binary constants are enclosed in single quotation marks and start with the letter "B", followed by the binary value. Separators are permitted between the digits.

Example:

Program code	Comment
`$MN_AUXFU_GROUP_SPEC='B10000001'`	; By assigning the binary constant, bit 0 and bit 7 are set in the machine data.

Note

The maximum number of characters is limited by the value range of the integer data type.

16.7 Operators and arithmetic functions

Operators

Arithmetic operators

System variables of the REAL and INT type can be linked by the following operators:

Operator	Meaning
+	Addition
-	Subtraction
*	Multiplication
/	• Division in synchronized actions: INT / INT ⇒ **INT** • Division in synchronized actions with REAL result by using the function ITOR(): ITOR(INT) / ITOR(INT) ⇒ **REAL** • Division in NC programs: INT / INT ⇒ **REAL**
DIV	Integer division: INT / INT ⇒ **INT**
MOD	Modulo division (only for type INT) supplies remainder of an INT division Example: 3 MOD 4 = 3

Note

Only variables of the same type may be linked by these operations.

Relational operators

Operator	Meaning
==	Equal to
>	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

Boolean operators

Operator	Meaning
NOT	NOT
AND	AND
OR	OR
XOR	Exclusive OR

Bit logic operators

Operator	Meaning
B_OR	Bit-by-bit OR
B_AND	Bit-by-bit AND
B_XOR	Bit-by-bit exclusive OR
B_NOT	Bit-by-bit negation

Priority of the operators

The operators have the following priorities for execution in the synchronized action (highest priority: 1):

Prio.	Operators	Meaning
1	NOT, B_NOT	Negation, bit-by-bit negation
2	*, /, DIV, MOD	Multiplication, division
3	+, -	Addition, subtraction
4	B_AND	Bit-by-bit AND
5	B_XOR	Bit-by-bit exclusive OR
6	B_OR	Bit-by-bit OR
7	AND	AND
8	XOR	Exclusive OR
9	OR	OR
10	<<	Concatenation of strings, result type STRING
11	==, <>, <, >, >=, <=	Relational operators

Note

It is strongly recommended that the individual operators are clearly prioritized by setting parentheses "(…)" when several operators are used in an expression.

Example of a condition with an expression with several operators:

Program code
`... WHEN ($AA_IM[X] > VALUE) AND ($AA_IM[Y] > VALUE1) DO ...`

Arithmetic functions

Operator	Meaning
SIN()	Sine
COS()	Cosine
TAN()	Tangent
ASIN()	Arc sine
ACOS()	Arc cosine
ATAN2()	Arc tangent 2
SQRT()	Square root

Operator	Meaning
ABS()	Absolute value
POT()	2nd power (square)
TRUNC()	Integer component The accuracy for comparison commands can be set using TRUNC
ROUND()	Round to an integer
LN()	Natural logarithm
EXP()	Exponential function

A detailed description of the functions can be found in:
References
Programming Manual, Job Planning; Section "Flexible NC programming" ff.

Indexing

The index of a system variable of type "Array of …" can in turn be a system variable. The index is also evaluated in the main run in the interpolator clock cycle.

Example

Program code
`... WHEN … DO $AC_PARAM[$AC_MARKER[1]]=3`

Restrictions

- It is not permissible to nest indices with further system variables.
- The index must not be formed via preprocessing variables. The following example is therefore **not** permitted since $P_EP is a preprocessing variable:
  $AC_PARAM[1] = $P_EP[$AC_MARKER[0]]

Tables 17

17.1 Operations

Note

Cycles

The list of operations contains all cycles, which occur in the NC program (G code), i.e. can be programmed in the program editor using masks - or must be programmed for loops without programming support. Cycles, which for reasons of compatibility, are still available in the control, however can no longer be edited using the SINUMERIK Operate program editor ("compatibility cycles") are not taken into account.

Operations A ... C

Operation	Type [1]	Meaning	W [2]	TP [3]	SA [4]	Description see [5]
[1] [2] [3] [4] [5] for explanations, see legend (Page 450).						
:	O	NC main block number, jump label termination, concatenation operator		+		PGAsl
*	O	Operator for multiplication		+		PGAsl
+	O	Operator for addition		+		PGAsl
-	O	Operator for subtraction		+		PGAsl
<	O	Comparison operator, less than		+		PGAsl
<<	O	Concatenation operator for strings		+		PGAsl
<=	O	Comparison operator, less than or equal to		+		PGAsl
=	O	Assignment operator		+		PGAsl
>=	O	Comparison operator, greater than or equal to		+		PGAsl
/	O	Operator for division		+		PGAsl
/0 … … /7		block is skipped (1st skip level) ... ... block is skipped (8th skip level)		+		PGsl
A	A	Axis name	m/s	+		PGAsl
A2	A	Tool orientation: RPY or Euler angle	s	+		PGAsl
A3	A	Tool orientation: 1st component of the direction vector	s	+		PGAsl
A4	A	Tool orientation: 1st component of the surface normal vector at start of block	s	+		PGAsl
A5	A	Tool orientation: 1st component of the surface normal vector at end of block	s	+		PGAsl
A6	A	Tool orientation: 1st component of the direction vector for taper's axis of rotation	s	+		PGAsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
A7	A	Tool orientation: 1st Vector component for intermediate orientation on peripheral surface of taper	s	+		PGAsl
ABS	F	Absolute value (amount)		+	+	PGAsl
AC	K	Absolute dimensions of coordinates/positions	s	+		PGsl
ACC	K	Effect of current axial acceleration	m	+	+	PGsl
ACCLIMA	K	Effect of current maximum axial acceleration	m	+	+	PGAsl
ACN	K	Absolute dimensions for rotary axes, approach position in negative direction	s	+		PGsl
ACOS	F	Arc cosine (trigon. function)		+	+	PGAsl
ACP	K	Absolute dimensions for rotary axes, approach position in positive direction	s	+		PGsl
ACTBLOCNO	P	Output of current block number of an alarm block, even if "current block display suppressed" (DISPLOF) is active!		+		PGAsl
ADDFRAME	F	Inclusion and possible activation of a measured frame		+	-	PGAsl, FB1sl (K2)
ADIS	A	Rounding clearance for path functions G1, G2, G3, ...	m	+		PGsl
ADISPOS	A	Rounding clearance for rapid traverse G0	m	+		PGsl
ADISPOSA	P	Size of the tolerance window for IPOBRKA	m	+	+	PGAsl
ALF	A	LIFTFAST angle	m	+		PGAsl
AMIRROR	G	Programmable mirroring	s	+		PGsl
AND	K	Logical AND		+		PGAsl
ANG	A	Contour angle	s	+		PGsl
AP	A	Polar angle	m/s	+		PGsl
APR	K	Read/show access protection		+		PGAsl
APRB	K	Read access right, OPI		+		PGAsl
APRP	K	Read access right, part program		+		PGAsl
APW	K	Write access protection		+		PGAsl
APWB	K	Write access right, OPI		+		PGAsl
APWP	K	Write access right, part program		+		PGAsl
APX	K	Definition of the access right for executing the specified language element		+		PGAsl
AR	A	Opening angle	m/s	+		PGsl
AROT	G	Programmable rotation	s	+		PGsl
AROTS	G	Programmable frame rotations with solid angles	s	+		PGsl
AS	K	Macro definition		+		PGAsl
ASCALE	G	Programmable scaling	s	+		PGsl
ASIN	F	Arithmetic function, arc sine		+	+	PGAsl
ASPLINE	G	Akima spline	m	+		PGAsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
ATAN2	F	Arc tangent 2		+	+	PGAsl
ATOL	A	Axis-specific tolerance for compressor functions, orientation smoothing and smoothing types	m	+		PGAsl
ATRANS	G	Additive programmable work offset	s	+		PGsl
AUXFUDEL	P	Delete auxiliary function channel-specifically from the global list		+	-	FB1sl (H2)
AUXFUDELG	P	Delete all auxiliary functions of an auxiliary function group channel-specifically from the global list		+	-	FB1sl (H2)
AUXFUMSEQ	P	Determine output sequence of M auxiliary functions		+	-	FB1sl (H2)
AUXFUSYNC	P	Generate a complete part program block for the channel-specific SERUPRO end ASUB as string from the global list of auxiliary functions		+	-	FB1sl (H2)
AX	K	Variable axis identifier	m/s	+		PGAsl
AXCTSWE	P	Rotate axis container		+	-	PGAsl
AXCTSWEC	P	Canceling enable for axis container rotation		+	+	PGAsl
AXCTSWED	P	Rotating axis container (command variant for commissioning!)		+	-	PGAsl
AXIS	K	Axis identifier, axis address		+		PGAsl
AXNAME	F	Converts input string into axis identifier		+	-	PGAsl
AXSTRING	F	Converts string spindle number		+	-	PGAsl
AXTOCHAN	P	Request axis for a specific channel. Possible from NC program and synchronized action.		+	+	PGAsl
AXTOSPI	F	Converts axis identifier into a spindle index		+	-	PGAsl
B	A	Axis name	m/s	+		PGAsl
B2	A	Tool orientation: RPY or Euler angle	s	+		PGAsl
B3	A	Tool orientation: 2nd component of the direction vector	s	+		PGAsl
B4	A	Tool orientation: 2nd component of the surface normal vector at start of block	s	+		PGAsl
B5	A	Tool orientation: 2nd component of the surface normal vector at end of block	s	+		PGAsl
B6	A	Tool orientation: 2nd component of the direction vector for taper's axis of rotation	s	+		PGAsl
B7	A	Tool orientation: 2nd Vector component for intermediate orientation on peripheral surface of taper	s	+		PGAsl
B_AND	O	Bit-by-bit AND		+		PGAsl
B_OR	O	Bit-by-bit OR		+		PGAsl
B_NOT	O	Bit-by-bit negation		+		PGAsl
B_XOR	O	Bit-by-bit exclusive OR		+		PGAsl
BAUTO	G	Definition of the first spline section by means of the next 3 points	m	+		PGAsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) **for explanations, see legend (Page 450).**						
BLOCK	K	Together with the keyword TO defines the program part to be processed in an indirect subprogram call		+		PGAsl
BLSYNC	K	Processing of interrupt routine is only to start with the next block change		+		PGAsl
BNAT 6)	G	Natural transition to first spline block	m	+		PGAsl
BOOL	K	Data type: Boolean value TRUE/FALSE or 1/0		+		PGAsl
BOUND	F	Tests whether the value falls within the defined value range. If the values are equal, the test value is returned.		+	+	PGAsl
BRISK 6)	G	Fast non-smoothed path acceleration	m	+		PGAsl
BRISKA	P	Switch on brisk path acceleration for the programmed axes		+	-	PGAsl
BSPLINE	G	B spline	m	+		PGAsl
BTAN	G	Tangential transition to first spline block	m	+		PGAsl
C	A	Axis name	m/s	+		PGAsl
C2	A	Tool orientation: RPY or Euler angle	s	+		PGAsl
C3	A	Tool orientation: 3rd component of the direction vector	s	+		PGAsl
C4	A	Tool orientation: 3rd component of the surface normal vector at start of block	s	+		PGAsl
C5	A	Tool orientation: 3rd component of the surface normal vector at end of block	s	+		PGAsl
C6	A	Tool orientation: 3rd component of the direction vector for taper's axis of rotation	s	+		PGAsl
C7	A	Tool orientation: 3rd Vector component for intermediate orientation on peripheral surface of taper	s	+		PGAsl
CAC	K	Absolute position approach		+		PGAsl
CACN	K	Absolute approach of the value listed in the table in negative direction		+		PGAsl
CACP	K	Absolute approach of the value listed in the table in positive direction		+		PGAsl
CALCDAT	F	Calculates radius and center point of circle from 3 or 4 points		+	-	PGAsl
CALCPOSI	F	Checking for protection area violation, working area limitation and software limits		+	-	PGAsl
CALL	K	Indirect subprogram call		+		PGAsl
CALLPATH	P	Programmable search path for subprogram calls		+	-	PGAsl
CANCEL	P	Cancel modal synchronized action		+	-	FBSYsl
CASE	K	Conditional program branch		+		PGAsl
CDC	K	Direct approach of a position		+		PGAsl
CDOF 6)	G	Switch off collision monitoring	m	+		PGsl

Operation	Type [1]	Meaning	W [2]	TP [3]	SA [4]	Description see [5]
[1] [2] [3] [4] [5] for explanations, see legend (Page 450).						
CDOF2	G	Switch off collision monitoring during 3D circumferential milling	m	+		PGsl
CDON	G	Activate collision monitoring	m	+		PGsl
CFC [6]	G	Constant feedrate on contour	m	+		PGsl
CFIN	G	Constant feedrate for internal radius only, not for external radius	m	+		PGsl
CFINE	F	Assignment of fine offset to a FRAME variable		+	-	PGAsl
CFTCP	G	Constant feedrate in tool center point (center point path)	m	+		PGsl
CHAN	K	Specify validity range for data		+		PGAsl
CHANDATA	P	Set channel number for channel data access		+	-	PGAsl
CHAR	K	Data type: ASCII character		+		PGAsl
CHF	A	Chamfer; value = length of chamfer	s	+		PGsl
CHKDM	F	Uniqueness check within a magazine		+	-	FBWsl
CHKDNO	F	Check for unique D numbers		+	-	PGAsl
CHR	A	Chamfer; value = length of chamfer in direction of movement		+		PGsl
CIC	K	Approach position by increments		+		PGAsl
CIP	G	Circular interpolation through intermediate point	m	+		PGsl
CLEARM	P	Reset one/several markers for channel coordination		+	+	PGAsl
CLRINT	P	Deselect interrupt		+	-	PGAsl
CMIRROR	F	Mirror on a coordinate axis		+	-	PGAsl
COARSEA	K	Motion end when "Exact stop coarse" reached	m	+		PGAsl
COLLPAIR	F	Check whether part of a collision pair		+		PGAsl
COMPCAD	G	Activate the compressor function COMPCAD	m	+		PGAsl
COMPCURV	G	Activate the compressor function COMPCURV	m	+		PGAsl
COMPLETE		Control instruction for reading and writing data		+		PGAsl
COMPOF [6]	G	Deactivate NC block compression	m	+		PGAsl
COMPON	G	Activate the compressor function COMPON	m	+		PGAsl
COMPSURF	G	Activate the compressor function COMPSURF	m	+		PGAsl
CONTDCON	P	Activate tabular contour decoding		+	-	PGAsl
CONTPRON	P	Activate reference preprocessing		+	-	PGAsl
CORROF	P	All active overlaid movements are deselected.		+	-	PGsl
CORRTRAFO	F	Modifying offset vectors or direction vectors for the orientation axes in the kinematic model of the machine		+	-	PGAsl
COS	F	Cosine (trigon. function)		+	+	PGAsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) **for explanations, see legend (Page 450).**						
COUPDEF	P	Definition ELG group/synchronous spindle group		+	-	PGAsl
COUPDEL	P	Delete ELG group		+	-	PGAsl
COUPOF	P	Deactivate ELG group / synchronous spindle pair		+	-	PGAsl
COUPOFS	P	Deactivate ELG group/synchronous spindle pair with stop of following spindle		+	-	PGAsl
COUPON	P	Activate ELG group / synchronous spindle pair		+	-	PGAsl
COUPONC	P	Transfer activation of ELG group/synchronous spindle pair with previous programming		+	-	PGAsl
COUPRES	P	Reset ELG group		+	-	PGAsl
CP 6)	G	Path motion	m	+		PGAsl
CPBC	K	Generic coupling: Block change criterion		+	+	FB3sl (M3)
CPDEF	K	Generic coupling: Creating a coupling module		+	+	FB3sl (M3)
CPDEL	K	Generic coupling: Deletion of a coupling module		+	+	FB3sl (M3)
CPFMOF	K	Generic coupling: Behavior of the following axis at complete switch-off		+	+	FB3sl (M3)
CPFMON	K	Generic coupling: Behavior of the following axis when switching on		+	+	FB3sl (M3)
CPFMSON	K	Generic coupling: Synchronization mode		+	+	FB3sl (M3)
CPFPOS	K	Generic coupling: Synchronized position of the following axis		+	+	FB3sl (M3)
CPFRS	K	Generic coupling: Coordinate reference system		+	+	FB3sl (M3)
CPLA	K	Generic coupling: Definition of a leading axis		+	-	FB3sl (M3)
CPLCTID	K	Generic coupling: Number of the curve table		+	+	FB3sl (M3)
CPLDEF	K	Generic coupling: Definition of a leading axis and creation of a coupling module		+	+	FB3sl (M3)
CPLDEL	K	Generic coupling: Deleting a leading axis of a coupling module		+	+	FB3sl (M3)
CPLDEN	K	Generic coupling: Denominator of the coupling factor		+	+	FB3sl (M3)
CPLINSC	K	Generic coupling: Scaling factor of the input value of a leading axis		+	+	FB3sl (M3)
CPLINTR	K	Generic coupling: Offset value of the input value of a leading axis		+	+	FB3sl (M3)
CPLNUM	K	Generic coupling: Numerator of the coupling factor		+	+	FB3sl (M3)
CPLOF	K	Generic coupling: Switching off a leading axis of a coupling module		+	+	FB3sl (M3)
CPLON	K	Generic coupling: Switching on a leading axis of a coupling module		+	+	FB3sl (M3)
CPLOUTSC	K	Generic coupling: Scaling factor for the output value of a coupling		+	+	FB3sl (M3)

Operation	Type [1]	Meaning	W [2]	TP [3]	SA [4]	Description see [5]
[1) 2) 3) 4) 5)] **for explanations, see legend (Page 450).**						
CPLOUTTR	K	Generic coupling: Offset value for the output value of a coupling		+	+	FB3sl (M3)
CPLPOS	K	Generic coupling: Synchronized position of the leading axis		+	+	FB3sl (M3)
CPLSETVAL	K	Generic coupling: Coupling reference		+	+	FB3sl (M3)
CPMALARM	K	Generic coupling: Suppression of special coupling-related alarm outputs		+	+	FB3sl (M3)
CPMBRAKE	K	Generic coupling: Response of the following axis to certain stop signals and stop commands		+	-	FB3sl (M3)
CPMPRT	K	Generic coupling: Coupling response at part program start under block search run via program test		+	+	FB3sl (M3)
CPMRESET	K	Generic coupling: Coupling behavior for RESET		+	+	FB3sl (M3)
CPMSTART	K	Generic coupling: Coupling behavior at part program start		+	+	FB3sl (M3)
CPMVDI	K	Generic coupling: Response of the following axis to certain NC/PLC interface signals		+	+	FB3sl (M3)
CPOF	K	Generic coupling: Switching off a coupling module		+	+	FB3sl (M3)
CPON	K	Generic coupling: Switching on a coupling module		+	+	FB3sl (M3)
CPRECOF [6]	G	Deactivate programmable contour accuracy	m	+		PGAsl
CPRECON	G	Activate programmable contour accuracy	m	+		PGAsl
CPRES	K	Generic coupling: Activates the configured data of the synchronous spindle coupling		+	-	FB3sl (M3)
CPROT	P	Activate / deactivate channel-specific protection zone		+	-	PGAsl
CPROTDEF	P	Definition of a channel-specific protection area		+	-	PGAsl
CPSETTYPE	K	Generic coupling: Coupling type		+	+	FB3sl (M3)
CPSYNCOP	K	Generic coupling: Threshold value of position synchronism "Coarse"		+	+	FB3sl (M3)
CPSYNCOP2	K	Generic coupling: Threshold value of position synchronism "Coarse" 2		+	+	FB3sl (M3)
CPSYNCOV	K	Generic coupling: Threshold value of velocity synchronism "Coarse"		+	+	FB3sl (M3)
CPSYNFIP	K	Generic coupling: Threshold value of position synchronism "Fine"		+	+	FB3sl (M3)
CPSYNFIP2	K	Generic coupling: Threshold value of position synchronism "Fine" 2		+	+	FB3sl (M3)
CPSYNFIV	K	Generic coupling: Threshold value of velocity synchronism "Fine"		+	+	FB3sl (M3)
CR	A	Circle radius	s	+		PGsl
CROT	F	Rotation of the current coordinate system		+	-	PGAsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
CROTS	F	Programmable frame rotations with solid angles (rotation in the specified axes)	s	+	-	PGsl
CRPL	F	Frame rotation in any plane		+	-	FB1sl (K2)
CSCALE	F	Scale factor for multiple axes		+	-	PGAsl
CSPLINE	F	Cubic spline	m	+		PGAsl
CT	G	Circle with tangential transition	m	+		PGsl
CTAB	F	Define following axis position according to leading axis position from curve table		+	+	PGAsl
CTABDEF	P	Activate table definition		+	-	PGAsl
CTABDEL	P	Clear curve table		+	-	PGAsl
CTABEND	P	Deactivate table definition		+	-	PGAsl
CTABEXISTS	F	Checks the curve table with number n		+	+	PGAsl
CTABFNO	F	Number of curve tables still possible in the memory		+	+	PGAsl
CTABFPOL	F	Number of polynomials still possible in the memory		+	+	PGAsl
CTABFSEG	F	Number of curve segments still possible in the memory		+	+	PGAsl
CTABID	F	Returns table number of the nth curve table		+	+	PGAsl
CTABINV	F	Define leading axis position according to following axis position from curve table		+	+	PGAsl
CTABISLOCK	F	Returns the lock state of the curve table with number n		+	+	PGAsl
CTABLOCK	P	Delete and overwrite, lock		+	+	PGAsl
CTABMEMTYP	F	Returns the memory in which curve table number n is created.		+	+	PGAsl
CTABMPOL	F	Max. number of polynomials still possible in the memory		+	+	PGAsl
CTABMSEG	F	Max. number of curve segments still possible in the memory		+	+	PGAsl
CTABNO	F	Number of defined curve tables in SRAM or DRAM		+	+	FB3sl (M3)
CTABNOMEM	F	Number of defined curve tables in SRAM or DRAM		+	+	PGAsl
CTABPERIOD	F	Returns the table periodicity of curve table number n		+	+	PGAsl
CTABPOL	F	Number of polynomials already used in the memory		+	+	PGAsl
CTABPOLID	F	Number of the curve polynomials used by the curve table with number n		+	+	PGAsl
CTABSEG	F	Number of curve segments already used in the memory		+	+	PGAsl
CTABSEGID	F	Number of the curve segments used by the curve table with number n		+	+	PGAsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) **for explanations, see legend (Page 450).**						
CTABSEV	F	Returns the final value of the following axis of a segment of the curve table		+	+	PGAsl
CTABSSV	F	Returns the initial value of the following axis of a segment of the curve table		+	+	PGAsl
CTABTEP	F	Returns the value of the leading axis at curve table end		+	+	PGAsl
CTABTEV	F	Returns the value of the the following axis at curve table end		+	+	PGAsl
CTABTMAX	F	Returns the maximum value of the following axis of the curve table		+	+	PGAsl
CTABTMIN	F	Returns the minimum value of the following axis of the curve table		+	+	PGAsl
CTABTSP	F	Returns the value of the leading axis at curve table start		+	+	PGAsl
CTABTSV	F	Returns the value of the following axis at curve table start		+	+	PGAsl
CTABUNLOCK	P	Revoke delete and overwrite lock		+	+	PGAsl
CTOL	A	Contour tolerance for compressor functions, orientation smoothing and smoothing types	m	+		PGAsl
CTRANS	F	Zero offset for multiple axes		+	-	PGAsl
CUT2D 6)	G	2D TRC	m	+		PGsl
CUT2DD	G	2½ D TRC in relation to the differential tool	m	+		PGsl
CUT2DF	G	2 D TRC relative to the current frame (inclined plane)	m	+		PGsl
CUT2DFD	G	2½ D TRC in relation to the differential tool relative to the current frame (inclined plane)	m	+		PGsl
CUT3DC	G	3D TRC for circumferential milling	m	+		PGAsl
CUT3DCC	G	3D TRC for circumferential milling taking into account a limitation surface with 3D radius compensation Contour on the machining surface	m	+		PGAsl
CUT3DCCD	G	3D TRC for circumferential milling taking into account a limitation surface with differential tool on the tool center point path Infeed to the limitation surface.	m	+		PGAsl
CUT3DCD	G	3D TRC in relation to a differential tool for circumferential milling	m	+		PGAsl
CUT3DF	G	3D TRC for face milling with change in orientation	m	+		PGAsl
CUT3DFD	G	3D TRC in relation to a differential tool for face milling with change in orientation	m	+		PGAsl
CUT3DFF	G	3D TRC for face milling with constant orientation. The tool orientation is the direction defined by G17 - G19 and, in some cases, rotated by a frame.	m	+		PGAsl

Operation	Type [1]	Meaning	W [2]	TP [3]	SA [4]	Description see [5]
[1] [2] [3] [4] [5] **for explanations, see legend (Page 450).**						
CUT3DFS	G	3D TRC for face milling with constant orientation. The tool orientation is defined by G17 - G19 and is not influenced by frames.	m	+		PGAsl
CUTCONOF [6]	G	Deactivate tool radius compensation	m	+		PGsl
CUTCONON	G	Activate tool radius compensation	m	+		PGsl
CUTMOD	A	Activate Modification of the offset data for rotatable tools (in connection with orientable tool carriers)	m	+		PGAsl
CUTMODK	A	Activate Modification of the offset data for rotatable tools (in connection with orientation transformations defined by kinematic chains)	m	+		PGAsl
CYCLE60	C (T)	Engraving cycle		+		PGAsl
CYCLE61	C (T)	Face milling		+		PGAsl
CYCLE62	C (T)	Contour call		+		PGAsl
CYCLE63	C (T)	Contour pocket milling		+		PGAsl
CYCLE64	C (T)	Contour pocket predrilling		+		PGAsl
CYCLE70	C (T)	Thread milling		+		PGAsl
CYCLE72	C (T)	Path milling		+		PGAsl
CYCLE76	C (T)	Milling the rectangular spigot		+		PGAsl
CYCLE77	C (T)	Circular spigot milling		+		PGAsl
CYCLE78	C (T)	Mill cutting thread		+		PGAsl
CYCLE79	C (T)	Multiple edge		+		PGAsl
CYCLE81	C (T)	Drilling, centering		+		PGAsl
CYCLE82	C (T)	Drilling, counterboring		+		PGAsl
CYCLE83	C (T)	Deep-hole drilling		+		PGAsl
CYCLE84	C (T)	Tapping without compensating chuck		+		PGAsl
CYCLE85	C (T)	Reaming		+		PGAsl
CYCLE86	C (T)	Boring		+		PGAsl
CYCLE92	C (T)	Parting		+		PGAsl
CYCLE95	C (T)	Stock removal along the contour		+		PGAsl
CYCLE98	C (T)	Thread chain		+		PGAsl
CYCLE99	C (T)	Thread cutting		+		PGAsl
CYCLE150	C (M)	Displaying/logging measurement results		+		BNMsl
CYCLE435	C (T)	Calculate dressing tool position		+		PGAsl
CYCLE495	C (T)	Form-truing		+		PGAsl
CYCLE750	C (A)	Internal operating cycle for CYCLE751... CYCLE759 (contains the MMC command for the actual function call)		-		FB3sl (T4)
CYCLE751	C (A)	Open / perform / close an optimization session		M		FB3sl (T4)
CYCLE752	C (A)	Add axis to an optimization session		M		FB3sl(T4)
CYCLE753	C (A)	Select optimization mode		M		FB3sl (T4)
CYCLE754	C (A)	Add / remove language block		M		FB3sl (T4)
CYCLE755	C (A)	Backing up/restoring data block		M		FB3sl (T4)

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
CYCLE756	C (A)	Activate optimization results		M		FB3sl (T4)
CYCLE757	C (A)	Store optimization data		M		FB3sl (T4)
CYCLE758	C (A)	Changing the parameter value		M		FB3sl (T4)
CYCLE759	C (A)	Read parameter value		M		FB3sl (T4)
CYCLE800	C (T)	Swiveling		+		PGAsl
CYCLE801	C (T)	Grid or frame		+		PGAsl
CYCLE802	C (T)	Arbitrary positions		+		PGAsl
CYCLE830	C (T)	Deep-hole drilling 2		+		PGAsl
CYCLE832	C (T)	High-Speed Settings		+		PGAsl
CYCLE840	C (T)	Tapping with compensating chuck		+		PGAsl
CYCLE899	C (T)	Open slot milling		+		PGAsl
CYCLE930	C (T)	Groove		+		PGAsl
CYCLE940	C (T)	Undercut forms		+		PGAsl
CYCLE951	C (T)	Stock removal		+		PGAsl
CYCLE952	C (T)	Contour grooving		+		PGAsl
CYCLE961	C (M)	Determine the position of a workpiece corner (inner or outer) and insert as work offset.		+		BNMsl
CYCLE971	C (M)	Calibrate tool probe, measure tool length and/ or tool radius (only for milling)		+		BNMsl
CYCLE973	C (M)	Calibrate a workpiece probe on a surface on the workpiece or in a groove (only for turning)		+		BNMsl
CYCLE974	C (M)	Determine the workpiece zero in the selected measuring axis, determine tool offset with 1-point measurement (only for turning).		+		BNMsl
CYCLE976	C (M)	Calibrate a workpiece probe in a calibration ring or on a calibration ball completely in the working plane or at an edge for a particular axis and direction		+		BNMsl
CYCLE977	C (M)	Determine the center in the plane as well as the width or the diameter		+		BNMsl
CYCLE978	C (M)	Measure the position of an edge in the work-piece coordinate system		+		BNMsl
CYCLE979	C (M)	Determine center in the plane, measure radi-us of circle segment.		+		BNMsl
CYCLE982	C (M)	Calibrate tool probe, measure turning drilling and milling tools (only for turning)		+		BNMsl
CYCLE994	C (M)	Determine the workpiece zero in the selected measuring axis with 2-point measurement (on-ly for turning).		+		BNMsl
CYCLE995	C (M)	Measure the angularity of the spindle on a ma-chine tool		+		BNMsl
CYCLE996	C (M)	Determine transformation-relevant data for kinematic transformations with rotary axes		+		BNMsl
CYCLE997	C (M)	Determine center and diameter of a ball, measure center of three distributed balls		+		BNMsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
CYCLE998	C (M)	Determine the angular position of a surface (plane) referred to the working plane, determine angle of edges in the workpiece coordinate system.		+		BNMsl
CYCLE4071	C (T)	Longitudinal grinding with infeed at the reversal point		+		PGAsl
CYCLE4072	C (T)	Longitudinal grinding with infeed at the reversal point and cancel signal		+		PGAsl
CYCLE4073	C (T)	Longitudinal grinding with continuous infeed		+		PGAsl
CYCLE4074	C (T)	Longitudinal grinding with continuous infeed and cancel signal		+		PGAsl
CYCLE4075	C (T)	Surface grinding with infeed at the reversal point		+		PGAsl
CYCLE4077	C (T)	Surface grinding with infeed at the reversal point and cancel signal		+		PGAsl
CYCLE4078	C (T)	Surface grinding with continuous infeed		+		PGAsl
CYCLE4079	C (T)	Surface grinding with intermittent infeed		+		PGAsl

Operations D ... F

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1)2)3)4)5) for explanations, see legend (Page 450).						
D	O	Tool offset number		+		PGsl
D0	O	With D0, offsets for the tool are ineffective		+		PGsl
DAC	K	Absolute non-modal axis-specific diameter programming	s	+		PGsl
DC	K	Absolute dimensions for rotary axes, approach position directly	s	+		PGsl
DCI	K	Assign data class I (= Individual) (only SINUMERIK 828D)		+		PGAsl
DCM	K	Assign data class M (= Manufacturer) (only SINUMERIK 828D)		+		PGAsl
DCU	K	Assign data class U (= User) (only SINUMERIK 828D)		+		PGAsl
DEF	K	Variable definition		+		PGAsl
DEFAULT	K	Branch in CASE branch		+		PGAsl
DEFINE	K	Keyword for macro definitions		+		PGAsl
DELAYFSTOF	P	Define the end of a stop delay section	m	+	-	PGAsl
DELAYFSTON	P	Define the start of a stop delay section	m	+	-	PGAsl
DELDL	F	Delete additive offsets		+	-	PGAsl
DELDTG	P	Delete distance-to-go		-	+	FBSYsl
DELETE	P	Delete the specified file. The file name can be specified with path and file identifier.		+	-	PGAsl

Operation	Type [1]	Meaning	W [2]	TP [3]	SA [4]	Description see [5]
1)2)3)4)5) for explanations, see legend (Page 450).						
DELMLOWNER	F	Delete owner magazine location of the tool		+	-	FBWsl
DELMLRES	F	Delete magazine location reservation		+	-	FBWsl
DELMT	P	Delete multitool		+	-	FBWsl
DELOBJ	F	Deletion of elements from kinematic chains, protection areas, protection area elements, collision pairs and transformation data		+		PGAsl
DELT	P	Delete Tool		+	-	FBWsl
DELTC	P	Delete toolholder data record		+	-	FBWsl
DELTOOLENV	F	Delete data records describing tool environ-ments		+	-	PGAsl
DIACYCOFA	K	Axis-specific modal diameter programming: OFF in cycles	m	+		FB1sl (P1)
DIAM90	G	Diameter programming for G90, radius pro-gramming for G91	m	+		PGAsl
DIAM90A	K	Axis-specific modal diameter programming for G90 and AC, radius programming for G91 and IC	m	+		PGsl
DIAMCHAN	K	Transfer of all axes from MD axis functions to diameter programming channel status		+		PGsl
DIAMCHANA	K	Transfer of the diameter programming chan-nel status		+		PGsl
DIAMCYCOF	G	Channel-specific diameter programming: OFF in cycles	m	+		FB1sl (P1)
DIAMOF [6]	G	Diameter programming: OFF Normal position, see machine manufacturer	m	+		PGsl
DIAMOFA	K	Axis-specific modal diameter programming: OFF Normal position, see machine manufacturer	m	+		PGsl
DIAMON	G	Diameter programming: ON	m	+		PGsl
DIAMONA	K	Axis-specific modal diameter programming: ON Activation, see machine manufacturer	m	+		PGsl
DIC	K	Relative non-modal axis-specific diameter programming	s	+		PGsl
DILF	O	Retraction path (length)	m	+		PGsl
DISABLE	P	Interrupt OFF		+	-	PGAsl
DISC	O	Transition circle overshoot tool radius com-pensation	m	+		PGsl
DISCL	O	Clearance between the end point of the fast infeed motion and the machining plane		+		PGsl
DISPLOF	PA	Suppress current block display		+		PGAsl
DISPLON	PA	Revoke suppression of the current block dis-play		+		PGAsl
DISPR	O	Path differential for repositioning	s	+		PGAsl
DISR	O	Distance for repositioning	s	+		PGAsl

Operation	Type [1]	Meaning	W [2]	TP [3]	SA [4]	Description see [5]
[1)2)3)4)5)] **for explanations, see legend (Page 450).**						
DISRP	O	Distance between the retraction plane and the machining plane during smooth approach and retraction		+		PGsl
DITE	O	Thread run-out path	m	+		PGsl
DITS	O	Thread run-in path	m	+		PGsl
DIV	K	Integer division		+		PGAsl
DL	O	Select location-dependent additive tool offset (DL, total set-up offset)	m	+		PGAsl
DO	O	Keyword for synchronized action, triggers action when condition is fulfilled		-	+	FBSYsl
DRFOF	P	Deactivation of handwheel offsets (DRF)	m	+	-	PGsl
DRIVE	G	Velocity-dependent path acceleration	m	+		PGAsl
DRIVEA	P	Activate knee-shaped acceleration characteristic for the programmed axes		+	-	PGAsl
DYNFINISH	G	Dynamic response for smooth finishing	m	+		PGAsl
DYNNORM [6]	G	Standard dynamic response	m	+		PGAsl
DYNPOS	G	Dynamic response for positioning mode, tapping	m	+		PGAsl
DYNROUGH	G	Dynamic response for roughing	m	+		PGAsl
DYNSEMIFIN	G	Dynamic response for finishing	m	+		PGAsl
DZERO	P	Marks all D numbers of the TO unit as invalid		+	-	PGAsl
EAUTO	G	Definition of the last spline section by means of the last 3 points	m	+		PGAsl
EGDEF	P	Definition of an electronic gear		+	-	PGAsl
EGDEL	P	Delete coupling definition for the following axis		+	-	PGAsl
EGOFC	P	Turn off electronic gear continuously		+	-	PGAsl
EGOFS	P	Turn off electronic gear selectively		+	-	PGAsl
EGON	P	Turn on electronic gear		+	-	PGAsl
EGONSYN	P	Turn on electronic gear		+	-	PGAsl
EGONSYNE	P	Turn on electronic gear, with specification of approach mode		+	-	PGAsl
ELSE	K	Program branch, if IF condition not fulfilled		+		PGAsl
ENABLE	P	Interrupt ON		+	-	PGAsl
ENAT [6]	G	Natural transition to next traversing block	m	+		PGAsl
ENDFOR	K	End line of FOR counter loop		+		PGAsl
ENDIF	K	End line of IF branch		+		PGAsl
ENDLABEL	K	End label for part program repetitions with REPEAT		+		PGAsl, FB1sl (K1)
ENDLOOP	K	End line of endless program loop LOOP		+		PGAsl
ENDPROC	K	End line of program with start line PROC		+		
ENDWHILE	K	End line of WHILE loop		+		PGAsl
ESRR	P	Parameterizing drive-autonomous ESR retraction in the drive		+		PGAsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1)2)3)4)5) for explanations, see legend (Page 450).						
ESRS	P	Parameterizing drive-autonomous ESR shut-down in the drive		+		PGAsl
ETAN	G	Tangential transition to next traversing block at spline begin	m	+		PGAsl
EVERY	K	Execute synchronized action on transition of condition from FALSE to TRUE		-	+	FBSYsl
EX	K	Keyword for value assignment in exponential notation		+		PGAsl
EXECSTRING	P	Transfer of a string variable with the executing part program line		+	-	PGAsl
EXECTAB	P	Execute an element from a motion table		+	-	PGAsl
EXECUTE	P	Program execution ON		+	-	PGAsl
EXP	F	Exponential function ex		+	+	PGAsl
EXTCALL	O	Execute external subprogram		+	+	PGAsl
EXTCLOSE	P	Closing external device / file that was opened for writing		+	-	PGAsl
EXTERN	K	Declaration of a subprogram with parameter transfer		+		PGAsl
EXTOPEN	P	Opening external device / file for the channel for writing		+	-	PGAsl
F	O	Feedrate value (in conjunction with G4 the dwell time is also programmed with F)		+	+	PGsl
FA	K	Axial feedrate	m	+	+	PGsl
FAD	O	Infeed rate for soft approach and retraction		+		PGsl
FALSE	K	Logical constant: Incorrect		+	+	PGAsl
FB	O	Non-modal feedrate		+		PGsl
FCTDEF	P	Define polynomial function		+	-	PGAsl
FCUB	G	Feedrate variable according to cubic spline	m	+		PGAsl
FD	O	Path feedrate for handwheel override	s	+		PGsl
FDA	K	Axis feedrate for handwheel override	s	+		PGsl
FENDNORM 6)	G	Corner deceleration OFF	m	+		PGAsl
FFWOF 6)	G	Feedforward control OFF	m	+		PGAsl
FFWON	G	Feedforward control ON	m	+		PGAsl
FGREF	K	Reference radius for rotary axes or path reference factors for orientation axes (vector interpolation)	m	+		PGsl
FGROUP	P	Definition of axis/axes with path feedrate		+	-	PGsl
FI	K	Parameter for access to frame data: Fine offset		+		PGAsl
FIFOCTRL	G	Control of preprocessing buffer	m	+		PGAsl
FILEDATE	P	Returns date of most recent write access to file		+	-	PGAsl

