Return to Library

PROGRAMMING MANUAL

for

MAZATROL MATRIX

For INTEGREX IV

Programming EIA/ISO

MANUAL No. : H740PB0030E

Serial No. :

Before using this machine and equipment, fully understand the contents of this manual to ensure proper operation. Should any questions arise, please ask the nearest Technical Center or Technology Center.

IMPORTANT NOTICE

1. **Be sure to observe the safety precautions described in this manual and the contents of the safety plates on the machine and equipment. Failure may cause serious personal injury or material damage. Please replace any missing safety plates as soon as possible.**
2. **No modifications are to be performed that will affect operation safety. If such modifications are required, please contact the nearest Technical Center or Technology Center.**
3. **For the purpose of explaining the operation of the machine and equipment, some illustrations may not include safety features such as covers, doors, etc. Before operation, make sure all such items are in place.**
4. **This manual was considered complete and accurate at the time of publication, however, due to our desire to constantly improve the quality and specification of all our products, it is subject to change or modification. If you have any questions, please contact the nearest Technical Center or Technology Center.**
5. **Always keep this manual near the machinery for immediate use.**
6. **If a new manual is required, please order from the nearest Technical Center or Technology Center with the manual No. or the machine name, serial No. and manual name.**

Issued by *Manual Publication Section, Yamazaki Mazak Corporation*, Japan

01. 2006

Notes:

SAFETY PRECAUTIONS

Preface

Safety precautions relating to the CNC unit (in the remainder of this manual, referred to simply as the NC unit) that is provided in this machine are explained below. Not only the persons who create programs, but also those who operate the machine must thoroughly understand the contents of this manual to ensure safe operation of the machine.

Read all these safety precautions, even if your NC model does not have the corresponding functions or optional units and a part of the precautions do not apply.

Rule

1. This section contains the precautions to be observed as to the working methods and states usually expected. Of course, however, unexpected operations and/or unexpected working states may take place at the user site.
   During daily operation of the machine, therefore, the user must pay extra careful attention to its own working safety as well as to observe the precautions described below.
2. Although this manual contains as great an amount of information as it can, since it is not rare for the user to perform the operations that overstep the manufacturer-assumed ones, not all of “what the user cannot perform” or “what the user must not perform” can be fully covered in this manual with all such operations taken into consideration beforehand.
   It is to be understood, therefore, that functions not clearly written as “executable” are “inexecutable” functions.
3. The meanings of our safety precautions to DANGER, WARNING, and CAUTION are as follows:

DANGER : Failure to follow these instructions could result in loss of life.

WARNING : Failure to observe these instructions could result in serious harm to a human life or body.

CAUTION : Failure to observe these instructions could result in minor injuries or serious machine damage.

Return to Library

Basics

WARNING

- After turning power on, keep hands away from the keys, buttons, or switches of the operating panel until an initial display has been made.
- Before proceeding to the next operations, fully check that correct data has been entered and/or set. If the operator performs operations without being aware of data errors, unexpected operation of the machine will result.
- Before machining workpieces, perform operational tests and make sure that the machine operates correctly. No workpieces must be machined without confirmation of normal operation. Closely check the accuracy of programs by executing override, single-block, and other functions or by operating the machine at no load. Also, fully utilize tool path check, solid check, and other functions, if provided.
- Make sure that the appropriate feed rate and rotational speed are designated for the particular machining requirements. Always understand that since the maximum usable feed rate and rotational speed are determined by the specifications of the tool to be used, those of the workpiece to be machined, and various other factors, actual capabilities differ from the machine specifications listed in this manual. If an inappropriate feed rate or rotational speed is designated, the workpiece or the tool may abruptly move out from the machine.
- Before executing correction functions, fully check that the direction and amount of correction are correct. Unexpected operation of the machine will result if a correction function is executed without its thorough understanding.
- Parameters are set to the optimum standard machining conditions prior to shipping of the machine from the factory. In principle, these settings should not be modified. If it becomes absolutely necessary to modify the settings, perform modifications only after thoroughly understanding the functions of the corresponding parameters. Modifications usually affect any program. Unexpected operation of the machine will result if the settings are modified without a thorough understanding.

Remarks on the cutting conditions recommended by the NC

WARNING

- Before using the following cutting conditions:

  - Cutting conditions that are the result of the MAZATROL Automatic Cutting Conditions Determination Function
  - Cutting conditions suggested by the Machining Navigation Function
  - Cutting conditions for tools that are suggested to be used by the Machining Navigation Function

  Confirm that every necessary precaution in regards to safe machine setup has been taken – especially for workpiece fixturing/clamping and tool setup.
- Confirm that the machine door is securely closed before starting machining.
  Failure to confirm safe machine setup may result in serious injury or death.

Programming

WARNING

- Fully check that the settings of the coordinate systems are correct. Even if the designated program data is correct, errors in the system settings may cause the machine to operate in unexpected places and the workpiece to abruptly move out from the machine in the event of contact with the tool.
- During surface velocity hold control, as the current workpiece coordinates of the surface velocity hold control axes approach zeroes, the spindle speed increases significantly. For the lathe, the workpiece may even come off if the chucking force decreases. Safety speed limits must therefore be observed when designating spindle speeds.
- Even after inch/metric system selection, the units of the programs, tool information, or parameters that have been registered until that time are not converted. Fully check these data units before operating the machine. If the machine is operated without checks being performed, even existing correct programs may cause the machine to operate differently from the way it did before.
- If a program is executed that includes the absolute data commands and relative data commands taken in the reverse of their original meaning, totally unexpected operation of the machine will result. Recheck the command scheme before executing programs.
- If an incorrect plane selection command is issued for a machine action such as arc interpolation or fixed-cycle machining, the tool may collide with the workpiece or part of the machine since the motions of the control axes assumed and those of actual ones will be interchanged. (This precaution applies only to NC units provided with EIA functions.)
- The mirror image, if made valid, changes subsequent machine actions significantly. Use the mirror image function only after thoroughly understanding the above. (This precaution applies only to NC units provided with EIA functions.)
- If machine coordinate system commands or reference position returning commands are issued with a correction function remaining made valid, correction may become invalid temporarily. If this is not thoroughly understood, the machine may appear as if it would operate against the expectations of the operator. Execute the above commands only after making the corresponding correction function invalid. (This precaution applies only to NC units provided with EIA functions.)
- The barrier function performs interference checks based on designated tool data. Enter the tool information that matches the tools to be actually used. Otherwise, the barrier function will not work correctly.
- The system of G-code and M-code commands differs, especially for turning, between the machines of INTEGREX e-Series and the other turning machines.
  Issuance of the wrong G-code or M-code command results in totally non-intended machine operation. Thoroughly understand the system of G-code and M-code commands before using this system.

Sample program	Machines of INTEGREX e-Series	Turning machines
`S1000M3`	The milling spindle rotates at 1000 min^{-1}.	The turning spindle rotates at 1000 min^{-1}.
`S1000M203`	The turning spindle rotates at 1000 min^{-1}.	The milling spindle rotates at 1000 min^{-1}.

- For the machines of INTEGREX e-Series, programmed coordinates can be rotated using an index unit of the MAZATROL program and a G68 command (coordinate rotate command) of the EIA program. However, for example, when the B-axis is rotated through 180 degrees around the Y-axis to implement machining with the turning spindle No. 2, the plus side of the X-axis in the programmed coordinate system faces downward and if the program is created ignoring this fact, the resulting movement of the tool to unexpected positions may incite collisions.
  To create the program with the plus side of the X-axis oriented in an upward direction, use the mirror function of the WPC shift unit or the mirror imaging function of G-code command (G50.1, G51.1).
- After modifying the tool data specified in the program, be sure to perform the tool path check function, the solid check function, and other functions, and confirm that the program operates properly. The modification of tool data may cause even a field-proven machining program to change in operational status.
  If the user operates the machine without being aware of any changes in program status, interference with the workpiece could arise from unexpected operation.
  For example, if the cutting edge of the tool during the start of automatic operation is present inside the clearance-including blank (unmachined workpiece) specified in the common unit of the MAZATROL program, care is required since the tool will directly move from that position to the approach point because of no obstructions being judged to be present on this path.
  For this reason, before starting automatic operation, make sure that the cutting edge of the tool during the start of automatic operation is present outside the clearance-including workpiece specified in the common unit of the MAZATROL program.

CAUTION

- If axis-by-axis independent positioning is selected and simultaneously rapid feed selected for each axis, movements to the ending point will not usually become linear. Before using these functions, therefore, make sure that no obstructions are present on the path.