Operation	Type [1]	Meaning	W [2]	TP [3]	SA [4]	Description see [5]
1)2)3)4)5) for explanations, see legend (Page 450).						
FILEINFO	P	Returns summary information listing FILE-DATE, FILESIZE, FILESTAT, and FILETIME		+	-	PGAsl
FILESIZE	P	Returns current file size		+	-	PGAsl
FILESTAT	P	Returns file status of rights for read, write, execute, display, delete (rwxsd)		+	-	PGAsl
FILETIME	P	Returns time of most recent write access to file		+	-	PGAsl
FINEA	K	End of motion when "Exact stop fine" reached	m	+		PGAsl
FL	K	Limit velocity for synchronized axis	m	+		PGsl
FLIN	G	Feed linear variable	m	+		PGAsl
FMA	K	Multiple feedrates axial	m	+		PGsl
FNORM [6]	G	Feedrate normal to DIN 66025	m	+		PGAsl
FOC	K	Non-modal torque/force limitation	s	-	+	FBSYsl
FOCOF	K	Switch off modal torque/force limitation	m	-	+	FBSYsl
FOCON	K	Switch on modal torque/force limitation	m	-	+	FBSYsl
FOR	K	Counter loop with fixed number of passes		+		PGAsl
FP	O	Fixed point: Number of fixed point to be approached	s	+		PGsl
FPO	K	Feedrate characteristic programmed via a polynomial		+		PGAsl
FPR	P	Rotary axis identifier		+	-	PGsl
FPRAOF	P	Deactivate revolutional feedrate		+	-	PGsl
FPRAON	P	Activate revolutional feedrate		+	-	PGsl
FRAME	K	Data type for the definition of coordinate systems		+		PGAsl
FRC	O	Feedrate for radius and chamfer	s	+		PGsl
FRCM	O	Feedrate for radius and chamfer, modal	m	+		PGsl
FROM	K	The action is executed if the condition is fulfilled once and as long as the synchronized action is active		-	+	FBSYsl
FTOC	P	Change fine tool offset		-	+	FBSYsl
FTOCOF [6]	G	Online fine tool offset OFF	m	+		PGAsl
FTOCON	G	Online fine tool offset ON	m	+		PGAsl
FXS	K	Travel to fixed stop ON	m	+	+	PGsl
FXST	K	Torque limit for travel to fixed stop	m	+	+	PGsl
FXSW	K	Monitoring window for travel to fixed stop		+	+	PGsl
FZ	K	Tooth feedrate	m	+		PGsl

Operations G ... L

Operation	Type [1]	Meaning	W [2]	TP [3]	SA [4]	Description see [5]
[1] [2] [3] [4] [5] for explanations, see legend (Page 450).						
G0	G	Linear interpolation with rapid traverse (rapid traverse motion)	m	+		PGsl
G1 [6]	G	Linear interpolation with feedrate (linear interpolation)	m	+		PGsl
G2	G	Circular interpolation clockwise	m	+		PGsl
G3	G	Circular interpolation counter-clockwise	m	+		PGsl
G4	G	Dwell time, preset	s	+		PGsl
G5	G	Oblique plunge-cut grinding	s	+		PGAsl
G7	G	Compensatory motion during oblique plunge-cut grinding	s	+		PGAsl
G9	G	Exact stop - deceleration	s	+		PGsl
G17 [6]	G	Selection of working plane X/Y	m	+		PGsl
G18	G	Selection of working plane Z/X	m	+		PGsl
G19	G	Selection of working plane Y/Z	m	+		PGsl
G25	G	Lower working area limitation	s	+		PGsl
G26	G	Upper working area limitation	s	+		PGsl
G33	G	Thread cutting with constant lead	m	+		PGsl
G34	G	Thread cutting with linear increasing lead	m	+		PGsl
G35	G	Thread cutting with linear decreasing lead	m	+		PGsl
G40 [6]	G	Tool radius compensation OFF	m	+		PGsl
G41	G	Tool radius compensation left of contour	m	+		PGsl
G42	G	Tool radius compensation right of contour	m	+		PGsl
G53	G	Suppression of current zero offset (non-modal)	s	+		PGsl
G54	G	1st settable zero offset	m	+		PGsl
G55	G	2nd settable zero offset	m	+		PGsl
G56	G	3rd settable zero offset	m	+		PGsl
G57	G	4th settable zero offset	m	+		PGsl
G58 (840D sl)	G	Absolute programmable work offset (coarse offset)	s	+		PGsl
G58 (828D)	G	5th settable zero offset	m	+		PGsl
G59 (840D sl)	G	Additive programmable work offset (fine offset)	s	+		PGsl
G59 (828D)	G	6th settable zero offset	m	+		PGsl
G60 [6]	G	Exact stop - deceleration	m	+		PGsl
G62	G	Corner deceleration at inside corners when tool radius offset is active (G41, G42)	m	+		PGAsl
G63	G	Tapping with compensating chuck	s	+		PGsl
G64	G	Continuous-path mode	m	+		PGsl
G70	G	Inch dimensions for geometric specifications (lengths)	m	+	+	PGsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
G71 6)	G	Metric dimensions for geometric specifications (lengths)	m	+	+	PGsl
G74	G	Search for reference	s	+		PGsl
G75	G	Fixed point approach	s	+		PGsl
G90 6)	G	Absolute dimensions	m/s	+		PGsl
G91	G	Incremental dimensions	m/s	+		PGsl
G93	G	Inverse-time feedrate rpm	m	+		PGsl
G94 6)	G	Linear feedrate F in mm/min or inch/min and degree/min	m	+		PGsl
G95	G	Revolutional feedrate F in mm/rev or inch/rev	m	+		PGsl
G96	G	Revolutional feedrate (as for G95) and constant cutting rate	m	+		PGsl
G97	G	Revolutional feedrate and constant spindle speed (constant cutting rate OFF)	m	+		PGsl
G110	G	Pole programming relative to the last programmed setpoint position	s	+		PGsl
G111	G	Pole programming relative to zero of current workpiece coordinate system	s	+		PGsl
G112	G	Pole programming relative to the last valid pole	s	+		PGsl
G140 6)	G	SAR approach direction defined by G41/G42	m	+		PGsl
G141	G	SAR approach direction to left of contour	m	+		PGsl
G142	G	SAR approach direction to right of contour	m	+		PGsl
G143	G	SAR approach direction tangent-dependent	m	+		PGsl
G147	G	Soft approach with straight line	s	+		PGsl
G148	G	Soft retraction with straight line	s	+		PGsl
G153	G	Suppression of current frames including basic frame	s	+		PGsl
G247	G	Soft approach with quadrant	s	+		PGsl
G248	G	Soft retraction with quadrant	s	+		PGsl
G290 6)	G	Switch over to SINUMERIK mode ON	m	+		FBWsl
G291	G	Switch over to ISO2/3 mode ON	m	+		FBWsl
G331	G	Rigid tapping, positive lead, clockwise	m	+		PGsl
G332	G	Rigid tapping, negative lead, counter-clockwise	m	+		PGsl
G335	G	Turning a convex thread in clockwise direction	m	+		PGsl
G336	G	Turning a convex thread in counter-clockwise direction	m	+		PGsl
G340 6)	G	Spatial approach block (depth and in plane at the same time (helix))	m	+		PGsl
G341	G	Initial infeed on perpendicular axis (z), then approach in plane	m	+		PGsl
G347	G	Soft approach with semicircle	s	+		PGsl
G348	G	Soft retraction with semicircle	s	+		PGsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
G450 6)	G	Transition circle	m	+		PGsl
G451	G	Intersection of equidistances	m	+		PGsl
G460 6)	G	Activation of collision detection for the approach and retraction block	m	+		PGsl
G461	G	Insertion of a circle into the TRC block	m	+		PGsl
G462	G	Insertion of a straight line into the TRC block	m	+		PGsl
G500 6)	G	Deactivation of all adjustable frames, basic frames are active	m	+		PGsl
G505 ... G599	G	5 ... 99 Settable work offset	m	+		PGsl
G601 6)	G	Block change at exact stop fine	m	+		PGsl
G602	G	Block change at exact stop coarse	m	+		PGsl
G603	G	Block change at IPO block end	m	+		PGsl
G621	G	Corner deceleration at all corners	m	+		PGAsl
G641	G	Continuous-path mode with smoothing as per distance criterion (= programmable rounding clearance)	m	+		PGsl
G642	G	Continuous-path mode with smoothing within the defined tolerances	m	+		PGsl
G643	G	Continuous-path mode with smoothing within the defined tolerances (block-internal)	m	+		PGsl
G644	G	Continuous-path mode with smoothing with maximum possible dynamic response	m	+		PGsl
G645	G	Continuous-path mode with smoothing and tangential block transitions within the defined tolerances	m	+		PGsl
G700	G	Inch dimensions for geometric and technological specifications (lengths, feedrate)	m	+	+	PGsl
G710 6)	G	Metric dimensions for geometric and technological specifications (lengths, feedrate)	m	+	+	PGsl
G810 6), ..., G819	G	G group reserved for the OEM user		+		PGAsl
G820 6), ..., G829	G	G group reserved for the OEM user		+		PGAsl
G931	G	Feedrate specified by means of traversing time, deactivate constant path velocity	m	+		
G942	G	Freeze linear feedrate and constant cutting rate or spindle speed	m	+		
G952	G	Freeze revolutional feedrate and constant cutting rate or spindle speed	m	+		
G961	G	Linear feedrate (as for G94) and constant cutting rate	m	+		PGsl
G962	G	Linear feedrate or revolutional feedrate and constant cutting rate	m	+		PGsl
G971	G	Linear feedrate and constant spindle speed (constant cutting rate OFF)	m	+		PGsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
G972	G	Linear feedrate or revolutional feedrate and constant spindle speed (constant cutting rate OFF)	m	+		PGsl
G973	G	Revolutional feedrate without spindle speed limitation and constant spindle speed (G97 without LIMS for ISO mode)	m	+		PGsl
GEOAX	P	Assign new channel axes to geometry axes 1 - 3		+	-	PGAsl
GET	P	Replace enabled axis between channels		+	+	PGAsl
GETACTT	F	Gets active tool from a group of tools with the same name		+	-	FBWsl
GETACTTD	F	Gets the T number associated with an absolute D number		+	-	PGAsl
GETD	P	Replace axis directly between channels		+	-	PGAsl
GETDNO	F	Returns the D number of a cutting edge (CE) of a tool (T)		+	-	PGAsl
GETEXET	P	Reading of the loaded T number		+	-	FBWsl
GETFREELOC	P	Find a free space in the magazine for a given tool		+	-	FBWsl
GETSELT	P	Return selected T number		+	-	FBWsl
GETT	F	Get T number for tool name		+	-	FBWsl
GETTCOR	F	Read out tool lengths and/or tool length components		+	-	PGAsl
GETTENV	F	Read T, D and DL numbers		+	-	PGAsl
GETVARAP	F	Read access rights to a system/user variable		+	-	PGAsl
GETVARDFT	F	Read default value of a system/user variable		+	-	PGAsl
GETVARLIM	F	Read limit values of a system/user variable		+	-	PGAsl
GETVARPHU	F	Read physical unit of a system/user variable		+	-	PGAsl
GETVARTYP	F	Read data type of a system/user variable		+	-	PGAsl
GFRAME0 ... GFRAME100	G	Activation of the grinding frame <n> of the data management in channel	m	+		PGsl
GOTO	K	Jump operation first forward then backward (direction initially to end of program and then to beginning of program)		+		PGAsl
GOTOB	K	Jump backward (toward the beginning of the program)		+		PGAsl
GOTOC	K	As GOTO, but suppress alarm 14080 "Jump destination not found"		+		PGAsl
GOTOF	K	Jump forward (toward the end of the program)		+		PGAsl
GOTOS	K	Jump back to beginning of program		+		PGAsl
GP	K	Keyword for the indirect programming of position attributes		+		PGAsl
GROUP_ADDEND	C (T)	End of trial cut addition		+		PGAsl
GROUP_BEGIN	C (T)	Beginning of program group		+		PGAsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) **for explanations, see legend (Page 450).**						
GROUP_END	C (T)	End of program group		+		PGAsl
GWPSOF	P	Deselect constant grinding wheel peripheral speed (GWPS)	s	+	-	PGsl
GWPSON	P	Select constant grinding wheel peripheral speed (GWPS)	s	+	-	PGsl
H...	O	Auxiliary function output to the PLC		+	+	PGsl/FB1sl (H2)
HOLES1	C (T)	Row of holes		+		PGAsl
HOLES2	C (T)	Circle of holes		+		PGAsl
I	O	Interpolation parameters	s	+		PGsl
I1	O	Intermediate point coordinate	s	+		PGsl
IC	K	Incremental dimensions	s	+		PGsl
ICYCOF	P	All blocks of a technology cycle are processed in one interpolation cycle following ICYCOF		+	+	FBSYsl
ICYCON	P	Each block of a technology cycle is processed in a separate interpolation cycle following ICY-CON		+	+	FBSYsl
ID	K	Identifier for modal synchronized actions	m	-	+	FBSYsl
IDS	K	Identifier for modal static synchronized actions		-	+	FBSYsl
IF	K	Introduction of a conditional jump in the part program/technology cycle		+	+	PGAsl
INDEX	F	Define index of character in input string		+	-	PGAsl
INICF	K	Initialization of variables for NEWCONF		+		PGAsl
INIPO	K	Initialization of variables at POWER ON		+		PGAsl
INIRE	K	Initialization of variables at reset		+		PGAsl
INIT	P	Selection of a particular NC program for execution in a particular channel		+	-	PGAsl
INITIAL		Generation of an INI file across all areas		+		PGAsl
INT	K	Data type: Integer with sign		+		PGAsl
INTERSEC	F	Calculate intersection between two contour elements		+	-	PGAsl
INVCCW	G	Trace involute, counter-clockwise	m	+		PGsl
INVCW	G	Trace involute, clockwise	m	+		PGsl
INVFRAME	F	Calculate the inverse frame from a frame		+	-	FB1sl (K2)
IP	K	Variable interpolation parameter		+		PGAsl
IPOBRKA	P	Motion criterion from braking ramp activation	m	+	+	
IPOENDA	K	End of motion when “IPO stop” reached	m	+		PGAsl
IPTRLOCK	P	Freeze start of the untraceable program section at next machine function block.	m	+	-	PGAsl
IPTRUNLOCK	P	Set end of untraceable program section at current block at time of interruption.	m	+	-	PGAsl
IR	O	Center of circle coordinate (X axis) when turning a convex thread		+		PGsl
ISAXIS	F	Check if geometry axis 1 specified as parameter		+	-	PGAsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
ISD	O	Insertion depth	m	+		PGAsl
ISFILE	F	Check whether the file exists in the NC application memory		+	-	PGAsl
ISNUMBER	F	Check whether the input string can be converted to a number		+	-	PGAsl
ISOCALL	K	Indirect call of a program programmed in an ISO language		+		PGAsl
ISVAR	F	Check whether the transfer parameter contains a variable declared in the NC		+	-	PGAsl
J	O	Interpolation parameters	s	+		PGsl
J1	O	Intermediate point coordinate	s	+		PGsl
JERKA	P	Activate acceleration response set via MD for programmed axes		+	-	
JERKLIM	K	Reduction or overshoot of maximum axial jerk	m	+		PGAsl
JERKLIMA	K	Reduction or overshoot of maximum axial jerk	m	+	+	PGAsl
JR	O	Center of circle coordinate (Y axis) when turning a convex thread		+		PGsl
K	O	Interpolation parameters	s	+		PGsl
K1	O	Intermediate point coordinate	s	+		PGsl
KONT	G	Travel around contour on tool offset	m	+		PGsl
KONTC	G	Approach/retract with continuous-curvature polynomial	m	+		PGsl
KONTT	G	Approach/retract with continuous-tangent polynomial	m	+		PGsl
KR	O	Center of circle coordinate (Z axis) when turning a convex thread		+		PGsl
L	O	Subprogram number	s	+	+	PGAsl
LEAD	O	Lead angle 1st basic tool orientation 2nd orientation polynomials	m	+		PGAsl
LEADOF	P	Axial master value coupling OFF		+	+	PGAsl
LEADON	P	Axial master value coupling on		+	+	PGAsl
LENTOAX	F	Provides information about the assignment of tool lengths L1, L2, and L3 of the active tool to the abscissa, ordinate and applicate		+	-	PGAsl
LFOF 6)	G	Fast retraction for thread cutting OFF	m	+		PGsl
LFON	G	Fast retraction for thread cutting ON	m	+		PGsl
LFPOS	G	Retraction of the axis declared with POLFMASK or POLFMLIN to the absolute axis position programmed with POLF	m	+		PGsl
LFTXT 6)	G	The plane of the retraction movement for fast retraction is determined from the path tangent and the current tool direction	m	+		PGsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
LFWP	G	The plane of the retraction movement for fast retraction is determined by the current working plane (G17/G18/G19)	m	+		PGsl
LIFTFAST	K	Fast retraction		+		PGsl
LIMS	K	Speed limitation for G96/G961 and G97	m	+		PGsl
LLI	K	Lower limit value of variables		+		PGAsl
LN	F	Natural logarithm		+	+	PGAsl
LOCK	P	Disable synchronized action with ID (stop technology cycle)		-	+	FBSYsl
LONGHOLE	C (T)	Elongated hole		+		PGAsl
LOOP	K	Introduction of an endless loop		+		PGAsl

Operations M ... R

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
M0		Programmed stop		+	+	PGsl
M1		Optional stop		+	+	PGsl
M2		End of program, main program (as M30)		+	+	PGsl
M3		CW spindle rotation		+	+	PGsl
M4		CCW spindle rotation		+	+	PGsl
M5		Spindle stop		+	+	PGsl
M6		Tool change		+	+	PGsl
M17		End of subprogram		+	+	PGsl
M19		Spindle positioning to the position entered in SD43240		+	+	PGsl
M30		End of program, main program (as M2)		+	+	PGsl
M40		Automatic gear change		+	+	PGsl
M41 ... M45		Gear stage 1 ... 5		+	+	PGsl
M70		Transition to axis mode		+	+	PGsl
MASLDEF	P	Define master/slave axis grouping		+	+	PGAsl
MASLDEL	P	Uncouple master/slave axis grouping and clear grouping definition		+	+	PGAsl
MASLOF	P	Deactivation of a temporary coupling		+	+	PGAsl
MASLOFS	P	Deactivation of a temporary coupling with automatic slave axis stop		+	+	PGAsl
MASLON	P	Activation of a temporary coupling		+	+	PGAsl
MATCH	F	Search for string in string		+	-	PGAsl
MAXVAL	F	Larger value of two variables (arithm. function)		+	+	PGAsl
MCALL	K	Modal subprogram call		+		PGAsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) **for explanations, see legend (Page 450).**						
MEAC	K	Continuous axial measurement without delete distance-to-go	s	+	+	PGAsl
MEAFRAME	F	Frame calculation from measuring points		+	-	PGAsl
MEAS	O	Measurement with deletion of distance-to-go	s	+		PGAsl
MEASA	K	Axial measurement with delete distance-to-go	s	+	+	PGAsl
MEASURE	F	Calculation method for workpiece and tool measurement		+	-	FB1sl (M5)
MEAW	O	Measurement without delete distance-to-go	s	+		PGAsl
MEAWA	K	Axial measurement without delete distance-to-go	s	+	+	PGAsl
MI	K	Access to frame data: Mirroring		+		PGAsl
MINDEX	F	Define index of character in input string		+	-	PGAsl
MINVAL	F	Smaller value of two variables (arithm. function)		+	+	PGAsl
MIRROR	G	Programmable mirroring	s	+		PGAsl
MMC	P	Call the dialog window interactively from the part program on the HMI		+	-	PGAsl
MOD	K	Modulo division		+		PGAsl
MODAXVAL	F	Determine modulo position of a modulo rotary axis		+	-	PGAsl
MOV	K	Start positioning axis		-	+	FBSYsl
MOVT	O	Specify end point of a traversing motion in the tool direction				FB1(K2)
MSG	P	Programmable messages	m	+	-	PGsl
MVTOOL	P	Language command to move tool		+	-	FBWsl
N	O	NC auxiliary block number		+		PGsl
NAMETOINT	F	Determining the system variable index		+		PGAsl
NC	K	Specify validity range for data		+		PGAsl
NEWCONF	P	Apply modified machine data (corresponds to "Activate machine data")		+	-	PGAsl
NEWMT	F	Create new multitool		+	-	FBWsl
NEWT	F	Create new tool		+	-	FBWsl
NORM 6)	G	Standard setting in starting point and end point with tool offset	m	+		PGsl
NOT	K	Logic NOT (negation)		+		PGAsl
NPROT	P	Machine-specific protection area ON/OFF		+	-	PGAsl
NPROTDEF	P	Definition of a machine-specific protection area		+	-	PGAsl
NUMBER	F	Convert input string to number		+	-	PGAsl
OEMIPO1	G	OEM interpolation 1	m	+		PGAsl
OEMIPO2	G	OEM interpolation 2	m	+		PGAsl
OF	K	Keyword in CASE branch		+		PGAsl
OFFN	O	Allowance on the programmed contour	m	+		PGsl

Operation	Type [1]	Meaning	W [2]	TP [3]	SA [4]	Description see [5]
[1) 2) 3) 4) 5)] **for explanations, see legend (Page 450).**						
OMA1	O	OEM address 1	m	+		PGAsl
OMA2	O	OEM address 2	m	+		PGAsl
OMA3	O	OEM address 3	m	+		PGAsl
OMA4	O	OEM address 4	m	+		PGAsl
OMA5	O	OEM address 5	m	+		PGAsl
OR	K	Logic operator, OR operation		+		PGAsl
ORIAXES	G	Linear interpolation of machine axes or orientation axes	m	+		PGAsl
ORIAXPOS	G	Orientation angle via virtual orientation axes with rotary axis positions	m	+		PGAsl
ORIC [6]	G	Orientation changes at outside corners are superimposed on the circle block to be inserted	m	+		PGAsl
ORICONCCW	G	Interpolation on a circular peripheral surface in CCW direction	m	+		PGAsl/FB3sl (F3)
ORICONCW	G	Interpolation on a circular peripheral surface in CW direction	m	+		PGAsl/FB3sl (F4)
ORICONIO	G	Interpolation on a circular peripheral surface with intermediate orientation setting	m	+		PGAsl/FB3sl (F4)
ORICONTO	G	Interpolation on circular peripheral surface in tangential transition (final orientation)	m	+		PGAsl/FB3sl (F5)
ORICURVE	G	Interpolation of orientation with specification of motion of two contact points of tool	m	+		PGAsl/FB3sl (F6)
ORID	G	Orientation changes are performed before the circle block	m	+		PGAsl
ORIEULER [6]	G	Orientation angle via Euler angle	m	+		PGAsl
ORIMKS	G	Tool orientation in the machine coordinate system	m	+		PGAsl
ORIPATH	G	Tool orientation in relation to path	m	+		PGAsl
ORIPATHS	G	Tool orientation in relation to path, blips in the orientation characteristic are smoothed	m	+		PGAsl
ORIPLANE	G	Interpolation in a plane (corresponds to ORIVECT), large-radius circular interpolation	m	+		PGAsl
ORIRESET	P	Initial tool orientation with up to 3 orientation axes		+	-	PGAsl
ORIROTA [6]	G	Angle of rotation to an absolute direction of rotation	m	+		PGAsl
ORIROTC	G	Tangential rotational vector in relation to path tangent	m	+		PGAsl
ORIROTR	G	Angle of rotation relative to the plane between the start and end orientation	m	+		PGAsl
ORIROTT	G	Angle of rotation relative to the change in the orientation vector	m	+		PGAsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
ORIRPY	G	Orientation angle via RPY angle (XYZ)	m	+		PGAsl
ORIRPY2	G	Orientation angle via RPY angle (ZYX)	m	+		PGAsl
ORIS	O	Change in orientation	m	+		PGAsl
ORISOF 6)	G	Smoothing of the orientation characteristic OFF	m	+		PGAsl
ORISOLH	F	Calculate orientations		+		PGAsl
ORISON	G	Smoothing of the orientation characteristic ON	m	+		PGAsl
ORIVECT 6)	G	Large-circle interpolation (identical to ORI-PLANE)	m	+		PGAsl
ORIVIRT1	G	Orientation angle via virtual orientation axes (definition 1)	m	+		PGAsl
ORIVIRT2	G	Orientation angle via virtual orientation axes (definition 1)	m	+		PGAsl
ORIWKS 6)	G	Tool orientation in the workpiece coordinate system	m	+		PGAsl
OS	K	Oscillation on/off		+		PGAsl
OSB	K	Oscillating: Starting point	m	+		FB1sl (P5)
OSC	G	Continuous tool orientation smoothing	m	+		PGAsl
OSCILL	K	Axis: 1 - 3 infeed axes	m	+		PGAsl
OSCTRL	K	Oscillation options	m	+		PGAsl
OSD	G	Smoothing of tool orientation by specifying smoothing distance with SD	m	+		PGAsl
OSE	K	Oscillation end position	m	+		PGAsl
OSNSC	K	Oscillating: Number of spark-out cycles	m	+		PGAsl
OSOF 6)	G	Tool orientation smoothing OFF	m	+		PGAsl
OSP1	K	Oscillating: Left reversal point	m	+		PGAsl
OSP2	K	Oscillation right reversal point	m	+		PGAsl
OSS	G	Tool orientation smoothing at end of block	m	+		PGAsl
OSSE	G	Tool orientation smoothing at start and end of block	m	+		PGAsl
OST	G	Smoothing of tool orientation by specifying angular tolerance in degrees with SD (maximum deviation from programmed orientation characteristic)	m	+		PGAsl
OST1	K	Oscillating: Stopping point in left reversal point	m	+		PGAsl
OST2	K	Oscillating: Stopping point in right reversal point	m	+		PGAsl
OTOL	A	Orientation tolerance for compressor functions, orientation smoothing and smoothing types	m	+		PGAsl
OVR	K	Speed offset	m	+		PGAsl
OVRA	K	Axial speed offset	m	+	+	PGAsl
OVRRAP	K	Rapid traverse override	m	+		PGAsl
P	O	Number of subprogram repetitions		+		PGAsl

Operation	Type [1)]	Meaning	W [2)]	TP [3)]	SA [4)]	Description see [5)]
[1) 2) 3) 4) 5)] **for explanations, see legend (Page 450).**						
PAROT	G	Align workpiece coordinate system on workpiece	m	+		PGsl
PAROTOF [6)]	G	Deactivate frame rotation in relation to workpiece	m	+		PGsl
PCALL	K	Call subprograms with absolute path and parameter transfer		+		PGAsl
PDELAYOF	G	Punching with delay OFF	m	+		PGAsl
PDELAYON [6)]	G	Punching with delay ON	m	+		PGAsl
PHI	K	Angle of rotation of the orientation around the direction axis of the taper		+		PGAsl
PHU	K	Physical unit of a variable		+		PGAsl
PL	O	1. B spline: Node clearance 2. Polynomial interpolation Length of the parameter interval for polynomial interpolation	s	+		PGAsl
PM	K	Per minute		+		PGsl
PO	K	Polynomial coefficient for polynomial interpolation	s	+		PGAsl
POCKET3	C (T)	Milling the rectangular pocket		+		PGAsl
POCKET4	C (T)	Milling the circular pocket		+		PGAsl
POLF	K	LIFTFAST retraction position	m	+		PGsl/PGAsl
POLFA	P	Start retraction position of single axes with $AA_ESR_TRIGGER	m	+	+	PGsl
POLFMASK	P	Enable axes for retraction without a connection between the axes	m	+	-	PGsl
POLFMLIN	P	Enable axes for retraction with a linear connection between the axes	m	+	-	PGsl
POLY	G	Polynomial interpolation	m	+		PGAsl
POLYPATH	P	Polynomial interpolation can be selected for the AXIS or VECT axis groups	m	+	-	PGAsl
PON	G	Punching ON	m	+		PGAsl
PONS	G	Punching ON in interpolation cycle	m	+		PGAsl
POS	K	Axis positioning		+	+	PGsl
POSA	K	Position axis across block boundary		+	+	PGsl
POSM	P	Position magazine		+	-	FBWsl
POSMT	P	Position multitool on toolholder at location number		+	-	FBWsl
POSP	K	Positioning axis in parts (oscillation)		+		PGsl
POSRANGE	F	Determine whether the currently interpolated position setpoint of an axis is located in a window at a predefined reference position		+	+	FBSYsl
POT	F	Square (arithmetic function)		+	+	PGAsl
PR	K	Per revolution		+		PGsl
PREPRO	PA	Identify subprograms with preparation		+		PGAsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
PRESETON	P	Actual value setting with loss of the referencing status		+	+	PGAsl
PRESETONS	P	Actual value setting with loss of the referencing status		+	+	PGAsl
PRIO	K	Keyword for setting the priority for interrupt processing		+		PGAsl
PRLOC	K	Initialization of variables at reset only after local change		+		PGAsl
PROC	K	First operation in a program		+		PGAsl
PROTA	P	Request for a recalculation of the collision model		+		PGAsl
PROTD	F	Calculating the distance between two protection areas		+		PGAsl
PROTS	P	Setting the protection area status		+		PGAsl
PSI	K	Opening angle of the taper		+		PGAsl
PTP	G	Point-to-point motion (PTP travel)	m	+		PGAsl
PTPG0	G	Point-to-point motion only with G0, otherwise path motion CP	m	+		PGAsl
PTPWOC	G	Point-to-point motion without compensation movements caused by changes in orientation	m	+		PGAsl
PUNCHACC	P	Travel-dependent acceleration for nibbling		+	-	PGAsl
PUTFTOC	P	Tool fine offset for parallel dressing		+	-	PGAsl
PUTFTOCF	P	Tool fine offset dependent on a function for parallel dressing defined with FCTDEF		+	-	PGAsl
PW	O	B spline, point weight	s	+		PGAsl
QU	K	Fast additional (auxiliary) function output		+		PGsl
R...	O	Arithmetic parameter also as settable address identifier and with numerical extension		+		PGAsl
RAC	K	Absolute non-modal axis-specific radius programming	s	+		PGsl
RDISABLE	P	Read-in disable		-	+	FBSYsl
READ	P	Reads one or more lines in the specified file and stores the information read in the array		+	-	PGAsl
REAL	K	Data type: Floating-point variable with sign (real numbers)		+		PGAsl
REDEF	K	Redefinition of system variables, user variables, and NC language commands		+		PGAsl
RELEASE	P	Release machine axes for axis exchange		+	+	PGAsl
REP	K	Keyword for initialization of all elements of an array with the same value		+		PGAsl
REPEAT	K	Repetition of a program loop		+		PGAsl
REPEATB	K	Repetition of a program line		+		PGAsl
REPOSA	G	Linear repositioning with all axes	s	+		PGAsl
REPOSH	G	Repositioning with semicircle	s	+		PGAsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
REPOSHA	G	Repositioning with all axes; geometry axes in semicircle	s	+		PGAsl
REPOSL	G	Linear repositioning	s	+		PGAsl
REPOSQ	G	Repositioning in a quadrant	s	+		PGAsl
REPOSQA	G	Linear repositioning with all axes, geometry axes in quadrant	s	+		PGAsl
RESETMON	P	Language command for setpoint activation		+	-	FBWsl
RET	P	End of subprogram		+	+	PGAsl
RETB	P	End of subprogram		+	+	PGAsl
RIC	K	Relative non-modal axis-specific radius programming	s	+		PGsl
RINDEX	F	Define index of character in input string		+	-	PGAsl
RMB	G	Repositioning to start of block	m	+		PGAsl
RMBBL	G	Repositioning to start of block	s	+		PGAsl
RME	G	Repositioning to end of block	m	+		PGAsl
RMEBL	G	Repositioning to end of block	s	+		PGAsl
RMI 6)	G	Repositioning to interrupt point	m	+		PGAsl
RMIBL 6)	G	Repositioning to interrupt point	s	+		PGAsl
RMN	G	Repositioning to the nearest path point	m	+		PGAsl
RMNBL	G	Repositioning to the nearest path point	s	+		PGAsl
RND	O	Round the contour corner	s	+		PGsl
RNDM	O	Modal rounding	m	+		PGsl
ROT	G	Programmable rotation	s	+		PGsl
ROTS	G	Programmable frame rotations with solid angles	s	+		PGsl
ROUND	F	Rounding of decimal places		+	+	PGAsl
ROUNDUP	F	Rounding up of an input value		+	+	PGAsl
RP	O	Polar radius	m/s	+		PGsl
RPL	O	Rotation in the plane	s	+		PGsl
RT	K	Parameter for access to frame data: Rotation		+		PGAsl
RTLIOF	G	G0 without linear interpolation (single-axis interpolation)	m	+		PGsl
RTLION 6)	G	G0 with linear interpolation	m	+		PGsl

Operations S ... Z

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
S	A	Spindle speed (with G4, G96/G961 different meaning)	m/s	+	+	PGsl
SAVE	PA	Attribute for saving information when subprograms are called		+		PGAsl
SBLOF	P	Suppress single block		+	-	PGAsl
SBLON	P	Revoke suppression of single block		+	-	PGAsl
SC	K	Parameter for access to frame data: Scaling		+		PGAsl
SCALE	G	Programmable scaling	s	+		PGsl
SCC	K	Selective assignment of transverse axis to G96/G961/G962. Axis identifiers may take the form of geometry, channel or machine axes.		+		PGsl
SCPARA	K	Program servo parameter set		+	+	PGAsl
SD	A	Spline degree	s	+		PGAsl
SET	K	Keyword for initialization of all elements of an array with listed values		+		PGAsl
SETAL	P	Set alarm		+	+	PGAsl
SETDNO	F	Assign the D number of a cutting edge (CE) of a tool (T)		+	-	PGAsl
SETINT	K	Define which interrupt routine is to be activated when an NC input is present		+		PGAsl
SETM	P	Setting of markers in dedicated channel		+	+	PGAsl
SETMS	P	Reset to the master spindle defined in machine data		+	-	PGsl
SETMS(n)	P	Set spindle n as master spindle		+		PGsl
SETMTH	P	Set master toolholder number		+	-	FBWsl
SETPIECE	P	Set piece number for all tools assigned to the spindle		+	-	FBWsl
SETTA	P	Activate tool from wear group		+	-	FBWsl
SETTCOR	F	Modification of tool components taking all supplementary conditions into account		+	-	PGAsl
SETTIA	P	Deactivate tool from wear group		+	-	FBWsl
SF	A	Starting point offset for thread cutting	m	+		PGsl
SIN	F	Sine (trigon. function)		+	+	PGAsl
SIRELAY	F	Activate the safety functions parameterized with SIRELIN, SIRELOUT, and SIRELTIME		-	+	FBSIsl
SIRELIN	P	Initialize input variables of function block		+	-	FBSIsl
SIRELOUT	P	Initialize output variables of function block		+	-	FBSIsl
SIRELTIME	P	Initialize timers of function block		+	-	FBSIsl
SLOT1	C (T)	Longitudinal groove		+		PGAsl
SLOT2	C (T)	Circumferential groove		+		PGAsl
SOFT	G	Soft path acceleration	m	+		PGAsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
SOFTA	P	Activate jerk-limited axis acceleration for the programmed axes		+	-	PGAsl
SON	G	Nibbling ON	m	+		PGAsl
SONS	G	Nibbling ON in interpolation cycle	m	+		PGAsl
SPATH 6)	G	Path reference for FGROUP axes is arc length	m	+		PGAsl
SPCOF	P	Switch master spindle or spindle(s) from position control to speed control	m	+	-	PGsl
SPCON	P	Switch master spindle or spindle(s) from speed control to position control	m	+	-	PGAsl
SPI	F	Converts spindle number into axis identifier		+	-	PGAsl
SPIF1 6)	G	Fast NC inputs/outputs for punching/nibbling byte 1	m	+		FB2sl (N4)
SPIF2	G	Fast NC inputs/outputs for punching/nibbling byte 2	m	+		FB2sl (N4)
SPLINEPATH	P	Define spline grouping		+	-	PGAsl
SPN	A	Number of path sections per block	s	+		PGAsl
SPOF 6)	G	Stroke OFF, nibbling, punching OFF	m	+		PGAsl
SPOS	K	Spindle position	m	+	+	PGsl
SPOSA	K	Spindle position across block boundaries	m	+		PGsl
SPP	A	Length of a path section	m	+		PGAsl
SPRINT	F	Returns an input string formatted		+		PGAsl
SQRT	F	Square root (arithmetic function)		+	+	PGAsl
SR	A	Oscillation retraction path for synchronized action	s	+		PGsl
SRA	K	Oscillation retraction path with external input axial for synchronized action	m	+		PGsl
ST	A	Oscillation sparking-out time for synchronized action	s	+		PGsl
STA	K	Oscillation sparking-out time axial for synchronized action	m	+		PGsl
START	P	Start selected programs simultaneously in several channels from current program		+	-	PGAsl
STARTFIFO 6)	G	Execute; fill preprocessing memory simultaneously	m	+		PGAsl
STAT		Position of joints	s	+		PGAsl
STOLF	K	G0 tolerance factor	m	+		PGAsl
STOPFIFO	G	Stop machining; fill preprocessing memory until STARTFIFO is detected, preprocessing memory is full or end of program	m	+		PGAsl
STOPRE	P	Preprocessing stop until all prepared blocks in the main run are executed		+	-	PGAsl
STOPREOF	P	Revoke preprocessing stop		-	+	FBSYsl
STRING	K	Data type: Character string		+		PGAsl