Return to Library

Operations

WARNING

- Single-block, feed hold, and override functions can be made invalid using system variables #3003 and #3004. Execution of this means the important modification that makes the corresponding operations invalid. Before using these variables, therefore, give thorough notification to related persons. Also, the operator must check the settings of the system variables before starting the above operations.
- If manual intervention during automatic operation, machine locking, the mirror image function, or other functions are executed, the workpiece coordinate systems will usually be shifted. When making machine restart after manual intervention, machine locking, the mirror image function, or other functions, consider the resulting amounts of shift and take the appropriate measures. If operation is restarted without any appropriate measures being taken, collision with the tool or workpiece may occur.
- Use the dry run function to check the machine for normal operation at no load. Since the feed rate at this time becomes a dry run rate different from the program-designated feed rate, the axes may move at a feed rate higher than the programmed value.
- After operation has been stopped temporarily and insertion, deletion, updating, or other commands executed for the active program, unexpected operation of the machine may result if that program is restarted. No such commands should, in principle, be issued for the active program.

CAUTION

- During manual operation, fully check the directions and speeds of axial movement.
- For a machine that requires manual homing, perform manual homing operations after turning power on. Since the software-controlled stroke limits will remain ineffective until manual homing is completed, the machine will not stop even if it oversteps the limit area. As a result, serious machine damage will result.
- Do not designate an incorrect pulse multiplier when performing manual pulse handle feed operations. If the multiplier is set to 1000 times and the handle operated inadvertently, axial movement will become faster than that expected.

OPERATIONAL WARRANTY FOR THE NC UNIT

The warranty of the manufacturer does not cover any trouble arising if the NC unit is used for its non-intended purpose. Take notice of this when operating the unit.

Examples of the trouble arising if the NC unit is used for its non-intended purpose are listed below.

1. Trouble associated with and caused by the use of any commercially available software products (including user-created ones)
2. Trouble associated with and caused by the use of any Windows operating systems
3. Trouble associated with and caused by the use of any commercially available computer equipment

Operating Environment

1. Ambient temperature

During machine operation: 0° to 50°C (0° to 122°F)

2. Relative humidity

During machine operation: 10 to 75% (without bedewing)

Note: As humidity increases, insulation deteriorates causing electrical component parts to deteriorate quickly.

Return to Library

CONTENTS

Page

Return to Library

Return to Library

Return to Library

Return to Library

Return to Library

Return to Library

- NOTE -

1 INTRODUCTION

EIA/ISO programs executed by the CNC unit include two modes: One is based on the G-code series T (designed for turning machines), and the other is based on the G-code series M (designed for machining centers).
Depending on the types of machines, the G-code series T and M are used as follows:

G-code series T for the INTEGREX-IV machines, and

G-code series M for the INTEGREX-*e* machines.

This manual gives descriptions in general with respect to the G-code series T designed for turning machines.

- NOTE -

2 UNITS OF PROGRAM DATA INPUT

2-1 Units of Program Data Input

The movements on coordinate axes are to be commanded in the MDI mode or machining program. The movement data are expressed in millimeters, inches or degrees.

2-2 Units of Data Setting

Various data commonly used for control axes, such as offsetting data, must be set for the machine to perform an operation as desired.
The units of data setting and those of program data input are listed below.

	Linear axis		Rotational axis
	Metric system	Inch system	
Units of program data input	0.0001 mm	0.00001 in.	0.0001 deg
Units of data setting	0.0001 mm	0.00001 in.	0.0001 deg

Note 1: Inch/metric selection can be freely made using either bit 4 of parameter **F91** ("0" for metric, "1" for inches; validated through power-off and -on) or G-code commands (G20, G21).
Selection using the G-code commands is valid only for program data input.
Variables and offsetting data (such as tool offsetting data) should therefore be set beforehand using the appropriate unit (inch or metric) for the particular machining requirements.

Note 2: Metric data and inch data cannot be used at the same time.

2-3 Ten-Fold Program Data

Using a predetermined parameter, machining program data can be processed as set in units of one micron. There may be cases that a machining program which has been set in units of one micron is to be used with a numerical control unit based on 0.1 micron increments. In such cases, use of this parameter allows the machine to perform the required machining operations without rewriting the program.
Use bit 0 of user parameter **F91** for this purpose.
All types of coordinate data (axis movement data) not provided with the decimal point will be multiplied by a factor of 10. This does not apply, indeed, to preset tool-offsetting data designated with addresses H and D.

Control axis	Program command	Moving distance when program commands are executed			Program applicability (A) → (B)
		NC (A) for which the program was prepared	MAZATROL (B)		
			Bit 0 of **F91** = 0	Bit 0 of **F91** = 1	
Linear axis	X1 (Y1 / Z1)	1 micron	0.1 micron	1 micron	Applicable
Rotational axis	B1	0.001°	0.0001°	0.001°	Applicable

- NOTE -

3 DATA FORMATS

3-1 Tape Codes

This numerical control unit (in the remainder of this manual, referred to as the NC unit) uses command information that consists of letters of the alphabet (A, B, C .... Z), numerics (0, 1, 2 .... 9), and signs (+, –, /, and so on). These alphanumerics and signs are referred to collectively as characters. On paper tape, these characters are represented as a combination of a maximum of eight punched holes.
Such a representation is referred to as a code.
The NC unit uses either the EIA codes (RS-244-A) or the ISO codes (R-840).

Note 1: Codes not included in the tape codes shown in Fig. 3-1 will result in an error when they are read.

Note 2: Of all codes specified as the ISO codes but not specified as the EIA codes, only the following codes can be designated using the data I/O (Tape) parameters **TAP9** to **TAP14**:

[Bracket Open
] Bracket Close
Sharp
∗ Asterisk
= Equal sign
: Colon

However, you cannot designate codes that overlap existing ones or that result in parity error.

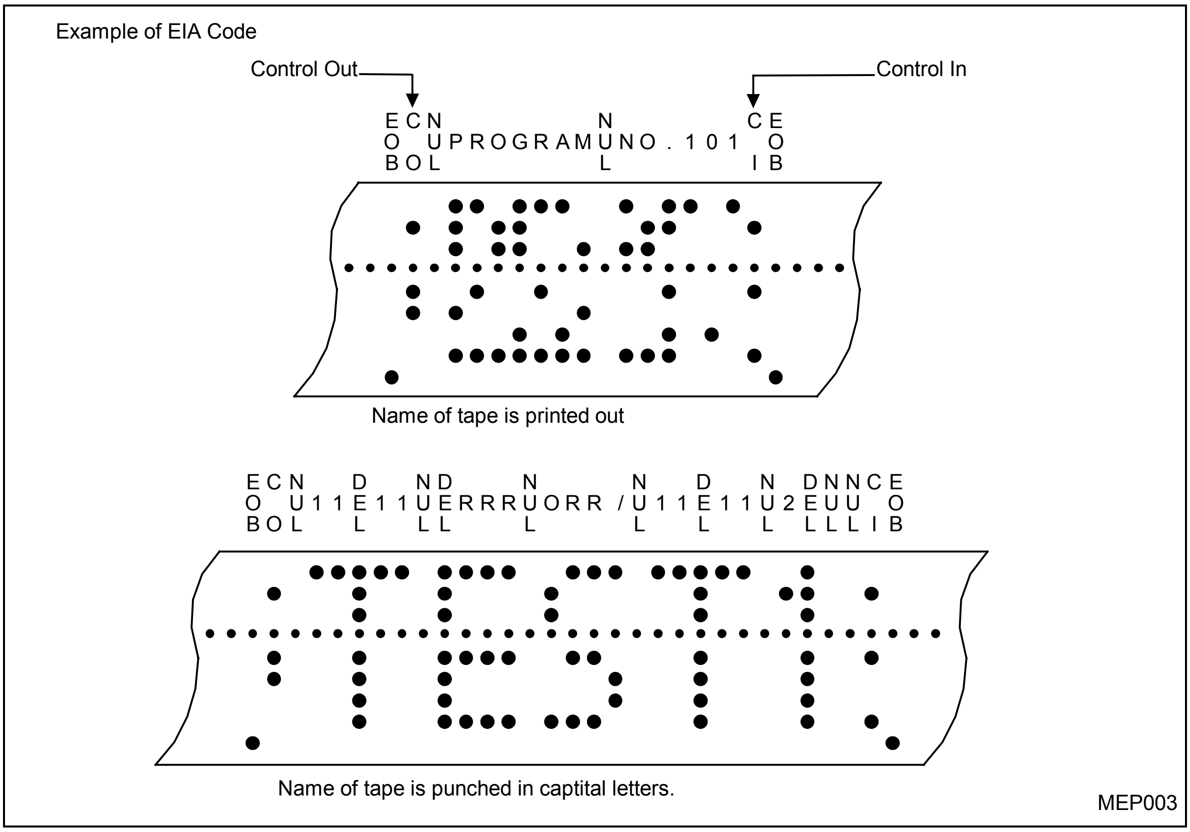
Note 3: EIA/ISO code identification is made automatically according to the first EOB/LF code appearing after the NC unit has been reset. (EOB: End Of Block, LF: Line Feed)