Operation	Type [1]	Meaning	W [2]	TP [3]	SA [4]	Description see [5]
[1) 2) 3) 4) 5)] **for explanations, see legend (Page 450).**						
STRINGIS	F	Checks the present scope of NC language and the NC cycle names, user variables, macros, and label names belonging specifically to this command to establish whether these exist, are valid, defined or active.		+	-	PGAsl
STRLEN	F	Define string length		+	-	PGAsl
SUBSTR	F	Define index of character in input string		+	-	PGAsl
SUPA	G	Suppression of current zero offset, including programmed offsets, system frames, handwheel offsets (DRF), external zero offset, and overlaid movement	s	+		PGsl
SVC	K	Tool cutting rate	m	+		PGsl
SYNFCT	P	Evaluation of a polynomial as a function of a condition in the motion-synchronous action		-	+	FBSYsl
SYNR	K	The variable is read synchronously, i.e. at the time of execution		+		PGAsl
SYNRW	K	The variable is read and written synchronously, i.e. at the time of execution		+		PGAsl
SYNW	K	The variable is written synchronously, i.e. at the time of execution		+		PGAsl
T	A	Call tool (only change if specified in machine data; otherwise M6 command necessary)		+		PGsl
TAN	F	Tangent (trigon. function)		+	+	PGAsl
TANG	P	Tangential control: Define coupling		+	-	PGAsl
TANGDEL	P	Tangential control: Delete coupling		+	-	PGAsl
TANGOF	P	Tangential control: Deactivate coupling		+	-	PGAsl
TANGON	P	Tangential control: Activate coupling		+	-	PGAsl
TCA (828D: _TCA)	P	Tool selection/tool change irrespective of tool status		+	-	FBWsl
TCARR	A	Request toolholder (number "m")		+		PGAsl
TCI	P	Load tool from buffer into magazine		+	-	FBWsl
TCOABS [6]	G	Determine tool length components from the current tool orientation	m	+		PGAsl
TCOFR	G	Determine tool length components from the orientation of the active frame	m	+		PGAsl
TCOFRX	G	Determine tool orientation of an active frame on selection of tool, tool points in X direction	m	+		PGAsl
TCOFRY	G	Determine tool orientation of an active frame on selection of tool, tool points in Y direction	m	+		PGAsl
TCOFRZ	G	Determine tool orientation of an active frame on selection of tool, tool points in Z direction	m	+		PGAsl
THETA	A	Angle of rotation	s	+		PGAsl
TILT	A	Tilt angle	m	+		PGAsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
TLIFT	P	Tangential control: Activate intermediate block generation		+	-	PGAsl
TML	P	Tool selection with magazine location number		+	-	FBWsl
TMOF	P	Deselect tool monitoring		+	-	PGAsl
TMON	P	Activate tool monitoring		+	-	PGAsl
TO	K	Designates the end value in a FOR counter loop		+		PGAsl
TOFF	K	Tool length offset in the direction of the tool length component that is effective parallel to the geometry axis specified in the index.	m	+		PGsl
TOFFL	K	Tool length offset in the direction of the tool length component L1, L2 or L3	m	+		PGsl
TOFFOF	P	Deactivate online tool offset		+	-	PGAsl
TOFFON	P	Activate online tool length offset		+	-	PGAsl
TOFFR	A	Tool radius offset	m	+		PGsl
TOFRAME	G	Align Z axis of the WCS by rotating the frame parallel to the tool orientation	m	+		PGsl
TOFRAMEX	G	Align X axis of the WCS by rotating the frame parallel to the tool orientation	m	+		PGsl
TOFRAMEY	G	Align Y axis of the WCS by rotating the frame parallel to the tool orientation	m	+		PGsl
TOFRAMEZ	G	As TOFRAME	m	+		PGsl
TOLOWER	F	Convert the letters of a string into lowercase		+	-	PGAsl
TOOLENV	F	Save current states which are of significance to the evaluation of the tool data stored in the memory		+	-	PGAsl
TOOLGNT	F	Determine number of tools of a tool group		+	-	FBWsl
TOOLGT	F	Determine T number of a tool from a tool group		+	-	FBWsl
TOROT	G	Align Z axis of the WCS by rotating the frame parallel to the tool orientation	m	+		PGsl
TOROTOF 6)	G	Frame rotations in tool direction OFF	m	+		PGsl
TOROTX	G	Align X axis of the WCS by rotating the frame parallel to the tool orientation	m	+		PGsl
TOROTY	G	Align Y axis of the WCS by rotating the frame parallel to the tool orientation	m	+		PGsl
TOROTZ	G	As TOROT	m	+		PGsl
TOUPPER	F	Convert the letters of a string into uppercase		+	-	PGAsl
TOWBCS	G	Wear values in the basic coordinate system (BCS)	m	+		PGAsl
TOWKCS	G	Wear values in the coordinate system of the tool head for kinetic transformation (differs from machine coordinate system through tool rotation)	m	+		PGAsl
TOWMCS	G	Wear values in machine coordinate system	m	+		PGAsl
TOWSTD 6)	G	Initial setting value for offsets in tool length	m	+		PGAsl

Operation	Type [1]	Meaning	W [2]	TP [3]	SA [4]	Description see [5]
[1] [2] [3] [4] [5] for explanations, see legend (Page 450).						
TOWTCS	G	Wear values in the tool coordinate system (toolholder ref. point T at the tool holder)	m	+		PGAsl
TOWWCS	G	Wear values in workpiece coordinate system	m	+		PGAsl
TR	K	Offset component of a frame variable		+		PGAsl
TRAANG	P	Transformation inclined axis		+	-	PGAsl
TRACON	P	Cascaded transformation		+	-	PGAsl
TRACYL	P	Cylinder: Peripheral surface transformation		+	-	PGAsl
TRAFOOF	P	Deactivate active transformations in the channel		+	-	PGAsl
TRAFOON	P	Activate transformation defined with kinematic chains		+	-	PGAsl
TRAILOF	P	Asynchronous coupled motion OFF		+	+	PGAsl
TRAILON	P	Asynchronous coupled motion ON		+	+	PGAsl
TRANS	G	Absolute programmable work offset	s	+		PGsl
TRANSMIT	P	Pole transformation (face machining)		+	-	PGAsl
TRAORI	P	4-axis, 5-axis transformation, generic transformation		+	-	PGAsl
TRUE	K	Logical constant: True		+		PGAsl
TRUNC	F	Truncation of decimal places		+	+	PGAsl
TU		Axis angle	s	+		PGAsl
TURN	A	Number of turns for helix	s	+		PGsl
ULI	K	Upper limit value of variables		+		PGAsl
UNLOCK	P	Enable synchronized action with ID (continue technology cycle)		-	+	FBSYsl
UNTIL	K	Condition for end of REPEAT loop		+		PGAsl
UPATH	G	Path reference for FGROUP axes is curve parameter	m	+		PGAsl
VAR	K	Keyword: Type of parameter transfer		+		PGAsl
VELOLIM	K	Reduction of the maximum axial velocity	m	+		PGAsl
VELOLIMA	K	Reduction or increase of the maximum axial velocity of the following axis	m	+	+	PGAsl
WAITC	P	Wait for the coupling block change criterion to be fulfilled for the axes/spindles		+	-	PGAsl
WAITE	P	Wait for end of program in another channel.		+	-	PGAsl
WAITENC	P	Wait for synchronized or restored axis positions		+	-	PGAsl
WAITM	P	Wait for marker in specified channel; terminate previous block with exact stop.		+	-	PGAsl
WAITMC	P	Wait for marker in specified channel; exact stop only if the other channels have not yet reached the marker.		+	-	PGAsl
WAITP	P	Wait for end of travel of the positioning axis		+	-	PGsl
WAITS	P	Wait for spindle position to be reached		+	-	PGsl

Operation	Type 1)	Meaning	W 2)	TP 3)	SA 4)	Description see 5)
1) 2) 3) 4) 5) for explanations, see legend (Page 450).						
WALCS0 6)	G	Workpiece coordinate system working area limitation deselected	m	+	-	PGsl
WALCS1	G	WCS working area limitation group 1 active	m	+	-	PGsl
WALCS2	G	WCS working area limitation group 2 active	m	+	-	PGsl
WALCS3	G	WCS working area limitation group 3 active	m	+	-	PGsl
WALCS4	G	WCS working area limitation group 4 active	m	+	-	PGsl
WALCS5	G	WCS working area limitation group 5 active	m	+	-	PGsl
WALCS6	G	WCS working area limitation group 6 active	m	+	-	PGsl
WALCS7	G	WCS working area limitation group 7 active	m	+	-	PGsl
WALCS8	G	WCS working area limitation group 8 active	m	+	-	PGsl
WALCS9	G	WCS working area limitation group 9 active	m	+	-	PGsl
WALCS10	G	WCS working area limitation group 10 active	m	+	-	PGsl
WALIMOF	G	BCS working area limitation OFF	m	+	-	PGsl
WALIMON 6)	G	BCS working area limitation ON	m	+	-	PGsl
WHEN	K	The action is executed once whenever the condition is fulfilled.		-	+	FBSYsl
WHENEVER	K	The action is executed cyclically in each interpolator cycle when the condition is fulfilled.		-	+	FBSYsl
WHILE	K	Start of WHILE program loop		+		PGAsl
WRITE	P	Write text to file system. Appends a block to the end of the specified file.		+	-	PGAsl
WRTPR	P	Write string in OPI variable		+	-	PGsl
X	A	Axis name	m/s	+	+	PGsl
XOR	O	Logic exclusive OR		+		PGAsl
Y	A	Axis name	m/s	+	+	PGsl
Z	A	Axis name	m/s	+	+	PGsl

Legend

[1] Type of operation:

A Address

Identifier to which a value is assigned (e.g. OVR=10). There are also some addresses that switch on or off a function without value assignment (e.g. CPLON and CPLOF).

C (A) AST cycle

Predefined NC program for automatic post optimization (tuning) with AST (= Automatic Servo Tuning). Parameters are used to adapt to the specific optimization situation; these parameters are transferred at the call.

C (M) Measuring cycle

Predefined NC program in which a specific, generally valid, measuring operation, such as determining the inner diameter of a cylindrical workpiece, is programmed. Parameters are used to adapt to the specific measurement situation; these parameters are transferred at the call.

C (T) Technological cycle

Predefined NC program in which a specific, generally valid, machining operation, such as tapping of a thread or milling a pocket, is programmed. The adaptation to a specific machine situation is realized via parameters that are transferred to the cycle during the call.

F Predefined function (supplies a return value)

The call of the predefined function can be an operand in an expression.

G G command

The G commands are divided into G groups. Only one G command of a group can be programmed in a block. A G command can be either modal (until it is canceled by another command of the same group) or only effective for the block in which it is programmed (non-modal).

K Keyword

Identifier that defines the syntax of a block. No value is assigned to a keyword, and no NC function can be switched on/off with a keyword.

Examples: Control structures (IF, ELSE, ENDIF, WHEN, ...), program execution (GOTOB, GOTO, RET ...)

O Operator

Operator for a mathematical, comparison or logical operation

P Predefined procedure (does not supply a return value)

PA Program attribute

Program attributes are at the end of the definition line of a subprogram:
`PROC <program name>(...) <program attribute>`

They determine the behavior during execution of the subprogram.

[2] Effectiveness of the operation:

m Modal

s Non-modal

[3] Programmability in part program:

+ Programmable

- Not programmable

M Programmable only by the machine manufacturer

[4] Programmability in synchronized actions:

+	Programmable
-	Not programmable
T	Programmable only in technology cycles

[5] Reference to the document containing the detailed description of the operation:

PGsl	Programming Manual, Fundamentals
PGAsl	Programming Manual, Job Planning
BNMsl	Programming Manual Measuring Cycles
BHDsl	Operating Manual, Turning
BHFsl	Operating Manual, Milling
FB1sl ( )	Function Manual, Basic Functions (with the alphanumeric abbreviation of the corresponding function description in brackets)
FB2sl ( )	Function Manual, Extended Functions (with the alphanumeric abbreviation of the corresponding function description in brackets)
FB3sl ( )	Function Manual, Special Functions (with the alphanumeric abbreviation of the corresponding function description in brackets)
FBSIsl	Function Manual, Safety Integrated
FBSYsl	Function Manual, Synchronized Actions
FBWsl	Function Manual, Tool Management

[6] Default setting at beginning of program (factory settings of the control, if nothing else programmed).

Figure 17-1 Meaning of footnotes in the tables of operations

17.2 Operations: Availability for SINUMERIK 828D

Note

Cycles

Cycles are marked as "optional" if they depend on the following options that require a license:

- Extended technology functions (article number: 6FC5800-0AP58-0YB0)
- Measuring cycles (article number: 6FC5800-0AP28-0YB0)
- Measuring kinematics (article number: 6FC5800-0AP18-0YB0)
- SINUMERIK Grinding Advanced (article number: 6FC5800-0AS35-0YB0)

Not marked, if cycles only contain partial functionalities as a result of option "Extended technology functions".

17.2.1 Control version milling / turning

Operations A ... C

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
:	●	●	●	●	●	●	●	●
*	●	●	●	●	●	●	●	●
+	●	●	●	●	●	●	●	●
-	●	●	●	●	●	●	●	●
<	●	●	●	●	●	●	●	●
<<	●	●	●	●	●	●	●	●
<=	●	●	●	●	●	●	●	●
=	●	●	●	●	●	●	●	●
>=	●	●	●	●	●	●	●	●
/	●	●	●	●	●	●	●	●
/0 … /7	●	●	●	●	●	●	●	●
A	●	●	●	●	●	●	●	●
A2	-	-	-	-	-	-	-	-
A3	-	-	-	-	-	-	-	-
A4	-	-	-	-	-	-	-	-
A5	-	-	-	-	-	-	-	-
A6	-	-	-	-	-	-	-	-
A7	-	-	-	-	-	-	-	-
ABS	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
AC	●	●	●	●	●	●	●	●
ACC	●	●	●	●	●	●	●	●
ACCLIMA	●	●	●	●	●	●	●	●
ACN	●	●	●	●	●	●	●	●
ACOS	●	●	●	●	●	●	●	●
ACP	●	●	●	●	●	●	●	●
ACTBLOCNO	●	●	●	●	●	●	●	●
ADDFRAME	●	●	●	●	●	●	●	●
ADIS	●	●	●	●	●	●	●	●
ADISPOS	●	●	●	●	●	●	●	●
ADISPOSA	●	●	●	●	●	●	●	●
ALF	●	●	●	●	●	●	●	●
AMIRROR	●	●	●	●	●	●	●	●
AND	●	●	●	●	●	●	●	●
ANG	●	●	●	●	●	●	●	●
AP	●	●	●	●	●	●	●	●
APR	●	●	●	●	●	●	●	●
APRB	●	●	●	●	●	●	●	●
APRP	●	●	●	●	●	●	●	●
APW	●	●	●	●	●	●	●	●
APWB	●	●	●	●	●	●	●	●
APWP	●	●	●	●	●	●	●	●
APX	●	●	●	●	●	●	●	●
AR	●	●	●	●	●	●	●	●
AROT	●	●	●	●	●	●	●	●
AROTS	●	●	●	●	●	●	●	●
AS	●	●	●	●	●	●	●	●
ASCALE	●	●	●	●	●	●	●	●
ASIN	●	●	●	●	●	●	●	●
ASPLINE	○	○	○	○	○	○	○	○
ATAN2	●	●	●	●	●	●	●	●
ATOL	●	●	●	●	●	●	●	●
ATRANS	●	●	●	●	●	●	●	●
AUXFUDEL	●	●	●	●	●	●	●	●
AUXFUDELG	●	●	●	●	●	●	●	●
AUXFUMSEQ	●	●	●	●	●	●	●	●
AUXFUSYNC	●	●	●	●	●	●	●	●
AX	●	●	●	●	●	●	●	●
AXCTSWE	-	-	-	-	-	-	-	-

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
AXCTSWEC	-	-	-	-	-	-	-	-
AXCTSWED	-	-	-	-	-	-	-	-
AXIS	●	●	●	●	●	●	●	●
AXNAME	●	●	●	●	●	●	●	●
AXSTRING	●	●	●	●	●	●	●	●
AXTOCHAN	●	●	●	●	●	●	●	●
AXTOSPI	●	●	●	●	●	●	●	●
B	●	●	●	●	●	●	●	●
B2	-	-	-	-	-	-	-	-
B3	-	-	-	-	-	-	-	-
B4	-	-	-	-	-	-	-	-
B5	-	-	-	-	-	-	-	-
B6	-	-	-	-	-	-	-	-
B7	-	-	-	-	-	-	-	-
B_AND	●	●	●	●	●	●	●	●
B_OR	●	●	●	●	●	●	●	●
B_NOT	●	●	●	●	●	●	●	●
B_XOR	●	●	●	●	●	●	●	●
BAUTO	○	○	○	○	○	○	○	○
BLOCK	●	●	●	●	●	●	●	●
BLSYNC	●	●	●	●	●	●	●	●
BNAT	○	○	○	○	○	○	○	○
BOOL	●	●	●	●	●	●	●	●
BOUND	●	●	●	●	●	●	●	●
BRISK	●	●	●	●	●	●	●	●
BRISKA	●	●	●	●	●	●	●	●
BSPLINE	○	○	○	○	○	○	○	○
BTAN	○	○	○	○	○	○	○	○
C	●	●	●	●	●	●	●	●
C2	-	-	-	-	-	-	Channel axis name	Channel ax-is name
C3	-	-	-	-	-	-	-	-
C4	-	-	-	-	-	-	-	-
C5	-	-	-	-	-	-	-	-
C6	-	-	-	-	-	-	-	-
C7	-	-	-	-	-	-	-	-
CAC	●	●	●	●	●	●	●	●
CACN	●	●	●	●	●	●	●	●
CACP	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
CALCDAT	●	●	●	●	●	●	●	●
CALCPOSI	●	●	●	●	●	●	●	●
CALL	●	●	●	●	●	●	●	●
CALLPATH	●	●	●	●	●	●	●	●
CANCEL	●	●	●	●	●	●	●	●
CASE	●	●	●	●	●	●	●	●
CDC	●	●	●	●	●	●	●	●
CDOF	-	-	-	-	-	-	-	-
CDOF2	-	-	-	-	-	-	-	-
CDON	-	-	-	-	-	-	-	-
CFC	●	●	●	●	●	●	●	●
CFIN	●	●	●	●	●	●	●	●
CFINE	●	●	●	●	●	●	●	●
CFTCP	●	●	●	●	●	●	●	●
CHAN	●	●	●	●	●	●	●	●
CHANDATA	●	●	●	●	●	●	●	●
CHAR	●	●	●	●	●	●	●	●
CHF	●	●	●	●	●	●	●	●
CHKDM	●	●	●	●	●	●	●	●
CHKDNO	●	●	●	●	●	●	●	●
CHR	●	●	●	●	●	●	●	●
CIC	●	●	●	●	●	●	●	●
CIP	●	●	●	●	●	●	●	●
CLEARM	-	-	-	-	-	○	-	●
CLRINT	●	●	●	●	●	●	●	●
CMIRROR	●	●	●	●	●	●	●	●
COARSEA	●	●	●	●	●	●	●	●
COLLPAIR	-	-	-	-	-	-	-	-
COMPCAD	●	-	●	-	●	●	-	-
COMPCURV	●	-	●	-	●	●	-	-
COMPLETE	●	●	●	●	●	●	●	●
COMPOF	●	-	●	-	●	●	-	-
COMPON	●	-	●	-	●	●	-	-
COMPSURF	-	-	○	-	○	○	-	-
CONTDCON	●	●	●	●	●	●	●	●
CONTPRON	●	●	●	●	●	●	●	●
CORROF	●	●	●	●	●	●	●	●
CORRTRAFO	-	-	-	-	-	-	-	-
COS	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
COUPDEF	-	○	-	○	-	-	○	○
COUPDEL	-	○	-	○	-	-	○	○
COUPOF	-	○	-	○	-	-	○	○
COUPOFS	-	○	-	○	-	-	○	○
COUPON	-	○	-	○	-	-	○	○
COUPONC	-	○	-	○	-	-	○	○
COUPRES	-	○	-	○	-	-	○	○
CP	●	●	●	●	●	●	●	●
CPBC	○	○	○	○	○	○	○	○
CPDEF	○	○	○	○	○	○	○	○
CPDEL	○	○	○	○	○	○	○	○
CPFMOF	○	○	○	○	○	○	○	○
CPFMON	○	○	○	○	○	○	○	○
CPFMSON	○	○	○	○	○	○	○	○
CPFPOS	○	○	○	○	○	○	○	○
CPFRS	○	○	○	○	○	○	○	○
CPLA	○	○	○	○	○	○	○	○
CPLCTID	○	○	○	○	○	○	○	○
CPLDEF	○	○	○	○	○	○	○	○
CPLDEL	○	○	○	○	○	○	○	○
CPLDEN	○	○	○	○	○	○	○	○
CPLINSC	○	○	○	○	○	○	○	○
CPLINTR	○	○	○	○	○	○	○	○
CPLNUM	○	○	○	○	○	○	○	○
CPLOF	○	○	○	○	○	○	○	○
CPLON	○	○	○	○	○	○	○	○
CPLOUTSC	○	○	○	○	○	○	○	○
CPLOUTTR	○	○	○	○	○	○	○	○
CPLPOS	○	○	○	○	○	○	○	○
CPLSETVAL	○	○	○	○	○	○	○	○
CPMALARM	○	○	○	○	○	○	○	○
CPMBRAKE	○	○	○	○	○	○	○	○
CPMPRT	○	○	○	○	○	○	○	○
CPMRESET	○	○	○	○	○	○	○	○
CPMSTART	○	○	○	○	○	○	○	○
CPMVDI	○	○	○	○	○	○	○	○
CPOF	○	○	○	○	○	○	○	○
CPON	○	○	○	○	○	○	○	○
CPRECOF	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
CPRECON	●	●	●	●	●	●	●	●
CPRES	○	○	○	○	○	○	○	○
CPROT	●	●	●	●	●	●	●	●
CPROTDEF	●	●	●	●	●	●	●	●
CPSETTYPE	○	○	○	○	○	○	○	○
CPSYNCOP	○	○	○	○	○	○	○	○
CPSYNCOP2	○	○	○	○	○	○	○	○
CPSYNCOV	○	○	○	○	○	○	○	○
CPSYNFIP	○	○	○	○	○	○	○	○
CPSYNFIP2	○	○	○	○	○	○	○	○
CPSYNFIV	○	○	○	○	○	○	○	○
CR	●	●	●	●	●	●	●	●
CROT	●	●	●	●	●	●	●	●
CROTS	●	●	●	●	●	●	●	●
CRPL	●	●	●	●	●	●	●	●
CSCALE	●	●	●	●	●	●	●	●
CSPLINE	○	○	○	○	○	○	○	○
CT	●	●	●	●	●	●	●	●
CTAB	-	-	-	-	-	-	-	-
CTABDEF	-	-	-	-	-	-	-	-
CTABDEL	-	-	-	-	-	-	-	-
CTABEND	-	-	-	-	-	-	-	-
CTABEXISTS	-	-	-	-	-	-	-	-
CTABFNO	-	-	-	-	-	-	-	-
CTABFPOL	-	-	-	-	-	-	-	-
CTABFSEG	-	-	-	-	-	-	-	-
CTABID	-	-	-	-	-	-	-	-
CTABINV	-	-	-	-	-	-	-	-
CTABISLOCK	-	-	-	-	-	-	-	-
CTABLOCK	-	-	-	-	-	-	-	-
CTABMEMTYP	-	-	-	-	-	-	-	-
CTABMPOL	-	-	-	-	-	-	-	-
CTABMSEG	-	-	-	-	-	-	-	-
CTABNO	-	-	-	-	-	-	-	-
CTABNOMEM	-	-	-	-	-	-	-	-
CTABPERIOD	-	-	-	-	-	-	-	-
CTABPOL	-	-	-	-	-	-	-	-
CTABPOLID	-	-	-	-	-	-	-	-
CTABSEG	-	-	-	-	-	-	-	-

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
CTABSEGID	-	-	-	-	-	-	-	-
CTABSEV	-	-	-	-	-	-	-	-
CTABSSV	-	-	-	-	-	-	-	-
CTABTEP	-	-	-	-	-	-	-	-
CTABTEV	-	-	-	-	-	-	-	-
CTABTMAX	-	-	-	-	-	-	-	-
CTABTMIN	-	-	-	-	-	-	-	-
CTABTSP	-	-	-	-	-	-	-	-
CTABTSV	-	-	-	-	-	-	-	-
CTABUNLOCK	-	-	-	-	-	-	-	-
CTOL	●	●	●	●	●	●	●	●
CTRANS	●	●	●	●	●	●	●	●
CUT2D	●	●	●	●	●	●	●	●
CUT2DD	●	●	●	●	●	●	●	●
CUT2DF	●	●	●	●	●	●	●	●
CUT2DFD	●	●	●	●	●	●	●	●
CUT3DC	-	-	-	-	-	-	-	-
CUT3DCC	-	-	-	-	-	-	-	-
CUT3DCCD	-	-	-	-	-	-	-	-
CUT3DCD	-	-	-	-	-	-	-	-
CUT3DF	-	-	-	-	-	-	-	-
CUT3DFD	-	-	-	-	-	-	-	-
CUT3DFF	-	-	-	-	-	-	-	-
CUT3DFS	-	-	-	-	-	-	-	-
CUTCONOF	●	●	●	●	●	●	●	●
CUTCONON	●	●	●	●	●	●	●	●
CUTMOD	●	●	●	●	●	●	●	●
CUTMODK	-	-	-	-	-	-	-	-
CYCLE60	○	○	●	●	●	●	●	●
CYCLE61	●	●	●	●	●	●	●	●
CYCLE62	●	●	●	●	●	●	●	●
CYCLE63	○	○	●	●	●	●	●	●
CYCLE64	○	○	●	●	●	●	●	●
CYCLE70	○	○	●	●	●	●	●	●
CYCLE72	●	●	●	●	●	●	●	●
CYCLE76	●	●	●	●	●	●	●	●
CYCLE77	●	●	●	●	●	●	●	●
CYCLE78	○	○	●	●	●	●	●	●
CYCLE79	○	○	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
CYCLE81	●	●	●	●	●	●	●	●
CYCLE82	●	●	●	●	●	●	●	●
CYCLE83	●	●	●	●	●	●	●	●
CYCLE84	●	●	●	●	●	●	●	●
CYCLE85	●	●	●	●	●	●	●	●
CYCLE86	●	●	●	●	●	●	●	●
CYCLE92	●	●	●	●	●	●	●	●
CYCLE95	●	●	●	●	●	●	●	●
CYCLE98	●	●	●	●	●	●	●	●
CYCLE99	●	●	●	●	●	●	●	●
CYCLE150	○	○	○	○	○	○	○	○
CYCLE435	-	-	-	-	-	-	-	-
CYCLE495	-	-	-	-	-	-	-	-
CYCLE750	●	●	●	●	●	●	●	●
CYCLE751	●	●	●	●	●	●	●	●
CYCLE752	●	●	●	●	●	●	●	●
CYCLE753	●	●	●	●	●	●	●	●
CYCLE754	●	●	●	●	●	●	●	●
CYCLE755	●	●	●	●	●	●	●	●
CYCLE756	●	●	●	●	●	●	●	●
CYCLE757	●	●	●	●	●	●	●	●
CYCLE758	●	●	●	●	●	●	●	●
CYCLE759	●	●	●	●	●	●	●	●
CYCLE800	-	-	●	●	●	●	●	●
CYCLE801	●	●	●	●	●	●	●	●
CYCLE802	●	●	●	●	●	●	●	●
CYCLE830	○	○	●	●	●	●	●	●
CYCLE832	●	●	●	●	●	●	●	●
CYCLE840	●	●	●	●	●	●	●	●
CYCLE899	○	○	●	●	●	●	●	●
CYCLE930	●	●	●	●	●	●	●	●
CYCLE940	●	●	●	●	●	●	●	●
CYCLE951	●	●	●	●	●	●	●	●
CYCLE952	○	○	●	●	●	●	●	●
CYCLE961	○	○	○	○	○	○	○	○
CYCLE971	○	○	○	○	○	○	○	○
CYCLE973	○	○	○	○	○	○	○	○
CYCLE974	○	○	○	○	○	○	○	○
CYCLE976	○	○	○	○	○	○	○	○

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
CYCLE977	○	○	○	○	○	○	○	○
CYCLE978	○	○	○	○	○	○	○	○
CYCLE979	○	○	○	○	○	○	○	○
CYCLE982	○	○	○	○	○	○	○	○
CYCLE994	○	○	○	○	○	○	○	○
CYCLE995	○	○	○	○	○	○	○	○
CYCLE996	○	-	○	-	○	○	-	-
CYCLE997	○	○	○	○	○	○	○	○
CYCLE998	○	○	○	○	○	○	○	○
CYCLE4071	-	-	-	-	-	-	-	-
CYCLE4072	-	-	-	-	-	-	-	-
CYCLE4073	-	-	-	-	-	-	-	-
CYCLE4074	-	-	-	-	-	-	-	-
CYCLE4075	-	-	-	-	-	-	-	-
CYCLE4077	-	-	-	-	-	-	-	-
CYCLE4078	-	-	-	-	-	-	-	-
CYCLE4079	-	-	-	-	-	-	-	-

Operations D ... F

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
D	●	●	●	●	●	●	●	●
D0	●	●	●	●	●	●	●	●
DAC	●	●	●	●	●	●	●	●
DC	●	●	●	●	●	●	●	●
DCI	●	●	●	●	●	●	●	●
DCM	●	●	●	●	●	●	●	●
DCU	●	●	●	●	●	●	●	●
DEF	●	●	●	●	●	●	●	●
DEFINE	●	●	●	●	●	●	●	●
DEFAULT	●	●	●	●	●	●	●	●
DELAYFSTON	●	●	●	●	●	●	●	●
DELAYFSTOF	●	●	●	●	●	●	●	●
DELDL	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
DELDTG	●	●	●	●	●	●	●	●
DELETE	●	●	●	●	●	●	●	●
DELMLOWNER	●	●	●	●	●	●	●	●
DELMLRES	●	●	●	●	●	●	●	●
DELMT	●	●	●	●	●	●	●	●
DELOBJ	-	-	-	-	-	-	-	-
DELT	●	●	●	●	●	●	●	●
DELTC	●	●	●	●	●	●	●	●
DELTOOLENV	●	●	●	●	●	●	●	●
DIACYCOFA	●	●	●	●	●	●	●	●
DIAM90	●	●	●	●	●	●	●	●
DIAM90A	●	●	●	●	●	●	●	●
DIAMCHAN	●	●	●	●	●	●	●	●
DIAMCHANA	●	●	●	●	●	●	●	●
DIAMCYCOF	●	●	●	●	●	●	●	●
DIAMOF	●	●	●	●	●	●	●	●
DIAMOFA	●	●	●	●	●	●	●	●
DIAMON	●	●	●	●	●	●	●	●
DIAMONA	●	●	●	●	●	●	●	●
DIC	●	●	●	●	●	●	●	●
DILF	●	●	●	●	●	●	●	●
DISABLE	●	●	●	●	●	●	●	●
DISC	●	●	●	●	●	●	●	●
DISCL	●	●	●	●	●	●	●	●
DISPLOF	●	●	●	●	●	●	●	●
DISPLON	●	●	●	●	●	●	●	●
DISPR	●	●	●	●	●	●	●	●
DISR	●	●	●	●	●	●	●	●
DISRP	●	●	●	●	●	●	●	●
DITE	●	●	●	●	●	●	●	●
DITS	●	●	●	●	●	●	●	●
DIV	●	●	●	●	●	●	●	●
DL	-	-	-	-	-	-	-	-
DO	●	●	●	●	●	●	●	●
DRFOF	●	●	●	●	●	●	●	●
DRIVE	●	●	●	●	●	●	●	●
DRIVEA	●	●	●	●	●	●	●	●
DYNFINISH	●	●	●	●	●	●	●	●
DYNNORM	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
DYNPOS	●	●	●	●	●	●	●	●
DYNROUGH	●	●	●	●	●	●	●	●
DYNSEMIFIN	●	●	●	●	●	●	●	●
DZERO	●	●	●	●	●	●	●	●
EAUTO	○	○	○	○	○	○	○	○
EGDEF	-	○	-	○	-	-	○	○
EGDEL	-	○	-	○	-	-	○	○
EGOFC	-	○	-	○	-	-	○	○
EGOFS	-	○	-	○	-	-	○	○
EGON	-	○	-	○	-	-	○	○
EGONSYN	-	○	-	○	-	-	○	○
EGONSYNE	-	○	-	○	-	-	○	○
ELSE	●	●	●	●	●	●	●	●
ENABLE	●	●	●	●	●	●	●	●
ENAT	○	○	○	○	○	○	○	○
ENDFOR	●	●	●	●	●	●	●	●
ENDIF	●	●	●	●	●	●	●	●
ENDLABEL	●	●	●	●	●	●	●	●
ENDLOOP	●	●	●	●	●	●	●	●
ENDPROC	●	●	●	●	●	●	●	●
ENDWHILE	●	●	●	●	●	●	●	●
ESRR	○	○	○	○	○	○	○	○
ESRS	○	○	○	○	○	○	○	○
ETAN	○	○	○	○	○	○	○	○
EVERY	●	●	●	●	●	●	●	●
EX	●	●	●	●	●	●	●	●
EXECSTRING	●	●	●	●	●	●	●	●
EXECTAB	●	●	●	●	●	●	●	●
EXECUTE	●	●	●	●	●	●	●	●
EXP	●	●	●	●	●	●	●	●
EXTCALL	●	●	●	●	●	●	●	●
EXTCLOSE	●	●	●	●	●	●	●	●
EXTERN	●	●	●	●	●	●	●	●
EXTOPEN	●	●	●	●	●	●	●	●
F	●	●	●	●	●	●	●	●
FA	●	●	●	●	●	●	●	●
FAD	●	●	●	●	●	●	●	●
FALSE	●	●	●	●	●	●	●	●
FB	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
FCTDEF	●	●	●	●	●	●	●	●
FCUB	●	●	●	●	●	●	●	●
FD	●	●	●	●	●	●	●	●
FDA	●	●	●	●	●	●	●	●
FENDNORM	●	●	●	●	●	●	●	●
FFWOF	●	●	●	●	●	●	●	●
FFWON	●	●	●	●	●	●	●	●
FGREF	●	●	●	●	●	●	●	●
FGROUP	●	●	●	●	●	●	●	●
FI	●	●	●	●	●	●	●	●
FIFOCTRL	●	●	●	●	●	●	●	●
FILEDATE	●	●	●	●	●	●	●	●
FILEINFO	●	●	●	●	●	●	●	●
FILESIZE	●	●	●	●	●	●	●	●
FILESTAT	●	●	●	●	●	●	●	●
FILETIME	●	●	●	●	●	●	●	●
FINEA	●	●	●	●	●	●	●	●
FL	●	●	●	●	●	●	●	●
FLIN	●	●	●	●	●	●	●	●
FMA	●	●	●	●	●	●	●	●
FNORM	●	●	●	●	●	●	●	●
FOCOF	○	○	○	○	○	○	○	○
FOCON	○	○	○	○	○	○	○	○
FOR	●	●	●	●	●	●	●	●
FP	●	●	●	●	●	●	●	●
FPO	-	-	-	-	-	-	-	-
FPR	●	●	●	●	●	●	●	●
FPRAOF	●	●	●	●	●	●	●	●
FPRAON	●	●	●	●	●	●	●	●
FRAME	●	●	●	●	●	●	●	●
FRC	●	●	●	●	●	●	●	●
FRCM	●	●	●	●	●	●	●	●
FROM	●	●	●	●	●	●	●	●
FTOC	●	●	●	●	●	●	●	●
FTOCOF	●	●	●	●	●	●	●	●
FTOCON	●	●	●	●	●	●	●	●
FXS	●	●	●	●	●	●	●	●
FXST	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
FXSW	●	●	●	●	●	●	●	●
FZ	●	●	●	●	●	●	●	●

Operations G ... L

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
G0	●	●	●	●	●	●	●	●
G1	●	●	●	●	●	●	●	●
G2	●	●	●	●	●	●	●	●
G3	●	●	●	●	●	●	●	●
G4	●	●	●	●	●	●	●	●
G5	●	●	●	●	●	●	●	●
G7	●	●	●	●	●	●	●	●
G9	●	●	●	●	●	●	●	●
G17	●	●	●	●	●	●	●	●
G18	●	●	●	●	●	●	●	●
G19	●	●	●	●	●	●	●	●
G25	●	●	●	●	●	●	●	●
G26	●	●	●	●	●	●	●	●
G33	●	●	●	●	●	●	●	●
G34	●	●	●	●	●	●	●	●
G35	●	●	●	●	●	●	●	●
G40	●	●	●	●	●	●	●	●
G41	●	●	●	●	●	●	●	●
G42	●	●	●	●	●	●	●	●
G53	●	●	●	●	●	●	●	●
G54	●	●	●	●	●	●	●	●
G55	●	●	●	●	●	●	●	●
G56	●	●	●	●	●	●	●	●
G57	●	●	●	●	●	●	●	●
G58	→ G505							
G59	→ G506							
G60	●	●	●	●	●	●	●	●
G62	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
G63	●	●	●	●	●	●	●	●
G64	●	●	●	●	●	●	●	●
G70	●	●	●	●	●	●	●	●
G71	●	●	●	●	●	●	●	●
G74	●	●	●	●	●	●	●	●
G75	●	●	●	●	●	●	●	●
G90	●	●	●	●	●	●	●	●
G91	●	●	●	●	●	●	●	●
G93	●	●	●	●	●	●	●	●
G94	●	●	●	●	●	●	●	●
G95	●	●	●	●	●	●	●	●
G96	●	●	●	●	●	●	●	●
G97	●	●	●	●	●	●	●	●
G110	●	●	●	●	●	●	●	●
G111	●	●	●	●	●	●	●	●
G112	●	●	●	●	●	●	●	●
G140	●	●	●	●	●	●	●	●
G141	●	●	●	●	●	●	●	●
G142	●	●	●	●	●	●	●	●
G143	●	●	●	●	●	●	●	●
G147	●	●	●	●	●	●	●	●
G148	●	●	●	●	●	●	●	●
G153	●	●	●	●	●	●	●	●
G247	●	●	●	●	●	●	●	●
G248	●	●	●	●	●	●	●	●
G290	●	●	●	●	●	●	●	●
G291	●	●	●	●	●	●	●	●
G331	●	●	●	●	●	●	●	●
G332	●	●	●	●	●	●	●	●
G335	●	●	●	●	●	●	●	●
G336	●	●	●	●	●	●	●	●
G340	●	●	●	●	●	●	●	●
G341	●	●	●	●	●	●	●	●
G347	●	●	●	●	●	●	●	●
G348	●	●	●	●	●	●	●	●
G450	●	●	●	●	●	●	●	●
G451	●	●	●	●	●	●	●	●
G460	●	●	●	●	●	●	●	●
G461	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
G462	●	●	●	●	●	●	●	●
G500	●	●	●	●	●	●	●	●
G505 ... G599	●	●	●	●	●	●	●	●
G601	●	●	●	●	●	●	●	●
G602	●	●	●	●	●	●	●	●
G603	●	●	●	●	●	●	●	●
G621	●	●	●	●	●	●	●	●
G641	●	●	●	●	●	●	●	●
G642	●	●	●	●	●	●	●	●
G643	●	●	●	●	●	●	●	●
G644	●	●	●	●	●	●	●	●
G645	●	●	●	●	●	●	●	●
G700	●	●	●	●	●	●	●	●
G710	●	●	●	●	●	●	●	●
G810 ... G819	-	-	-	-	-	-	-	-
G820 ... G829	-	-	-	-	-	-	-	-
G931	●	●	●	●	●	●	●	●
G942	●	●	●	●	●	●	●	●
G952	●	●	●	●	●	●	●	●
G961	●	●	●	●	●	●	●	●
G962	●	●	●	●	●	●	●	●
G971	●	●	●	●	●	●	●	●
G972	●	●	●	●	●	●	●	●
G973	●	●	●	●	●	●	●	●
GEOAX	●	●	●	●	●	●	●	●
GET	●	●	●	●	●	●	●	●
GETACTT	●	●	●	●	●	●	●	●
GETACTTD	●	●	●	●	●	●	●	●
GETD	-	-	-	-	-	○	-	●
GETDNO	●	●	●	●	●	●	●	●
GETEXET	●	●	●	●	●	●	●	●
GETFREELOC	●	●	●	●	●	●	●	●
GETSELT	●	●	●	●	●	●	●	●
GETT	●	●	●	●	●	●	●	●
GETTCOR	●	●	●	●	●	●	●	●
GETTENV	●	●	●	●	●	●	●	●
GETVARAP	●	●	●	●	●	●	●	●
GETVARDFT	●	●	●	●	●	●	●	●
GETVARLIM	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
GETVARPHU	●	●	●	●	●	●	●	●
GETVARTYP	●	●	●	●	●	●	●	●
GFRAME0 ... GFRAME100	-	-	-	-	-	-	-	-
GOTO	●	●	●	●	●	●	●	●
GOTOB	●	●	●	●	●	●	●	●
GOTOC	●	●	●	●	●	●	●	●
GOTOF	●	●	●	●	●	●	●	●
GOTOS	●	●	●	●	●	●	●	●
GP	●	●	●	●	●	●	●	●
GWPSOF	●	●	●	●	●	●	●	●
GROUP_ ADDEND	●	●	●	●	●	●	●	●
GROUP_BEGIN	●	●	●	●	●	●	●	●
GROUP_END	●	●	●	●	●	●	●	●
GWPSON	●	●	●	●	●	●	●	●
H...	●	●	●	●	●	●	●	●
HOLES1	●	●	●	●	●	●	●	●
HOLES2	●	●	●	●	●	●	●	●
I	●	●	●	●	●	●	●	●
I1	●	●	●	●	●	●	●	●
IC	●	●	●	●	●	●	●	●
ICYCOF	●	●	●	●	●	●	●	●
ICYCON	●	●	●	●	●	●	●	●
ID	●	●	●	●	●	●	●	●
IDS	●	●	●	●	●	●	●	●
IF	●	●	●	●	●	●	●	●
INDEX	●	●	●	●	●	●	●	●
INIPO	●	●	●	●	●	●	●	●
INIRE	●	●	●	●	●	●	●	●
INICF	●	●	●	●	●	●	●	●
INIT	-	-	-	-	-	○	-	●
INITIAL								
INT	●	●	●	●	●	●	●	●
INTERSEC	●	●	●	●	●	●	●	●
INVCCW	-	-	-	-	-	-	-	-
INVCW	-	-	-	-	-	-	-	-
INVFRAME	●	●	●	●	●	●	●	●
IP	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
IPOBRKA	●	●	●	●	●	●	●	●
IPOENDA	●	●	●	●	●	●	●	●
IPTRLOCK	●	●	●	●	●	●	●	●
IPTRUNLOCK	●	●	●	●	●	●	●	●
IR	●	●	●	●	●	●	●	●
ISAXIS	●	●	●	●	●	●	●	●
ISD	-	-	-	-	-	-	-	-
ISFILE	●	●	●	●	●	●	●	●
ISNUMBER	●	●	●	●	●	●	●	●
ISOCALL	●	●	●	●	●	●	●	●
ISVAR	●	●	●	●	●	●	●	●
J	●	●	●	●	●	●	●	●
J1	●	●	●	●	●	●	●	●
JERKA	●	●	●	●	●	●	●	●
JERKLIM	●	●	●	●	●	●	●	●
JERKLIMA	●	●	●	●	●	●	●	●
JR	●	●	●	●	●	●	●	●
K	●	●	●	●	●	●	●	●
K1	●	●	●	●	●	●	●	●
KONT	●	●	●	●	●	●	●	●
KONTC	●	●	●	●	●	●	●	●
KONTT	●	●	●	●	●	●	●	●
KR	●	●	●	●	●	●	●	●
L	●	●	●	●	●	●	●	●
LEAD Tool orientation Orientation polyn.	- -	- -	- -	- -	- -	- -	- -	- -
LEADOF	-	-	-	●	-	-	●	●
LEADON	-	-	-	●	-	-	●	●
LENTOAX	●	●	●	●	●	●	●	●
LFOF	●	●	●	●	●	●	●	●
LFON	●	●	●	●	●	●	●	●
LFPOS	●	●	●	●	●	●	●	●
LFTXT	●	●	●	●	●	●	●	●
LFWP	●	●	●	●	●	●	●	●
LIFTFAST	●	●	●	●	●	●	●	●
LIMS	●	●	●	●	●	●	●	●
LLI	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
LN	●	●	●	●	●	●	●	●
LOCK	●	●	●	●	●	●	●	●
LONGHOLE	●	●	●	●	●	●	●	●
LOOP	●	●	●	●	●	●	●	●