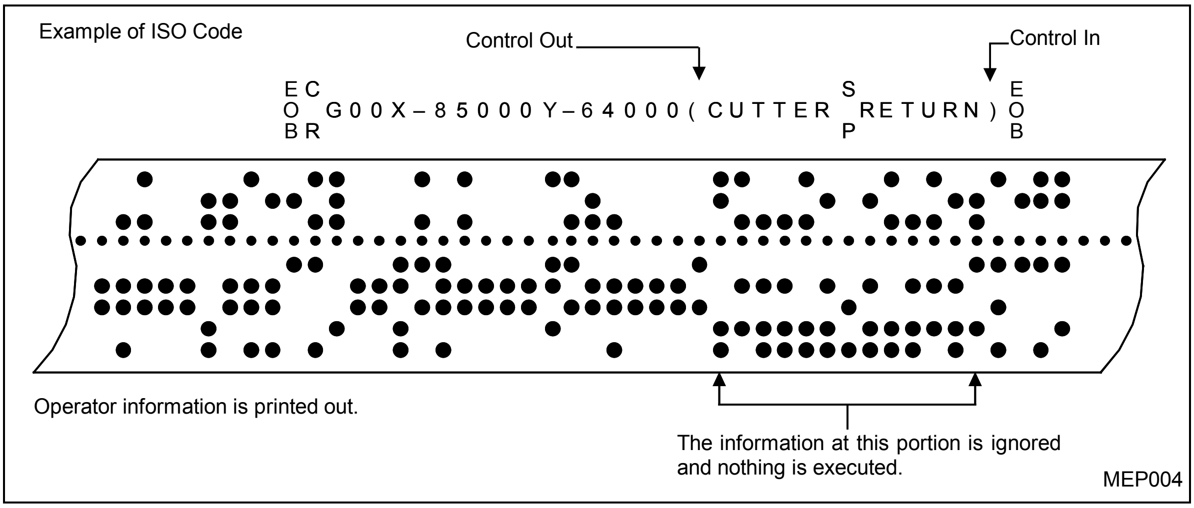
1. Significant information area (LABEL SKIP function)

During tape-based automatic operation, data storage into the memory, or data searching, the NC unit will ignore the entire information up to the first EOB code (;) in the tape when the unit is turned on or reset. That is, significant information in a tape refers to the information contained in the interval from the time a character or numeric code appears, following the first EOB code (;) after the NC unit has been reset, until a reset command is given.

2. Control Out, Control In

The entire information in the area from Control Out “(” to Control In “)” will be ignored in regard to machine control, while they will surely be displayed on the data display unit. Thus, this area can be used to contain information, such as the name and number of the command tape, that is not directly related to control.
During tape storage, however, the information in this area will also be stored. The NC unit will enter the Control In status when power is turned on.



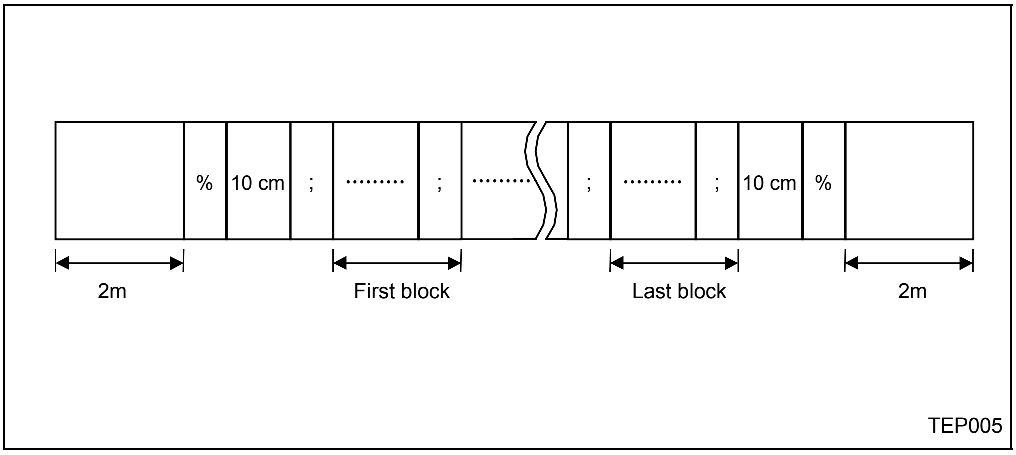


3. EOR code (%)

In general, the EOR (End Of Record) code is punched at both ends of a tape and has the following functions:

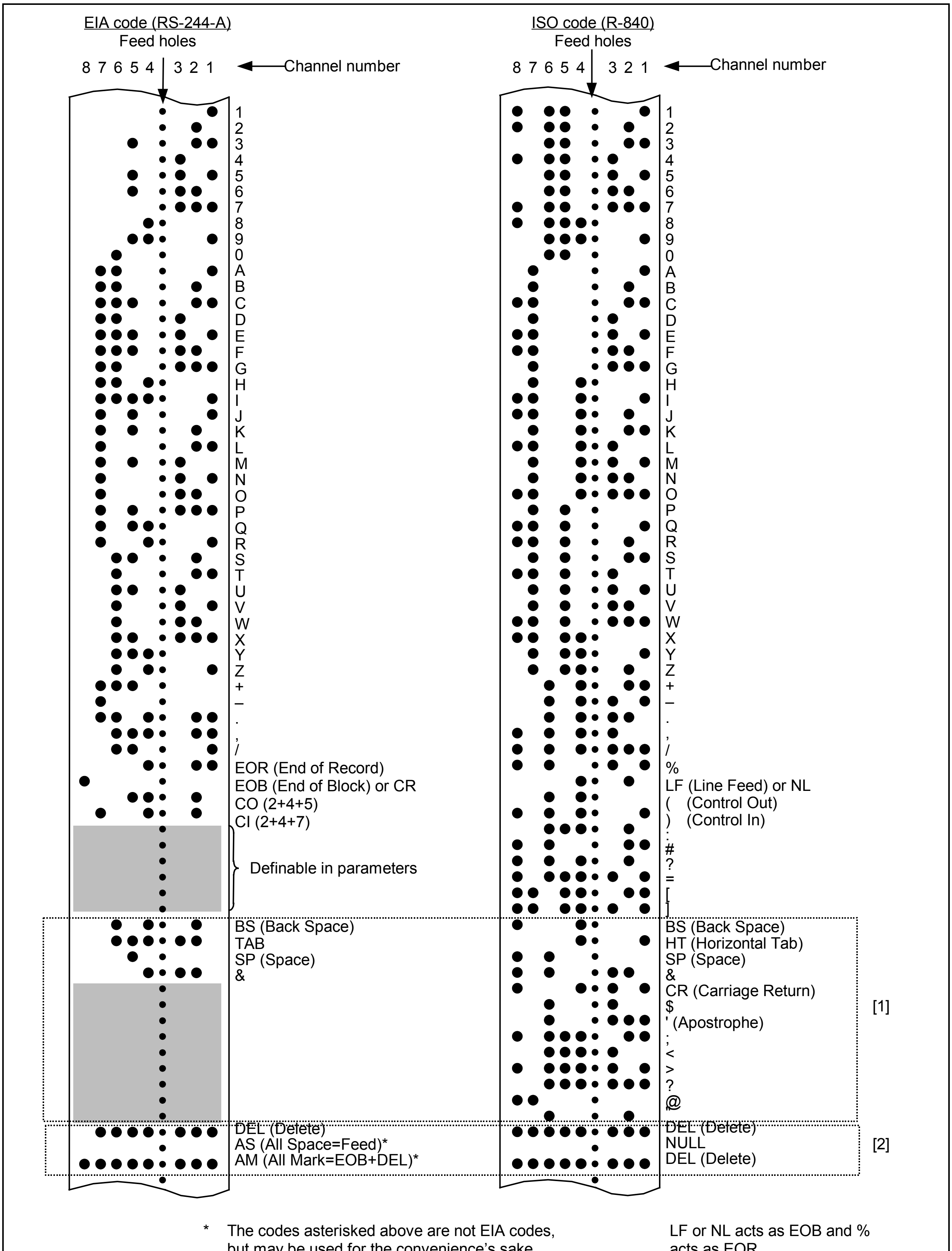
- To stop rewinding (only when a rewinding device is provided)
- To start rewinding during tape data search (only when a rewinding device is provided)
- To terminate the storage of tape data.

4. Tape creation method for tape operation (Only when a rewinding device is used)



The two meters of dummy at both ends and the EOR (%) at the head are not required when a rewinding device is used.

EIA/ISO identification is made automatically by detecting whether EOB or LF initially appears after the NC unit has been reset.



* The codes asterisked above are not EIA codes, but may be used for the convenience's sake.