Operations M ... R

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
M0	●	●	●	●	●	●	●	●
M1	●	●	●	●	●	●	●	●
M2	●	●	●	●	●	●	●	●
M3	●	●	●	●	●	●	●	●
M4	●	●	●	●	●	●	●	●
M5	●	●	●	●	●	●	●	●
M6	●	●	●	●	●	●	●	●
M17	●	●	●	●	●	●	●	●
M19	●	●	●	●	●	●	●	●
M30	●	●	●	●	●	●	●	●
M40	●	●	●	●	●	●	●	●
M41 ... M45	●	●	●	●	●	●	●	●
M70	●	●	●	●	●	●	●	●
MASLDEF	-	-	-	-	-	-	-	-
MASLDEL	-	-	-	-	-	-	-	-
MASLOF	-	-	-	-	-	-	-	-
MASLOFS	-	-	-	-	-	-	-	-
MASLON	-	-	-	-	-	-	-	-
MATCH	●	●	●	●	●	●	●	●
MAXVAL	●	●	●	●	●	●	●	●
MCALL	●	●	●	●	●	●	●	●
MEAC	-	-	-	-	-	-	-	-
MEAFRAME	●	●	●	●	●	●	●	●
MEAS	●	●	●	●	●	●	●	●
MEASA	-	-	-	-	-	-	-	-
MEASURE	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
MEAW	●	●	●	●	●	●	●	●
MEAWA	-	-	-	-	-	-	-	-
MI	●	●	●	●	●	●	●	●
MINDEX	●	●	●	●	●	●	●	●
MINVAL	●	●	●	●	●	●	●	●
MIRROR	●	●	●	●	●	●	●	●
MMC	●	●	●	●	●	●	●	●
MOD	●	●	●	●	●	●	●	●
MODAXVAL	●	●	●	●	●	●	●	●
MOV	●	●	●	●	●	●	●	●
MOVT	●	●	●	●	●	●	●	●
MSG	●	●	●	●	●	●	●	●
MVTOOL	●	●	●	●	●	●	●	●
N	●	●	●	●	●	●	●	●
NAMETOINT	●	●	●	●	●	●	●	●
NC	●	●	●	●	●	●	●	●
NEWCONF	●	●	●	●	●	●	●	●
NEWMT	●	●	●	●	●	●	●	●
NEWT	●	●	●	●	●	●	●	●
NORM	●	●	●	●	●	●	●	●
NOT	●	●	●	●	●	●	●	●
NPROT	●	●	●	●	●	●	●	●
NPROTDEF	●	●	●	●	●	●	●	●
NUMBER	●	●	●	●	●	●	●	●
OEMIPO1	-	-	-	-	-	-	-	-
OEMIPO2	-	-	-	-	-	-	-	-
OF	●	●	●	●	●	●	●	●
OFFN	●	●	●	●	●	●	●	●
OMA1	-	-	-	-	-	-	-	-
OMA2	-	-	-	-	-	-	-	-
OMA3	-	-	-	-	-	-	-	-
OMA4	-	-	-	-	-	-	-	-
OMA5	-	-	-	-	-	-	-	-
OR	●	●	●	●	●	●	●	●
ORIAXES	-	-	-	-	-	-	-	-
ORIAXPOS	-	-	-	-	-	-	-	-
ORIC	-	-	-	-	-	-	-	-
ORICONCCW	-	-	-	-	-	-	-	-
ORICONCW	-	-	-	-	-	-	-	-

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
ORICONIO	-	-	-	-	-	-	-	-
ORICONTO	-	-	-	-	-	-	-	-
ORICURVE	-	-	-	-	-	-	-	-
ORID	-	-	-	-	-	-	-	-
ORIEULER	-	-	-	-	-	-	-	-
ORIMKS	-	-	-	-	-	-	-	-
ORIPATH	-	-	-	-	-	-	-	-
ORIPATHS	-	-	-	-	-	-	-	-
ORIPLANE	-	-	-	-	-	-	-	-
ORIRESET	-	-	-	-	-	-	-	-
ORIROTA	-	-	-	-	-	-	-	-
ORIROTC	-	-	-	-	-	-	-	-
ORIROTR	-	-	-	-	-	-	-	-
ORIROTT	-	-	-	-	-	-	-	-
ORIRPY	-	-	-	-	-	-	-	-
ORIRPY2	-	-	-	-	-	-	-	-
ORIS	-	-	-	-	-	-	-	-
ORISOF	-	-	-	-	-	-	-	-
ORISOLH	-	-	-	-	-	-	-	-
ORISON	-	-	-	-	-	-	-	-
ORIVECT	-	-	-	-	-	-	-	-
ORIVIRT1	-	-	-	-	-	-	-	-
ORIVIRT2	-	-	-	-	-	-	-	-
ORIWKS	-	-	-	-	-	-	-	-
OS	-	-	-	-	-	-	-	-
OSB	-	-	-	-	-	-	-	-
OSC	-	-	-	-	-	-	-	-
OSCILL	-	-	-	-	-	-	-	-
OSCTRL	-	-	-	-	-	-	-	-
OSD	-	-	-	-	-	-	-	-
OSE	-	-	-	-	-	-	-	-
OSNSC	●	●	●	●	●	●	●	●
OSOF	-	-	-	-	-	-	-	-
OSP1	●	●	●	●	●	●	●	●
OSP2	●	●	●	●	●	●	●	●
OSS	-	-	-	-	-	-	-	-
OSSE	-	-	-	-	-	-	-	-
OST	-	-	-	-	-	-	-	-
OST1	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
OST2	●	●	●	●	●	●	●	●
OTOL	●	●	●	●	●	●	●	●
OVR	●	●	●	●	●	●	●	●
OVRA	●	●	●	●	●	●	●	●
OVRRAP	●	●	●	●	●	●	●	●
P	●	●	●	●	●	●	●	●
PAROT	●	●	●	●	●	●	●	●
PAROTOF	●	●	●	●	●	●	●	●
PCALL	●	●	●	●	●	●	●	●
PDELAYOF	-	-	-	-	-	-	-	-
PDELAYON	-	-	-	-	-	-	-	-
PHI	-	-	-	-	-	-	-	-
PHU	●	●	●	●	●	●	●	●
PL	-	-	-	-	-	-	-	-
PM	●	●	●	●	●	●	●	●
PO	-	-	-	-	-	-	-	-
POCKET3	●	●	●	●	●	●	●	●
POCKET4	●	●	●	●	●	●	●	●
POLF	●	●	●	●	●	●	●	●
POLFA	●	●	●	●	●	●	●	●
POLFMASK	●	●	●	●	●	●	●	●
POLFMLIN	●	●	●	●	●	●	●	●
POLY	-	-	-	-	-	-	-	-
POLYPATH	-	-	-	-	-	-	-	-
PON	-	-	-	-	-	-	-	-
PONS	-	-	-	-	-	-	-	-
POS	●	●	●	●	●	●	●	●
POSA	●	●	●	●	●	●	●	●
POSM	●	●	●	●	●	●	●	●
POSMT	●	●	●	●	●	●	●	●
POSP	●	●	●	●	●	●	●	●
POSRANGE	●	●	●	●	●	●	●	●
POT	●	●	●	●	●	●	●	●
PR	●	●	●	●	●	●	●	●
PREPRO	●	●	●	●	●	●	●	●
PRESETON	●	●	●	●	●	●	●	●
PRESETONS	●	●	●	●	●	●	●	●
PRIO	●	●	●	●	●	●	●	●
PRLOC	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
PROC	●	●	●	●	●	●	●	●
PROTA	●	●	●	●	●	●	●	●
PROTD	●	●	●	●	●	●	●	●
PROTS	●	●	●	●	●	●	●	●
PSI	-	-	-	-	-	-	-	-
PTP	●	●	●	●	●	●	●	●
PTPG0	●	●	●	●	●	●	●	●
PTPWOC	●	●	●	●	●	●	●	●
PUNCHACC	-	-	-	-	-	-	-	-
PUTFTOC	●	●	●	●	●	●	●	●
PUTFTOCF	●	●	●	●	●	●	●	●
PW	○	○	○	○	○	○	○	○
QU	●	●	●	●	●	●	●	●
R...	●	●	●	●	●	●	●	●
RAC	●	●	●	●	●	●	●	●
RDISABLE	●	●	●	●	●	●	●	●
READ	●	●	●	●	●	●	●	●
REAL	●	●	●	●	●	●	●	●
RELEASE	●	●	●	●	●	●	●	●
REP	●	●	●	●	●	●	●	●
REPEAT	●	●	●	●	●	●	●	●
REPEATB	●	●	●	●	●	●	●	●
REPOSA	●	●	●	●	●	●	●	●
REPOSH	●	●	●	●	●	●	●	●
REPOSHA	●	●	●	●	●	●	●	●
REPOSL	●	●	●	●	●	●	●	●
REPOSQ	●	●	●	●	●	●	●	●
REPOSQA	●	●	●	●	●	●	●	●
RESETMON	●	●	●	●	●	●	●	●
RET	●	●	●	●	●	●	●	●
RETB	●	●	●	●	●	●	●	●
RIC	●	●	●	●	●	●	●	●
RINDEX	●	●	●	●	●	●	●	●
RMB	●	●	●	●	●	●	●	●
RME	●	●	●	●	●	●	●	●
RMI	●	●	●	●	●	●	●	●
RMN	●	●	●	●	●	●	●	●
RND	●	●	●	●	●	●	●	●
RNDM	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
ROT	●	●	●	●	●	●	●	●
ROTS	●	●	●	●	●	●	●	●
ROUND	●	●	●	●	●	●	●	●
ROUNDUP	●	●	●	●	●	●	●	●
RP	●	●	●	●	●	●	●	●
RPL	●	●	●	●	●	●	●	●
RT	●	●	●	●	●	●	●	●
RTLIOF	●	●	●	●	●	●	●	●
RTLION	●	●	●	●	●	●	●	●

Operations S ... Z

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
S	●	●	●	●	●	●	●	●
SAVE	●	●	●	●	●	●	●	●
SBLOF	●	●	●	●	●	●	●	●
SBLON	●	●	●	●	●	●	●	●
SC	●	●	●	●	●	●	●	●
SCALE	●	●	●	●	●	●	●	●
SCC	●	●	●	●	●	●	●	●
SCPARA	●	●	●	●	●	●	●	●
SD	○	○	○	○	○	○	○	○
SET	●	●	●	●	●	●	●	●
SETAL	●	●	●	●	●	●	●	●
SETDNO	●	●	●	●	●	●	●	●
SETINT	●	●	●	●	●	●	●	●
SETM	-	-	-	-	-	○	-	●
SETMS	●	●	●	●	●	●	●	●
SETMS(n)	●	●	●	●	●	●	●	●
SETMTH	●	●	●	●	●	●	●	●
SETPIECE	●	●	●	●	●	●	●	●
SETTA	●	●	●	●	●	●	●	●
SETTCOR	●	●	●	●	●	●	●	●
SETTIA	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
SF	●	●	●	●	●	●	●	●
SIN	●	●	●	●	●	●	●	●
SIRELAY	-	-	-	-	-	-	-	-
SIRELIN	-	-	-	-	-	-	-	-
SIRELOUT	-	-	-	-	-	-	-	-
SIRELTIME	-	-	-	-	-	-	-	-
SLOT1	●	●	●	●	●	●	●	●
SLOT2	●	●	●	●	●	●	●	●
SOFT	●	●	●	●	●	●	●	●
SOFTA	●	●	●	●	●	●	●	●
SON	-	-	-	-	-	-	-	-
SONS	-	-	-	-	-	-	-	-
SPATH	●	●	●	●	●	●	●	●
SPCOF	●	●	●	●	●	●	●	●
SPCON	●	●	●	●	●	●	●	●
SPI	●	●	●	●	●	●	●	●
SPIF1	-	-	-	-	-	-	-	-
SPIF2	-	-	-	-	-	-	-	-
SPLINEPATH	○	○	○	○	○	○	○	○
SPN	-	-	-	-	-	-	-	-
SPOF	-	-	-	-	-	-	-	-
SPOS	●	●	●	●	●	●	●	●
SPOSA	●	●	●	●	●	●	●	●
SPP	-	-	-	-	-	-	-	-
SPRINT	●	●	●	●	●	●	●	●
SQRT	●	●	●	●	●	●	●	●
SR	●	●	●	●	●	●	●	●
SRA	●	●	●	●	●	●	●	●
ST	●	●	●	●	●	●	●	●
STA	●	●	●	●	●	●	●	●
START	-	-	-	-	-	○	-	●
STARTFIFO	●	●	●	●	●	●	●	●
STAT	●	●	●	●	●	●	●	●
STOLF	●	●	●	●	●	●	●	●
STOPFIFO	●	●	●	●	●	●	●	●
STOPRE	●	●	●	●	●	●	●	●
STOPREOF	●	●	●	●	●	●	●	●
STRING	●	●	●	●	●	●	●	●
STRINGFELD	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
STRINGIS	●	●	●	●	●	●	●	●
STRLEN	●	●	●	●	●	●	●	●
SUBSTR	●	●	●	●	●	●	●	●
SUPA	●	●	●	●	●	●	●	●
SVC	●	●	●	●	●	●	●	●
SYNFCT	●	●	●	●	●	●	●	●
SYNR	●	●	●	●	●	●	●	●
SYNRW	●	●	●	●	●	●	●	●
SYNW	●	●	●	●	●	●	●	●
T	●	●	●	●	●	●	●	●
TAN	●	●	●	●	●	●	●	●
TANG	-	-	-	-	-	-	-	-
TANGDEL	-	-	-	-	-	-	-	-
TANGOF	-	-	-	-	-	-	-	-
TANGON	-	-	-	-	-	-	-	-
TCA (828D: _TCA)	●	●	●	●	●	●	●	●
TCARR	●	-	●	-	●	●	-	-
TCI	●	●	●	●	●	●	●	●
TCOABS	●	-	●	-	●	●	-	-
TCOFR	●	-	●	-	●	●	-	-
TCOFRX	●	-	●	-	●	●	-	-
TCOFRY	●	-	●	-	●	●	-	-
TCOFRZ	●	-	●	-	●	●	-	-
THETA	-	-	-	-	-	-	-	-
TILT	-	-	-	-	-	-	-	-
TLIFT	-	-	-	-	-	-	-	-
TML	●	●	●	●	●	●	●	●
TMOF	●	●	●	●	●	●	●	●
TMON	●	●	●	●	●	●	●	●
TO	●	●	●	●	●	●	●	●
TOFF	●	●	●	●	●	●	●	●
TOFFL	●	●	●	●	●	●	●	●
TOFFOF	●	●	●	●	●	●	●	●
TOFFON	●	●	●	●	●	●	●	●
TOFFR	●	●	●	●	●	●	●	●
TOFRAME	●	●	●	●	●	●	●	●
TOFRAMEX	●	●	●	●	●	●	●	●
TOFRAMEY	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
TOFRAMEZ	●	●	●	●	●	●	●	●
TOLOWER	●	●	●	●	●	●	●	●
TOOLENV	●	●	●	●	●	●	●	●
TOOLGNT	●	●	●	●	●	●	●	●
TOOLGT	●	●	●	●	●	●	●	●
TOROT	●	●	●	●	●	●	●	●
TOROTOF	●	●	●	●	●	●	●	●
TOROTX	●	●	●	●	●	●	●	●
TOROTY	●	●	●	●	●	●	●	●
TOROTZ	●	●	●	●	●	●	●	●
TOUPPER	●	●	●	●	●	●	●	●
TOWBCS	●	-	●	-	●	●	-	-
TOWKCS	●	-	●	-	●	●	-	-
TOWMCS	●	-	●	-	●	●	-	-
TOWSTD	●	-	●	-	●	●	-	-
TOWTCS	●	-	●	-	●	●	-	-
TOWWCS	●	-	●	-	●	●	-	-
TR	●	●	●	●	●	●	●	●
TRAANG	-	-	-	-	-	-	○	○
TRACON	-	-	-	-	-	-	○	○
TRACYL	○	○	○	○	○	○	○	○
TRAFOOF	●	●	●	●	●	●	●	●
TRAFOON	-	-	-	-	-	-	-	-
TRAILOF	●	●	●	●	●	●	●	●
TRAILON	●	●	●	●	●	●	●	●
TRANS	●	●	●	●	●	●	●	●
TRANSMIT	○	○	○	○	○	○	○	○
TRAORI	-	-	-	-	-	-	-	-
TRUE	●	●	●	●	●	●	●	●
TRUNC	●	●	●	●	●	●	●	●
TU	●	●	●	●	●	●	●	●
TURN	●	●	●	●	●	●	●	●
ULI	●	●	●	●	●	●	●	●
UNLOCK	●	●	●	●	●	●	●	●
UNTIL	●	●	●	●	●	●	●	●
UPATH	●	●	●	●	●	●	●	●
VAR	●	●	●	●	●	●	●	●
VELOLIM	●	●	●	●	●	●	●	●
VELOLIMA	●	●	●	●	●	●	●	●

Operation	SINUMERIK 828D							
● Standard ○ Option - not available	SW24x(5) CNC-SW Milling Export (me42)	SW24x(5) CNC-SW Turning Export (te42)	SW26x(3) CNC-SW Milling Export (me62)	SW26x(3) CNC-SW Turning Export (te62)	SW28x(2) CNC-SW Milling Export (me821)	SW28x(1) CNC-SW Milling Adv. Export (me822)	SW28x(2) CNC-SW Turning Export (te821)	SW28x(1) CNC-SW Turning Adv. Export (te822)
WAITC	●	●	●	●	●	●	●	●
WAITE	-	-	-	-	-	○	-	●
WAITENC	●	●	●	●	●	●	●	●
WAITM	-	-	-	-	-	○	-	●
WAITMC	-	-	-	-	-	○	-	●
WAITP	●	●	●	●	●	●	●	●
WAITS	●	●	●	●	●	●	●	●
WALCS0	●	●	●	●	●	●	●	●
WALCS1	●	●	●	●	●	●	●	●
WALCS2	●	●	●	●	●	●	●	●
WALCS3	●	●	●	●	●	●	●	●
WALCS4	●	●	●	●	●	●	●	●
WALCS5	●	●	●	●	●	●	●	●
WALCS6	●	●	●	●	●	●	●	●
WALCS7	●	●	●	●	●	●	●	●
WALCS8	●	●	●	●	●	●	●	●
WALCS9	●	●	●	●	●	●	●	●
WALCS10	●	●	●	●	●	●	●	●
WALIMOF	●	●	●	●	●	●	●	●
WALIMON	●	●	●	●	●	●	●	●
WHEN	●	●	●	●	●	●	●	●
WHENEVER	●	●	●	●	●	●	●	●
WHILE	●	●	●	●	●	●	●	●
WORKPIECE	●	●	●	●	●	●	●	●
WRITE	●	●	●	●	●	●	●	●
WRTPR	●	●	●	●	●	●	●	●
X	●	●	●	●	●	●	●	●
XOR	●	●	●	●	●	●	●	●
Y	●	●	●	●	●	●	●	●
Z	●	●	●	●	●	●	●	●

17.2.2 Control versions grinding

Operations A ... C

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
:	●	●	●	●	●	●
*	●	●	●	●	●	●
+	●	●	●	●	●	●
-	●	●	●	●	●	●
<	●	●	●	●	●	●
<<	●	●	●	●	●	●
<=	●	●	●	●	●	●
=	●	●	●	●	●	●
>=	●	●	●	●	●	●
/	●	●	●	●	●	●
/0 … /7	●	●	●	●	●	●
A	●	●	●	●	●	●
A2	-	-	-	-	-	-
A3	-	-	-	-	-	-
A4	-	-	-	-	-	-
A5	-	-	-	-	-	-
A6	-	-	-	-	-	-
A7	-	-	-	-	-	-
ABS	●	●	●	●	●	●
AC	●	●	●	●	●	●
ACC	●	●	●	●	●	●
ACCLIMA	●	●	●	●	●	●
ACN	●	●	●	●	●	●
ACOS	●	●	●	●	●	●
ACP	●	●	●	●	●	●
ACTBLOCNO	●	●	●	●	●	●
ADDFRAME	●	●	●	●	●	●
ADIS	●	●	●	●	●	●
ADISPOS	●	●	●	●	●	●
ADISPOSA	●	●	●	●	●	●
ALF	●	●	●	●	●	●
AMIRROR	●	●	●	●	●	●
AND	●	●	●	●	●	●
ANG	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
AP	●	●	●	●	●	●
APR	●	●	●	●	●	●
APRB	●	●	●	●	●	●
APRP	●	●	●	●	●	●
APW	●	●	●	●	●	●
APWB	●	●	●	●	●	●
APWP	●	●	●	●	●	●
APX	●	●	●	●	●	●
AR	●	●	●	●	●	●
AROT	●	●	●	●	●	●
AROTS	●	●	●	●	●	●
AS	●	●	●	●	●	●
ASCALE	●	●	●	●	●	●
ASIN	●	●	●	●	●	●
ASPLINE	○	○	○	○	○	○
ATAN2	●	●	●	●	●	●
ATOL	●	●	●	●	●	●
ATRANS	●	●	●	●	●	●
AUXFUDEL	●	●	●	●	●	●
AUXFUDELG	●	●	●	●	●	●
AUXFUMSEQ	●	●	●	●	●	●
AUXFUSYNC	●	●	●	●	●	●
AX	●	●	●	●	●	●
AXCTSWE	-	-	-	-	-	-
AXCTSWEC	-	-	-	-	-	-
AXCTSWED	-	-	-	-	-	-
AXIS	●	●	●	●	●	●
AXNAME	●	●	●	●	●	●
AXSTRING	●	●	●	●	●	●
AXTOCHAN	●	●	●	●	●	●
AXTOSPI	●	●	●	●	●	●
B	●	●	●	●	●	●
B2	-	-	-	-	-	-
B3	-	-	-	-	-	-
B4	-	-	-	-	-	-
B5	-	-	-	-	-	-
B6	-	-	-	-	-	-
B7	-	-	-	-	-	-
B_AND	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
B_OR	●	●	●	●	●	●
B_NOT	●	●	●	●	●	●
B_XOR	●	●	●	●	●	●
BAUTO	○	○	○	○	○	○
BLOCK	●	●	●	●	●	●
BLSYNC	●	●	●	●	●	●
BNAT	○	○	○	○	○	○
BOOL	●	●	●	●	●	●
BOUND	●	●	●	●	●	●
BRISK	●	●	●	●	●	●
BRISKA	●	●	●	●	●	●
BSPLINE	○	○	○	○	○	○
BTAN	○	○	○	○	○	○
C	●	●	●	●	●	●
C2	-	-	Channel axis name	-	-	-
C3	-	-	-	-	-	-
C4	-	-	-	-	-	-
C5	-	-	-	-	-	-
C6	-	-	-	-	-	-
C7	-	-	-	-	-	-
CAC	●	●	●	●	●	●
CACN	●	●	●	●	●	●
CACP	●	●	●	●	●	●
CALCDAT	●	●	●	●	●	●
CALCPOSI	●	●	●	●	●	●
CALL	●	●	●	●	●	●
CALLPATH	●	●	●	●	●	●
CANCEL	●	●	●	●	●	●
CASE	●	●	●	●	●	●
CDC	●	●	●	●	●	●
CDOF	-	-	-	-	-	-
CDOF2	-	-	-	-	-	-
CDON	-	-	-	-	-	-
CFC	●	●	●	●	●	●
CFIN	●	●	●	●	●	●
CFINE	●	●	●	●	●	●
CFTCP	●	●	●	●	●	●
CHAN	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
CHANDATA	●	●	●	●	●	●
CHAR	●	●	●	●	●	●
CHF	●	●	●	●	●	●
CHKDM	●	●	●	●	●	●
CHKDNO	●	●	●	●	●	●
CHR	●	●	●	●	●	●
CIC	●	●	●	●	●	●
CIP	●	●	●	●	●	●
CLEARM	-	-	●	-	-	●
CLRINT	●	●	●	●	●	●
CMIRROR	●	●	●	●	●	●
COARSEA	●	●	●	●	●	●
COLLPAIR	-	-	-	-	-	-
COMPCAD	●	●	●	●	●	●
COMPCURV	●	●	●	●	●	●
COMPLETE	●	●	●	●	●	●
COMPOF	●	●	●	●	●	●
COMPON	●	●	●	●	●	●
COMPSURF	-	-	-	-	-	-
CONTDCON	●	●	●	●	●	●
CONTPRON	●	●	●	●	●	●
CORROF	●	●	●	●	●	●
CORRTRAFO	-	-	-	-	-	-
COS	●	●	●	●	●	●
COUPDEF	○	○	○	○	○	○
COUPDEL	○	○	○	○	○	○
COUPOF	○	○	○	○	○	○
COUPOFS	○	○	○	○	○	○
COUPON	○	○	○	○	○	○
COUPONC	○	○	○	○	○	○
COUPRES	○	○	○	○	○	○
CP	●	●	●	●	●	●
CPBC	○	○	○	○	○	○
CPDEF	○	○	○	○	○	○
CPDEL	○	○	○	○	○	○
CPFMOF	○	○	○	○	○	○
CPFMON	○	○	○	○	○	○
CPFMSON	○	○	○	○	○	○
CPFPOS	○	○	○	○	○	○

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
CPFRS	○	○	○	○	○	○
CPLA	○	○	○	○	○	○
CPLCTID	○	○	○	○	○	○
CPLDEF	○	○	○	○	○	○
CPLDEL	○	○	○	○	○	○
CPLDEN	○	○	○	○	○	○
CPLINSC	○	○	○	○	○	○
CPLINTR	○	○	○	○	○	○
CPLNUM	○	○	○	○	○	○
CPLOF	○	○	○	○	○	○
CPLON	○	○	○	○	○	○
CPLOUTSC	○	○	○	○	○	○
CPLOUTTR	○	○	○	○	○	○
CPLPOS	○	○	○	○	○	○
CPLSETVAL	○	○	○	○	○	○
CPMALARM	○	○	○	○	○	○
CPMBRAKE	○	○	○	○	○	○
CPMPRT	○	○	○	○	○	○
CPMRESET	○	○	○	○	○	○
CPMSTART	○	○	○	○	○	○
CPMVDI	○	○	○	○	○	○
CPOF	○	○	○	○	○	○
CPON	○	○	○	○	○	○
CPRECOF	●	●	●	●	●	●
CPRECON	●	●	●	●	●	●
CPRES	○	○	○	○	○	○
CPROT	●	●	●	●	●	●
CPROTDEF	●	●	●	●	●	●
CPSETTYPE	○	○	○	○	○	○
CPSYNCOP	○	○	○	○	○	○
CPSYNCOP2	○	○	○	○	○	○
CPSYNCOV	○	○	○	○	○	○
CPSYNFIP	○	○	○	○	○	○
CPSYNFIP2	○	○	○	○	○	○
CPSYNFIV	○	○	○	○	○	○
CR	●	●	●	●	●	●
CROT	●	●	●	●	●	●
CROTS	●	●	●	●	●	●
CRPL	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
CSCALE	●	●	●	●	●	●
CSPLINE	○	○	○	○	○	○
CT	●	●	●	●	●	●
CTAB	-	-	-	-	-	-
CTABDEF	-	-	-	-	-	-
CTABDEL	-	-	-	-	-	-
CTABEND	-	-	-	-	-	-
CTABEXISTS	-	-	-	-	-	-
CTABFNO	-	-	-	-	-	-
CTABFPOL	-	-	-	-	-	-
CTABFSEG	-	-	-	-	-	-
CTABID	-	-	-	-	-	-
CTABINV	-	-	-	-	-	-
CTABISLOCK	-	-	-	-	-	-
CTABLOCK	-	-	-	-	-	-
CTABMEMTYP	-	-	-	-	-	-
CTABMPOL	-	-	-	-	-	-
CTABMSEG	-	-	-	-	-	-
CTABNO	-	-	-	-	-	-
CTABNOMEM	-	-	-	-	-	-
CTABPERIOD	-	-	-	-	-	-
CTABPOL	-	-	-	-	-	-
CTABPOLID	-	-	-	-	-	-
CTABSEG	-	-	-	-	-	-
CTABSEGID	-	-	-	-	-	-
CTABSEV	-	-	-	-	-	-
CTABSSV	-	-	-	-	-	-
CTABTEP	-	-	-	-	-	-
CTABTEV	-	-	-	-	-	-
CTABTMAX	-	-	-	-	-	-
CTABTMIN	-	-	-	-	-	-
CTABTSP	-	-	-	-	-	-
CTABTSV	-	-	-	-	-	-
CTABUNLOCK	-	-	-	-	-	-
CTOL	●	●	●	●	●	●
CTRANS	●	●	●	●	●	●
CUT2D	●	●	●	●	●	●
CUT2DD	●	●	●	●	●	●
CUT2DF	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
CUT2DFD	●	●	●	●	●	●
CUT3DC	-	-	-	-	-	-
CUT3DCC	-	-	-	-	-	-
CUT3DCCD	-	-	-	-	-	-
CUT3DCD	-	-	-	-	-	-
CUT3DF	-	-	-	-	-	-
CUT3DFD	-	-	-	-	-	-
CUT3DFF	-	-	-	-	-	-
CUT3DFS	-	-	-	-	-	-
CUTCONOF	●	●	●	●	●	●
CUTCONON	●	●	●	●	●	●
CUTMOD	●	●	●	●	●	●
CUTMODK	-	-	-	-	-	-
CYCLE60	-	-	-	-	-	-
CYCLE61	-	-	-	-	-	-
CYCLE62	●	●	●	●	●	●
CYCLE63	-	-	-	-	-	-
CYCLE64	-	-	-	-	-	-
CYCLE70	-	-	-	-	-	-
CYCLE72	-	-	-	-	-	-
CYCLE76	-	-	-	-	-	-
CYCLE77	-	-	-	-	-	-
CYCLE78	-	-	-	-	-	-
CYCLE79	-	-	-	-	-	-
CYCLE81	-	-	-	-	-	-
CYCLE82	-	-	-	-	-	-
CYCLE83	-	-	-	-	-	-
CYCLE84	-	-	-	-	-	-
CYCLE85	-	-	-	-	-	-
CYCLE86	-	-	-	-	-	-
CYCLE92	-	-	-	-	-	-
CYCLE95	-	-	-	-	-	-
CYCLE98	-	-	-	-	-	-
CYCLE99	-	-	-	-	-	-
CYCLE150	-	-	-	-	-	-
CYCLE435	○	○	○	○	○	○
CYCLE495	○	○	○	○	○	○
CYCLE750	●	●	●	●	●	●
CYCLE751	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
CYCLE752	●	●	●	●	●	●
CYCLE753	●	●	●	●	●	●
CYCLE754	●	●	●	●	●	●
CYCLE755	●	●	●	●	●	●
CYCLE756	●	●	●	●	●	●
CYCLE757	●	●	●	●	●	●
CYCLE758	●	●	●	●	●	●
CYCLE759	●	●	●	●	●	●
CYCLE800	○	○	○	○	○	○
CYCLE801	-	-	-	-	-	-
CYCLE802	-	-	-	-	-	-
CYCLE830	-	-	-	-	-	-
CYCLE832	●	●	●	●	●	●
CYCLE840	-	-	-	-	-	-
CYCLE899	-	-	-	-	-	-
CYCLE930	-	-	-	-	-	-
CYCLE940	-	-	-	-	-	-
CYCLE951	-	-	-	-	-	-
CYCLE952	-	-	-	-	-	-
CYCLE961	-	-	-	-	-	-
CYCLE971	-	-	-	-	-	-
CYCLE973	-	-	-	-	-	-
CYCLE974	-	-	-	-	-	-
CYCLE976	-	-	-	-	-	-
CYCLE977	-	-	-	-	-	-
CYCLE978	-	-	-	-	-	-
CYCLE979	-	-	-	-	-	-
CYCLE982	-	-	-	-	-	-
CYCLE994	-	-	-	-	-	-
CYCLE995	-	-	-	-	-	-
CYCLE996	-	-	-	-	-	-
CYCLE997	-	-	-	-	-	-
CYCLE998	-	-	-	-	-	-
CYCLE4071	●	●	●	●	●	●
CYCLE4072	●	●	●	●	●	●
CYCLE4073	●	●	●	●	●	●
CYCLE4074	●	●	●	●	●	●
CYCLE4075	●	●	●	●	●	●
CYCLE4077	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
CYCLE4078	●	●	●	●	●	●
CYCLE4079	●	●	●	●	●	●

Operations D ... F

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
D	●	●	●	●	●	●
D0	●	●	●	●	●	●
DAC	●	●	●	●	●	●
DC	●	●	●	●	●	●
DCI	●	●	●	●	●	●
DCM	●	●	●	●	●	●
DCU	●	●	●	●	●	●
DEF	●	●	●	●	●	●
DEFINE	●	●	●	●	●	●
DEFAULT	●	●	●	●	●	●
DELAYFSTON	●	●	●	●	●	●
DELAYFSTOF	●	●	●	●	●	●
DELDL	●	●	●	●	●	●
DELDTG	●	●	●	●	●	●
DELETE	●	●	●	●	●	●
DELMLOWNER	●	●	●	●	●	●
DELMLRES	●	●	●	●	●	●
DELMT	-	-	-	-	-	-
DELOBJ	-	-	-	-	-	-
DELT	●	●	●	●	●	●
DELTC	●	●	●	●	●	●
DELTOOLENV	●	●	●	●	●	●
DIACYCOFA	●	●	●	●	●	●
DIAM90	●	●	●	●	●	●
DIAM90A	●	●	●	●	●	●
DIAMCHAN	●	●	●	●	●	●
DIAMCHANA	●	●	●	●	●	●
DIAMCYCOF	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
DIAMOF	●	●	●	●	●	●
DIAMOFA	●	●	●	●	●	●
DIAMON	●	●	●	●	●	●
DIAMONA	●	●	●	●	●	●
DIC	●	●	●	●	●	●
DILF	●	●	●	●	●	●
DISABLE	●	●	●	●	●	●
DISC	●	●	●	●	●	●
DISCL	●	●	●	●	●	●
DISPLOF	●	●	●	●	●	●
DISPLON	●	●	●	●	●	●
DISPR	●	●	●	●	●	●
DISR	●	●	●	●	●	●
DISRP	●	●	●	●	●	●
DITE	●	●	●	●	●	●
DITS	●	●	●	●	●	●
DIV	●	●	●	●	●	●
DL	-	-	-	-	-	-
DO	●	●	●	●	●	●
DRFOF	●	●	●	●	●	●
DRIVE	●	●	●	●	●	●
DRIVEA	●	●	●	●	●	●
DYNFINISH	●	●	●	●	●	●
DYNNORM	●	●	●	●	●	●
DYNPOS	●	●	●	●	●	●
DYNROUGH	●	●	●	●	●	●
DYNSEMIFIN	●	●	●	●	●	●
DZERO	●	●	●	●	●	●
EAUTO	○	○	○	○	○	○
EGDEF	○	○	○	○	○	○
EGDEL	○	○	○	○	○	○
EGOFC	○	○	○	○	○	○
EGOFS	○	○	○	○	○	○
EGON	○	○	○	○	○	○
EGONSYN	○	○	○	○	○	○
EGONSYNE	○	○	○	○	○	○
ELSE	●	●	●	●	●	●
ENABLE	●	●	●	●	●	●
ENAT	○	○	○	○	○	○