LF or NL acts as EOB and % acts as EOR.

MEP006

Fig. 3-1 Tape codes

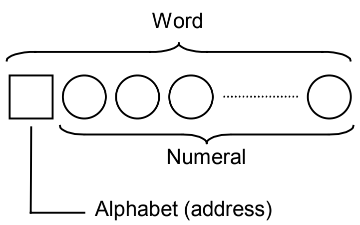
Codes in section [1] will only be stored as tape data when they are present in a comment section, and ignored elsewhere in the significant information area.
Codes in section [2] are non-operative and will always be ignored (but undergo the parity-V check).
A dotted area indicates that the EIA Standard provides no corresponding codes.

3-2 Program Formats

A format predetermined for assigning control information to the NC unit is referred to as a program format. The program format used for our NC unit is word address format.

1. Words and addresses

A word is a set of characters arranged as shown below, and information is processed in words.



Word configuration

The alphabetic character at the beginning of a word is referred to as an address, which defines the meaning of its succeeding numeric information.

Table 3-1 Type and format of words

Item			Metric command	Inch command
Program No.			O8	
Sequence No.			N5	
Preparatory function			G3 or G21	
Moving axis	Input unit	0.0001 mm (deg.), 0.00001 in.	X+54 Y+54 Z+54 α+54	X+45 Y+45 Z+45 α+45
Auxiliary axis		0.0001 mm (deg.), 0.00001 in.	I+54 J+54 K+54	I+45 J+45 K+45
Dwell		0.001 mm (rev), 0.0001 in.	X54 P8 U54	
Feed		0.0001 mm (deg.)/min, 0.00001 in./min	F54 (per minute) F33 (per revolution)	F45 (per minute) F24 (per revolution)
Fixed cycle		0.0001 mm (deg.), 0.00001 in.	R+54 Q54 P8 L4	R+45 Q45 P8 L4
Tool offset			T1 or T2	
Miscellaneous function			M3 × 4	
Spindle function			S5	
Tool function			T4 or T6	
No. 2 miscellaneous function			B8, A8 or C8	
Subprogram			P4 Q5 L4	
Variables number			#5	

1. Code O8 here indicates that program number can be set as an unsigned integer of eight digits following O, and for X+54, “+” indicates that the value can be signed (negative) and the two-digit number (54) indicates that the decimal point can be used and that five digits before and four after the decimal point are effective (5 + 4 = 9 digits are effective for a designation without decimal point).
2. The alpha sign (α) denotes additional axis address. +44 will be used when α is specified for rotational axis.
3. The number of digits in the words is checked by the maximum number of digits in the addresses.
4. When data with decimal point is used for address for which decimal input is not available, decimal figures will be ignored.
5. If the number of integral digits exceeds the specified format, an alarm will result.
6. If the number of decimal digits exceed the specified format, the excess will be rounded.

2. Blocks

A block, unit of instruction, contains a number of words which constitute information necessary for the NC machine to perform an operation. The end of each block must be indicated by an EOB (End Of Block) code.

3. Programs

A number of blocks form one program.

4. Program end

M02, M30, M99, M998, M999 or % is used as program end code.

3-3 Tape Data Storage Format

As with tape operation, tape data to be stored into the memory can be either of ISO or EIA code. The first EOB code read in after resetting is used by the NC unit for automatic identification of the code system ISO or EIA.
The area of tape data to be stored into the memeory is, if the NC unit has been reset, from the character immediately succeeding the first EOB code the EOR code, and in all other cases, from the current tape position to the EOR code. Usually, therefore, start tape data storage operation after resetting the NC unit.

3-4 Optional Block Skip

1. Function and purpose

Optional block skip is a function that selectively ignores that specific block within a machining program which begins with the slash code “/”.
Any block beginning with “/” will be ignored if the **[BLOCK SKIP]** menu function is set to ON, or will be executed if the menu function is set to OFF.
For example, if all blocks are to be executed for a type of parts but specific blocks are not to be executed for another type, then different parts can be machined using one and the same program that contains the “/” code at the beginning of the specific blocks.

2. Operating notes

1. Blocks that have already been read into the pre-read buffer cannot be skipped.
2. This function is valid even during sequence number search.
3. During tape data storage (input) or output, all blocks, including those having a "/" code, are in- or outputted, irrespective of the status of the **[BLOCK SKIP]** menu function.

3-5 Program Number, Sequence Number and Block Number : O, N

Program numbers, sequence numbers, and block numbers are used to monitor the execution status of a machining program or to call a machining program or a specific process within a machining program.
Program numbers are assigned to command blocks as required. A program number must be set using the letter O (address) and a numeric of a maximum of eight digits that follow O.
Sequence numbers identify command blocks forming a machining program. A sequence number must be set using the letter N (address) and a numeric of a maximum of five digits that follow N.
Block numbers are counted automatically within the NC unit, and reset to 0 each time a program number or a sequence number is read. These numbers will be counted up by one if the block to be read does not have an assigned program number or sequence number.
All blocks of a machining program, therefore, can be uniquely defined by combining program number, sequence number, and block number as shown in the table below.

NC input machining program	NC MONITOR display		
	Program No.	Sequence No.	Block No.
O1234 (DEMO. PROG);	1234	0	0
N100 G00 X120. Z100.;	1234	100	0
G98 S1000;	1234	100	1
N102 G71 P210 Q220 I0.2 K0.2 D0.5 F600;	1234	102	0
N200 G98 S1200 F300;	1234	200	0
N210 G01 X0 Z95.;	1234	210	0
G01 X20.;	1234	210	1
G03 X50. Z80. K-15.;	1234	210	2
G01 Z55.;	1234	210	3
G02 X80. Z40. I15.;	1234	210	4
G01 X100.;	1234	210	5
G01 Z30.;	1234	210	6
G02 Z10. K-15.;	1234	210	7
N220 G01 Z0;	1234	220	0
N230 G00 X120. Z150.;	1234	230	0
N240 M02;	1234	240	0
%	1234	240	0

3-6 Parity-H/V

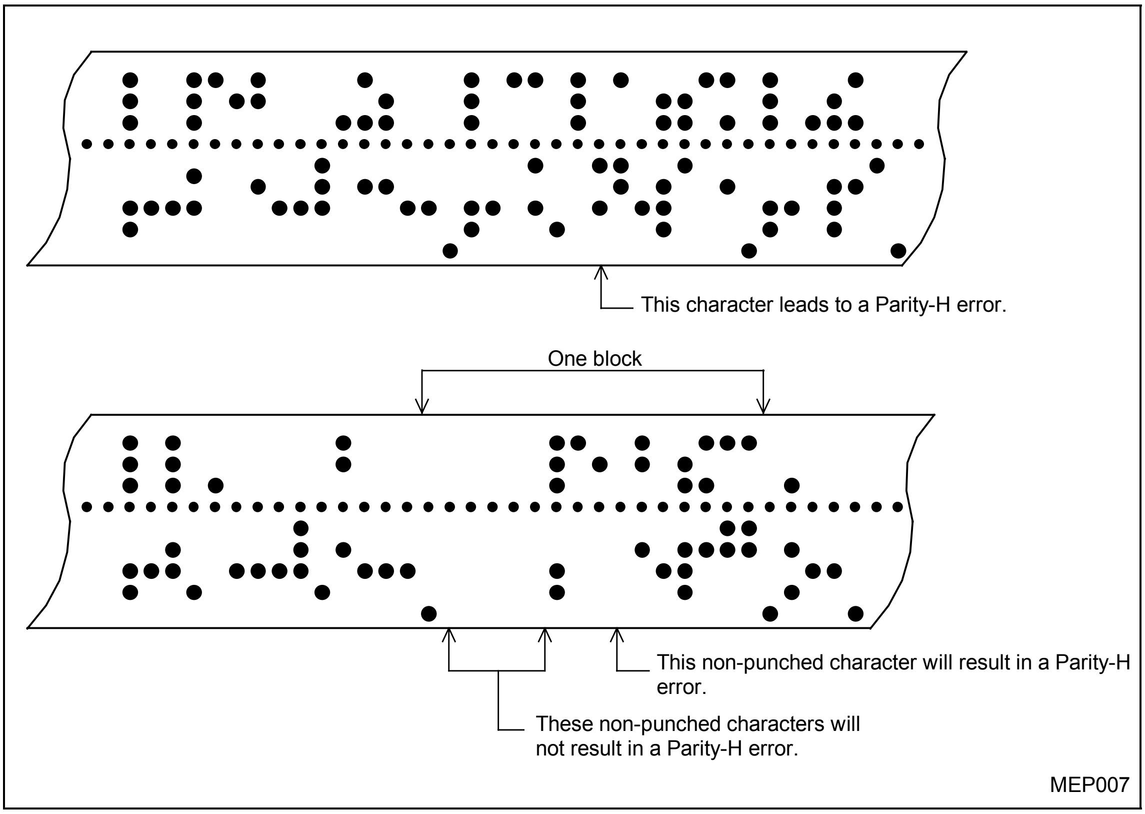
One method of checking if the tape is correctly created is by parity checks. Parity checks are performed to check a tape for errors in punched codes, that is, for punching errors. There are two types of parity checks: parity-H and parity-V.

1. Parity-H check

Parity-H checks are intended to check the quantity of punched holes which form one character, and performed during tape operation, tape loading, and sequence-number searching.
A parity-H error occurs in the following cases:

- ISO Codes
  If a code with an odd number of punched holes is present in the significant information area.
- EIA Codes
  If a code with an even number of punched holes is present in the significant information area or if non-punched holes (sprockets only) are present after a significant code in one block.

Example 1: Parity-H error (for EIA codes)



If a parity-H error occurs, the tape will stop at the position next to the error code.

2. Parity-V check

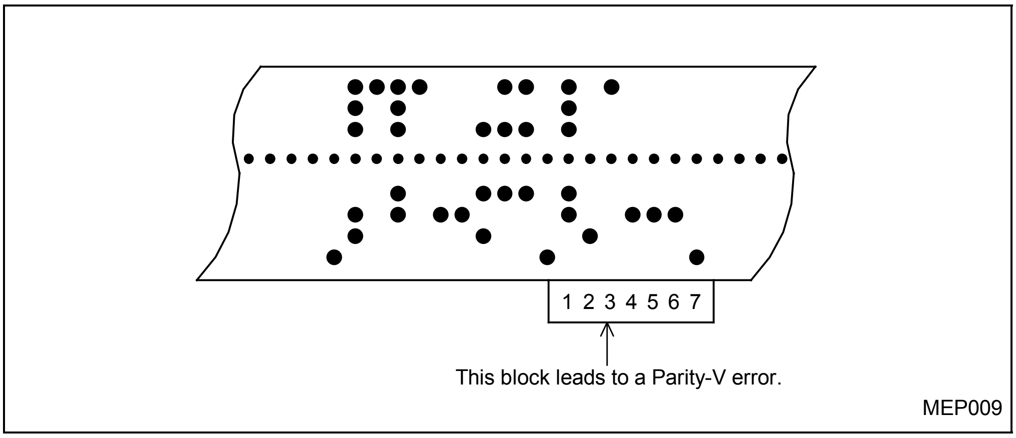
Parity-V checks will be performed during tape operation, tape loading, or sequence-number searching, if parity-V check item on the **PARAMETER** display is set to ON. Parity-V during memory operation, however, will not be checked.

A parity-V error occurs in the following case:

If an odd number of codes are present in the significant information area from the first significant code in the vertical direction to the EOB code (;), that is, if an odd number of characters are present in one block.

In the event of a parity-V error, the tape stops at a code next to the EOB (;).

Example 2: An example of parity-V error



Note 1: During a parity-V check, some types of code are not counted as characters. See Fig. 3-1, “Tape codes” for further details.

Note 2: Space codes in the area from the first EOB code to the first address code or slash code “/” are not subjected to counting for parity-V check.

3-7 List of G-Codes

G functions are described in the list below.

Function	G-code series		Group
	T	M	
Positioning	■G00	■G00	01
Linear interpolation	■G01	■G01	01
Threading with C-axis interpolation	G01.1	G01.1	01
Circular interpolation (CW)	G02	G02	01
Circular interpolation (CCW)	G03	G03	01
Spiral interpolation (CW)	G02.1	G02.1	01
Spiral interpolation (CCW)	G03.1	G03.1	01
Dwell	G04	G04	00
High-speed machining mode	G05	G05	00
Fine spline interpolation	G06.1	G06.1	01
NURBS interpolation	G06.2	G06.2	01
Virtual-axis interpolation	G07	G07	00
Cylindrical interpolation	G07.1	G07.1	00
Exact-stop check	G09	G09	00
Data setting mode ON	G10	G10	00
Command address OFF	G10.1	G10.1	00
Data setting mode OFF	G11	G11	00
Polar coordinate interpolation ON	G12.1	G12.1	26
Polar coordinate interpolation OFF	▲G13.1	▲G13.1	26
X-Y plane selection	■G17	■G17	02
Z-X plane selection	■G18	■G18	02
Y-Z plane selection	■G19	■G19	02
Inch command	■G20	■G20	06
Metric command	■G21	■G21	06
Pre-move stroke check ON	G22	G22	04
Pre-move stroke check OFF	▲G23	▲G23	04
Reference point check	G27	G27	00
Reference point return	G28	G28	00
Return from reference point	G29	G29	00
Return to 2nd, 3rd and 4th reference points	G30	G30	00
Skip function	G31	G31	00
Multi-step skip 1	G31.1	G31.1	00
Multi-step skip 2	G31.2	G31.2	00
Multi-step skip 3	G31.3	G31.3	00
Thread cutting (straight, taper)	G32	G33	01
Variable lead thread cutting	G34	G34	01
Hole machining pattern cycle (on a circle)	G234.1	G34.1	00
Hole machining pattern cycle (on a line)	G235	G35	00
Hole machining pattern cycle (on an arc)	G236	G36	00
Hole machining pattern cycle (on a grid)	G237.1	G37.1	00
Automatic tool length measurement	—	G37	00
Vector selection for tool radius compensation	—	G38	00
Corner arc for tool radius compensation	—	G39	00
Nose R/Tool radius compensation OFF	▲G40	▲G40	07
Nose R/Tool radius compensation (left)	G41	G41	07

Function	G-code series		Group
	T	M	
3-D tool radius compensation (left)	G41.2	G41.2	07
Nose R/Tool radius compensation (right)	G42	G42	07
3-D tool radius compensation (right)	G42.2	G42.2	07
Tool length offset (+)	—	G43	08
Tool tip point control (Type 1) ON	G43.4	G43.4	08
Tool tip point control (Type 2) ON	G43.5	G43.5	08
Tool length offset (–)	—	G44	08
Tool position offset, extension	—	G45	00
Tool position offset, reduction	—	G46	00
Tool position offset, double extension	—	G47	00
Tool position offset, double reduction	—	G48	00
Tool position offset OFF	—	▲G49	08
Coordinate system setting/Spindle clamp speed setting	G50	G92	00
Scaling OFF	—	▲G50	11
Scaling ON	—	G51	11
Mirror image OFF	—	▲G50.1	19
Mirror image ON	—	G51.1	19
Polygonal machining mode OFF	▲G50.2	▲G50.2	23
Polygonal machining mode ON	G51.2	G51.2	23
Local coordinate system setting	G52	G52	00
MAZATROL coordinate system cancel	■G52.5	—	00
Machine coordinate system selection	G53	G53	00
MAZATROL coordinate system selection	■G53.5	—	00
Selection of workpiece coordinate system 1	▲G54	▲G54	12
Selection of workpiece coordinate system 2	G55	G55	12
Selection of workpiece coordinate system 3	G56	G56	12
Selection of workpiece coordinate system 4	G57	G57	12
Selection of workpiece coordinate system 5	G58	G58	12
Selection of workpiece coordinate system 6	G59	G59	12
Additional workpiece coordinate systems	G54.1	G54.1	12
Selection of fixture offset	—	G54.2	23
One-way positioning	G60	G60	00
Exact stop mode	G61	G61	13
High-accuracy mode (Geometry compensation)	G61.1	G61.1	13
Automatic corner override	G62	G62	13
Tapping mode	G63	G63	13
Cutting mode	▲G64	▲G64	13
User macro single call	G65	G65	00
User macro modal call A	G66	G66	14
User macro modal call B	G66.1	G66.1	14
User macro modal call OFF	▲G67	▲G67	14
Programmed coordinate rotation ON	—	G68	16
Programmed coordinate rotation OFF	—	G69	16
3-D coordinate conversion ON	G68.5	G68	16
3-D coordinate conversion OFF	▲G69.5	▲G69	16
Finishing cycle	G70	G270	09
Longitudinal roughing cycle	G71	G271	09
Transverse roughing cycle	G72	G272	09