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
ENDFOR	●	●	●	●	●	●
ENDIF	●	●	●	●	●	●
ENDLABEL	●	●	●	●	●	●
ENDLOOP	●	●	●	●	●	●
ENDPROC	●	●	●	●	●	●
ENDWHILE	●	●	●	●	●	●
ESRR	○	○	○	○	○	○
ESRS	○	○	○	○	○	○
ETAN	○	○	○	○	○	○
EVERY	●	●	●	●	●	●
EX	●	●	●	●	●	●
EXECSTRING	●	●	●	●	●	●
EXECTAB	●	●	●	●	●	●
EXECUTE	●	●	●	●	●	●
EXP	●	●	●	●	●	●
EXTCALL	●	●	●	●	●	●
EXTCLOSE	●	●	●	●	●	●
EXTERN	●	●	●	●	●	●
EXTOPEN	●	●	●	●	●	●
F	●	●	●	●	●	●
FA	●	●	●	●	●	●
FAD	●	●	●	●	●	●
FALSE	●	●	●	●	●	●
FB	●	●	●	●	●	●
FCTDEF	●	●	●	●	●	●
FCUB	●	●	●	●	●	●
FD	●	●	●	●	●	●
FDA	●	●	●	●	●	●
FENDNORM	●	●	●	●	●	●
FFWOF	●	●	●	●	●	●
FFWON	●	●	●	●	●	●
FGREF	●	●	●	●	●	●
FGROUP	●	●	●	●	●	●
FI	●	●	●	●	●	●
FIFOCTRL	●	●	●	●	●	●
FILEDATE	●	●	●	●	●	●
FILEINFO	●	●	●	●	●	●
FILESIZE	●	●	●	●	●	●
FILESTAT	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
FILETIME	●	●	●	●	●	●
FINEA	●	●	●	●	●	●
FL	●	●	●	●	●	●
FLIN	●	●	●	●	●	●
FMA	●	●	●	●	●	●
FNORM	●	●	●	●	●	●
FOCOF	○	○	○	○	○	○
FOCON	○	○	○	○	○	○
FOR	●	●	●	●	●	●
FP	●	●	●	●	●	●
FPO	-	-	-	-	-	-
FPR	●	●	●	●	●	●
FPRAOF	●	●	●	●	●	●
FPRAON	●	●	●	●	●	●
FRAME	●	●	●	●	●	●
FRC	●	●	●	●	●	●
FRCM	●	●	●	●	●	●
FROM	●	●	●	●	●	●
FTOC	●	●	●	●	●	●
FTOCOF	●	●	●	●	●	●
FTOCON	●	●	●	●	●	●
FXS	●	●	●	●	●	●
FXST	●	●	●	●	●	●
FXSW	●	●	●	●	●	●
FZ	●	●	●	●	●	●

Operations G ... L

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
G0	●	●	●	●	●	●
G1	●	●	●	●	●	●
G2	●	●	●	●	●	●
G3	●	●	●	●	●	●
G4	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
G5	●	●	●	●	●	●
G7	●	●	●	●	●	●
G9	●	●	●	●	●	●
G17	●	●	●	●	●	●
G18	●	●	●	●	●	●
G19	●	●	●	●	●	●
G25	●	●	●	●	●	●
G26	●	●	●	●	●	●
G33	●	●	●	●	●	●
G34	●	●	●	●	●	●
G35	●	●	●	●	●	●
G40	●	●	●	●	●	●
G41	●	●	●	●	●	●
G42	●	●	●	●	●	●
G53	●	●	●	●	●	●
G54	●	●	●	●	●	●
G55	●	●	●	●	●	●
G56	●	●	●	●	●	●
G57	●	●	●	●	●	●
G58	→ G505					
G59	→ G506					
G60	●	●	●	●	●	●
G62	●	●	●	●	●	●
G63	●	●	●	●	●	●
G64	●	●	●	●	●	●
G70	●	●	●	●	●	●
G71	●	●	●	●	●	●
G74	●	●	●	●	●	●
G75	●	●	●	●	●	●
G90	●	●	●	●	●	●
G91	●	●	●	●	●	●
G93	●	●	●	●	●	●
G94	●	●	●	●	●	●
G95	●	●	●	●	●	●
G96	●	●	●	●	●	●
G97	●	●	●	●	●	●
G110	●	●	●	●	●	●
G111	●	●	●	●	●	●
G112	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
G140	●	●	●	●	●	●
G141	●	●	●	●	●	●
G142	●	●	●	●	●	●
G143	●	●	●	●	●	●
G147	●	●	●	●	●	●
G148	●	●	●	●	●	●
G153	●	●	●	●	●	●
G247	●	●	●	●	●	●
G248	●	●	●	●	●	●
G290	●	●	●	●	●	●
G291	-	-	-	-	-	-
G331	●	●	●	●	●	●
G332	●	●	●	●	●	●
G335	●	●	●	●	●	●
G336	●	●	●	●	●	●
G340	●	●	●	●	●	●
G341	●	●	●	●	●	●
G347	●	●	●	●	●	●
G348	●	●	●	●	●	●
G450	●	●	●	●	●	●
G451	●	●	●	●	●	●
G460	●	●	●	●	●	●
G461	●	●	●	●	●	●
G462	●	●	●	●	●	●
G500	●	●	●	●	●	●
G505 ... G599	●	●	●	●	●	●
G601	●	●	●	●	●	●
G602	●	●	●	●	●	●
G603	●	●	●	●	●	●
G621	●	●	●	●	●	●
G641	●	●	●	●	●	●
G642	●	●	●	●	●	●
G643	●	●	●	●	●	●
G644	●	●	●	●	●	●
G645	●	●	●	●	●	●
G700	●	●	●	●	●	●
G710	●	●	●	●	●	●
G810 ... G819	-	-	-	-	-	-
G820 ... G829	-	-	-	-	-	-

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
G931	●	●	●	●	●	●
G942	●	●	●	●	●	●
G952	●	●	●	●	●	●
G961	●	●	●	●	●	●
G962	●	●	●	●	●	●
G971	●	●	●	●	●	●
G972	●	●	●	●	●	●
G973	●	●	●	●	●	●
GEOAX	●	●	●	●	●	●
GET	●	●	●	●	●	●
GETACTT	●	●	●	●	●	●
GETACTTD	●	●	●	●	●	●
GETD	-	-	●	-	-	●
GETDNO	●	●	●	●	●	●
GETEXET	●	●	●	●	●	●
GETFREELOC	●	●	●	●	●	●
GETSELT	●	●	●	●	●	●
GETT	●	●	●	●	●	●
GETTCOR	●	●	●	●	●	●
GETTENV	●	●	●	●	●	●
GETVARAP	●	●	●	●	●	●
GETVARDFT	●	●	●	●	●	●
GETVARLIM	●	●	●	●	●	●
GETVARPHU	●	●	●	●	●	●
GETVARTYP	●	●	●	●	●	●
GFRAME0 ... GFRAME100	< 50	< 100	< 100	< 50	< 100	< 100
GOTO	●	●	●	●	●	●
GOTOB	●	●	●	●	●	●
GOTOC	●	●	●	●	●	●
GOTOF	●	●	●	●	●	●
GOTOS	●	●	●	●	●	●
GP	●	●	●	●	●	●
GWPSOF	●	●	●	●	●	●
GROUP_ADDEND	●	●	●	●	●	●
GROUP_BEGIN	●	●	●	●	●	●
GROUP_END	●	●	●	●	●	●
GWPSON	●	●	●	●	●	●
H...	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
HOLES1	-	-	-	-	-	-
HOLES2	-	-	-	-	-	-
I	●	●	●	●	●	●
I1	●	●	●	●	●	●
IC	●	●	●	●	●	●
ICYCOF	●	●	●	●	●	●
ICYCON	●	●	●	●	●	●
ID	●	●	●	●	●	●
IDS	●	●	●	●	●	●
IF	●	●	●	●	●	●
INDEX	●	●	●	●	●	●
INIPO	●	●	●	●	●	●
INIRE	●	●	●	●	●	●
INICF	●	●	●	●	●	●
INIT	-	-	●	-	-	●
INITIAL						
INT	●	●	●	●	●	●
INTERSEC	●	●	●	●	●	●
INVCCW	-	-	-	-	-	-
INVCW	-	-	-	-	-	-
INVFRAME	●	●	●	●	●	●
IP	●	●	●	●	●	●
IPOBRKA	●	●	●	●	●	●
IPOENDA	●	●	●	●	●	●
IPTRLOCK	●	●	●	●	●	●
IPTRUNLOCK	●	●	●	●	●	●
IR	●	●	●	●	●	●
ISAXIS	●	●	●	●	●	●
ISD	-	-	-	-	-	-
ISFILE	●	●	●	●	●	●
ISNUMBER	●	●	●	●	●	●
ISOCALL	-	-	-	-	-	-
ISVAR	●	●	●	●	●	●
J	●	●	●	●	●	●
J1	●	●	●	●	●	●
JERKA	●	●	●	●	●	●
JERKLIM	●	●	●	●	●	●
JERKLIMA	●	●	●	●	●	●
JR	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
K	●	●	●	●	●	●
K1	●	●	●	●	●	●
KONT	●	●	●	●	●	●
KONTC	●	●	●	●	●	●
KONTT	●	●	●	●	●	●
KR	●	●	●	●	●	●
L	●	●	●	●	●	●
LEAD						
Tool orientation	-	-	-	-	-	-
Orientation polynomial	-	-	-	-	-	-
LEADOF	-	-	-	-	-	-
LEADON	-	-	-	-	-	-
LENTOAX	●	●	●	●	●	●
LFOF	●	●	●	●	●	●
LFON	●	●	●	●	●	●
LFPOS	●	●	●	●	●	●
LFTXT	●	●	●	●	●	●
LFWP	●	●	●	●	●	●
LIFTFAST	●	●	●	●	●	●
LIMS	●	●	●	●	●	●
LLI	●	●	●	●	●	●
LN	●	●	●	●	●	●
LOCK	●	●	●	●	●	●
LONGHOLE	-	-	-	-	-	-
LOOP	●	●	●	●	●	●

Operations M ... R

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
M0	●	●	●	●	●	●
M1	●	●	●	●	●	●
M2	●	●	●	●	●	●
M3	●	●	●	●	●	●
M4	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
M5	●	●	●	●	●	●
M6	●	●	●	●	●	●
M17	●	●	●	●	●	●
M19	●	●	●	●	●	●
M30	●	●	●	●	●	●
M40	●	●	●	●	●	●
M41 ... M45	●	●	●	●	●	●
M70	●	●	●	●	●	●
MASLDEF	-	-	-	-	-	-
MASLDEL	-	-	-	-	-	-
MASLOF	-	-	-	-	-	-
MASLOFS	-	-	-	-	-	-
MASLON	-	-	-	-	-	-
MATCH	●	●	●	●	●	●
MAXVAL	●	●	●	●	●	●
MCALL	●	●	●	●	●	●
MEAC	-	○	○	-	○	○
MEAFRAME	●	●	●	●	●	●
MEAS	●	●	●	●	●	●
MEASA	-	○	○	-	○	○
MEASURE	●	●	●	●	●	●
MEAW	●	●	●	●	●	●
MEAWA	-	○	○	-	○	○
MI	●	●	●	●	●	●
MINDEX	●	●	●	●	●	●
MINVAL	●	●	●	●	●	●
MIRROR	●	●	●	●	●	●
MMC	●	●	●	●	●	●
MOD	●	●	●	●	●	●
MODAXVAL	●	●	●	●	●	●
MOV	●	●	●	●	●	●
MOVT	●	●	●	●	●	●
MSG	●	●	●	●	●	●
MVTOOL	●	●	●	●	●	●
N	●	●	●	●	●	●
NAMETOINT	●	●	●	●	●	●
NCK	●	●	●	●	●	●
NEWCONF	●	●	●	●	●	●
NEWMT	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
NEWT	-	-	-	-	-	-
NORM	●	●	●	●	●	●
NOT	●	●	●	●	●	●
NPROT	●	●	●	●	●	●
NPROTDEF	●	●	●	●	●	●
NUMBER	●	●	●	●	●	●
OEMIPO1	-	-	-	-	-	-
OEMIPO2	-	-	-	-	-	-
OF	●	●	●	●	●	●
OFFN	●	●	●	●	●	●
OMA1	-	-	-	-	-	-
OMA2	-	-	-	-	-	-
OMA3	-	-	-	-	-	-
OMA4	-	-	-	-	-	-
OMA5	-	-	-	-	-	-
OR	●	●	●	●	●	●
ORIAXES	-	-	-	-	-	-
ORIAXPOS	-	-	-	-	-	-
ORIC	-	-	-	-	-	-
ORICONCCW	-	-	-	-	-	-
ORICONCW	-	-	-	-	-	-
ORICONIO	-	-	-	-	-	-
ORICONTO	-	-	-	-	-	-
ORICURVE	-	-	-	-	-	-
ORID	-	-	-	-	-	-
ORIEULER	-	-	-	-	-	-
ORIMKS	-	-	-	-	-	-
ORIPATH	-	-	-	-	-	-
ORIPATHS	-	-	-	-	-	-
ORIPLANE	-	-	-	-	-	-
ORIRESET	-	-	-	-	-	-
ORIROTA	-	-	-	-	-	-
ORIROTC	-	-	-	-	-	-
ORIROTR	-	-	-	-	-	-
ORIROTT	-	-	-	-	-	-
ORIRPY	-	-	-	-	-	-
ORIRPY2	-	-	-	-	-	-
ORIS	-	-	-	-	-	-
ORISOF	-	-	-	-	-	-

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
ORISOLH	-	-	-	-	-	-
ORISON	-	-	-	-	-	-
ORIVECT	-	-	-	-	-	-
ORIVIRT1	-	-	-	-	-	-
ORIVIRT2	-	-	-	-	-	-
ORIWKS	-	-	-	-	-	-
OS	○	○	○	○	○	○
OSB	○	○	○	○	○	○
OSC	-	-	-	-	-	-
OSCILL	○	○	○	○	○	○
OSCTRL	○	○	○	○	○	○
OSD	-	-	-	-	-	-
OSE	○	○	○	○	○	○
OSNSC	●	●	●	●	●	●
OSOF	-	-	-	-	-	-
OSP1	●	●	●	●	●	●
OSP2	●	●	●	●	●	●
OSS	-	-	-	-	-	-
OSSE	-	-	-	-	-	-
OST	-	-	-	-	-	-
OST1	●	●	●	●	●	●
OST2	●	●	●	●	●	●
OTOL	●	●	●	●	●	●
OVR	●	●	●	●	●	●
OVRA	●	●	●	●	●	●
OVRRAP	●	●	●	●	●	●
P	●	●	●	●	●	●
PAROT	●	●	●	●	●	●
PAROTOF	●	●	●	●	●	●
PCALL	●	●	●	●	●	●
PDELAYOF	-	-	-	-	-	-
PDELAYON	-	-	-	-	-	-
PHI	-	-	-	-	-	-
PHU	●	●	●	●	●	●
PL	-	-	-	-	-	-
PM	●	●	●	●	●	●
PO	-	-	-	-	-	-
POCKET3	-	-	-	-	-	-
POCKET4	-	-	-	-	-	-

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
POLF	●	●	●	●	●	●
POLFA	●	●	●	●	●	●
POLFMASK	●	●	●	●	●	●
POLFMLIN	●	●	●	●	●	●
POLY	-	-	-	-	-	-
POLYPATH	-	-	-	-	-	-
PON	-	-	-	-	-	-
PONS	-	-	-	-	-	-
POS	●	●	●	●	●	●
POSA	●	●	●	●	●	●
POSM	●	●	●	●	●	●
POSMT	-	-	-	-	-	-
POSP	●	●	●	●	●	●
POSRANGE	●	●	●	●	●	●
POT	●	●	●	●	●	●
PR	●	●	●	●	●	●
PREPRO	●	●	●	●	●	●
PRESETON	●	●	●	●	●	●
PRESETONS	●	●	●	●	●	●
PRIO	●	●	●	●	●	●
PRLOC	●	●	●	●	●	●
PROC	●	●	●	●	●	●
PROTA	●	●	●	●	●	●
PROTD	●	●	●	●	●	●
PROTS	●	●	●	●	●	●
PSI	-	-	-	-	-	-
PTP	●	●	●	●	●	●
PTPG0	●	●	●	●	●	●
PTPWOC	●	●	●	●	●	●
PUNCHACC	-	-	-	-	-	-
PUTFTOC	●	●	●	●	●	●
PUTFTOCF	●	●	●	●	●	●
PW	○	○	○	○	○	○
QU	●	●	●	●	●	●
R...	●	●	●	●	●	●
RAC	●	●	●	●	●	●
RDISABLE	●	●	●	●	●	●
READ	●	●	●	●	●	●
REAL	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
RELEASE	●	●	●	●	●	●
REP	●	●	●	●	●	●
REPEAT	●	●	●	●	●	●
REPEATB	●	●	●	●	●	●
REPOSA	●	●	●	●	●	●
REPOSH	●	●	●	●	●	●
REPOSHA	●	●	●	●	●	●
REPOSL	●	●	●	●	●	●
REPOSQ	●	●	●	●	●	●
REPOSQA	●	●	●	●	●	●
RESETMON	●	●	●	●	●	●
RET	●	●	●	●	●	●
RETB	●	●	●	●	●	●
RIC	●	●	●	●	●	●
RINDEX	●	●	●	●	●	●
RMB	●	●	●	●	●	●
RME	●	●	●	●	●	●
RMI	●	●	●	●	●	●
RMN	●	●	●	●	●	●
RND	●	●	●	●	●	●
RNDM	●	●	●	●	●	●
ROT	●	●	●	●	●	●
ROTS	●	●	●	●	●	●
ROUND	●	●	●	●	●	●
ROUNDUP	●	●	●	●	●	●
RP	●	●	●	●	●	●
RPL	●	●	●	●	●	●
RT	●	●	●	●	●	●
RTLIOF	●	●	●	●	●	●
RTLION	●	●	●	●	●	●

Operations S ... Z

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
S	●	●	●	●	●	●
SAVE	●	●	●	●	●	●
SBLOF	●	●	●	●	●	●
SBLON	●	●	●	●	●	●
SC	●	●	●	●	●	●
SCALE	●	●	●	●	●	●
SCC	●	●	●	●	●	●
SCPARA	●	●	●	●	●	●
SD	○	○	○	○	○	○
SET	●	●	●	●	●	●
SETAL	●	●	●	●	●	●
SETDNO	●	●	●	●	●	●
SETINT	●	●	●	●	●	●
SETM	-	-	●	-	-	●
SETMS	●	●	●	●	●	●
SETMS(n)	●	●	●	●	●	●
SETMTH	●	●	●	●	●	●
SETPIECE	●	●	●	●	●	●
SETTA	●	●	●	●	●	●
SETTCOR	●	●	●	●	●	●
SETTIA	●	●	●	●	●	●
SF	●	●	●	●	●	●
SIN	●	●	●	●	●	●
SIRELAY	-	-	-	-	-	-
SIRELIN	-	-	-	-	-	-
SIRELOUT	-	-	-	-	-	-
SIRELTIME	-	-	-	-	-	-
SLOT1	-	-	-	-	-	-
SLOT2	-	-	-	-	-	-
SOFT	●	●	●	●	●	●
SOFTA	●	●	●	●	●	●
SON	-	-	-	-	-	-
SONS	-	-	-	-	-	-
SPATH	●	●	●	●	●	●
SPCOF	●	●	●	●	●	●
SPCON	●	●	●	●	●	●
SPI	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
SPIF1	-	-	-	-	-	-
SPIF2	-	-	-	-	-	-
SPLINEPATH	○	○	○	○	○	○
SPN	-	-	-	-	-	-
SPOF	-	-	-	-	-	-
SPOS	●	●	●	●	●	●
SPOSA	●	●	●	●	●	●
SPP	-	-	-	-	-	-
SPRINT	●	●	●	●	●	●
SQRT	●	●	●	●	●	●
SR	●	●	●	●	●	●
SRA	●	●	●	●	●	●
ST	●	●	●	●	●	●
STA	●	●	●	●	●	●
START	-	-	●	-	-	●
STARTFIFO	●	●	●	●	●	●
STAT	●	●	●	●	●	●
STOLF	●	●	●	●	●	●
STOPFIFO	●	●	●	●	●	●
STOPRE	●	●	●	●	●	●
STOPREOF	●	●	●	●	●	●
STRING	●	●	●	●	●	●
STRINGFELD	●	●	●	●	●	●
STRINGIS	●	●	●	●	●	●
STRLEN	●	●	●	●	●	●
SUBSTR	●	●	●	●	●	●
SUPA	●	●	●	●	●	●
SVC	●	●	●	●	●	●
SYNFCT	●	●	●	●	●	●
SYNR	●	●	●	●	●	●
SYNRW	●	●	●	●	●	●
SYNW	●	●	●	●	●	●
T	●	●	●	●	●	●
TAN	●	●	●	●	●	●
TANG	○	○	○	○	○	○
TANGDEL	○	○	○	○	○	○
TANGOF	○	○	○	○	○	○
TANGON	○	○	○	○	○	○

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
TCA (828D: _TCA)	●	●	●	●	●	●
TCARR	●	●	●	●	●	●
TCI	●	●	●	●	●	●
TCOABS	●	●	●	●	●	●
TCOFR	●	●	●	●	●	●
TCOFRX	●	●	●	●	●	●
TCOFRY	●	●	●	●	●	●
TCOFRZ	●	●	●	●	●	●
THETA	-	-	-	-	-	-
TILT	-	-	-	-	-	-
TLIFT	○	○	○	○	○	○
TML	●	●	●	●	●	●
TMOF	●	●	●	●	●	●
TMON	●	●	●	●	●	●
TO	●	●	●	●	●	●
TOFF	●	●	●	●	●	●
TOFFL	●	●	●	●	●	●
TOFFOF	●	●	●	●	●	●
TOFFON	●	●	●	●	●	●
TOFFR	●	●	●	●	●	●
TOFRAME	●	●	●	●	●	●
TOFRAMEX	●	●	●	●	●	●
TOFRAMEY	●	●	●	●	●	●
TOFRAMEZ	●	●	●	●	●	●
TOLOWER	●	●	●	●	●	●
TOOLENV	●	●	●	●	●	●
TOOLGNT	●	●	●	●	●	●
TOOLGT	●	●	●	●	●	●
TOROT	●	●	●	●	●	●
TOROTOF	●	●	●	●	●	●
TOROTX	●	●	●	●	●	●
TOROTY	●	●	●	●	●	●
TOROTZ	●	●	●	●	●	●
TOUPPER	●	●	●	●	●	●
TOWBCS	●	●	●	●	●	●
TOWKCS	●	●	●	●	●	●
TOWMCS	●	●	●	●	●	●
TOWSTD	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
TOWTCS	●	●	●	●	●	●
TOWWCS	●	●	●	●	●	●
TR	●	●	●	●	●	●
TRAANG	○	○	○	-	-	-
TRACON	○	○	○	-	-	-
TRACYL	○	○	○	○	○	○
TRAFOOF	●	●	●	●	●	●
TRAFOON	-	-	-	-	-	-
TRAILOF	●	●	●	●	●	●
TRAILON	●	●	●	●	●	●
TRANS	●	●	●	●	●	●
TRANSMIT	○	○	○	○	○	○
TRAORI	-	-	-	-	-	-
TRUE	●	●	●	●	●	●
TRUNC	●	●	●	●	●	●
TU	●	●	●	●	●	●
TURN	●	●	●	●	●	●
ULI	●	●	●	●	●	●
UNLOCK	●	●	●	●	●	●
UNTIL	●	●	●	●	●	●
UPATH	●	●	●	●	●	●
VAR	●	●	●	●	●	●
VELOLIM	●	●	●	●	●	●
VELOLIMA	●	●	●	●	●	●
WAITC	●	●	●	●	●	●
WAITE	-	-	●	-	-	●
WAITENC	●	●	●	●	●	●
WAITM	-	-	●	-	-	●
WAITMC	-	-	●	-	-	●
WAITP	●	●	●	●	●	●
WAITS	●	●	●	●	●	●
WALCS0	●	●	●	●	●	●
WALCS1	●	●	●	●	●	●
WALCS2	●	●	●	●	●	●
WALCS3	●	●	●	●	●	●
WALCS4	●	●	●	●	●	●
WALCS5	●	●	●	●	●	●
WALCS6	●	●	●	●	●	●
WALCS7	●	●	●	●	●	●

Operation	SINUMERIK 828D					
● Standard ○ Option - not available	SW24x(5) CNC-SW G-Tech Export (gce42)	SW26x(3) CNC-SW G-Tech Export (gce62)	SW28x(1) CNC-SW G-Tech Adv. Export (gce82)	SW24x(5) CNC-SW G-Tech Export (gse42)	SW26x(3) CNC-SW G-Tech Export (gse62)	SW28x(1) CNC-SW G-Tech Adv. Export (gse82)
WALCS8	●	●	●	●	●	●
WALCS9	●	●	●	●	●	●
WALCS10	●	●	●	●	●	●
WALIMOF	●	●	●	●	●	●
WALIMON	●	●	●	●	●	●
WHEN	●	●	●	●	●	●
WHENEVER	●	●	●	●	●	●
WHILE	●	●	●	●	●	●
WORKPIECE	●	●	●	●	●	●
WRITE	●	●	●	●	●	●
WRTPR	●	●	●	●	●	●
X	●	●	●	●	●	●
XOR	●	●	●	●	●	●
Y	●	●	●	●	●	●
Z	●	●	●	●	●	●

17.3 Addresses

17.3.1 Address letters

Letter	Meaning	Numeric extension
A	Settable address identifier	x
B	Settable address identifier	x
C	Settable address identifier	x
D	Selection/deselection of tool length compensation, tool cutting edge	
E	Settable address identifier	x
F	Feedrate Dwell time in seconds	x
G	G command	
H	H function	x
I	Settable address identifier	x
J	Settable address identifier	x
K	Settable address identifier	x
L	Subprogram name, subprogram call	
M	M function	x
N	Subblock number	
O	Unassigned	
P	Number of program runs	
Q	Settable address identifier	x
R	Variable identifier (R parameter) Settable address identifier (without numeric extension)	x
S	Spindle value Dwell time in spindle revolutions	x x
T	Tool number	x
U	Settable address identifier	x
V	Settable address identifier	x
W	Settable address identifier	x
X	Settable address identifier	x
Y	Settable address identifier	x
Z	Settable address identifier	x
%	Start character and separator for file transfer	
:	Main block number	
/	Skip identifier	

17.3.2 Fixed addresses

Fixed addresses without axial extension

Address identifier	Address type	Modal/ non-modal	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	Data type of the assigned value
L	Subpro-gram num-ber	s									Unsigned INT
P	Number of subprogram repetitions	s									Unsigned INT
N	Block num-ber	s									Unsigned INT
G	G command	See list of the G func-tions									Unsigned INT
F	Feed, dwell time	m, s	x							x	Unsigned REAL
OVR	Override	m									Unsigned REAL
OVRRAP	Override for rapid tra-verse veloci-ty	m									Unsigned REAL
S	Spindle, dwell time	m, s								x	Unsigned REAL
SPOS	Spindle po-sition	m				x	x	x			REAL
SCC	Assignment of a trans-verse axis to G96 /G961/G962	m									REAL
SPOSA	Spindle po-sition across block boun-daries	m				x	x	x			REAL
T	Tool number	m								x	Unsigned INT

Address identifier	Address type	Modal/ non-modal	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	Data type of the assigned value
D	Offset number	m								x	Unsigned INT
M, H	Auxiliary functions	s								x	M: unsigned INT H: REAL

Fixed addresses with axial extension

Address identifier	Address type	Modal/ non-modal	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	Data type of the assigned value
AX	Variable axis identifier	1)	x	x	x	x	x	x			REAL
IP	Variable interpolation parameter	s	x	x	x	x	x				REAL
POS	Positioning axis	m	x	x	x	x	x	x	x		REAL
POSA	Positioning axis across block boundaries	m	x	x	x	x	x	x	x		REAL
POSP	Positioning axis in parts (oscillation)	m	x	x	x	x	x	x			REAL: End position Real: Partial length INT: Option
MOV	Start positioning axis	m	x	x	x	x	x	x	x		REAL
PO	Polynomial coefficient	s	x	x		x	x	x			Unsigned REAL
FA	Axial feedrate	m	x							x	Unsigned REAL
FL	Axial feedrate limit	m	x								Unsigned REAL
OVRA	Axial override	m	x								Unsigned REAL
ACC	Axial acceleration	m									Unsigned REAL
VELOLIM	Axial velocity limitation	m									Unsigned REAL

Address identifier	Address type	Modal/ non-modal	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	Data type of the assigned value
JERKLIM	Axial jerk limitation	m									Unsigned REAL
ACCLIMA	Axial acceleration limitation of following axis	m									Unsigned REAL
VELOLIMA	Axial velocity limitation of following axis	m									Unsigned REAL
JERKLIMA	Axial jerk limitation of following axis	m									Unsigned REAL
FMA	Synchronous feedrate axial	m									Unsigned REAL
STA	Sparking out time axial	m									Unsigned REAL
SRA	Retraction path on external input axial	m									Unsigned REAL
OS	Oscillation ON/OFF	m									Unsigned INT
OST1	Stopping time at left reversal point (oscillation)	m									REAL
OST2	Stopping time at right reversal point (oscillation)	m									REAL
OSP1	Left reversal point (oscillation)	m	x	x	x	x	x	x			REAL
OSP2	Right reversal point (oscillation)	m	x	x	x	x	x	x			REAL
OSB	position	m	x	x	x	x	x	x			REAL
OSE	Oscillation end position	m	x	x	x	x	x	x			REAL

Address identifier	Address type	Modal/ non-modal	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	Data type of the assigned value
OSNSC	Number of spark-out cycles (oscillation)	m									Unsigned INT
OSCTRL	Oscillation options	m									Unsigned INT: Setting options, unsigned INT: Reset options
OSCILL	Axis assignment for oscillation, activate oscillation	m									Axis: 1 - 3 infeed axes
FDA	Axis feed for handwheel override	s	x								Unsigned REAL
FGREF	Reference radius	m	x	x							Unsigned REAL
POLF	LIFTFAST position	m	x	x							Unsigned REAL
FXS	Travel to fixed stop ON	m									Unsigned INT
FXST	Torque limit for travel to fixed stop	m									REAL
FXSW	Monitoring window for travel to fixed stop	m									REAL
FOC	Travel with limited torque, non-modal	s									REAL
FOCON	Travel with limited torque ON, modal	m									REAL
FOCOF	Travel with limited torque OFF, modal	m									REAL

Address identifier	Address type	Modal/ non-modal	G70/ G71	G700/ G710	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	QU	Data type of the assigned value
MEASA	Axial measurement with delete distance-to-go	s									INT Mode and 1 - 4 trigger events
MEAWA	Axial measurement without delete distance-to-go	s									INT Mode and 1 - 4 trigger events
MEAC	Cyclic measuring	s									INT Mode and 1 - 4 trigger events

[1)] Absolute end points: Modal, incremental end points: Non-modal, otherwise modal/non-modal depending on the G function that determines the syntax.

17.3.3 Settable addresses

Address identifier (default setting)	Address type	Modal/ non-modal	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	Max. number	Data type of the assigned value
Axis values and end points											
X, Y, Z, A, B, C	Axis	[1)]	x	x	x	x				8	REAL
AP	Polar angle	m/s [1)]	x	x	x					1	REAL
RP	Polar radius	m/s [1)]	x	x	x					1	Unsigned REAL
Tool orientation											
A2, B2, C2	Euler angle or RPY angle	s								3	REAL
A3, B3, C3	Components of the directional vector	s								3	REAL
A4, B4, C4	Components of the surface normal vector at the start of the block	s								3	REAL

Address identifier (default setting)	Address type	Modal/ non-modal	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	Max. number	Data type of the assigned value
A5, B5, C5	Components of the surface normal vector at the end of the block	s								3	REAL
A6, B6, C6	Components of the direction vector for the rotary axis of the taper	s								3	REAL
A7, B7, C7	Vector components for the intermediate orientation on the peripheral surface of a taper	s								3	REAL
LEAD	Lead angle	m								1	REAL
THETA	Angle of rotation, rotation around the tool direction	m		x	x					1	REAL
TILT	Tilt angle	m								1	REAL
ORIS	Orientation change (in relation to the path)	m								1	REAL
Interpolation parameters											
I, J, K	Interpolation parameter intermediate point coordinate	s		x 2)	x 2)					3	REAL
I1, J1, K1		s	x	x	x					3	REAL
RPL	Rotation in the plane	s								1	REAL
CR	Circle radius	s								1	Unsigned REAL
AR	Opening angle	s								1	Unsigned REAL
TURN	Number of turns for helix	s								1	Unsigned INT
PL	Parameter interval length	s								1	Unsigned REAL
PW	weight	s								1	Unsigned REAL
SD	Spline degree	m								1	Unsigned INT
TU	Axis angle	s								1	Unsigned INT

Address identifier (default setting)	Address type	Modal/ non-modal	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	Max. number	Data type of the assigned value
STAT	Position of joints	m								1	Unsigned INT
SF	Starting point offset for thread cutting	m								1	REAL
DISCL	Safety clearance SAR	s								1	Unsigned REAL
DISR	Repositioning clearance / SAR clearance	s								1	Unsigned REAL
DISPR	Path differential for repositioning	s								1	Unsigned REAL
ALF	Rapid lift angle	m								1	Unsigned INT
DILF	Rapid lift length	m								1	REAL
FP	Fixed point: Number of fixed point to be approached	s								1	Unsigned INT
RNDM	Modal rounding	m								1	Unsigned REAL
RND	Non-modal rounding	s								1	Unsigned REAL
CHF	Chamfer non-modal	s								1	Unsigned REAL
CHR	Chamfer in original direction of motion	s								1	Unsigned REAL
ANG	Contour angle	s								1	REAL
ISD	Insertion depth	m								1	REAL
DISC	Transition circle overshoot tool radius compensation	m								1	Unsigned REAL
OFFN	Offset contour normal	m								1	REAL
DITS	Thread run-in path	m								1	REAL
DITE	Thread run-out path	m								1	REAL
Corner rounding criteria											
ADIS	Rounding clearance	m								1	Unsigned REAL

Address identifier (default setting)	Address type	Modal/ non-modal	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	Max. number	Data type of the assigned value
ADISPOS	Rounding clearance for rapid traverse	m								1	Unsigned REAL
Measurement											
MEAS	Measurement with touch-trigger probe	s								1	Unsigned INT
MEAW	Measurement with touch-trigger probe without deletion of distance-to-go	s								1	Unsigned INT
Axis, spindle behavior											
LIMS	Spindle speed limitation	m								1	Unsigned REAL
COARSEA	Block change behavior: Exact stop coarse axial	m									
FINEA	Block change behavior: Exact stop fine axial	m									
IPOENDA	Block change behavior: Interpolator stop axial	m									
DIACYCOFA	Transverse axis: Axial diameter programming OFF in cycles	m									
DIAM90A	Transverse axis: Axial diameter programming for G90	m									
DIAMCHAN	Transverse axis: Transfer of all transverse axes in the diameter programming channel status	m									
DIAMCHANA	Transverse axis: Transfer of the diameter programming channel status	m									

Address identifier (default setting)	Address type	Modal/ non-modal	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	Max. number	Data type of the assigned value
DIAMOFA	Transverse axis: Axial diameter programming OFF	m									
DIAMONA	Transverse axis: Axial diameter programming ON	m									
GP	Position: Indirect programming of position attributes	m									
Feedrates											
FAD	Speed of the slow feed movement	s						x		1	Unsigned REAL
FD	Path feedrate for handwheel override	s								1	Unsigned REAL
FRC	Feedrate for radius and chamfer	s								1	Unsigned REAL
FRCM	Feedrate for radius and chamfer, modal	m								1	Unsigned REAL
FB	Non-modal feedrate	s								1	Unsigned REAL
Nibbling/punching											
SPN	Number of path sections per block	s								1	INT
SPP	Length of a path section	m								1	REAL
Grinding											
ST	Sparking-out time	s								1	Unsigned REAL
SR	Retraction path	s								1	Unsigned REAL
Tool selection											
TCARR	Tool carrier	m								1	INT
Tool management											

Address identifier (default setting)	Address type	Modal/ non-modal	G90/ G91	IC	AC	DC, ACN, ACP	CIC, CAC, CDC, CACN, CACP	PR, PM	QU	Max. number	Data type of the assigned value
DL	Total tool offset	m								1	INT
OEM addresses											
OMA1	OEM address 1	m		x	x	x				1	REAL
OMA2	OEM address 2	m		x	x	x				1	REAL
OMA3	OEM address 3	m		x	x	x				1	REAL
OMA4	OEM address 4	m		x	x	x				1	REAL
OMA5	OEM address 5	m		x	x	x				1	REAL
Miscellaneous											
CUTMOD	Modification of the offset data for rotatable tools (in combination with orientable tool carriers)	m									INT
CUTMODK	Modification of the offset data for rotatable tools (in combination with orientation transformations that have been defined by means of kinematic chains)	m									STRING
TOFF	Tool length offset parallel to the specified geometry axis	m									REAL
TOFFL	Tool length offset in the direction of the tool length component L1, L2 or L3	m									REAL
TOFFR	Tool radius offset	m									REAL

[1] Absolute end points: Modal, incremental end points: non-modal, otherwise modal/non-modal depending on the G command that determines the syntax.

[2] As circle center points, IPO parameters act incrementally. They can be programmed in absolute mode with AC. The address modification is ignored when the parameters have other meanings (e.g. thread lead).

17.4 G commands

The G commands are divided into G groups. In part programs or synchronized actions, in a block, only a G command of a G group can be written. A G command can be modal or non-modal.

Modal: up to programming another G command of the same G group.

G groups

- G group 1 ... 15 (Page 517)
- G group 16 ... 30 (Page 517)
- G group 31 ... 45 (Page 517)
- G group 46 ... 62 (Page 517)
- G groups: Table legend (Page 517)

G group 1: Modally valid motion commands						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
G0	1	Rapid traverse	+	m		
G1	2	Linear interpolation (linear interpolation)	+	m	x	
G2	3	Circular interpolation clockwise	+	m		
G3	4	Circular interpolation counter-clockwise	+	m		
CIP	5	Circular interpolation through intermediate point	+	m		
ASPLINE	6	Akima spline	+	m		
BSPLINE	7	B spline	+	m		
CSPLINE	8	Cubic spline	+	m		
POLY	9	Polynomial interpolation	+	m		
G33	10	Thread cutting with constant lead	+	m		
G331	11	Tapping	+	m		
G332	12	Retraction (tapping)	+	m		
OEMIPO1	13	Reserved	+	m		
OEMIPO2	14	Reserved	+	m		
CT	15	Circle with tangential transition	+	m		
G34	16	Thread cutting with linear increasing lead	+	m		
G35	17	Thread cutting with linear decreasing lead	+	m		
INVCW	18	Involute interpolation clockwise	+	m		
INVCCW	19	Counter-clockwise involute interpolation	+	m		
G335	20	Turning a convex thread in clockwise direction	+	m		
G336	21	Turning a convex thread in counter-clockwise direction	+	m		

G group 2: Non-modally valid motion, dwell time						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
G4	1	Dwell time, preset	-	s		
G63	2	Tapping without synchronization	-	s		
G74	3	Reference point approach with synchronization	-	s		
G75	4	Fixed-point approach	-	s		
REPOSL	5	Linear repositioning	-	s		
REPOSQ	6	Repositioning in a quadrant	-	s		
REPOSH	7	Repositioning in semicircle	-	s		
REPOSA	8	Linear repositioning with all axes	-	s		
REPOSQA	9	Linear repositioning with all axes, geometry axes in quadrant	-	s		
REPOSHA	10	Repositioning with all axes; geometry axes in semicircle	-	s		
G147	11	Approach contour with straight line	-	s		
G247	12	Approach contour with quadrant	-	s		
G347	13	Approach contour with semicircle	-	s		
G148	14	Leave contour with straight line	-	s		
G248	15	Leave contour with quadrant	-	s		
G348	16	Leave contour with semicircle	-	s		
G5	17	Oblique plunge-cut grinding	-	s		
G7	18	Compensatory motion during oblique plunge-cut grinding	-	s		

G group 3: Programmable frame, working area limitation and pole programming						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
TRANS	1	TRANSLATION: Programmable offset	-	s		
ROT	2	ROTATION: Programmable rotation	-	s		
SCALE	3	SCALE: Programmable scaling	-	s		
MIRROR	4	MIRROR: Programmable mirroring	-	s		
ATRANS	5	Additive TRANSLATION: Additive programmable translation	-	s		
AROT	6	Additive ROTATION: Programmable rotation	-	s		
ASCALE	7	Additive SCALE: Programmable scaling	-	s		
AMIRROR	8	Additive MIRROR: Programmable mirroring	-	s		
-	9	Unassigned	-	-		
G25	10	Minimum working area limitation/spindle speed limitation	-	s		
G26	11	Maximum working area limitation/spindle speed limitation	-	s		
G110	12	Pole programming relative to the last programmed setpoint position	-	s		

G group 3: Programmable frame, working area limitation and pole programming						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
G111	13	Polar programming relative to origin of current workpiece coordinate system	-	s		
G112	14	Pole programming relative to the last valid pole	-	s		
G58	15	840D sl: Absolute programmable work offset 828D: 5. Settable work offset	-	s m		
G59	16	840D sl: Additive programmable work offset 828D: 6. Settable work offset	-	s m		
ROTS	17	Rotation with solid angle	-	s		
AROTS	18	Additive rotation with solid angle	-	s		

G group 4: FIFO						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
STARTFIFO	1	Start FIFO Execute and simultaneously fill preprocessing memory	+	m	x	
STOPFIFO	2	STOP FIFO Stop machining; fill preprocessing memory until STARTFIFO is detected, FIFO is full or end of program	+	m		
FIFOCTRL	3	Activation of automatic preprocessing memory control	+	m		

G group 6: Plane selection						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
G17	1	Plane selection 1. – 2. Geometry axis	+	m	x	
G18	2	Plane selection 3. – 1. Geometry axis	+	m		
G19	3	Plane selection 2. – 3. Geometry axis	+	m		

G group 7: Tool radius compensation						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
G40	1	No tool radius compensation	+	m	x	
G41	2	Tool radius compensation left of contour	-	m		
G42	3	Tool radius compensation right of contour	-	m		

G group 8: Settable work offset						
G command	**No.** 1)	**Meaning**	**MD20150** 2)	**W** 3)	**STD** 4)	
					SAG	**MH**
G500	1	Deactivation of settable zero offset (G54 to G57, G505 to G599)	+	m	x	
G54	2	1st settable zero offset	+	m		
G55	3	2nd settable zero offset	+	m		
G56	4	3rd settable zero offset	+	m		
G57	5	4th settable zero offset	+	m		
G505	6	5th settable zero offset	+	m		
...	...	...	+	m		
G599	100	99th settable zero offset	+	m		
Each of the G commands in this G group is used to activate an adjustable user frame $P_UIFR[ ]. G54 corresponds to frame $P_UIFR[1], G505 corresponds to frame $P_UIFR[5]. The number of adjustable user frames and therefore the number of G commands in this G group can be set using machine data MD28080 $MC_MM_NUM_USER_FRAMES.						

G group 9: Frame suppression						
G command	**No.** 1)	**Meaning**	**MD20150** 2)	**W** 3)	**STD** 4)	
					SAG	**MH**
G53	1	Suppression of current frames: Programmable frame including system frame for TOROT and TOFRAME and active adjustable frame (G54 to G57, G505 to G599)	-	s		
SUPA	2	As for G153 including suppression of system frames for actual-value setting, scratching, ext. zero offset, PAROT including handwheel offsets (DRF), [external zero offset], overlaid movement	-	s		
G153	3	As for G53 including suppression of all channel-specific and/or NCU-global basic frames	-	s		

G group 10: Exact stop - continuous-path mode						
G command	**No.** 1)	**Meaning**	**MD20150** 2)	**W** 3)	**STD** 4)	
					SAG	**MH**
G60	1	Exact stop	+	m	x	
G64	2	Continuous-path mode	+	m		
G641	3	Continuous-path mode with smoothing according to distance criterion (= programmable rounding clearance)	+	m		
G642	4	Continuous-path mode with smoothing within the defined tolerances	+	m		
G643	5	Continuous-path mode with smoothing within the defined tolerances (block-internal)	+	m		

G group 10: Exact stop - continuous-path mode						
G command	No. [1]	Meaning	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G644	6	Continuous-path mode with smoothing with maximum possible dynamic response	+	m		
G645	7	Continuous-path mode with smoothing and tangential block transitions within defined tolerances	+	m		

G group 11: Exact stop, non-modal						
G command	No. [1]	Meaning	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G9	1	Exact stop	-	s		

G group 12: Block change criteria at exact stop (G60/G9)						
G command	No. [1]	Meaning	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G601	1	Block change at exact stop fine	+	m	x	
G602	2	Block change at exact stop coarse	+	m		
G603	3	Block change at IPO block end	+	m		

G group 13: Workpiece measuring inch/metric						
G command	No. [1]	Meaning	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G70	1	Input system inches (length)	+	m		
G71	2	Input system metric mm (lengths)	+	m	x	
G700	3	Input system inch, inch/min (lengths + velocity + system variable)	+	m		
G710	4	Input system metric mm, mm/min (lengths + velocity + system variable)	+	m		

G group 14: Workpiece measuring absolute/incremental						
G command	No. [1]	Meaning	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G90	1	Absolute dimension	+	m	x	
G91	2	Incremental dimensions	+	m		

G group 15: Feed type						
G command	No. [1]	Meaning	MD20150 [2]	W [3]	STD [4]	
					SAG	MH
G93	1	Inverse-time feedrate rpm	+	m		
G94	2	Linear feedrate in mm/min, inch/min	+	m	x	

G group 15: Feed type						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
G95	3	Revolutional feedrate in mm/rev, inch/rev	+	m		
G96	4	Revolutional feedrate (as for G95) and constant cutting rate	+	m		
G97	5	Revolutional feedrate and constant spindle speed (constant cutting rate OFF)	+	m		
G931	6	Feedrate specified by means of traversing time, deactivate constant path velocity	+	m		
G961	7	Linear feedrate (as for G94) and constant cutting rate	+	m		
G971	8	Linear feedrate and constant spindle speed (constant cutting rate OFF)	+	m		
G942	9	Linear feedrate and constant cutting rate or constant spindle speed	+	m		
G952	10	Revolutional feedrate and constant cutting rate or constant spindle speed	+	m		
G962	11	Linear feedrate or revolutional feedrate and constant cutting rate	+	m		
G972	12	Linear feedrate or revolutional feedrate and constant spindle speed (constant cutting rate OFF)	+	m		
G973	13	Revolutional feedrate without spindle speed limitation and constant spindle speed (G97 without LIMS for ISO mode)	+	m		

G group 16: Feedrate override at inside and outside curvature						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
CFC	1	Constant feedrate at contour effective for internal and external radius	+	m	x	
CFTCP	2	Constant feedrate in tool center point (center point path)	+	m		
CFIN	3	Constant feedrate for internal radius only, acceleration for external radius	+	m		

G group 17: Approach and retraction response, tool offset						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
NORM	1	Normal position at starting and end points	+	m	x	
KONT	2	Travel around contour at starting and end points	+	m		
KONTT	3	Approach/retraction with constant tangent	+	m		
KONTC	4	Approach/retraction with constant curvature	+	m		

G group 18: Corner behavior, tool offset						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
G450	1	Transition circle (tool travels around workpiece corners on a circular path)	+	m	x	
G451	2	Intersection of equidistant paths (tool backs off from the workpiece corner)	+	m		

G group 19: Curve transition at beginning of spline						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
BNAT	1	Natural transition to first spline block	+	m	x	
BTAN	2	Tangential transition to first spline block	+	m		
BAUTO	3	Definition of the first spline section by means of the next 3 points	+	m		

G group 20: Curve transition at end of spline						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
ENAT	1	Natural transition to next traversing block	+	m	x	
ETAN	2	Tangential transition to next traversing block	+	m		
EAUTO	3	Definition of the last spline section by means of the last 3 points	+	m		

G group 21: Acceleration profile						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
BRISK	1	Fast non-smoothed path acceleration	+	m	x	
SOFT	2	Soft smoothed path acceleration	+	m		
DRIVE	3	Velocity-dependent path acceleration	+	m		

G group 22: Tool offset type						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
CUT2D	1	2½D TRC	+	m	x	
CUT2DF	2	2½D TRC relative to the current frame (inclined plane)	+	m		
CUT3DC	3	3D TRC for circumferential milling	+	m		
CUT3DF	4	3D TRC for face milling with change in orientation	+	m		
CUT3DFS	5	3D TRC for face milling with constant orientation. The tool orientation is defined by G17 - G19 and is not influenced by frames.	+	m		

G group 22: Tool offset type						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
CUT3DFF	6	3D TRC for face milling with constant orientation. The tool orientation is the direction defined by G17 - G19 and, in some case, rotated by a frame.	+	m		
CUT3DCC	7	3D TRC for the circumferential milling taking into account a limitation surface with 3D radius compensation: Contour on the machining surface	+	m		
CUT3DCCD	8	3D TRC for the circumferential milling taking into account a limitation surface with a differential tool on the tool center-point path: Infeed to the limitation surface	+	m		
CUT2DD	9	2½D TRC referred to a differential tool	+	m		
CUT2DFD	10	2½D TRC referred to a differential tool relative to the current frame (inclined plane)	+	m		
CUT3DCD	11	3D TRC for circumferential milling referred to a differential tool	+	m		
CUT3DFD	12	3D TRC for face milling with orientation change referred to a differential tool	+	m		

G group 23: Collision monitoring at inside contours						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
CDOF	1	Collision detection OFF	+	m	x	
CDON	2	Collision detection ON	+	m		
CDOF2	3	Collision detection OFF for 3D circumferential milling	+	m		

G group 24: Precontrol						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
FFWOF	1	Feedforward control OFF	+	m	x	
FFWON	2	Feedforward control ON	+	m		

G group 25: Tool orientation reference						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
ORIWKS	1	Tool orientation in workpiece coordinate system (WCS)	+	m	x	
ORIMKS	2	Tool orientation in machine coordinate system (MCS)	+	m		

G group 26: Repositioning mode for REPOS (modal)						
G command	No. [1)]	Meaning	MD20150 [2)]	W [3)]	STD [4)]	
					SAG	MM
RMB	1	Repositioning to start of block	-	m		
RMI	2	Repositioning to interrupt point	-	m	x	
RME	3	Repositioning to end of block	-	m		
RMN	4	Repositioning to the nearest path point	-	m		

G group 27: Tool offset for change in orientation at outside corners						
G command	No. [1)]	Meaning	MD20150 [2)]	W [3)]	STD [4)]	
					SAG	MM
ORIC	1	Orientation changes at outside corners are superimposed on the circle block to be inserted	+	m	x	
ORID	2.	Orientation changes are performed before the circle block	+	m		

G group 28: Working area limitation						
G command	No. [1)]	Meaning	MD20150 [2)]	W [3)]	STD [4)]	
					SAG	MM
WALIMON	1	Working area limitation ON	+	m	x	
WALIMOF	2	Working area limitation OFF	+	m		

G group 29: Radius/diameter programming						
G command	No. [1)]	Meaning	MD20150 [2)]	W [3)]	STD [4)]	
					SAG	MM
DIAMOF	1	Modal channel-specific diameter programming OFF Deactivation activates channel-specific radius programming.	+	m	x	
DIAMON	2	Modal independent channel-specific diameter programming ON The effect is independent of the programmed dimensions mode (G90/G91).	+	m		
DIAM90	3	Modal dependent channel-specific diameter programming ON The effect is dependent on the programmed dimensions mode (G90/G91).	+	m		
DIAMCYCOF	4	Modal channel-specific diameter programming during cycle processing OFF	+	m		

G group 30: NC block compression						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
COMPOF	1	NC block compression OFF	+	m	x	
COMPON	2	Compressor function COMPON ON	+	m		
COMPCURV	3	Compressor function COMPCURV ON	+	m		
COMPCAD	4	Compressor function COMPCAD ON	+	m		
COMPSURF	5	COMPSURF EIN compressor function	+	m		

G group 31: OEM G commands						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
G810	1	OEM G command	-	m		
G811	2	OEM G command	-	m		
G812	3	OEM G command	-	m		
G813	4	OEM G command	-	m		
G814	5	OEM G command	-	m		
G815	6	OEM G command	-	m		
G816	7	OEM G command	-	m		
G817	8	OEM G command	-	m		
G818	9	OEM G command	-	m		
G819	10	OEM G command	-	m		
Two G groups are reserved for the OEM user. This enables the OEM to program functions that can be customized.						

G group 32: OEM G commands						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
G820	1	OEM G command	-	m		
G821	2	OEM G command	-	m		
G822	3	OEM G command	-	m		
G823	4	OEM G command	-	m		
G824	5	OEM G command	-	m		
G825	6	OEM G command	-	m		
G826	7	OEM G command	-	m		
G827	8	OEM G command	-	m		
G828	9	OEM G command	-	m		
G829	10	OEM G command	-	m		
Two G groups are reserved for the OEM user. This enables the OEM to program functions that can be customized.						

G group 33: Settable fine tool offset						
G command	**No.** 1)	**Meaning**	**MD20150** 2)	**W** 3)	**STD** 4)	
					SAG	**MM**
FTOCOF	1	Online fine tool offset OFF	+	m	x	
FTOCON	2	Online fine tool offset ON	-	m		

G group 34: Tool orientation smoothing						
G command	**No.** 1)	**Meaning**	**MD20150** 2)	**W** 3)	**STD** 4)	
					SAG	**MM**
OSOF	1	Tool orientation smoothing OFF	+	m	x	
OSC	2	Continuous tool orientation smoothing	+	m		
OSS	3	Tool orientation smoothing at end of block	+	m		
OSSE	4	Tool orientation smoothing at start and end of block	+	m		
OSD	5	Block-internal smoothing with specification of path length	+	m		
OST	6	Block-internal smoothing with specification of angular tolerance	+	m		

G group 35: Punching and nibbling						
G command	**No.** 1)	**Meaning**	**MD20150** 2)	**W** 3)	**STD** 4)	
					SAG	**MM**
SPOF	1	Stroke OFF, nibbling and punching OFF	+	m	x	
SON	2	Nibbling ON	+	m		
PON	3	Punching ON	+	m		
SONS	4	Nibbling ON in interpolation cycle	-	m		
PONS	5	Punching ON in interpolation cycle	-	m		

G group 36: Punching with delay						
G command	**No.** 1)	**Meaning**	**MD20150** 2)	**W** 3)	**STD** 4)	
					SAG	**MM**
PDELAYON	1	Punching with delay ON	+	m	x	
PDELAYOF	2	Punching with delay OFF	+	m		

G group 37: Feedrate profile						
G command	**No.** 1)	**Meaning**	**MD20150** 2)	**W** 3)	**STD** 4)	
					SAG	**MM**
FNORM	1	Feedrate normal to DIN 66025	+	m	x	
FLIN	2	Feed linear variable	+	m		
FCUB	3	Feedrate variable according to cubic spline	+	m		

G group 38: Assignment of fast inputs/outputs for punching/nibbling						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
SPIF1	1	Fast NC inputs/outputs for punching/nibbling byte 1	+	m	x	
SPIF2	2	Fast NC inputs/outputs for punching/nibbling byte 2	+	m		

G group 39: Programmable contour accuracy						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
CPRECOF	1	Programmable contour accuracy OFF	+	m	x	
CPRECON	2	Programmable contour accuracy ON	+	m		

G group 40: Tool radius compensation constant						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
CUTCONOF	1	Constant tool radius compensation OFF	+	m	x	
CUTCONON	2	Constant tool radius compensation ON	+	m		

G group 41: Interruptible thread cutting						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
LFOF	1	Interruptible thread cutting OFF	+	m	x	
LFON	2	Interruptible thread cutting ON	+	m		

G group 42: Tool carrier						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
TCOABS	1	Determine tool length components from the current tool orientation	+	m	x	
TCOFR	2	Determine tool length components from the orientation of the active frame	+	m		
TCOFRZ	3	Determine tool orientation of an active frame on selection of tool, tool points in Z direction	+	m		
TCOFRY	4	Determine tool orientation of an active frame on selection of tool, tool points in Y direction	+	m		
TCOFRX	5	Determine tool orientation of an active frame on selection of tool, tool points in X direction		m		

G group 43: SAR approach direction						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
G140	1	SAR approach direction defined by G41/G42	+	m	x	
G141	2	SAR approach direction to left of contour	+	m		
G142	3	SAR approach direction to right of contour	+	m		
G143	4	SAR approach direction tangent-dependent	+	m		

G group 44: SAR path segmentation						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
G340	1	Spatial approach block; in other words, infeed depth and approach in plane in one block	+	m	x	
G341	2	Start with infeed on perpendicular axis (Z), then approach in plane	+	m		

G group 45: Path reference for FGROUP axes						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MM
SPATH	1	Path reference for FGROUP axes is arc length	+	m	x	
UPATH	2	Path reference for FGROUP axes is curve parameter	+	m		

G group 46: Plane selection for fast retraction						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
LFTXT	1	The plane is determined from the path tangent and the current tool orientation	+	m	x	
LFWP	2	The plane is determined by the current working plane (G17/G18/G19)	+	m		
LFPOS	3	Axial retraction to a position	+	m		

G group 47: Mode switchover for external NC code						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
G290	1	Activate SINUMERIK language mode	+	m	x	
G291	2	Activate ISO language mode	+	m		

G group 48: Approach and retraction response with tool radius compensation						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
G460	1	Collision detection for approach and retraction block ON	+	m	x	
G461	2	Extend border block with arc if no intersection in TRC block	+	m		
G462	3	Extend border block with straight line if no intersection in TRC block	+	m		

G group 49: Point-to-point motion						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
CP	1	Path motion	+	m	x	
PTP	2	Point-to-point motion (synchronized axis motion)	+	m		
PTPG0	3	Point-to-point motion only with G0, otherwise path motion CP	+	m		
PTPWOC	4	Point-to-point motion without compensationary motion, which is caused by orientation changes	+	m		

G group 50: Orientation programming						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
ORIEULER	1	Orientation angle via Euler angle	+	m	x	
ORIRPY	2	Orientation angle via RPY angle (rotation sequence XYZ)	+	m		
ORIVIRT1	3	Orientation angle via virtual orientation axes (definition 1)	+	m		
ORIVIRT2	4	Orientation angle via virtual orientation axes (definition 2)	+	m		
ORIAXPOS	5	Orientation angle via virtual orientation axes with rotary axis positions	+	m		
ORIRPY2	6	Orientation angle via RPY angle (rotation sequence ZYX)	+	m		

G group 51: Interpolation type for orientation programming						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
ORIVECT	1	Large-circle interpolation (identical to ORIPLANE)	+	m	x	
ORIAXES	2	Linear interpolation of machine axes or orientation axes	+	m		
ORIPATH	3	Tool orientation trajectory referred to path	+	m		
ORIPLANE	4	Interpolation in plane (identical to ORIVECT)	+	m		

G group 51: Interpolation type for orientation programming						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
ORICONCW	5	Interpolation on the peripheral surface of a taper in the clockwise direction	+	m		
ORI-CONCCW	6	Interpolation on the peripheral surface of a taper in the counter-clockwise direction	+	m		
ORICONIO	7	Interpolation on a conical peripheral surface with intermediate orientation setting	+	m		
ORICONTO	8	Interpolation on a peripheral surface of the cone with tangential transition	+	m		
ORICURVE	9	Interpolation with additional space curve for orientation	+	m		
ORIPATHS	10	Tool orientation in relation to the path, kinks in the orientation characteristic are smoothed	+	m		

Table 17-1

G group 52: Frame rotation in relation to workpiece						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
PAROTOF	1	Frame rotation in relation to workpiece OFF	+	m	x	
PAROT	2	Frame rotation in relation to workpiece ON The workpiece coordinate system is aligned on the workpiece.	+	m		

G group 53: Frame rotation in relation to tool						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
TOROTOF	1	Frame rotation in relation to tool OFF	+	m	x	
TOROT	2	Align Z axis of the WCS by rotating the frame parallel to the tool orientation	+	m		
TOROTZ	3	Like TOROT	+	m		
TOROTY	4	Align Y axis of the WCS by rotating the frame parallel to the tool orientation	+	m		
TOROTX	5	Align X axis of the WCS by rotating the frame parallel to the tool orientation	+	m		
TOFRAME	6	Align Z axis of the WCS by rotating the frame parallel to the tool orientation	+	m		
TOFRAMEZ	7	Like TOFRAME	+	m		
TOFRAMEY	8	Align Y axis of the WCS by rotating the frame parallel to the tool orientation	+	m		
TOFRAMEX	9	Align X axis of the WCS by rotating the frame parallel to the tool orientation	+	m		

G group 54: Vector rotation for polynomial programming						
G command	No. [1)]	Meaning	MD20150 [2)]	W [3)]	STD [4)]	
					SAG	MH
ORIROTA	1	Vector rotation absolute	+	m	x	
ORIROTR	2	Vector rotation relative	+	m		
ORIROTT	3	Vector rotation tangential	+	m		
ORIROTC	4	Tangential rotational vector in relation to path tangent	+	m		

G group 55: Rapid traverse with/without linear interpolation						
G command	No. [1)]	Meaning	MD20150 [2)]	W [3)]	STD [4)]	
					SAG	MH
RTLION	1	Rapid traverse motion with linear interpolation ON	+	m	x	
RTLIOF	2	Rapid traverse motion with linear interpolation OFF Rapid traverse motion is achieved with single-axis interpolation.	+	m		

G group 56: Taking into account tool wear						
G command	No. [1)]	Meaning	MD20150 [2)]	W [3)]	STD [4)]	
					SAG	MH
TOWSTD	1	Initial setting value for offsets in tool length	+	m	x	
TOWMCS	2	Wear values in the machine coordinate system (MCS)	+	m		
TOWWCS	3	Wear values in the workpiece coordinate system (WCS)	+	m		
TOWBCS	4	Wear values in the basic coordinate system (BCS)	+	m		
TOWTCS	5	Wear values in the tool coordinate system (toolholder ref. point T at the toolholder)	+	m		
TOWKCS	6	Wear values in the coordinate system of the tool head for kinetic transformation (differs from machine coordinate system through tool rotation)	+	m		

G group 57: Corner deceleration						
G command	No. [1)]	Meaning	MD20150 [2)]	W [3)]	STD [4)]	
					SAG	MH
FENDNORM	1	Corner deceleration OFF	+	m	x	
G62	2	Corner deceleration at inside corners when tool radius compensation is active (G41/G42)	+	m		
G621	3	Corner deceleration at all corners	+	m		

G group 59: Dynamic response mode for path interpolation						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
DYNNORM	1	Standard dynamic response, as before	+	m	x	
DYNPOS	2	Positioning mode, tapping	+	m		
DYNROUGH	3	Roughing	+	m		
DYNSEMIFIN	4	Finishing	+	m		
DYNFINISH	5	Smooth finishing	+	m		

G group 60: Working area limitation						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
WALCS0	1	Workpiece coordinate system working area limitation OFF	+	m	x	
WALCS1	2	WCS working area limitation group 1 active	+	m		
WALCS2	3	WCS working area limitation group 2 active	+	m		
WALCS3	4	WCS working area limitation group 3 active	+	m		
WALCS4	5	WCS working area limitation group 4 active	+	m		
WALCS5	6	WCS working area limitation group 5 active	+	m		
WALCS6	7	WCS working area limitation group 6 active	+	m		
WALCS7	8	WCS working area limitation group 7 active	+	m		
WALCS8	9	WCS working area limitation group 8 active	+	m		
WALCS9	10	WCS working area limitation group 9 active	+	m		
WALCS10	11	WCS working area limitation group 10 active	+	m		

G group 61: Tool orientation smoothing						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
ORISOF	1	Tool orientation smoothing OFF	+	m	x	
ORISON	2	Tool orientation smoothing ON	+	m		

G group 62: Repositioning mode for REPOS (non-modal)						
G command	No. 1)	Meaning	MD20150 2)	W 3)	STD 4)	
					SAG	MH
RMBBL	1	Repositioning to start of block	-	s		
RMIBL	2	Repositioning to interrupt point	-	s	x	
RMEBL	3	Repositioning to end of block	-	s		
RMNBL	4	Repositioning to the nearest path point	-	s		

G group 64: Grinding frames						
G command	**No.** [1)]	**Meaning** **Active grinding frame in the channel $P_GFRAME =**	**MD20150** [2)]	**W** [3)]	**STD** [4)]	
					SAG	**MH**
GFRAME[0]	1	Grinding frame of the data management $P_GFR[0] (null frame)	+	m	x	
GFRAME[1]	2	Grinding frame of the data management $P_GFR[1]	+	m		
GFRAME[2]	3	Grinding frame of the data management $P_GFR[2]	+	m		
...	...		+	m		
GFRAME[100]	101	Grinding frame of the data management $P_GFR[100]	+	m		

Legend

1) Internal number (e.g. for PLC interface)

2) Configurability of the G command as a reset setting for the G group on power up, reset or end of part program (with MD20150 $MC_GCODE_RESET_VALUES):

+ Configurable
- Not configurable

3) Effectiveness of the G command:

m modal
s Non-modal

4) Reset setting, see the following machine data:

- MD20149GCODE_RESET_S_VALUES (reset position of G groups (fix))
- MD20150 $MC_GCODE_RESET_VALUES (reset position of the G groups)
- MD20151GCODE_RESET_S_MODE (reset behavior of G groups (fix))
- MD20152 $MC_GCODE_RESET_MODE (reset behavior of G groups)
- MD20154 $MC_EXTERN_GCODE_RESET_VALUES (reset position of the G groups in ISO mode)
- MD20156 $MC_EXTERN_GCODE_RESET_MODE (reset behavior of external G groups)

SAG Default setting **S**iemens **AG**

MM Default setting **M**achine **M**anufacturer (see machine manufacturer's specifications)

17.5 Predefined procedures

The call of a predefined procedure triggers the execution of a predefined NC function. A predefined procedure does **not** supply a return value in contrast to a predefined function.

Coordinate system					
Identifier	**Parameter**				**Explanation**
	1.	**2.**	**3. - 15.**	**4. - 16.**	
PRESETON	AXIS *): Axis identifier of machine axis	REAL: Preset offset G700/G710 context	As 1 ...	As 2 ...	Set actual value for the programmed axes with loss of the referencing status
PRESETONS	AXIS *): Axis identifier of machine axis	REAL: Preset offset G700/G710 context	As 1 ...	As 2 ...	Set actual value for the programmed axes without loss of the referencing status
DRFOF					Deletes the DRF offset for all axes assigned to the channel.

) As a general rule, geometry or special axis identifiers can also be used instead of the machine axis identifier, as long as the reference is unambiguous.

Axis groupings					
Identifier	**Parameter**				**Explanation**
GEOAX	**1.**	**2.**	**3. / 5.**	**4. / 6.**	Selection of a parallel coordinate system
	INT: Geometry axis number 1 - 3	AXIS: Channel axis identifier	As 1	As 2	
FGROUP	**1. – 8.**				Variable F value reference: Definition of the axes to which the path feed refers Maximum number of axes: 8 The default setting for the F value reference is activated with FGROUP ( ) without parameters
	AXIS: Channel axis identifier				
SPLINEPATH	**1.**	**2. – 9.**			Definition of the spline grouping Maximum number of axes: 8
	INT: Spline grouping (must be 1)	AXIS: Geometry of additional identifier			
POLYPATH	**1.**	**2.**			Activation of the polynomial interpolation for selective axis groups
	STRING	STRING			

Coupled motion							
Identifier	**Parameter**						**Explanation**
	1.	**2.**	**3.**	**4.**	**5.**	**6.**	
TANG	AXIS: Axis name following axis	AXIS: Leading axis 1	AXIS: Leading axis 2	REAL: Coupling factor	CHAR: Option: "B": Tracking in the BCS "W": Tracking in the WCS	CHAR optimization: "S": Standard "P": Autom. with rounding clearance, angle tolerance	Tangential control: Define coupling The tangent for the follow-up is determined by the two master axes specified. The coupling factor specifies the relationship between a change in the angle of tangent and the following axis. It is usually 1.
TANGON	AXIS: Axis name following axis	REAL: Offset angle	REAL: Rounding clearance	REAL: Angular tolerance			Tangential control: Activate coupling
TANGOF	AXIS: Axis name following axis						Tangential control: Deactivate coupling
TLIFT	AXIS: Tracked axis						Tangential control: Activate intermediate block generation
TRAILON	AXIS: Following axis	AXIS: Leading axis	REAL: Coupling factor				Asynchronous coupled motion OFF
TRAILOF	AXIS: Following axis	AXIS: Leading axis					Deactivate coupled-axis motion
TANGDEL	AXIS: Following axis						Tangential control: Delete coupling

Curve tables						
Identifier	**Parameter**					**Explanation**
	1.	**2.**	**3.**	**4.**	**5.**	
CTABDEF	AXIS: Following axis	AXIS: Leading axis	INT: Table number	INT: Behavior at edges of the definition range	STRING: Specification of the storage location	Table definition ON The following motion blocks determine the curve table
CTABEND	AXIS: Following axis	AXIS: Leading axis	INT: Table number	INT: Behavior at edges of the definition range		Table definition OFF
CTABDEL	INT: Table number n	INT: Table number m	STRING: Specification of the storage location			Clear curve table
CTABLOCK	INT: Table number n					Locks the curve table with number n, i.e. this table cannot be deleted/ overwritten
CTABUNLOCK	INT: Table number n					Releases the table with the number n protected with CTABLOCK again
LEADON	AXIS: Following axis	AXIS: Leading axis	INT: Table number			Master value coupling ON
LEADOF	AXIS: Following axis	AXIS: Leading axis				Master value coupling OFF

Axial acceleration profile		
Identifier	**Parameter**	**Explanation**
	1. – 8.	
BRISKA	AXIS	Activate stepped axis acceleration for the programmed axes
SOFTA	AXIS	Activate jerk-limited axis acceleration for the programmed axes
DRIVEA	AXIS	Activate knee-shaped acceleration characteristic for the programmed axes
JERKA	AXIS	The acceleration behavior set in machine data $MA_AX_JERK_ENABLE is active for the programmed axes

Revolutional feedrate			
Identifier	**Parameter**		**Explanation**
FPRAON	**1.**	**2.**	Axial revolutional feedrate ON
	AXIS: Axis for which revolutional fee-drate is activated	AXIS: Axis/spindle from which revolu-tional feedrate is derived. If no axis has been program-med, the revolutional feedrate is derived from the master spin-dle.	
FPRAOF	**1. - n.**		Axial revolutional feedrate OFF The revolutional feedrate can be de-activated for several axes simultane-ously. You can program as many ax-es as are permitted in a block.
	AXIS: Axes for which revolutional feedrate is deactivated		
FPR	**1.**		Selection of a rotary axis or spindle from which the revolutional feedrate of the path is derived for G95. The setting made with FPR is modal.
	AXIS: Axis/spindle from which revolu-tional feedrate is derived. If no axis has been program-med, the revolutional feedrate is derived from the master spin-dle.		

Transformations				
Identifier	**Parameter**			**Explanation**
	1.	**2.**	**3.**	
TRACYL	REAL: Working diam-eter	INT: Number of the transformation		Cylinder: Peripheral surface transformation Several transformations can be set per channel. The transformation number specifies which transforma-tion is to be activated. If the 2nd parameter is omitted, the transformation grouping set via MD is activated.
TRANSMIT	INT: Number of the transformation			Transmit: Polar transformation Several transformations can be set per channel. The transformation number specifies which transforma-tion is to be activated. If the parameter is omitted, the transformation group defined in the MD is activated.

Transformations				
Identifier	**Parameter**			**Explanation**
	1.	**2.**	**3.**	
TRAANG	REAL: Angle	INT: Number of the transformation		Transformation inclined axis Several transformations can be set per channel. The transformation number specifies which transformation is to be activated. If the 2nd parameter is omitted, the transformation grouping set via MD is activated. If the angle is not programmed (TRAANG (,2) or TRAANG) the last angle applies modally.
TRAORI	INT: Number of the transformation			4-axis, 5-axis transformation Several transformations can be set per channel. The transformation number specifies which transformation is to be activated.
TRACON	INT: Number of the transformation	REAL: Further parameters, MD-dependent		Cascaded transformation The meaning of the parameters depends on the type of cascading.
TRAFOOF				Deactivate transformation
TRAFOON	STRING: Name of the transformation data set	REAL: Reference or working diameter (TRACYL only)	BOOL: With/without groove side offset (TRACYL only)	Activate a transformation defined with kinematic chains

Spindle			
Identifier	**Parameter**		**Explanation**
	1	**2. - n.**	
SPCON	INT: Spindle number	INT: Spindle number	Switch to position-controlled spindle operation.
SPCOF	INT: Spindle number	INT: Spindle number	Switch to speed-controlled spindle operation.
SETMS	INT: Spindle number		Declaration of spindle as master spindle for the current channel With SETMS( ), the machine data default applies automatically without any need for parameterization.

Grinding		
Identifier	Parameter	Explanation
	1.	
GWPSON	INT: Spindle number	Constant grinding wheel peripheral speed ON If the spindle number is not programmed, the grinding wheel peripheral speed for the spindle of the active tool is selected.
GWPSOF	INT: Spindle number	Constant grinding wheel peripheral speed OFF If the spindle number is not programmed, the grinding wheel peripheral speed for the spindle of the active tool is deselected.
TMON	INT: T number	Grinding-specific tool monitoring ON If no T number is programmed, monitoring is activated for the active tool.
TMOF	INT: T number	Tool monitoring OFF If no T number is programmed, monitoring is deactivated for the active tool.

Stock removal					
Identifier	Parameter				Explanation
	1.	2.	3.	4.	
CONTPRON	REAL [,11]: Contour table	CHAR: Machining type	INT: Number of relief cuts	INT: Status of the calculation	Activate reference preprocessing The contour programs or NC blocks which are called in the following steps are divided into individual movements and stored in the contour table. The number of relief cuts is returned.
CONTDCON	REAL [, 6]: Contour table	INT: Machining direction			Contour decoding The blocks for a contour are stored in a named table with one table line per block and coded to save memory.
EXECUTE	INT: Error status				Activate program execution This switches back to normal program execution from reference point editing mode or after setting up a protection area.

Execute table		
Identifier	Parameter	Explanation
	1.	
EXECTAB	REAL [11]: Element from motion table	Execute an element from a motion table

Protection areas						
Identifier	**Parameter**					**Explanation**
	1.	**2.**	**3.**	**4.**	**5.**	
CPROTDEF	INT: Number of the protection area	BOOL: TRUE: Tool-related protection area	INT: 0: 4th and 5th parameters are not evaluated 1: 4th parameter is evaluated 2: 5th parameter is evaluated 3: 4th and 5th parameters are evaluated	REAL: Limit in plus direction	REAL: Limit in minus direction	Definition of a channel-specific protection area
NPROTDEF	INT: Number of the protection area	BOOL: TRUE: Tool-related protection area	INT: 0: 4th and 5th parameters are not evaluated 1: 4th parameter is evaluated 2: 5th parameter is evaluated 3: 4th and 5th parameters are evaluated	REAL: Limit in plus direction	REAL: Limit in minus direction	Definition of a machine-specific protection area

Protection areas						
Identifier	**Parameter**					**Explanation**
	1.	**2.**	**3.**	**4.**	**5.**	
CPROT	INT: Number of the protection area	INT: Option 0: Protection area OFF 1: Preactivate protection area 2: Protection area ON 3: Preactivate protection area with conditional stop, only with protection areas active	REAL: Offset of the protection area in the first geometry axis	REAL: Offset of the protection area in the second geometry axis	REAL: Offset of the protection area in the third geometry axis	Channel-specific protection area ON/OFF
NPROT	INT: Number of the protection area	INT: Option 0: Protection area OFF 1: Preactivate protection area 2: Protection area ON 3: Preactivate protection area with conditional stop, only with protection areas active	REAL: Offset of the protection area in the first geometry axis	REAL: Offset of the protection area in the second geometry axis	REAL: Offset of the protection area in the third geometry axis	Machine-specific protection area ON/OFF

Preprocessing / single block		
Identifier	**Parameter**	**Explanation**
STOPRE		Preprocessing stop until all prepared blocks in the main run are executed
SBLOF		Suppress single-block processing
SBLON		Cancel suppression of the single-block processing

Interrupts		
Identifier	**Parameter**	**Explanation**
	1.	
DISABLE	INT: Number of the interrupt input	Deactivates the interrupt routine assigned to the specified hardware input. Fast retraction is not executed. The assignment between the hardware input and the interrupt routine made with SETINT remains valid and can be reactivated with ENABLE.
ENABLE	INT: Number of the interrupt input	Reactivation of the interrupt routine assignment deactivated with DISABLE.
CLRINT	INT: Number of the interrupt input	Delete assignment of interrupt routines and attributes to an interrupt input. The interrupt routine is deactivated and no reaction occurs when the interrupt is generated.

Synchronized actions		
Identifier	**Parameter**	**Explanation**
	1. - n.	
CANCEL	INT: Number of the synchronized action	Aborts the modal synchronized action with the specified ID. Several IDs, separated by commas, can be specified.

Function definition					
Identifier	**Parameter**				**Explanation**
	1.	**2.**	**3.**	**4.-7.**	
FCTDEF	INT: Function number	REAL: Lower limit value	REAL: Upper limit value	REAL: Coefficients a0 - a3	Define polynomial function This is evaluated in SYFCT or PUTFTOCF.

Communication			
Identifier	**Parameter**		**Explanation**
	1.	**2.**	
MMC	STRING: Command	CHAR: Acknowledgement mode*) "N": Without acknowledgement "S": Synchronous acknowledgement "A": Asynchronous acknowledgement	Command to HMI command Interpreter for the configuration of windows via NC program

) Commands are acknowledged on request from the executing component (channel, NC, etc.).