Function	G-code series		Group
	T	M	
Contour-parallel roughing cycle	G73	G273	09
Longitudinal cut-off cycle	G74	G274	09
Transverse cut-off cycle	G75	G275	09
Compound thread-cutting cycle	G76	G276	09
Fixed cycle OFF	▲G80	▲G80	09
Front driling cycle	G83	G283	09
Front tapping cycle	G84	G284	09
Front synchronous tapping cycle	G84.2	G284.2	09
Front boring cycle	G85	G285	09
Outside driling cycle	G87	G287	09
Outside tapping cycle	G88	G288	09
Outside synchronous tapping cycle	G88.2	G288.2	09
Outside boring cycle	G89	G289	09
Fixed cycle A (Longitudinal turning cycle)	G90	G290	09
Threading cycle	G92	G292	09
Fixed cycle B (Transverse turning cycle)	G94	G294	09
Fixed cycle (Chamfering cutter 1, CW)	—	G71.1	09
Fixed cycle (Chamfering cutter 2, CCW)	—	G72.1	09
Fixed cycle (High-speed deep-hole drilling)	—	G73	09
Fixed cycle (Reverse tapping)	—	G74	09
Fixed cycle (Boring 1)	—	G75	09
Fixed cycle (Boring 2)	—	G76	09
Fixed cycle (Back spot facing)	—	G77	09
Fixed cycle (Boring 3)	—	G78	09
Fixed cycle (Boring 4)	—	G79	09
Fixed cycle (Spot drilling)	—	G81	09
Fixed cycle (Drilling)	—	G82	09
Fixed cycle (Deep-hole drilling)	—	G83	09
Fixed cycle (Tapping)	—	G84	09
Fixed cycle (Synchronous tapping)	—	G84.2	09
Fixed cycle (Synchronous reverse tapping)	—	G84.3	09
Fixed cycle (Reaming)	—	G85	09
Fixed cycle (Boring 5)	—	G86	09
Fixed cycle (Back boring)	—	G87	09
Fixed cycle (Boring 6)	—	G88	09
Fixed cycle (Boring 7)	—	G89	09
Absolute data input	—	■G90	03
Incremental data input	—	■G91	03
Workpiece coordinate system rotation	—	G92.5	00
Inverse time feed	G93	G93	05
Constant peripheral speed control ON	■G96	■G96	17
Constant peripheral speed control OFF	■G97	■G97	17
Feed per minute (asynchronous)	■G98	■G94	05
Feed per revolution (synchronous)	■G99	■G95	05
Initial point level return in fixed cycles	—	▲G98	10
R-point level return in fixed cycles	—	G99	10
Single program multi-system control	G109	G109	00

Function	G-code series		Group
	T	M	
Cross machining control ON	G110	G110	20
Cross machining control OFF	G111	G111	20
M, S, T, B output to opposite system	G112	G112	00
Hob milling mode OFF	G113	G113	23
Hob milling mode ON	G114.3	G114.3	23
Polar coordinate input ON	G122	G16	18
Polar coordinate input OFF	G123	G15	18
X-axis radial command ON	G122.1	—	00
X-axis radial command OFF	▲G123.1	—	00
Selection between diameter and radius data input	—	G10.9	
Tornado cycle	G130	G130	
Measurement macro, workpiece/coordinate measurement	G136	G136	
Compensation macro	G137	G137	

Notes:

1. The codes marked with ▲ are selected in each group when the power is turned ON or executing reset for initializing modal.
2. The codes marked with ■ are able to be selected by a parameter as an initial modal which is to become valid when the power is turned ON or executing reset for initializing modal. Changeover of inch/metric system, however, can be made valid only by turning the power ON.
3. G-codes of group 00 are those which are not modal, and they are valid only for commanded blocks.
4. If a G-code not given in the G-code list is commanded, an alarm is displayed. And if a G-code without corresponding option is commanded, an alarm is displayed (**808 MIS-SET G CODE**).
5. If G-codes belong to different groups each other, any G-code can be commanded in the same block. The G-codes are then processed in order of increasing group number. If two or more G-codes belonging to the same group are commanded in the same block, a G-code commanded last is valid.

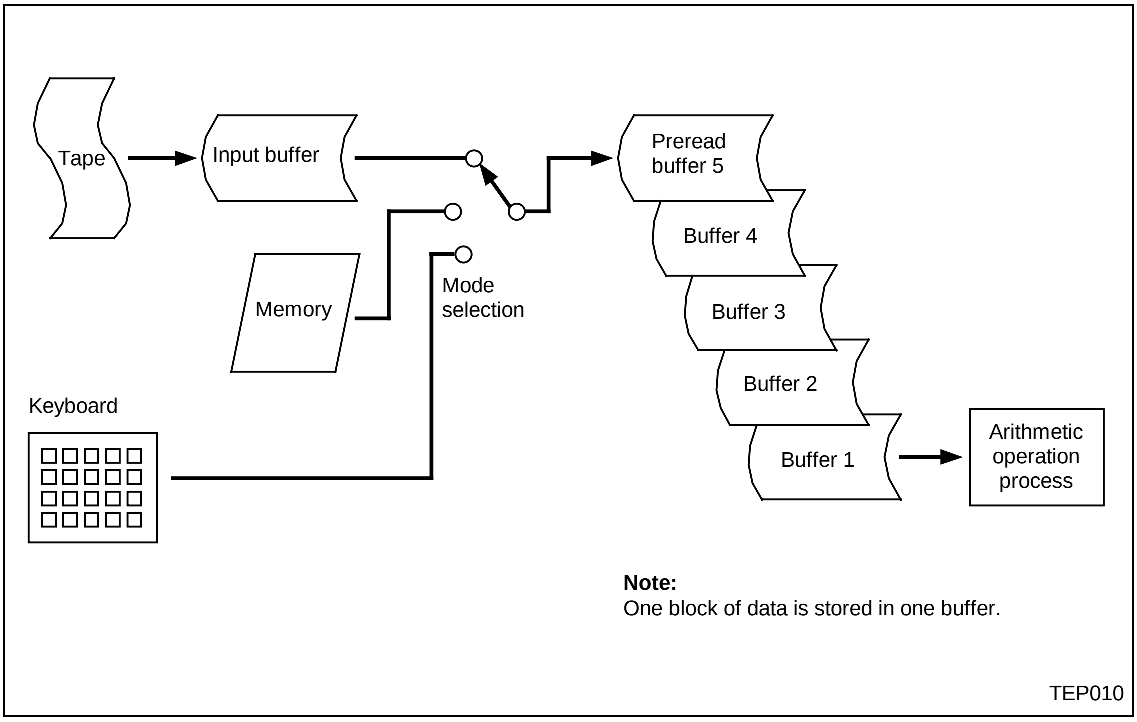
- NOTE -

4 BUFFER REGISTERS

4-1 Input Buffer

1. Overview

During tape operation or RS-232C operation, when the preread buffer becomes empty, the contents of the input buffer will be immediately shifted into the pre-read buffer and, following this, if the memory capacity of the input buffer diminishes to 248 × 4 characters or less, next data (up to 248 characters) will be preread from the tape and then stored into the input buffer.
The input buffer makes block-to-block connections smooth by eliminating any operational delays due to the tape-reading time of the tape reader.
These favorable results of prereading, however, will be obtained only if the execution time of the block is longer than the tape-reading time of the next block.



2. Detailed description

- The memory capacity of the input buffer is 248 × 5 characters (including the EOB code).
- The contents of the input buffer register are updated in 248-character units.
- Only the significant codes in the significant information area are read into the buffer.
- Codes, including “(” and “)”, that exist between Control Out and Control In, are read into the input buffer. Even if optional block skip is valid, codes from / to EOB will also be read into the input buffer.
- The contents of the buffer are cleared by a reset command.

4-2 Preread Buffer

1. Overview

During automatic operation, one block of data is usually preread to ensure smooth analysis of the program. During tool nose radius compensation, however, maximal five blocks of data are preread to calculate crossing point or to check the interference.

2. Detailed description

- One block of data is stored into the prepared buffer.
- Only the significant codes in the significant information area are read into the pre-read buffer.
- Codes existing between Control Out and Control In are not read into the pre-read buffer. If optional block skip is valid, codes from / to EOB will not also be read into the pre-read buffer.
- The contents of the buffer are cleared by a reset command.
- If the single block operation mode is selected during continuous operation, processing will stop after pre-reading the next block data.

Return to Library

5 POSITION PROGRAMMING

5-1 Dimensional Data Input Method

5-1-1 Absolute/Incremental data input (Series T)

In the use of G-code series T, absolute and incremental data input methods are distinguished by axis addresses as shown in the table below.

<table>
<tr><th colspan="2"></th><th>Command system</th><th>Remarks</th></tr>
<tr><td rowspan="4">Absolute data</td><td>X-axis</td><td>Address X</td><td rowspan="8">- The address corresponding to the desired axis is to be set by machine parameter.
- Absolute and incremental data can be used together in the same block.
- Address of incremental data input for A- and B-axes does not exist.</td></tr>
<tr><td>Z-axis</td><td>Address Z</td></tr>
<tr><td>C-aixs</td><td>Address C</td></tr>
<tr><td>Y-aixs</td><td>Address Y</td></tr>
<tr><td rowspan="4">Incremental data</td><td>X-axis</td><td>Address U</td></tr>
<tr><td>Z-axis</td><td>Address W</td></tr>
<tr><td>C-aixs</td><td>Address H</td></tr>
<tr><td>Y-aixs</td><td>Address V</td></tr>
</table>

Example: X____W____;

Incremental data input for the Z-axis

Absolute data input for the X-axis

5-1-2 Absolute/Incremental data input: G90/G91 (Series M)

1. Function and purpose

Setting of G90 or G91 allows succeeding dimensional data to be processed as absolute data or incremental data.
Setting of arc radius (with address R) or arc center position (with addresses I, J, K) for circular interpolation, however, must always refer to incremental data input, irrespective of preceding G90 command.

2. Programming format

G90 (or G91) Xx_1 Yy_1 Zz_1 $\alpha\alpha_1$ (α: Additional axis)
where G90: Absolute data input
G91: Incremental data input

3. Detailed description

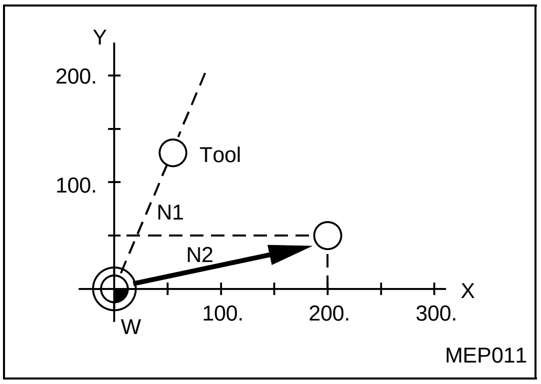
1. In the absolute data mode, axis movement will be performed to the program-designated position within the workpiece coordinate system, irrespective of the current position.

```
N1 G90G00X0 Y0
```

In the incremental data mode, axis movement will be performed through the program-designated distance as relative data with respect to the current position.

```
N2 G91G01X200. Y50. F100
```

```
N2 G90G01X200. Y50. F100
```



MEP011

Commands for a movement from the origin of the workpiece coordinate system are given with the same values, irrespective of whether the absolute data mode or the incremental data mode is used.

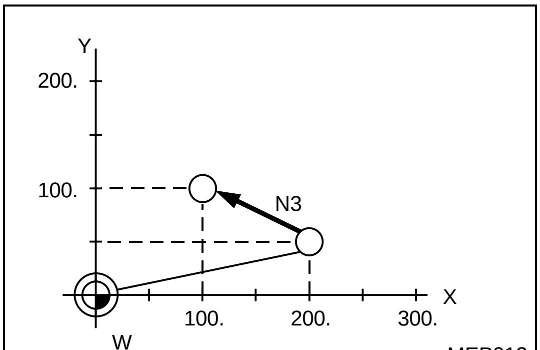
2. The last G90 or G91 command works as a modal one for the following blocks.

(G90)
```
N3 X100. Y100.
```

This block will perform a movement to the position of X = 100 and Y = 100 in the workpiece coordinate system.

(G91)
```
N3 X-100. Y50.
```

This block will perform a movement of –100 on the X-axis and +50 on the Y-axis, and thus result in a movement to the position of X = 100 and Y = 100.

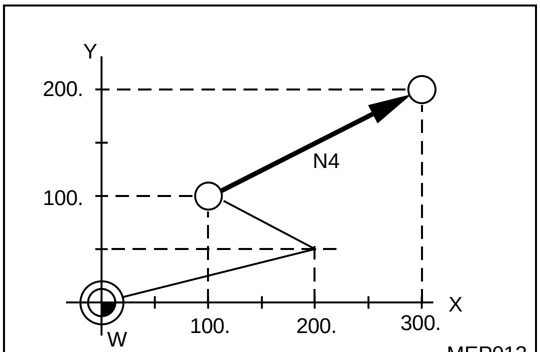


MEP012

3. Multiple G90 or G91 commands can be set in one block, and thus only a specific address can be set as absolute data or incremental data.

```
N4 G90X300. G91Y100.
```

In this example, dimensional data X300 preceded by G90 will be processed as an absolute data input, and Y100 preceded by G91 as an incremental data input. Therefore, this block will result in a movement to the position of X = 300 and Y = 200 (100 + 100) in the workpiece coordinate system.



MEP013

Moreover, G91 (incremental data input mode) will work for the succeeding blocks.

4. Either the absolute data mode or the incremental data mode can be freely selected as initial mode by setting the bit 2 of user parameter **F93**.

5. Even in the MDI (Manual Data Input) mode, G90 and G91 will also be handled as modal commands.

5-2 Inch/Metric Selection: G20/G21

1. Function and purpose

Inch command/metric command selection is possible with G-code commands.

2. Programming format

G20: Inch command selection
G21: Metric command selection

3. Detailed description

1. Changeover between G20 and G21 is effective only for linear axes; it is meaningless for rotational axes.

Example: Preset unit of data input and G20/G21 (for decimal-point input type I)

Axis	Example	Initial Inch (parameter) OFF		Initial Inch (parameter) ON	
		G21	G20	G21	G20
X	X100	0.0100 mm	0.0254 mm	0.00039 inches	0.00100 inches
Y	Y100	0.0100 mm	0.0254 mm	0.00039 inches	0.00100 inches
Z	Z100	0.0100 mm	0.0254 mm	0.00039 inches	0.00100 inches
B	B100	0.0100 deg	0.0100 deg	0.0100 deg	0.0100 deg

2. To perform G20/G21 changeover in a program, you must first convert variables, parameters, and offsetting data (such as tool length/tool position/tool diameter offsetting data) according to the unit of data input for the desired system (inch or metric) and then set all these types of data either on each data setting display or using the programmed parameter input function.

Example: If Initial inch selection is OFF and offsetting data is 0.05 mm, the offsetting data must be converted to 0.002 (0.05 ÷ 25.4 ≈ 0.002) before changing the G21 mode over to the G20 mode.

3. In principle, G20/G21 selection should be done before machining. If you want this changeover to be performed in the middle of the program, temporarily stop the program by an M00 command after G20 or G21 and convert the offsetting data as required.

Example:

```
G21  G92  Xx1  Yy1  Zz1
      :     :
      :     :
G20  G92  Xx2  Yy2  Zz2
     M00  →   Convert offsetting data here.
      :
     F10  →   Set an F (Feed rate) command anew.
```

Note: Do not fail to give an F command appropriate to the new unit system after changeover between G20 and G21. Otherwise, axis movements would be performed using the last F value before the changeover, without any conversion, on the basis of the new unit system.

4. Whether G20 or G21 is to be selected upon switching-on can be specified by the bit 4 of user parameter **F91** (Initial Inch parameter).

5-3 Decimal Point Input

1. Function and purpose

The decimal point can be used to determin the units digit (mm or inch) of dimensional data or feed rate.

2. Programming format

○○○○○.○○○○ Metric system

○○○○.○○○○○ Inch system

3. Detailed description

1. Decimal-point commands are valid only for the distance, angle, time, speed, and scaling factor (only after G51) that have been set in the machining program.
2. As listed in the table below, the meaning of command data without the decimal point differs between decimal-point input types I and II according to the type of command unit system.

Command	Command unit × 10	Type I	Type II
X1	OFF	0.0001 (mm, inches, deg)	1.0000 (mm, inches, deg)
	ON	0.0010 (mm, inches, deg)	1.0000 (mm, inches, deg)

3. Decimal-point commands are only valid for addresses X, Y, Z, U, V, W, A, B, C, I, J, K, E, F, P, Q and R, where address P only refers to a scaling factor.
4. The number of effective digits for each type of decimal-point command is as follows:

	Move command (Linear)		Move command (Rotational)		Feed rate		Dwell	
	Integral part	Decimal part	Integral part	Decimal part	Integral part	Decimal part	Integral part	Decimal part
mm	0. - 99999.	.0000 - .9999	0. - 99999.	.0000 - .9999	0. - 200000.	.0000 - .9999	0. - 99999.	.000 - .999
inch	0. - 9999.	.00000 - .99999	0. - 99999. (359.)	.0000 - .9999	0. - 20000.	.00000 - .99999	0. - 99999.	.000 - .999

5. Decimal-point commands are also valid for definition of variables data used in subprograms.
6. For data which can be, but is not specified with the decimal point, either the minimum program data input unit or mm (or in.) unit can be selected using bit 5 of parameter **F91**.
7. A decimal-point command issued for an address which does not accept the decimal point will be processed as data that consists of an integral part only. That is, all decimal digits will be ignored. Addresses that do not accept the decimal point are D, H, L, M, N, O, S and T. All types of variables command data are handled as the data having the decimal point.