Program coordination					
Identifier	**Parameter**				**Explanation**
INIT	**1.**	**2.**	**3.**		Selection of an NC program for execution in a channel
	INIT: Channel number **or** channel name from MD20000*)	STRING: Path specification	CHAR: Acknowledgement mode**)		
	1. - n.				
START	INT: Channel number **or** channel name from MD20000*)				Start selected programs simultaneously in several channels from current program This command has no effect for the own channel
WAITE	INT: Channel number **or** channel name from MD20000*)				Wait for end of program in one or more other channels
	1.	**2. - n.**			
WAITM	INT: Marker number	INT: Channel number **or** channel name from MD20000*)			Wait until a marker is reached in the specified channels The previous block is terminated with exact stop
WAITMC	INT: Marker number	INT: Channel number **or** channel name from MD20000*)			Wait until a marker is reached in the specified channels An exact stop is initiated only if the other channels have not yet reached the marker
	1. - n.				
SETM	INT: Marker number				Set one or more markers for the channel coordination The processing in own channel is not affected by this
CLEARM	INT: Marker number				Delete one or more markers for the channel coordination The processing in own channel is not affected by this
	1. - n.				
WAITP	AXIS: Axis identifier				Wait until the specified positioning axes that were previously programmed with POSA, reach their programmed end point

Program coordination					
Identifier	**Parameter**				**Explanation**
WAITS	INT: Spindle number				Wait until the specified spindles that were previously programmed with SPOSA, reach their programmed end point
RET	**1.**	**2.**	**3.**	**4.**	End of subprogram with no function output to the PLC If the 1st parameter (jump destination) is specified, the return jump is first made to the block after the calling block. The target is then sought depending on the programming (RET or RETB) according to the following strategy: • RET: Search in the direction of the end of the program. A search is made toward the start of the program if the search was not successful. • RETB: Search in the direction of the start of the program. A search is made toward the end of the program if the search was not successful.
	INT (or STRING): Jump target (block no./ marker) for return	INT: 0: Return jump to jump destination from 1st par. > 0: Return to the following block	INT: Number of subprogram levels to be skipped	BOOL: Return to first block in the main program	
RETB	INT (or STRING): Jump target (block no./ marker) for return	INT: 0: Return jump to jump destination from 1st par. > 0: Return to the following block	INT: Number of subprogram levels to be skipped	BOOL: Return to first block in the main program	
	1. - n.				
GET	AXIS: Axis identifier ***)				Assign machine axis(axes) The specified axes must be released in the other channel with RELEASE
GETD	AXIS: Axis identifier ***)				Assign machine axis(axes) directly The specified axes must **not** be released with RELEASE
RELEASE	AXIS: Axis identifier ***)				Release machine axis(axes)
	1.	**2.**	**3.**	**4.**	
PUTFTOC	REAL: Offset value	INT: Parameter number	INT: Channel number **or** channel name from MD20000*)	INT: Spindle number	Change of fine tool compensation

Program coordination					
Identifier	**Parameter**				**Explanation**
PUTFTOCF	INT: No. of the function	VAR REAL: Reference value	INT: Pa-rameter number	INT: Channel number **or** channel name from MD20000*)	Change of fine tool compensation depending on a function defined with FCTDEF (max. 3rd degree polynomial) The number used here must be specified in FCTDEF
AXTOCHAN	**1.**	**2.**	**3. - n.**	**4. - m.**	Axes transferred to other channels
	AXIS: Axis identifi-er	INT: Channel number **or** channel name from MD20000*)	As 1 ...	As 2 ...	

) Instead of channel numbers, the channel names defined via MD20000 $MC_CHAN_NAME can also be programmed.

**) Commands are acknowledged on request from the executing component (channel, NC, etc.).

***) The SPI function can be used to program a spindle instead of an axis. E.g. GET(SPI(1))

Data access		
Identifier	**Parameter**	**Explanation**
CHANDATA	**1.**	Set channel number for channel data access (only permitted in the initialization block). The following access refers to the channel set with CHANDATA.
	INT: Channel num-ber	
NEWCONF		Accept changed machine data

Messages			
Identifier	**Parameter**		**Explanation**
	1.	**2.**	
MSG	STRING: Message	INT: Execution	Output arbitrary character string as message on the user in-terface
WRTPR	STRING: Character string	INT: Execution	Writing string in OPI variable

File access						
Identifier	**Parameter**					**Explanation**
READ	**1.**	**2.**	**3.**	**4.**	**5.**	Read blocks from file system
	VAR INT: Error	CHAR[160]: File name	INT: Start line of the file section to be read	INT: Number of lines to be read	VAR CHAR[255]: Variable array in which the read information is stored	
WRITE	**1.**	**2.**	**3.**	**4.**		Write block to file system (or to an external device/file)
	VAR INT: Error	CHAR[160]: File name	STRING: Device/file for external output	CHAR[200]: Block		
DELETE	**1.**	**2.**				Delete file
	VAR INT: Error	CHAR[160]: File name				

Alarms			
Identifier	**Parameter**		**Explanation**
	1.	**2.**	
SETAL	INT: Alarm number (cycle alarms)	STRING: Character string	Set alarm A character string with up to four parameters can be specified in addition to the alarm number. The following predefined parameters are available: %1 = channel number %2 = block number, label %3 = text index for cycle alarms %4 = additional alarm parameters

Tool management				
Identifier	**Parameter**			**Explanation**
	1.	**2.**		
DELDL	INT: T no.	INT: D no.		Delete all additive offsets of the tool edge (or of a tool if D is not specified)
DELT	STRING[32]: Tool identifier	INT: Duplo no.		Delete tool Duplo number can be omitted

Tool management							
Identifier	**Parameter**						**Explanation**
DELTC	INT: Data set no. n	INT: Data set no. m					Delete tool carrier data set number n to m
DZERO							Set D numbers of all tools of the TO unit assigned to the channel to invalid
	1.	**2.**	**3.**	**4.**	**5.**	**6.**	
GETFREELOC	VAR INT: Magazine no. (return value)	VAR INT: Location no. (return value)	INT: T no.	INT: Reference magazine no.	CHAR: Specification dep. on 4th parameter	INT: Reservation mode	Find empty location for a tool
	1.	**2.**					
GETSELT	VAR INT: T no. (return value)	INT: Spindle no.					Returns the T number of the tool preselected for the spindle
GETEXET	VAR INT: T no. (return value)	INT: Spindle no.					Returns the T number of the tool active from the point of view of the NC program
GETTENV	STRING: Name of the tool environment	INT ARRAY[3]: Return values					Reads the T, D and DL numbers stored in a tool environment
	1.	**2.**	**3.**	**4.**			
POSM	INT: No. of the location for positioning	INT: No. of the magazine to be moved	INT: Location no. of the internal magazine	INT: Magazine no. of the internal magazine			Position magazine
RESETMON	VAR INT: Status = result of the operation (return value)	INT: Internal T no.	INT: D no. of the tool	INT: Optional bit-coded parameter			Set actual value of tool to setpoint
SETDNO	**1.**	**2.**	**3.**				Set offset number (D) of the cutting edge of the tool (T)
	INT: T no.	INT: Cutting edge no.	INT: D no.				

Tool management						
Identifier	**Parameter**					**Explanation**
SETMTH	**1.**					Set tool carrier no.
	INT: Tool carrier no.					
SETPIECE	**1.**	**2.**				Decrement workpiece counter of the spindle Update the count monitoring data of the tools associated with the machining process
	INT: Value used when decrementing	INT: Spindle no.				
	1.	**2.**	**3.**	**4.**		
SETTA	VAR INT: Status = result of the operation (return value)	INT: Magazine no.	INT: Wear group no.	INT: Tool subgroup		Activate tool from wear group
SETTIA	VAR INT: Status = result of the operation (return value)	INT: Magazine no.	INT: Wear group no.	INT: Tool subgroup		Deactivate tool from wear group
TCA	**1.**	**2.**	**3.**			Tool selection/change irrespective of the tool status
	STRING[32]: Tool identifier	INT: Duplo no.	INT: Tool carrier no.			
TCI	**1.**	**2.**				Load tool from buffer into the magazine
	INT: No. of the buffer	INT: Tool carrier no.				
MVTOOL	**1.**	**2.**	**3.**	**4.**	**5.**	Language command to move tool
	INT: Status	INT: Magazine no.	INT: Location no.	INT: Magazine no. after moving	INT: Target location no. after moving	

Tool orientation				
Identifier	**Parameter**			**Explanation**
	1.	**2.**	**3.**	
ORIRESET	REAL: Initial setting, 1st geometry axis	REAL: Initial setting, 2nd geometry axis	REAL: Initial setting, 3rd geometry axis	Initial setting of the tool orientation

Synchronous spindle							
Identifier	**Parameter**						**Explanation**
	1.	**2.**	**3.**	**4.**	**5.**	**6.**	
COUPDEF	AXIS: Following spindle	AXIS: Leading spindle	REAL: Numerator of transmission ratio	REAL: Denominator of transmission ratio	STRING[8]: Block change behavior	STRING[2]: Coupling type	Define synchronous spindle grouping
COUPDEL	AXIS: Following spindle	AXIS: Leading spindle					Delete synchronous spindle grouping
COUPRES	AXIS: Following spindle	AXIS: Leading spindle					Reset coupling parameters to configured MD and SD values
COUPON	AXIS: Following spindle	AXIS: Leading spindle	REAL: Switch-on position of the following spindle				Switch-on synchronous spindle coupling If a switch-on position is specified for the following spindle (angular offset between FS and LS that refers -- absolutely or incrementally -- to the zero degree position of the LS in the positive direction of rotation), the coupling is only switched on when the specified position is crossed.
COUPONC	AXIS: Following spindle	AXIS: Leading spindle					Switch-on synchronous spindle coupling With COUPONC, the currently active speed of the following spindle is taken over when switching on the coupling (M3/M4 S..).

Synchronous spindle							
Identifier	Parameter						Explanation
	1.	2.	3.	4.	5.	6.	
COUPOF	AXIS: Following spindle	AXIS: Leading spindle	REAL: Switch-off position of the following spindle (absolute)	REAL: Switch-off position of the leading spindle (absolute)			Switch-off synchronous spindle coupling If positions are specified, the coupling is only cancelled when all the specified positions have been overtraveled The following spindle continues to revolve at the last speed programmed before deactivation of the coupling
COUPOFS	AXIS: Following spindle	AXIS: Leading spindle	REAL: Switch-off position of the following spindle (absolute)				Switch off the synchronous spindle coupling with stop of the following spindle If a position is specified, the coupling is only cancelled when the specified position is crossed
WAITC	AXIS: Following spindle	STRING [8]: Block change behavior	AXIS: Following spindle	STRING[8]: Block change behavior			Wait until the coupling block change criterion for the spindles (max. 2) has been fulfilled If the block change behavior is not specified, the block change behavior specified in the definition with COUPDEF applies

Electronic gear					
Identifier	Parameter				Explanation
EGDEL	1.				Delete coupling definition for the following axis
	AXIS: Following axis				
EGDEF	1.	2. / 4. / 6. / 8. / 10.	3. / 5. / 7. / 9. / 11.		Definition of an electronic gear
	AXIS: Following axis	AXIS: Leading axis	INT: Coupling type		

Electronic gear									
Identifier	**Parameter**								**Explanation**
EGON	**1.**	**2.**	**3. / 6. / 9. / 12. / 15.**	**4. / 7. / 10. / 13. / 16.**	**5. / 8. / 11. / 14. / 17.**				Electronic gear ON without synchronization
	AXIS: Following axis	STRING: Block change behavior	AXIS: Leading axis	REAL: Numerator of the coupling factor	REAL: Denominator of the coupling factor				
EGONSYN	**1.**	**2.**	**3.**	**4. / 8. / 12. / 16. / 20.**	**5. / 9. / 13. / 17. / 21.**	**6. / 10. / 14. / 18. / 22.**	**7. / 11. / 15. / 19. / 23.**		Electronic gear ON with synchronization
	AXIS: Following axis	STRING: Block change behavior	REAL: Synchronized position of the following axis	AXIS: Leading axis	REAL: Synchronized position of the leading axis	REAL: Numerator of the coupling factor	REAL: Denominator of the coupling factor		
EGONSYNE	**1.**	**2.**	**3.**	**4.**	**5. / 9. / 13. / 17. / 21.**	**6. / 10. / 14. / 18. / 22.**	**7. / 11. / 15. / 19. / 23.**	**8. / 12. / 16. / 20. / 24.**	Electronic gear ON with synchronization and specification of the approach mode
	AXIS: Following axis	STRING: Block change behavior	REAL: Synchronized position of the following axis	STRING: Approach mode	AXIS: Leading axis	REAL: Synchronized position of the leading axis	REAL: Numerator of the coupling factor	REAL: Denominator of the coupling factor	
EGOFS	**1.**	**2. - n.**							Turn off electronic gear selectively
	AXIS: Following axis	AXIS: Leading axis							
EGOFC	**1.**								Switch off electronic gear (version only for spindles)
	AXIS: Following spindle								

Nibbling					
Identifier	**Parameter**				**Explanation**
	1.	**2.**	**3.**	**4.**	
PUNCHAAC	REAL: Minimum hole spacing	REAL: Initial acceleration	REAL: Maximum hole spacing	REAL: Final acceleration	Activate travel-dependent acceleration

Information functions in the passive file system				
Identifier	**Parameter**			**Explanation**
	1.	**2.**	**3.**	
FILEDATE	VAR INT: Error message	CHAR[160]: File name	VAR CHAR[8]: Date in the format "dd.mm.yy"	Returns the date of the last write access to a file
FILETIME	VAR INT: Error message	CHAR[160]: File name	VAR CHAR[8]: Time in the format "hh.mm.ss"	Returns the time of the last write access to a file
FILESIZE	VAR INT: Error message	CHAR[160]: File name	VAR INT: File size	Returns the current size of a file
FILESTAT	VAR INT: Error message	CHAR[160]: File name	VAR CHAR[5]: Date in the format "rwxsd"	Returns the status of a file with respect to the following rights: • Read (r: read) • Write (w: write) • Execute (x: execute) • Show (s: show) • Delete (d: delete)
FILEINFO	VAR INT: Error message	CHAR[160]: File name	VAR CHAR[32]: Date in the format "rwxsd nnnnnnnn dd.mm.yy hh:mm:ss"	Returns the sum of the information for a file that can be read out via FILEDATE, FILETIME, FILESIZE, and FILESTAT

Axis container		
Identifier	**Parameter**	**Explanation**
	1. - n.	
AXCTSWE	AXIS: Axis container	Rotate axis container
AXCTSWED	AXIS: Axis container	Rotate axis container (command variant for commissioning!)
AXCTSWEC:	AXIS: Axis container	Cancel enable for axis container rotation

Master/slave coupling		
Identifier	Parameter	Explanation
	1. - n.	
MASLON	AXIS: Axis identifier	Switch on master/slave coupling
MASLOF	AXIS: Axis identifier	Separate master/slave coupling
MASLOFS	AXIS: Axis identifier	Separate master/slave coupling and automatically brake slave spindles
MASLDEF	AXIS: Axis identifier	Define master/slave coupling The last axis is the master axis
MASLDEL	AXIS: Axis identifier	Separate master/slave coupling and delete the definition of the grouping

Online tool length offset			
Identifier	Parameter		Explanation
	1.	2.	
TOFFON	AXIS: Offset direction	REAL: Offset value in offset direction	Activate online tool length offset in the specified offset direction
TOFFOF	AXIS: Offset direction		Reset online tool length offset in the specified offset direction

SERUPRO		
Identifier	Parameter	Explanation
IPTRLOCK		Start of untraceable program section
IPTRUNLOCK		End of search-suppressed program section

Retraction		
Identifier	Parameter	Explanation
	1. - n.	
POLFMASK	AXIS: Geometry or machine axis name	Enable axes for rapid retraction (without a connection between the axes)
POLFMLIN	AXIS: Geometry or machine axis name	Enable axes for linear rapid retraction

Retraction				
Identifier	**Parameter**			**Explanation**
POLFA	**1.**	**2.**	**3.**	Retraction position for single axes
	AXIS: Channel axis identifier	INT: Type	REAL: Value	

Collision avoidance			
Identifier	**Parameter**		**Explanation**
	1.		
PROTA	STRING: "R"		Request for a recalculation of the collision model
PROTS	**1.**	**2. - n.**	Set protection zone status
	CHAR: Status	STRING: Protection zone name	

17.6 Predefined procedures in synchronized actions

The following predefined procedures are only available in synchronized actions.

Synchronous procedures		
Identifier	**Parameter**	**Explanation**
STOPREOF		Revoke preprocessing stop A synchronized action with a STOPREOF command causes a preprocessing stop after the next output block (= block for the main run). The preprocessing stop is canceled with the end of the output block or when the STOPREOF condition is fulfilled. All synchronized action operations with the STOPREOF command are therefore interpreted as having been executed.
RDISABLE		Read-in disable
DELDTG	**1.** AXIS: Axis for axial delete distance-to-go (optional). If the axis is omitted, delete distance-to-go is triggered for the path distance.	Delete distance-to-go A synchronized action with a DELDTG command causes a preprocessing stop after the next output block (= block for the main run). The preprocessing stop is canceled with the end of the output block or when the first DELDTG condition is fulfilled. The axial distance to the destination point on an axial delete distance-to-go is stored in $AA_DELT[axis]; the distance-to-go is stored in $AC_DELT.

Program coordination of technology cycles		
Identifier	**Parameter**	**Explanation**
	1.	
LOCK	INT: ID of the synchronized action to be disabled	Lock synchronized action with ID or stop technology cycle One or more IDs can be programmed
UNLOCK	INT: ID of the synchronized action to be unlocked	Unlock synchronized action with ID or continue technology cycle One or more IDs can be programmed
ICYCON		Each block of a technology cycle is processed in a separate interpolation cycle following ICYCON
ICYCOF		All blocks of a technology cycle are processed in one interpolation cycle following ICYCOF

Polynomial functions						
Identifier	**Parameter**					**Explanation**
SYNFCT	**1.**	**2.**	**3.**			If the condition in the motion-synchronous action is fulfilled, the polynomial determined by the first expression is evaluated at the input variable. The upper and lower range of the value is limited and the result variable is assigned.
	INT: Number of the polynomial function defined with FCTDEF	VAR REAL: Result variable *)	VAR REAL: Input variable **)			
FTOC	**1.**	**2.**	**3.**	**4.**	**5.**	Change of fine tool compensation depending on a function defined with FCTDEF (max. 3rd degree polynomial). The number used here must be specified in FCTDEF.
	INT: Number of the polynomial function defined with FCTDEF	VAR REAL: Input variable **)	INT: Length 1, 2, 3	INT: Channel number	INT: Spindle number	

*) Only special system variables are permissible as a result variable (see Function Manual Synchronized Actions).

**) Only special system variables are permissible as input variable (see Function Manual Synchronized Actions).

17.7 Predefined functions

The call of a predefined function triggers the execution of a predefined NC function, which in contrast to the predefined procedure, supplies a return value. The call of the predefined function can be an operand in an expression.

Coordinate system						
Identifier	**Return value**	**Parameter**				**Explanation**
		1.	**2.**	**3. - 15.**	**4. - 16.**	
CTRANS	FRAME	AXIS: Axis identifier	REAL: Offset	AXIS: Axis identifier	REAL: Offset	Translation: Zero offset COARSE for multiple axes
CFINE	FRAME	AXIS: Axis identifier	REAL: Offset	AXIS: Axis identifier	REAL: Offset	Translation: Zero offset for FINE multiple axes
CSCALE	FRAME	AXIS: Axis identifier	REAL: Scale factor	AXIS: Axis identifier	REAL: Scale factor	Scale: Scale factor for multiple axes
		1.	**2.**	**3. and 5.**	**4. and 6.**	
CROT	FRAME	AXIS: Axis identifier	REAL: Rotation	AXIS: Axis identifier	REAL: Rotation	Rotation: Rotation of the current coordinate system Maximum number of parameters: 6 (one axis identifier and one value per geometry axis)
CROTS	FRAME	AXIS: Axis identifier	REAL: Rotation with solid angle	AXIS: Axis identifier	REAL: Rotation with solid angle	Rotation: Rotation of the current coordinate system with solid angle Maximum number of parameters: 6 (one axis identifier and one value per geometry axis)
CMIRROR		**1.**	**2. – 8.**			Mirror: Mirror on a coordinate axis
	FRAME	AXIS	AXIS			

Coordinate system						
Identifier	**Return value**	**Parameter**				**Explanation**
		1.	**2.**			
CRPL	FRAME	INT: Rotary axis	REAL: Angle of rotation			Frame rotation in any plane
ADDFRAME	INT: 0: OK 1: Specified target (string) is wrong 2: Target frame is not configured 3: Rotation in frame is not permitted	FRAME: Additively measured or calculated frame	STRING: Specified target frame			Calculates the target frame specified by the string The target frame is calculated in such a way that the new complete frame appears as a chain of the old complete frame and the transferred frame.
INVFRAME	FRAME	**1.**				Calculates the inverse frame from a frame The frame chaining of a frame with its inverse frame always results in a zero frame
		FRAME				
MEAFRAME	FRAME	**1.**	**2.**	**3.**		Frame calculation from 3 measuring points in space
		REAL[3,3]: Coordinates of the measured spatial points	REAL[3,3]: Coordinates of the specified points	VAR REAL: Variable with which the information on the quality of FRAME calculation is returned		

Geometry functions					
Identifier	Return value	Parameter			Explanation
		1.	2.	3.	
CALCDAT	BOOL: Error status	VAR REAL [n, 2]: Table (abscissa, ordinate) of points 1 to n	INT: Number of points	VAR REAL [3]: Result: Abscissa, ordinate and radius of calculated circle center point	Calculates the center point coordinates and the radius of the circle from 3 or 4 points The points must be different.
INTERSEC	BOOL: Error status	VAR REAL [11]: First contour element	VAR REAL [11]: Second contour element	VAR REAL [2]: Result vector for the intersection coordinates: Abscissa and ordinate	Calculates the intersection coordinates between two contour elements. The error status indicates whether an intersection was found.

Curve table functions								
Identifier	Return value	Parameter						Explanation
		1.	2.	3.	4.	5.	6.	
CTAB	REAL: Following axis position	REAL: Leading axis position	INT: Table number	VAR REAL[ ]: Pitch result	AXIS: Following axis for scaling	AXIS: Leading axis for scaling		Determines the following axis position to the specified leading axis position from the curve table. If parameters 4/5 are not programmed, calculation is with standard scaling.
CTABINV	REAL: Leading axis position	REAL: Following axis position	REAL: Leading position	INT: Table number	VAR REAL[ ]: Pitch result	AXIS: Following axis for scaling	AXIS: Leading axis for scaling	Determines the leading axis position to the specified following axis position from the curve table. If parameters 5/6 are not programmed, calculation is with standard scaling.
CTABID	INT: Curve table number	INT: Entry number in memory	STRING: Storage location: "SRAM", "DRAM"					Determines the curve table number entered under the specified number in the memory.

Curve table functions								
Identifier	Return value	Parameter						Explanation
		1.	2.	3.	4.	5.	6.	
CTABISLOCK	INT: Lock state	INT: Table number						Determines the lock state of the curve table: > 0: Table is locked 1: CTABLOCK 2: Active coupling 3: CTABLOCK and active coupling 0: Table is not locked -1: Table does not exist
CTABEXISTS	INT: Existence	INT: Table number						Determines the existence of the curve table in the static or dynamic NC memory: 0: FALSE 1: TRUE
CTABMEMTYP	INT: Storage location	INT: Table number						Determines the storage location of the curve table: 1: DRAM 0: SRAM -1: Table does not exist
CTABPERIOD	INT: Periodicity	INT: Table number						Determines the periodicity of the curve table: 0: Not periodic 1: Periodic in leading axis 2: Periodic in leading and following axis -1: Table does not exist
CTABNO	INT: Number of curve tables							Determines the number of defined curve tables (in static and dynamic NC memory)
CTABNOMEM	INT: Number of curve tables	STRING: Storage location: "SRAM", "DRAM"						Determines the number of defined curve tables in the specified memory

Curve table functions								
Identifier	**Return value**	**Parameter**						**Explanation**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
CTABFNO	INT: Number of tables	STRING: Storage location: "SRAM", "DRAM"						Determines the number of curve tables still possible in the specified memory
CTABSEG	INT: Number of curve segments	STRING: Storage location: "SRAM", "DRAM"	STRING: Segment type: "L": Linear "P": Polynomial					Determines the number of curve segments used of the specified segment type in the specified memory >=0: Number -1: Invalid memory type If parameter 2 is not programmed, the sum of the linear and polynomial segments is output.
CTABFSEG	INT: Number of curve segments	STRING: Storage location: "SRAM", "DRAM"	STRING: Segment type: "L": Linear "P": Polynomial					Determines the number of still possible curve segments of the specified segment type in the specified memory >=0: Number -1: Invalid memory type
CTABSEGID	INT: Number of curve segments	INT: Table number	STRING: Segment type: "L": Linear "P": Polynomial					Determines the number of curve segments of the specified segment type that are used by the curve table >=0: Number -1: Table does not exist
CTABMSEG	INT: Number of curve segments	STRING: Storage location: "SRAM", "DRAM"	STRING: Segment type: "L": Linear "P": Polynomial					Determines the maximum possible number of curve segments of the specified segment type in the specified memory >=0: Number -1: Table does not exist

Curve table functions								
Identifier	**Return value**	**Parameter**						**Explanation**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
CTABPOL	INT: Number of curve polynomials	STRING: Storage location: "SRAM", "DRAM"						Determines the number of used curve polynomials in the specified memory >=0: Number -1: Table does not exist
CTABPOLID	INT: Number of curve polynomials	INT: Table number						Determines the number of curve polynomials used by the curve table >=0: Number -1: Table does not exist
CTABFPOL	INT: Number of curve polynomials	STRING: Storage location: "SRAM", "DRAM"						Determines the maximum possible number of curve polynomials in the specified memory: >=0: Number -1: Table does not exist
CTABMPOL	INT: Number of curve polynomials	STRING: Storage location: "SRAM", "DRAM"						Determines the maximum possible number of curve polynomials in the specified memory: >=0: Number -1: Table does not exist
CTABSSV	REAL: Following axis position	REAL: Leading axis position	INT: Table number	VAR REAL[ ]: Pitch result	AXIS: Following axis for scaling	AXIS: Leading axis for scaling		Determines the following axis position at the start of the curve segment belonging to the specified leading axis value
CTABSEV	REAL: Following axis position	REAL: Leading axis position	INT: Table number	VAR REAL[ ]: Pitch result	AXIS: Following axis for scaling	AXIS: Leading axis for scaling		Determines the following axis position at the end of the curve segment belonging to the specified leading axis value
CTABTSV	REAL: Following axis position	INT: Table number	VAR REAL[ ]: Pitch result at start of the table	AXIS: Following axis				Determines the following axis position at the start of the curve table.

Curve table functions								
Identifier	Return value	Parameter						Explanation
		1.	2.	3.	4.	5.	6.	
CTABTEV	REAL: Following axis position	INT: Table number	VAR REAL[ ]: Pitch result at end of the table	AXIS: Following axis				Determines the following axis position at the end of the curve table.
CTABTSP	REAL: Leading axis position	INT: Table number	VAR REAL[ ]: Pitch result at start of the table	AXIS: Leading axis				Determines the leading axis position at the start of the curve table.
CTABTEP	REAL: Leading axis position	INT: Table number	VAR REAL[ ]: Pitch result at end of the table	AXIS: Leading axis				Determines the leading axis position at the end of the curve table.
CTABTMIN	REAL: Minimum value	INT: Table number	REAL: Leading value interval lower limit	REAL: Leading value interval upper limit	AXIS: Following axis	AXIS: Leading axis		Determines the minimum value of the following axis in the entire definition range of the curve table or in a defined interval
CTABTMAX	REAL: Maximum value	INT: Table number	REAL: Leading value interval lower limit	REAL: Leading value interval upper limit	AXIS: Following axis	AXIS: Leading axis		Determines the maximum value of the following axis in the entire definition range of the curve table or in a defined interval
Note: The curve table functions can also be programmed in synchronized actions.								

Axis functions						
Identifier	Return value	Parameter				
		1.	2.	3.	4.	Explanation
AXNAME	AXIS: Axis identifier	STRING [ ]: Input string				Converts input string into axis identifier
AXSTRING	STRING[ ]: Axis name	AXIS: Axis identifier				Converts axis identifier into string

Axis functions						
Identifier	**Return value**	**Parameter**				
		1.	**2.**	**3.**	**4.**	**Explanation**
ISAXIS	BOOL: Axis present (TRUE) or not (FALSE)	INT: Number of the geometry axis (1 to 3)				Checks whether the geometry axes 1 to 3 specified as parameters are present in accordance with machine data MD20050 $MC_AXCONF_GEOAX_ASSIGN_TAB
SPI	AXIS: Axis identifier	INT: Spindle number				Converts spindle number into axis identifier
AXTOSPI	INT: Spindle number	AXIS: Axis identifier				Converts axis identifier into spindle number
MODAXVAL	REAL: modulo value	AXIS: Axis identifier	REAL: Axis position			From the entered axis position, calculates the modulo rest If the specified axis is not a modulo axis, the axis position is returned unchanged.
POSRANGE	BOOL: Position setpoint within the position window (TRUE) or not (FALSE)	AXIS: Axis identifier	REAL: Reference position in the coordinate system	REAL: Position window width	INT: Coordinate system	Determines whether the position setpoint of an axis is located in a window at a predefined reference position

Tool management					
Identifier	**Return value**	**Parameter**			**Explanation**
		1.	**2.**	**3.**	
CHKDM	INT: Status Result of the check:	INT: Magazine number	INT: D number		Checks the uniqueness of the D number within a magazine
CHKDNO	INT: Status Result of the check:	INT: T number of the 1st tool	INT: T number of the 2nd tool	INT: D number	Checks the uniqueness of the D number
GETACTT	INT: Status	INT: T number	STRING [32]: Tool name		Determines the active tool from a group of tools with the same name
GETACTTD	INT: Status Result of the check:	VAR INT: T number found (return value)	INT: D number		Determines the T number associated with an absolute D number
GETDNO	INT: D number	INT: T number	INT: Cutting edge number		Determines the D number of the cutting edge of tool T

Tool management					
Identifier	**Return value**	**Parameter**			**Explanation**
		1.	**2.**	**3.**	
GETT	INT: T number	STRING [32]: Tool name	INT: Duplo number		Determines the T number for the tool name
NEWT	INT: T number	STRING [32]: Tool name	INT: Duplo number		Sets up a new tool (provides the tool data) The duplo number can be omitted.
TOOLENV	INT: Status	STRING: Name			Stores the tool environment with the specified name in the static NC memory
DELTOOLENV	INT: Status	STRING: Name			Deletes the tool environment with the specified name in the static NC memory Deletes all tool environments if no name is specified.
GETTENV	INT: Status	STRING: Name	VAR INT: T number [0] D number [1] DL number [2]		Determines the T number, D number, and DL number from a tool environment with the specified name

Arithmetic					
Identifier	**Return value**	**Parameter**			**Explanation**
		1.	**2.**	**3.**	
SIN	REAL	REAL			Sine
ASIN	REAL	REAL			Arc sine
COS	REAL	REAL			Cosine
ACOS	REAL	REAL			Arc cosine
TAN	REAL	REAL			Tangent
ATAN2	REAL	REAL	REAL		Arc tangent 2
SQRT	REAL	REAL			Square root
POT	REAL	REAL			Square
TRUNC	REAL	REAL			Integer component
ROUND	REAL	REAL			Round down
ROUNDUP	REAL	REAL			Round up
ABS	REAL	REAL			Absolute value
LN	REAL	REAL			Natural logarithm
EXP	REAL	REAL			Exponential function e^x
MINVAL	REAL	REAL	REAL		Determines the smaller value of two parameters
MAXVAL	REAL	REAL	REAL		Determines the larger value of two parameters

Arithmetic					
Identifier	**Return value**	**Parameter**			**Explanation**
		1.	**2.**	**3.**	
BOUND	REAL: Check status	REAL: Lower limit	REAL: Upper limit	REAL: Reference value	Determines whether the reference value is within the limits.
Note: The arithmetic functions can also be programmed in synchronized actions. These arithmetic functions are calculated and evaluated in the main run. The synchronized action parameter $AC_PARAM[<n>] can also be used for calculations and as buffer.					

String functions					
Identifier	**Return value**	**Parameter**			**Explanation**
		1.	**2.**	**3.**	
ISNUMBER	BOOL	STRING: Input string			Checks whether the input string can be converted to a number.
NUMBER	REAL	STRING: Input string			Converts the input string into a number.
TOUPPER	STRING	STRING: Input string			Converts the input string into upper case
TOLOWER	STRING	STRING: Input string			Converts the input string into lower case
STRLEN	INT	STRING: Input string			Determines the length of the input string up to the end of the string (/0)
INDEX	INT	STRING: Input string	CHAR: Search characters		Determines the position of the character in the input string from left to right. The 1st character of the string from the left has the index 0.
RINDEX	INT	STRING: Input string	CHAR: Search characters		Determines the position of the character in the input string from right to left. The 1st character of the string from the right has the index 0.
MINDEX	INT	STRING: Input string	STRING: Search character		Determines the position of a character specified in the 2nd parameter in the input string from left to right. The 1st character of the input string from the left has the index 0.
SUBSTR	STRING	STRING: Input string	INT	INT	Determines the substring of the input string, defined by the start character (2nd parameter) and number of characters (3rd parameter).
SPRINT	STRING	STRING: Input string			Determines the formatted input string

Functions for measuring cycles								
Identifier	**Return value**	**Parameter**						**Explanation**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
CALCPOSI	INT: Status	REAL[3]: Starting position in the WCS	REAL[3]: Incremental path specification in relation to the starting position	REAL[5]: Minimum distances to the monitoring limits	REAL[3]: Return array for the poss. incr. distance	BOOL: Conversion of the measuring system Yes/ No	INT: Type of limit monitoring	Checks whether the geometry axes can traverse a defined path without violating the axis limits starting from a specified starting point. If the defined path cannot be traversed without violating limits, the maximum permissible value is returned.
GETTCOR	INT: Status	REAL [11]:	STRING: Tool length component: Coordinate system	STRING: Name of the tool environment	INT: Internal T no. of the tool	INT: Cutting-edge number (D no.) of the tool	INT: Number of the location-dependent offset (DL no. of the tool)	Determines the tool lengths and tool length components from tool environment or current environment
LENTOAX	INT: Status	INT[3]: Axis assignment of the geometry axes	REAL[3]: Matrix for mapping the tool lengths in the coordinate system	STRING: Coordinate system for the assignment				Determines information about the assignment of the tool lengths L1, L2, L3 of the active tool to abscissa, ordinate, applicate. The assignment to the geometry axes is affected by frames and the active plane (G17 - 19).

SETTCOR	INT: Status	**1.**	**2.**	**3.**	**4.**	**5.**	**6.**	**7.**	**8.**	**9.**	
		REAL [3]: Offset vector in space	STR.: Component identifier	INT: Component(s) to be corrected 0 - 11	INT: Type of write operation 0 - 3	INT: Index of the geometry axis	STRING: Name of the tool environment	INT: int. T No. of the tool	INT: D no. of the tool	INT: DL no. of the tool	Changes the tool components, considering all supplementary conditions that are included in the evaluation of the individual components

Other functions								
Identifier	**Return value**	**Parameter**						**Explanation**
		1.	**2.**	**3.**	**4.**	**5.**	**6.**	
STRINGIS	INT: Information about the string	STRING: Name of the element to be checked						Checks whether the specified string is available as element of the NC programming language in the current language scope.
ISVAR	BOOL: Variable known Yes/No	STRING: Name of the variable						Checks whether the transfer parameter contains a variable known in the NC (machine data, setting data, system variable, general variables such as GUDs).
GETVARTYP	INT: Data type	STRING: Name of the variable						Determines the data type of a system/user variable
GETVARPHU	INT: Numeric value of the physical unit	STRING: Name of the variable						Determines the physical unit of a system/ user variable
GETVARAP	INT: Protection level for access	STRING: Name of the variable	STRING: Type of access					Determines the access right to a system/user variable
GETVARLIM	INT: Status	STRING: Name of the variable	CHAR: Specifies which limit value should be read out	VAR REAL: Return of the limit value				Determines the lower/ upper limit value of a system/user variable
GETVARDFT	INT: Status	STRING: Name of the variable	VAR REAL / STRING/ FRAME: Return of the default value	INT: Index to the first dimension (optional)	INT: Index to the second dimension (optional)	INT: Index to the third dimension (optional)		Determines the default value of a system/user variable
COLLPAIR	INT: Check result	STRING: Name of the 1st protection area	STRING: Name of the 2nd protection area	BOOL: Alarm suppression (optional)				Checks whether part of a collision pair

Other functions								
Identifier	Return value	Parameter						Explanation
		1.	2.	3.	4.	5.	6.	
PROTD	REAL: Clearance of the two protection zones	STRING: Name of the 1st protection area	STRING: Name of the 2nd protection area	VAR REAL: Return value: 3-dimensional clearance vector	BOOL: Measuring system for clearance and clearance vector (optional)			Determines the clearance of the two specified protection areas.
DELOBJ	INT: Error number	STRING: Component type to be deleted	INT: Start index of the components to be deleted (optional)	INT: End index of the components to be deleted (optional)	BOOL: Alarm suppression (optional)			Deletes elements from kinematic chains, protection areas, protection area elements, collision pairs and transformation data
NAMETOINT	INT: System variable index	STRING: Name of the system variable array	STRING: Character string / name	BOOL: Alarm suppression (optional)				Determines the associated system variable index based on the character string
ORISOLH	INT: Error number	INT: Controls the behavior of the function	REAL: First angle	REAL: Second angle				Helps the user to set the rotary axis positions of a machine so that a turning tool can be brought into a defined, kinematic-independent position relative to the workpiece. Requirement: A 6-axis transformation is active that has been parameterized with kinematic chains.
CORRTRAFO	INT: Error number	REAL: Correction vector	INT: Element to be modiifed	INT: Correction mode	BOOL: Alarm suppression (optional)			Modifies offset vectors or direction vectors of the orientation axes in the kinematic model of the machine.

17.8 Currently set language in the HMI

The table below lists all of the languages available at the user interface.

The currently set language can be queried in the part program and in the synchronized actions using the following system variable:

$AN_LANGUAGE_ON_HMI = <value>

<value>	Language	Language code
1	German (Germany)	GER
2	French	FRA
3	English (Great Britain)	ENG
4	Spanish	ESP
6	Italian	ITA
7	Dutch	NLD
8	Simplified Chinese	CHS
9	Swedish	SVE
18	Hungarian	HUN
19	Finnish	FIN
28	Czech	CSY
50	Portuguese (Brazil)	PTB
53	Polish	PLK
55	Danish	DAN
57	Russian	RUS
68	Slovakian	SKY
72	Rumanian	ROM
80	Traditional Chinese	CHT
85	Korean	KOR
87	Japanese	JPN
89	Turkish	TRK

Note

$AN_LANGUAGE_ON_HMI is updated:

- after the system boots.
- after NC and/or PLC reset.
- after switching over to another NC within the scope of M2N.
- after changing over the language on the HMI.

Appendix A

A.1 List of abbreviations

A	
O	Output
ADI4	(Analog drive interface for 4 axes)
AC	Adaptive Control
ALM	Active Line Module
ARM	Rotating induction motor
AS	Automation system
ASCII	American Standard Code for Information Interchange: American coding standard for the exchange of information
ASIC	Application-Specific Integrated Circuit: User switching circuit
ASUB	Asynchronous subprogram
AUXFU	Auxiliary function: Auxiliary function
STL	Statement List
UP	User Program

B	
OP	Operating Mode
BAG	Mode group
BCD	Binary Coded Decimals: Decimal numbers encoded in binary code
BERO	Contact-less proximity switch
BI	Binector Input
BICO	Binector Connector
BIN	BINary files: Binary files
BIOS	Basic Input Output System
BCS	Basic Coordinate System
BO	Binector Output
OPI	Operator Panel Interface

C	
CAD	Computer-Aided Design
CAM	Computer-Aided Manufacturing
CC	Compile Cycle: Compile cycles
CEC	Cross Error Compensation
CI	Connector Input
CF Card	Compact Flash Card

C	
CNC	Computerized Numerical Control: Computer-Supported Numerical Control
CO	Connector Output
CoL	Certificate of License
COM	Communication
CPA	Compiler Projecting Data: Configuring data of the compiler
CRT	Cathode Ray Tube: picture tube
CSB	Central Service Board: PLC module
CU	Control Unit
CP	Communication Processor
CPU	Central Processing Unit: Central processing unit
CR	Carriage Return
CTS	Clear To Send: Ready to send signal for serial data interfaces
CUTCOM	Cutter radius Compensation: Tool radius compensation

D	
DAC	Digital-to-Analog Converter
DB	Data Block (PLC)
DBB	Data Block Byte (PLC)
DBD	Data Block Double word (PLC)
DBW	Data Block Word (PLC)
DBX	Data block bit (PLC)
DDE	Dynamic Data Exchange
DDS	Drive Data Set: Drive data set
DIN	Deutsche Industrie Norm
DIO	Data Input/Output: Data transfer display
DIR	Directory: Directory
DLL	Dynamic Link Library
DO	Drive Object
DPM	Dual Port Memory
DPR	Dual Port RAM
DRAM	Dynamic memory (non-buffered)
DRF	Differential Resolver Function: Differential revolver function (handwheel)
DRIVE-CLiQ	Drive Component Link with IQ
DRY	Dry Run: Dry run feedrate
DSB	Decoding Single Block: Decoding single block
DSC	Dynamic Servo Control / Dynamic Stiffness Control
DW	Data Word
DWORD	Double Word (currently 32 bits)

E	
I	Input
EES	Execution from External Storage
I/O	Input/Output
ENC	Encoder: Actual value encoder
EFP	Compact I/O module (PLC I/O module)
ESD	Electrostatic Sensitive Devices
EMC	ElectroMagnetic Compatibility
EN	European standard
ENC	Encoder: Actual value encoder
EnDat	Encoder interface
EPROM	Erasable Programmable Read Only Memory: Erasable, electrically programmable read-only memory
ePS Network Services	Services for Internet-based remote machine maintenance
EQN	Designation for an absolute encoder with 2048 sine signals per revolution
ES	Engineering System
ESR	Extended Stop and Retract
ETC	ETC key ">"; softkey bar extension in the same menu

F	
FB	Function Block (PLC)
FC	Function Call: Function Block (PLC)
FEPROM	Flash EPROM: Read and write memory
FIFO	First In First Out: Memory that works without address specification and whose data is read in the same order in which they was stored
FIPO	Fine interpolator
FPU	Floating Point Unit: Floating Point Unit
CRC	Cutter Radius Compensation
FST	Feed Stop: Feedrate stop
FBD	Function Block Diagram (PLC programming method)
FW	Firmware

G	
GC	Global Control (PROFIBUS: Broadcast telegram)
GDIR	Global part program memory
GEO	Geometry, e.g. geometry axis
GIA	Gear Interpolation dAta: Gear interpolation data
GND	Signal Ground
GP	Basic program (PLC)
GS	Gear Stage
GSD	Device master file for describing a PROFIBUS slave

G	
GSDML	Generic Station Description Markup Language: XML-based description language for creating a GSD file
GUD	Global User Data: Global user data

H	
HEX	Abbreviation for hexadecimal number
AuxF	Auxiliary function
HLA	Hydraulic linear drive
HMI	Human Machine Interface: SINUMERIK user interface
MSD	Main Spindle Drive
HW	Hardware

I	
IBN	Commissioning
ICA	Interpolatory compensation
IM	Interface Module: Interconnection module
IMR	Interface Module Receive: Interface module for receiving data
IMS	Interface Module Send: Interface module for sending data
INC	Increment: Increment
INI	Initializing Data: Initializing data
IPO	Interpolator
ISA	Industry Standard Architecture
ISO	International Standardization Organization

J	
JOG	Jogging: Setup mode

K	
K_V	Gain factor of control loop
K_p	Proportional gain
$K_Ü$	Transformation ratio
LAD	Ladder Diagram (PLC programming method)

L	
LAI	Logic Machine Axis Image: Logical machine axes image
LAN	Local Area Network
LCD	Liquid Crystal Display: Liquid crystal display
LED	Light Emitting Diode: Light-emitting diode
LF	Line Feed

L	
PMS	Position Measuring System
LR	Position controller
LSB	Least Significant Bit: Least significant bit
LUD	Local User Data: User data (local)

M	
MAC	Media Access Control
MAIN	Main program: Main program (OB1, PLC)
MB	Megabyte
MCI	Motion Control Interface
MCIS	Motion Control Information System
MCP	Machine Control Panel: Machine control panel
MD	Machine Data
MDA	Manual Data Automatic: Manual input
MDS	Motor Data Set: Motor data set
MSGW	Message Word
MCS	Machine Coordinate System
MM	Motor Module
MPF	Main Program File: Main program (NC)
MCP	Machine control panel

N	
NC	Numerical Control: Numerical control with block preparation, traversing range, etc.
NCU	Numerical Control Unit: NC hardware unit
NRK	Name for the operating system of the NC
IS	Interface Signal
NURBS	Non-Uniform Rational B-Spline
WO	Work Offset
NX	Numerical Extension: Axis expansion board

O	
OB	Organization block in the PLC
OEM	Original Equipment Manufacturer
OP	Operator Panel: Operating equipment
OPI	Operator Panel Interface: Interface for connection to the operator panel
OPT	Options: Options
OLP	Optical Link Plug: Fiber optic bus connector
OSI	Open Systems Interconnection: Standard for computer communications

P	
PIQ	Process Image Output
PII	Process Image Input
PC	Personal Computer
PCIN	Name of the SW for data exchange with the control
PCMCIA	Personal Computer Memory Card International Association: Plug-in memory card standardization
PCU	PC Unit: PC box (computer unit)
PG	Programming device
PKE	Parameter identification: Part of a PIV
PIV	Parameter identification: Value (parameterizing part of a PPO)
PLC	Programmable Logic Control: Adaptation control
PN	PROFINET
PNO	PROFIBUS user organization
PO	POWER ON
POU	Program Organization Unit
POS	Position/positioning
POSMO A	Positioning Motor Actuator: Positioning motor
POSMO CA	Positioning Motor Compact AC: Complete drive unit with integrated power and control module as well as positioning unit and program memory; AC infeed
POSMO CD	Positioning Motor Compact DC: Like CA but with DC infeed
POSMO SI	Positioning Motor Servo Integrated: Positioning motor, DC infeed
PPO	Parameter Process data Object: Cyclic data telegram for PROFIBUS DP transmission and "Variable speed drives" profile
PPU	Panel Processing Unit (central hardware for a panel-based CNC, e.g SINUMERIK 828D)
PROFIBUS	Process Field Bus: Serial data bus
PRT	Program Test
PSW	Program control word
PTP	Point-To-Point: Point-To-Point
PUD	Program global User Data: Program-global user variables
PZD	Process data: Process data part of a PPO

Q	
QEC	Quadrant Error Compensation

R	
RAM	Random Access Memory: Read/write memory
REF	REFerence point approach function
REPOS	REPOSition function
RISC	Reduced Instruction Set Computer: Type of processor with small instruction set and ability to process instructions at high speed
ROV	Rapid Override: Input correction

R	
RP	R Parameter, arithmetic parameter, predefined user variable
RPA	R Parameter Active: Memory area in the NC for R parameter numbers
RPY	Roll Pitch Yaw: Rotation type of a coordinate system
RTLI	Rapid Traverse Linear Interpolation: Linear interpolation during rapid traverse motion
RTS	Request To Send: Control signal of serial data interfaces
RTCP	Real Time Control Protocol

S	
SA	Synchronized Action
SBC	Safe Brake Control: Safe Brake Control
SBL	Single Block: Single block
SBR	Subroutine: Subprogram (PLC)
SD	Setting Data
SDB	System Data Block
SEA	Setting Data Active: Identifier (file type) for setting data
SERUPRO	SEarch RUn by PROgram test: Block search, program test
SFB	System Function Block
SFC	System Function Call
SGE	Safety-related input
SGA	Safety-related output
SH	Safe standstill
SIM	Single in Line Module
SK	Softkey
SKP	Skip: Function for skipping a part program block
SLM	Synchronous Linear Motor
SM	Stepper Motor
SMC	Sensor Module Cabinet Mounted
SME	Sensor Module Externally Mounted
SMI	Sensor Module Integrated
SPF	Sub Routine File: Subprogram (NC)
PLC	Programmable Logic Controller
SRAM	Static RAM (non-volatile)
TNRC	Tool Nose Radius Compensation
SRM	Synchronous Rotary Motor
LEC	Leadscrew Error Compensation
SSI	Serial Synchronous Interface: Synchronous serial interface
SSL	Block search
STW	Control word
GWPS	Grinding Wheel Peripheral Speed
SW	Software
SYF	System Files: System files
SYNACT	SYNchronized ACTion: Synchronized Action

T	
TB	Terminal Board (SINAMICS)
TCP	Tool Center Point: Tool tip
TCP/IP	Transport Control Protocol / Internet Protocol
TCU	Thin Client Unit
TEA	Testing Data Active: Identifier for machine data
TIA	Totally Integrated Automation
TM	Terminal Module (SINAMICS)
TO	Tool Offset: Tool offset
TOA	Tool Offset Active: Identifier (file type) for tool offsets
TRANSMIT	Transform Milling Into Turning: Coordination transformation for milling operations on a lathe
TTL	Transistor-Transistor Logic (interface type)
TZ	Technology cycle

U	
UFR	User Frame: Work offset
SR	Subprogram
USB	Universal Serial Bus
UPS	Uninterruptible Power Supply

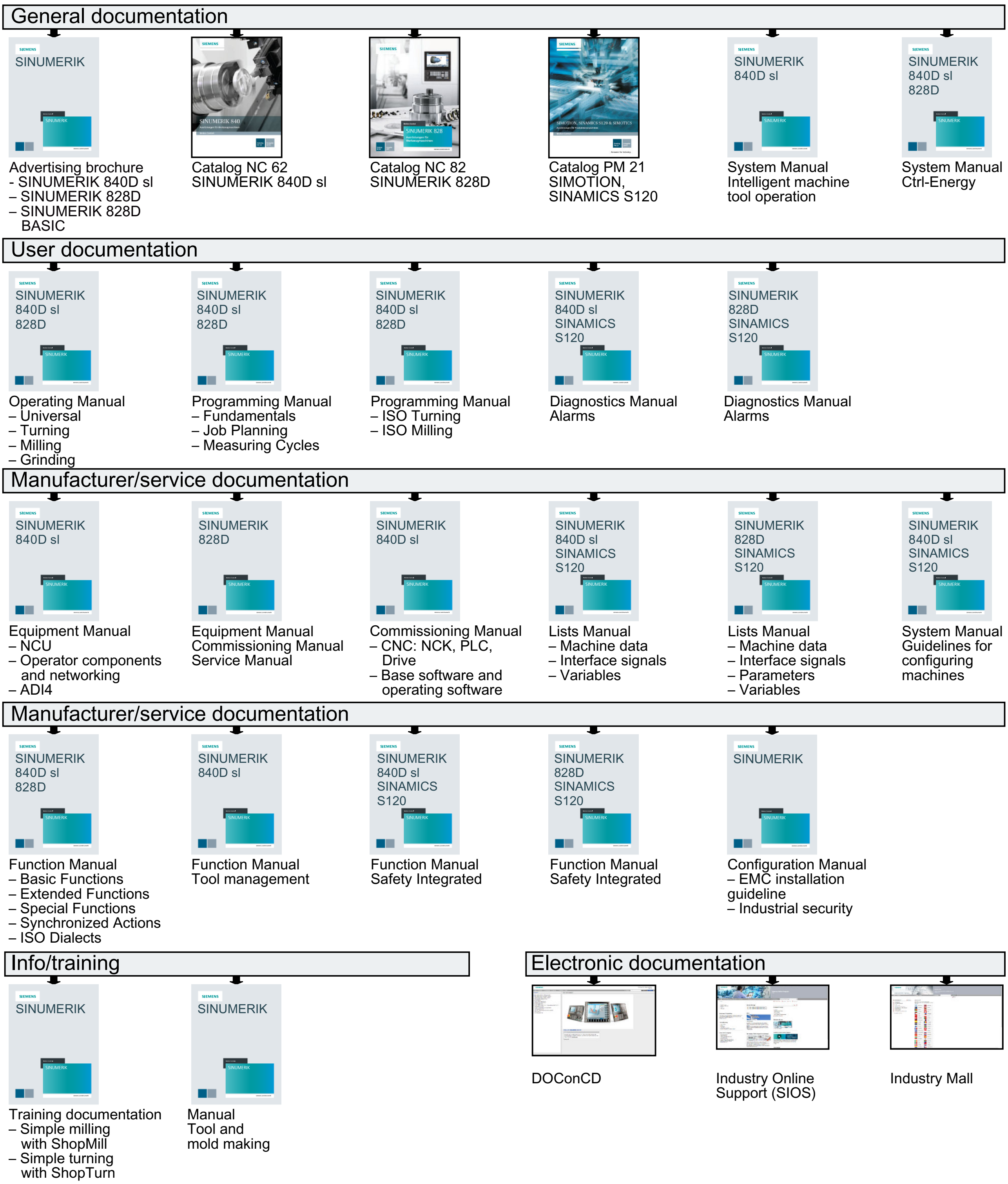
V	
VDI	Internal communication interface between NC and PLC
VDI	Verein Deutscher Ingenieure [Association of German Engineers]
VDE	Verband Deutscher Elektrotechniker [Association of German Electrical Engineers]
VI	Voltage Input
VO	Voltage Output
FDD	Feed Drive

W	
SAR	Smooth Approach and Retraction
WCS	Workpiece Coordinate System
T	Tool
TLC	Tool Length Compensation
WOP	Workshop-Oriented Programming
WPD	Workpiece Directory: Workpiece directory
TRC	Tool Radius Compensation
T	Tool
TO	Tool Offset
TM	Tool Management
TC	Tool change

X	
XML	Extensible Markup Language

Z	
WOA	Work Offset Active: Identifier for work offsets
ZSW	Status word (of drive)

A.2 Documentation overview



Glossary

Absolute dimensions

A destination for an axis motion is defined by a dimension that refers to the origin of the currently valid coordinate system. See → Incremental dimension

Acceleration with jerk limitation

In order to optimize the acceleration response of the machine whilst simultaneously protecting the mechanical components, it is possible to switch over in the machining program between abrupt acceleration and continuous (jerk-free) acceleration.

Address

An address is the identifier for a certain operand or operand range, e.g. input, output, etc.

Alarms

All → messages and alarms are displayed on the operator panel in plain text with date and time and the corresponding symbol for the deletion criterion. Alarms and messages are displayed separately.

1. Alarms and messages in the part program:
   Alarms and messages can be displayed in plain text directly from the part program.
2. Alarms and messages from the PLC:
   Alarms and messages for the machine can be displayed in plain text from the PLC program. No additional function block packages are required for this purpose.

Archiving

Reading out of files and/or directories on an **external** memory device.

Asynchronous subprogram

Part program that can be started asynchronously to (independently of) the current program status using an interrupt signal (e.g. "Rapid NC input" signal).

Automatic

Operating mode of the controller (block sequence operation according to DIN): Operating mode for NC systems in which a → subprogram is selected and executed continuously.

Auxiliary functions

Auxiliary functions enable → part programs to transfer → parameters to the → PLC, which then trigger reactions defined by the machine manufacturer.

Axes

In accordance with their functional scope, the CNC axes are subdivided into:

- Axes: Interpolating path axes
- Auxiliary axes: Non-interpolating feed and positioning axes with an axis-specific feedrate. Auxiliary axes are not involved in actual machining, e.g. tool feeder, tool magazine.

Axis address

See → Axis name

Axis name

To ensure clear identification, all channel and → machine axes of the control system must be designated with unique names in the channel and control system. The → geometry axes are called X, Y, Z. The rotary axes rotating around the geometry axes → are called A, B, C.

Backlash compensation

Compensation for a mechanical machine backlash, e.g. backlash on reversal for ball screws. Backlash compensation can be entered separately for each axis.

Backup battery

The backup battery ensures that the → user program in the → CPU is stored so that it is safe from power failure and so that specified data areas and bit memory, timers and counters are stored retentively.

Basic axis

Axis whose setpoint or actual value position forms the basis of the calculation of a compensation value.

Basic Coordinate System

Cartesian coordinate system which is mapped by transformation onto the machine coordinate system.

The programmer uses axis names of the basic coordinate system in the → part program. The basic coordinate system exists parallel to the → machine coordinate system if no → transformation is active. The difference lies in the → axis names.

Baud rate

Rate of data transfer (bits/s).

Blank

Workpiece as it is before it is machined.

Block

"Block" is the term given to any files required for creating and processing programs.

Block search

For debugging purposes or following a program abort, the "Block search" function can be used to select any location in the part program at which the program is to be started or resumed.

Booting

Loading the system program after power ON.

C axis

Axis around which the tool spindle describes a controlled rotational and positioning motion.

C spline

The C spline is the most well-known and widely used spline. The transitions at the interpolation points are continuous, both tangentially and in terms of curvature. 3rd order polynomials are used.

Channel

A channel is characterized by the fact that it can process a → part program independently of other channels. A channel exclusively controls the axes and spindles assigned to it. Part program runs of different channels can be coordinated through → synchronization.

Circular interpolation

The → tool moves on a circle between specified points on the contour at a given feedrate, and the workpiece is thereby machined.

CNC

See → NC

Computerized Numerical Control: includes the components → NC, → PLC, HMI, → COM.

CNC

See → NC

Computerized Numerical Control: includes the components → NC, → PLC, HMI, → COM.

COM

Component of the NC for the implementation and coordination of communication.

Compensation axis

Axis with a setpoint or actual value modified by the compensation value

Compensation table

Table containing interpolation points. It provides the compensation values of the compensation axis for selected positions on the basic axis.

Compensation value

Difference between the axis position measured by the encoder and the desired, programmed axis position.

Continuous-path mode

The objective of continuous-path mode is to avoid substantial deceleration of the → path axes at the part program block boundaries and to change to the next block at as close to the same path velocity as possible.

Contour

Contour of the → workpiece

Contour monitoring

The following error is monitored within a definable tolerance band as a measure of contour accuracy. An unacceptably high following error can cause the drive to become overloaded, for example. In such cases, an alarm is output and the axes are stopped.

Coordinate system

See → Machine coordinate system, → Workpiece coordinate system

CPU

Central processing unit, see → PLC

CU

Transformation ratio

Curvature

The curvature k of a contour is the inverse of radius r of the nestling circle in a contour point (k = 1/r).

Cycles

Protected subprograms for execution of repetitive machining operations on the → workpiece.

Data block

1. Data unit of the → PLC that → HIGHSTEP programs can access.
2. Data unit of the → NC: Data blocks contain data definitions for global user data. This data can be initialized directly when it is defined.

Data word

Two-byte data unit within a → data block.

Diagnostics

1. Operating area of the control.
2. The control has a self-diagnostics program as well as test functions for servicing purposes: status, alarm, and service displays

Dimensions specification, metric and inches

Position and pitch values can be programmed in inches in the machining program. Irrespective of the programmable dimensions (`G70`/`G71`), the control is set to a basic system.

DRF

Differential Resolver Function: NC function which generates an incremental work offset in Automatic mode in conjunction with an electronic handwheel.

Drive

The drive is the unit of the CNC that performs the speed and torque control based on the settings of the NC.

Dynamic feedforward control

Inaccuracies in the → contour due to following errors can be practically eliminated using dynamic, acceleration-dependent feedforward control. This results in excellent machining accuracy even at high → path velocities. Feedforward control can be selected and deselected on an axis-specific basis via the → part program.

Editor

The editor makes it possible to create, edit, extend, join, and import programs / texts / program blocks.

Exact stop

When an exact stop statement is programmed, the position specified in a block is approached exactly and, if necessary, very slowly. To reduce the approach time, → exact stop limits are defined for rapid traverse and feed.

Exact stop limit

When all path axes reach their exact stop limits, the control responds as if it had reached its precise destination point. A block advance of the → part program occurs.

External work offset

Work offset specified by the → PLC.

Fast retraction from the contour

When an interrupt occurs, a motion can be initiated via the CNC machining program, enabling the tool to be quickly retracted from the workpiece contour that is currently being machined. The retraction angle and the distance retracted can also be parameterized. An interrupt routine can also be executed following the fast retraction.

Feed override

The programmed velocity is overriden by the current velocity setting made via the → machine control panel or from the → PLC (0 to 200%). The feedrate can also be corrected by a programmable percentage factor (1 to 200%) in the machining program.

Finished-part contour

Contour of the finished workpiece. See → Raw part.

Fixed machine point

Point that is uniquely defined by the machine tool, e.g. machine reference point.

Fixed-point approach

Machine tools can approach fixed points such as a tool change point, loading point, pallet change point, etc. in a defined way. The coordinates of these points are stored in the control. The control moves the relevant axes in → rapid traverse, whenever possible.

Frame

A frame is an arithmetic rule that transforms one Cartesian coordinate system into another Cartesian coordinate system. A frame contains the following components: → work offset, → rotation, → scaling, → mirroring.

Geometry

Description of a → workpiece in the → workpiece coordinate system.

Geometry axis

The geometry axes form the 2 or 3-dimensional → workpiece coordinate system in which, in → part programs, the geometry of the workpiece is programmed.

Ground

Ground is taken as the total of all linked inactive parts of a device which will not become live with a dangerous contact voltage even in the event of a malfunction.

Helical interpolation

The helical interpolation function is ideal for machining internal and external threads using form milling cutters and for milling lubrication grooves.

The helix comprises two motions:

- Circular motion in one plane
- A linear motion perpendicular to this plane

High-level CNC language

The high-level language is used to write NC programs, → synchronized actions, and → cycles. It provides: control structures → user-defined variables, → system variables, → macro programming.

High-speed digital inputs/outputs

The digital inputs can be used for example to start fast CNC program routines (interrupt routines). High-speed, program-driven switching functions can be initiated via the digital CNC outputs

HIGHSTEP

Summary of programming options for → PLCs of the AS300/AS400 system.

HW Config

SIMATIC S7 tool for the configuration and parameterization of hardware components within an S7 project

Identifier

In accordance with DIN 66025, words are supplemented using identifiers (names) for variables (arithmetic variables, system variables, user variables), subprograms, key words, and words with multiple address letters. These supplements have the same meaning as the words with respect to block format. Identifiers must be unique. It is not permissible to use the same identifier for different objects.

Inch measuring system

Measuring system which defines distances in inches and fractions of inches.

Inclined surface machining

Drilling and milling operations on workpiece surfaces that do not lie in the coordinate planes of the machine can be performed easily using the function "inclined-surface machining".

Increment

Travel path length specification based on number of increments. The number of increments can be stored as → setting data or be selected by means of a suitably labeled key (i.e. 10, 100, 1000, 10000).

Incremental dimension

Incremental dimension: A destination for axis traversal is defined by a distance to be covered and a direction referenced to a point already reached. See → Absolute dimension.

Intermediate blocks

Motions with selected → tool offset (`G41`/`G42`) may be interrupted by a limited number of intermediate blocks (blocks without axis motions in the offset plane), whereby the tool offset can still be correctly compensated for. The permissible number of intermediate blocks which the controller reads ahead can be set in system parameters.

Interpolator

Logic unit of the → NC that defines intermediate values for the motion to be carried out in individual axes based on information on the end positions specified in the part program.

Interpolatory compensation

Mechanical deviations of the machine are compensated for by means of interpolatory compensation functions, such as → leadscrew error, sag, angularity, and temperature compensation.

Interrupt routine

Interrupt routines are special → subprograms that can be started by events (external signals) in the machining process. A part program block which is currently being worked through is interrupted and the position of the axes at the point of interruption is automatically saved.

Inverse-time feedrate

The time required for the path of a block to be traversed can also be programmed for the axis motion instead of the feed velocity (G93).

JOG

Operating mode of the control (setup mode): The machine can be set up in JOG mode. Individual axes and spindles can be traversed in JOG mode by means of the direction keys. Additional functions in JOG mode include: → Reference point approach, → Repos, and → Preset (set actual value).

Key switch

The key switch on the → machine control panel has four positions that are assigned functions by the operating system of the controller. The key switch has three different colored keys that can be removed in the specified positions.

Keywords

Words with specified notation that have a defined meaning in the programming language for → part programs.

KV

Servo gain factor, a control variable in a control loop.

Leading axis

The leading axis is the → gantry axis that exists from the point of view of the operator and programmer and, thus, can be influenced like a standard NC axis.

Leadscrew error compensation

Compensation for the mechanical inaccuracies of a leadscrew participating in the feed. The controller uses stored deviation values for the compensation.

Limit speed

Maximum/minimum (spindle) speed: The maximum speed of a spindle can be limited by specifying machine data, the → PLC or → setting data.

Linear axis

In contrast to a rotary axis, a linear axis describes a straight line.

Linear interpolation

The tool travels along a straight line to the destination point while machining the workpiece.

Load memory

The load memory is the same as the → working memory for the CPU 314 of the → PLC.

Look Ahead

The **Look Ahead** function is used to achieve an optimal machining speed by looking ahead over an assignable number of traversing blocks.

Machine axes

Physically existent axes on the machine tool.

Machine control panel

An operator panel on a machine tool with operating elements such as keys, rotary switches, etc., and simple indicators such as LEDs. It is used to directly influence the machine tool via the PLC.

Machine coordinate system

A coordinate system, which is related to the axes of the machine tool.

Machine zero

Fixed point of the machine tool to which all (derived) measuring systems can be traced back.

Machining channel

A channel structure can be used to shorten idle times by means of parallel motion sequences, e.g. moving a loading gantry simultaneously with machining. Here, a CNC channel must be regarded as a separate CNC control system with decoding, block preparation and interpolation.

Macro techniques

Grouping of a set of statements under a single identifier. The identifier represents the set of consolidated statements in the program.

Main block

A block preceded with ":" that contains all information to start the operating sequence in a → part program.

Main program

The term "main program" has its origins during the time when part programs were split strictly into main and → subprograms. This strict division no longer exists with today's SINUMERIK NC language. In principle, any part program in the channel can be selected and started. It then runs through in → program level 0 (main program level). Further part programs or → cycles as subprograms can be called up in the main program.

MDI

Operating mode of the control: Manual Data Input. In the MDI mode, individual program blocks or block sequences with no reference to a main program or subprogram can be input and executed immediately afterwards through actuation of the NC start key.

Messages

All messages programmed in the part program and → alarms detected by the system are displayed on the operator panel in plain text with date and time and the corresponding symbol for the deletion criterion. Alarms and messages are displayed separately.

Metric measuring system

Standardized system of units: For length, e.g. mm (millimeters), m (meters).

Mirroring

Mirroring reverses the signs of the coordinate values of a contour, with respect to an axis. It is possible to mirror with respect to more than one axis at a time.

Mode

An operating concept on a SINUMERIK control The following modes are defined: → Jog, → MDI, → Automatic.

Mode group

Axes and spindles that are technologically related can be combined into one mode group. Axes/spindles of a mode group can be controlled by one or more → channels. The same → mode type is always assigned to the channels of the mode group.

NC

Numerical Control component of the → CNC that executes the → part programs and coordinates the movements of the machine tool.

Network

A network is the connection of multiple S7-300 and other end devices, e.g. a programming device via a → connecting cable. A data exchange takes place over the network between the connected devices.

NRK

Numeric robotic kernel (operating system of → NC)

NURBS

The motion control and path interpolation that occurs within the control is performed based on NURBS (**N**on **U**niform **R**ational **B**-**S**plines). This provides a uniform procedure for all internal interpolations.

OEM

The scope for implementing individual solutions (OEM applications) has been provided for machine manufacturers, who wish to create their own user interface or integrate technology-specific functions in the control.

Offset memory

Data range in the control, in which the tool offset data is stored.

Oriented spindle stop

Stops the workpiece spindle in a specified angular position, e.g. in order to perform additional machining at a particular location.

Overall reset

In the event of an overall reset, the following memories of the → CPU are deleted:

- → Working memory
- Read/write area of → load memory
- → System memory
- → Backup memory

Override

Manual or programmable possibility of intervention that enables the user to override programmed feedrates or speeds in order to adapt them to a specific workpiece or material.

Part program

Series of statements to the NC that act in concert to produce a particular → workpiece. Likewise, this term applies to execution of a particular machining operation on a given → raw part.

Part program block

Part of a → part program that is demarcated by a line feed. There are two types: → main blocks and → subblocks.

Part program management

Part program management can be organized by → workpieces. The size of the user memory determines the number of programs and the amount of data that can be managed. Each file (programs and data) can be given a name consisting of a maximum of 24 alphanumeric characters.

Path axis

Path axes include all machining axes of the → channel that are controlled by the → interpolator in such a way that they start, accelerate, stop, and reach their end point simultaneously.

Path feedrate

Path feedrate affects → path axes. It represents the geometric sum of the feedrates of the → geometry axes involved.

Path velocity

The maximum programmable path velocity depends on the input resolution. For example, with a resolution of 0.1 mm the maximum programmable path velocity is 1000 m/min.

PCIN data transfer program

PCIN is a utility program for sending and receiving CNC user data (e.g. part programs, tool offsets) via the serial interface. The PCIN program can run under MS-DOS on standard industrial PCs.

Peripheral module

I/O modules represent the link between the CPU and the process.

I/O modules are:

- → Digital input/output modules
- → Analog input/output modules
- → Simulator modules

PLC

Programmable **L**ogic **C**ontroller: → Programmable logic controller. Component of → NC: Programmable control for processing the control logic of the machine tool.

PLC program memory

SINUMERIK 840D sl: The PLC user program, the user data and the basic PLC program are stored together in the PLC user memory.

PLC programming

The PLC is programmed using the **STEP 7** software. The STEP 7 programming software is based on the **WINDOWS** standard operating system and contains the STEP 5 programming functions with innovative enhancements.

Polar coordinates

A coordinate system which defines the position of a point on a plane in terms of its distance from the origin and the angle formed by the radius vector with a defined axis.

Polynomial interpolation

Polynomial interpolation enables a wide variety of curve characteristics to be generated, such as **straight line, parabolic, exponential functions** (SINUMERIK 840D sl).

Positioning axis

Axis that performs an auxiliary motion on a machine tool (e.g. tool magazine, pallet transport). Positioning axes are axes that do not interpolate with → path axes.

Pre-coincidence

Block change occurs already when the path distance approaches an amount equal to a specifiable delta of the end position.

Program block

Program blocks contain the main program and subprograms of → part programs.

Program level

A part program started in the channel runs as a → main program on program level 0 (main program level). Any part program called up in the main program runs as a → subprogram on a program level 1 ... n of its own.

Programmable frames

Programmable → frames enable dynamic definition of new coordinate system output points while the part program is being executed. A distinction is made between absolute definition using a new frame and additive definition with reference to an existing starting point.

Programmable logic controller

Programmable logic controllers (PLCs) are electronic controllers, the function of which is stored as a program in the control unit. This means that the layout and wiring of the device do not depend on the function of the controller. The programmable logic control has the same structure as a computer; it consists of a CPU (central module) with memory, input/output modules and an internal bus system. The peripherals and the programming language are matched to the requirements of the control technology.

Programmable working area limitation

Limitation of the motion space of the tool to a space defined by programmed limitations.

Programming key

Characters and character strings that have a defined meaning in the programming language for → part programs.

Protection zone

Three-dimensional zone within the → working area into which the tool tip must not pass.

Quadrant error compensation

Contour errors at quadrant transitions, which arise as a result of changing friction conditions on the guideways, can be virtually entirely eliminated with the quadrant error compensation. Parameterization of the quadrant error compensation is performed by means of a circuit test.

R parameters

Arithmetic parameter that can be set or queried by the programmer of the → part program for any purpose in the program.

Rapid traverse

The highest traverse velocity of an axis. It is used, for example, when the tool approaches the → workpiece contour from a resting position or when the tool is retracted from the workpiece contour. The rapid traverse velocity is set on a machine-specific basis using a machine data item.

Reference point

Machine tool position that the measuring system of the → machine axes references.

Rotary axis

Rotary axes apply a workpiece or tool rotation to a defined angular position.

Rotation

Component of a → frame that defines a rotation of the coordinate system around a particular angle.

Rounding axis

Rounding axes rotate a workpiece or tool to an angular position corresponding to an indexing grid. When a grid index is reached, the rounding axis is "in position".

RS-232-C

Serial interface for data input/output. Machining programs as well as manufacturer and user data can be loaded and saved via this interface.

Safety functions

The controller is equipped with permanently active monitoring functions that detect faults in the → CNC, the → PLC, and the machine in a timely manner so that damage to the workpiece, tool, or machine is largely prevented. In the event of a fault, the machining operation is interrupted and the drives stopped. The cause of the malfunction is logged and output as an alarm. At the same time, the PLC is notified that a CNC alarm has been triggered.

Scaling

Component of a → frame that implements axis-specific scale modifications.

Setting data

Data which communicates the properties of the machine tool to the NC as defined by the system software.

Softkey

A key, whose name appears on an area of the screen. The choice of softkeys displayed is dynamically adapted to the operating situation. The freely assignable function keys (softkeys) are assigned defined functions in the software.

Software limit switch

Software limit switches limit the traversing range of an axis and prevent an abrupt stop of the slide at the hardware limit switch. Two value pairs can be specified for each axis and activated separately by means of the → PLC.

Spline interpolation

With spline interpolation, the controller can generate a smooth curve characteristic from only a few specified interpolation points of a set contour.

Standard cycles

Standard cycles are provided for machining operations which are frequently repeated:

- For the drilling/milling technology
- For turning technology

The available cycles are listed in the "Cycle support" menu in the "Program" operating area. Once the desired machining cycle has been selected, the parameters required for assigning values are displayed in plain text.

Subblock

Block preceded by "N" containing information for a sequence, e.g. positional data.

Subprogram

The term "subprogram" has its origins during the time when part programs were split strictly into →main and subprograms. This strict division no longer exists with today's SINUMERIK NC language. In principle, any part program or any → cycle can be called up as a subprogram within another part program. It then runs through in the next → program level (x+1) (subprogram level (x+1)).

Synchronization

Statements in → part programs for coordination of sequences in different → channels at certain machining points.

Synchronized actions

1. Auxiliary function output
   During workpiece machining, technological functions (→ auxiliary functions) can be output from the CNC program to the PLC. For example, these auxiliary functions are used to control additional equipment for the machine tool, such as quills, grabbers, clamping chucks, etc.
2. Fast auxiliary function output
   For time-critical switching functions, the acknowledgement times for the → auxiliary functions can be minimized and unnecessary hold points in the machining process can be avoided.

Synchronized axes

Synchronized axes take the same time to traverse their path as the geometry axes take for their path.

Synchronized axis

A synchronized axis is the → gantry axis whose set position is continuously derived from the motion of the → leading axis and is, thus, moved synchronously with the leading axis. From the point of view of the programmer and operator, the synchronized axis "does not exist".

System memory

The system memory is a memory in the CPU in which the following data is stored:

- Data required by the operating system
- The operands timers, counters, markers

System variable

A variable that exists without any input from the programmer of a → part program. It is defined by a data type and the variable name preceded by the character **$**. See → User-defined variable.

Tapping without compensating chuck

This function allows threads to be tapped without a compensating chuck. By using the interpolating method of the spindle as a rotary axis and the drilling axis, threads can be cut to a precise final drilling depth, e.g. for blind hole threads (requirement: spindles in axis operation).

Text editor

See → Editor

TOA area

The TOA area includes all tool and magazine data. By default, this area coincides with the → channel area with regard to the access of the data. However, machine data can be used to specify that multiple channels share one → TOA unit so that common tool management data is then available to these channels.

TOA unit

Each → TOA area can have more than one TOA unit. The number of possible TOA units is limited by the maximum number of active → channels. A TOA unit includes exactly one tool data block and one magazine data block. In addition, a TOA unit can also contain a toolholder data block (optional).

Tool

Active part on the machine tool that implements machining (e.g. turning tool, milling tool, drill, LASER beam, etc.).

Tool nose radius compensation

Contour programming assumes that the tool is pointed. Because this is not actually the case in practice, the curvature radius of the tool used must be communicated to the controller which then takes it into account. The curvature center is maintained equidistantly around the contour, offset by the curvature radius.

Tool offset

Consideration of the tool dimensions in calculating the path.

Tool radius compensation

To directly program a desired → workpiece contour, the control must traverse an equistant path to the programmed contour taking into account the radius of the tool that is being used (`G41`/`G42`).

Transformation

Additive or absolute zero offset of an axis.

Travel range

The maximum permissible travel range for linear axes is ± 9 decades. The absolute value depends on the selected input and position control resolution and the unit of measurement (inch or metric).

User interface

The user interface (UI) is the display medium for a CNC in the form of a screen. It features horizontal and vertical softkeys.

User memory

All programs and data, such as part programs, subprograms, comments, tool offsets, and work offsets / frames, as well as channel and program user data, can be stored in the shared CNC user memory.

User program

User programs for the S7-300 automation systems are created using the programming language STEP 7. The user program has a modular layout and consists of individual blocks.

The basic block types are:

- Code blocks
  These blocks contain the STEP 7 commands.
- Data blocks
  These blocks contain constants and variables for the STEP 7 program.

User-defined variable

Users can declare their own variables for any purpose in the → part program or data block (global user data). A definition contains a data type specification and the variable name. See → System variable.

Variable definition

A variable definition includes the specification of a data type and a variable name. The variable names can be used to access the value of the variables.

Velocity control

In order to achieve an acceptable traverse rate in the case of very slight motions per block, an anticipatory evaluation over several blocks (→ Look Ahead) can be specified.

WinSCP

WinSCP is a freely available open source program for Windows for the transfer of files.

Work offset

Specifies a new reference point for a coordinate system through reference to an existing zero point and a → frame.

1. Settable
   A configurable number of settable work offsets are available for each CNC axis. The offsets - which are selected by means of G commands - take effect alternatively.
2. External
   In addition to all the offsets which define the position of the workpiece zero, an external work offset can be overridden by means of the handwheel (DRF offset) or from the PLC.
3. Programmable
   Work offsets can be programmed for all path and positioning axes using the `TRANS` statement.

Working area

Three-dimensional zone into which the tool tip can be moved on account of the physical design of the machine tool. See → Protection zone.

Working area limitation

With the aid of the working area limitation, the traversing range of the axes can be further restricted in addition to the limit switches. One value pair per axis may be used to describe the protected working area.

Working memory

The working memory is a RAM in the → CPU that the processor accesses when processing the application program.

Workpiece

Part to be made/machined by the machine tool.

Workpiece contour

Set contour of the → workpiece to be created or machined.

Workpiece coordinate system

The workpiece coordinate system has its starting point in the → workpiece zero-point. In machining operations programmed in the workpiece coordinate system, the dimensions and directions refer to this system.

Workpiece zero

The workpiece zero is the starting point for the → workpiece coordinate system. It is defined in terms of distances to the → machine zero.

Index

$

A

B

H

I

J

K

Q

R

S