4. Sample programs

A. Sample programs for addresses accepting the decimal point

Command category / Program example	For 1 = 1 μ	For 1 = 0.1 μ	1 = 1 mm
G0X123.45 (With the decimal point always given as the millimeter point)	X123.450 mm	X123.450 mm	X123.450 mm
G0X12345	X12.345 mm*	X1.2345 mm**	X12345.000 mm***
#111=123 #112=5.55 X#111 Y#112	X123.000 mm Y5.550 mm		
#113=#111+#112 (ADD)	#113 = 128.550		
#114=#111−#112 (SUBTRACT)	#114 = 117.450		
#115=#111.#112 (MULTIPLY)	#115 = 682.650		
#116=#111/#112 #117=#112/#111 (DIVIDE)	#116 = 22.162 #117 = 0.045		

* The least significant digit is given in 1 micron.
** The least significant digit is given in 0.1 micron.
*** The least significant digit is given in 1 mm.

Return to Library

B. Validity of decimal point for each address

Address	Decimal point command	Application	Remarks
A	Valid	Coordinate position data	
	Invalid	Rotary table Miscellaneous function code	
	Valid	Linear angle data	
B	Valid	Coordinate position data	
	Invalid	Rotary table Miscellaneous function code	
C	Valid	Coordinate position data	
	Invalid	Rotary table Miscellaneous function code	
	Valid	Corner chamfering amount	
D	Invalid	Offset number (tool position, tool length and tool diameter)	
E	Valid		
F	Valid	Feed rate	
G	Valid	Preparatory function code	
H	Invalid	Offset number (tool postion, tool length and tool diameter)	
	Invalid	Intra-subprogram sequence number	
I	Valid	Coordinate of arc center	
	Valid	Vector component for tool diameter offset	
J	Valid	Coordinate of arc center	
	Valid	Vector component for tool diameter offset	
K	Valid	Coordinate of arc center	
	Valid	Vector component for tool diamater offset	
	Valid	Knot for NURBS curve	
L	Invalid	Fixed cycle/subprogram repetition	
M	Invalid	Miscellaneous function code	
N	Invalid	Sequence number	
O	Invalid	Program number	

Address	Decimal point command	Application	Remarks
P	Invalid	Dwell time	
	Valid	Subprogram call number	
	Invalid	Number of helical pitches	
	Invalid	Offset amount (in G10)	
	Valid	Scaling factor	
	Invalid	Rank for NURBS curve	
Q	Valid	Cutting depth for deep-hole drilling cycle	
	Valid	Shift amount for back boring	
	Valid	Shift amount for fine boring	
R	Valid	R point in fixed cycle	
	Valid	Radius of an arc with R selected	
	Valid	Radius of an arc for corner rounding	
	Valid	Offset amount (in G10)	
	Valid	Weight for NURBS curve	
S	Invalid	Spindle function code	
T	Invalid	Tool function code	
U	Valid	Coordinate position data	
V	Valid	Coordinate position data	
W	Valid	Coordinate position data	
X	Valid	Coordinate position data	
	Valid	Dwell time	
Y	Valid	Coordinate position data	
Z	Valid	Coordinate position data	

Note: The decimal point is valid in all the arguments for a user macroprogram.

5-4 Polar Coordinate Input ON/OFF: G122/G123 [Series M: G16/G15]

1. Function and purpose

The end point of interpolation can be designated with polar coordinates (radius and angle). Polar coordinate input is available only in the mode of polar coordinate interpolation.

2. Programming format

G122........... Polar coordinate input ON (G-code group No. 18)
G123........... Polar coordinate input OFF (G-code group No. 18)

3. Detailed description

Even in the mode of polar coordinate input, positional commands for the axes that have no relation to the polar coordinate interpolation are available as ordinary commands.

In the mode of polar coordinate input, the length must always be designated in radius values, regardless of the modal state for radius/diameter data input (G122.1/G123.1). This also applies to the axes that have no relation to the polar coordinate interpolation.

The last modal state for radius/diameter data input before the G122 command will be restored automatically by the cancel command G123.

4. Sample program

```
G12.1;  ....................Polar coordinate interpolation ON
G122;   ....................Polar coordinate input ON
G01 X50.C30.F100;
G02 X50.C60.R50;
G123;   ....................Polar coordinate input OFF
G13.1;  ....................Polar coordinate interpolation OFF
```

5. Remarks

1. Enter polar coordinates with respect to the plane of polar coordinate interpolation.
2. Positive values (+) for angle data refer to measurement in the counterclockwise direction on the plane of polar coordinate interpolation.
3. Use address R to designate the radius for circular interpolation (G02 or G03).
4. If the G122 command is given without selecting the mode of polar coordinate interpolation (by G12.1), an alarm will occur.
5. If the polar coordinate interpolation mode is cancelled (by G13.1) during polar coordinate input, the mode of polar coordinate input will be cancelled together with the mode of polar coordinate interpolation.
6. G122 and G123 must be given in an independent block. That is, the block of G122 or G123 must not contain any other G-codes or addresses with the exception of N and P.
7. The following G-codes are available during polar coordinate input. An alarm will occur if any G-code other than these is specified.

   Available G-codes

G00	Positioning
G01	Linear interpolation
G02	Circular interpolation (CW)

G03	Circular interpolation (CCW)
G04	Dwell
G09	Exact-stop check
G13.1	Polar coordinate interpolation OFF
G15	Polar coordinate input OFF (in G-code series M)
G40-G42	Tool radius compensation
G61	Exact-stop mode
G64	Cutting mode
G65	User macro single call
G66	User macro modal call A
G66.1	User macro modal call B
G67	User macro modal call OFF
G80-G89	Fixed cycles for hole machining
G98	Asynchronous feed
G123	Polar coordinate input OFF

5-5 X-axis Radial Command ON/OFF: G122.1/G123.1 (Series T)

1. Function and purpose

The X-axis dimensions can be entered in radial values, instead of diametrical ones, by the aid of a preparatory function (G-code) in order to improve EIA/ISO programming efficiency for milling.

2. Programming format

G122.1........ X-axis radial data input ON (G-code group No. 25)
G123.1........ X-axis radial data input OFF (G-code group No. 25)

3. Detailed description

All the X-axis dimensions entered after G122.1 are processed as radial values until the command G123.1 is given for the restoration of diametrical data input mode for the X-axis.

4. Sample program

Program		Counter indication on **POSITION** display	Modal indication on **POSITION** display
⋮			
`G122.1;`	......X-axis radial data input ON		
⋮			
`G1X10.F100;`	......Radial dimension	`X20.`	`G122.1`
⋮			
`G123.1;`	......X-axis radial data input OFF		
⋮			
`G1X10.F100;`	......Diametrical dimension	`X10.`	`G123.1`
⋮			

5. Remarks

1. The counter indication on the **POSITION** display always refers to a diametrical value even in the mode of G122.1.
2. The selection of the G122.1 mode does not exercise any influence upon parameters, offset values, etc.
3. G123.1 is selected as the initial mode when the power is turned on.
4. Resetting causes the mode of G122.1 to be canceled and replaced by the G123.1 mode.

Return to Library

5. Even in the G122.1 mode the X-axis dimensions entered under the following modal functions are always processed as diametral values. Issuance of these G-code commands also cancels G122.1 mode:

   G7.1 Cylindrical interpolation
   G12.1 Polar coordinate interpolation ON
   G69.5 3-D coordinate conversion OFF
   G123 Polar coordinate input OFF
   G22 Pre-move stroke check ON

6. Even in the G123.1 mode the X-axis dimensions entered under the following modal functions are always processed as radial values (with diametrical indication on the **POSITON** display):

   G68.5 3-D coordinate conversion ON
   G122 Polar coordinate input ON

7. Various settings for software limits and barrier functions are not to be changed.

5-6 Selection between Diameter and Radius Data Input: G10.9 (Series M)

1. Function and purpose

The G10.9 command allows changeover between diameter data input and radius data input, facilitating the creation of the turning section in a compound machining program.

2. Programming format

G10.9 A_x_

A_x: Address of the axis for which diameter or radius data input is to be specified.
Numerical value = 0: Radius data input
1: Diameter data input

3. Remarks

1. Give the G10.9 command in a single-command block. Otherwise it may be ignored.
2. If the G10.9 command is not followed by an axis address, the alarm **807 ILLEGAL FORMAT** is caused. Also, the alarm **806 ILLEGAL ADDRESS** is caused if a rotational axis is specified in the G10.9 command.
3. Do not assign a decimal point to the numerical value that follows the axis address. Moreover, assigning a value other than 0 and 1 results in the alarm **809 ILLEGAL NUMBER INPUT**.
4. The G10.9 command only changes the method of programming the positional data for the particular axis. It does not affect various external data such as parameters, workpiece origin data, tool data, and tool offset data.
5. Irrespective of whether the absolute programming (G90) or the incremental programming (G91) is currently modal, designate the position in diameter values for the axis for which diameter data input has been selected.

Return to Library

4. Relationship to other G-codes

Diameter data input applies in general to the positional data of the specified axis.

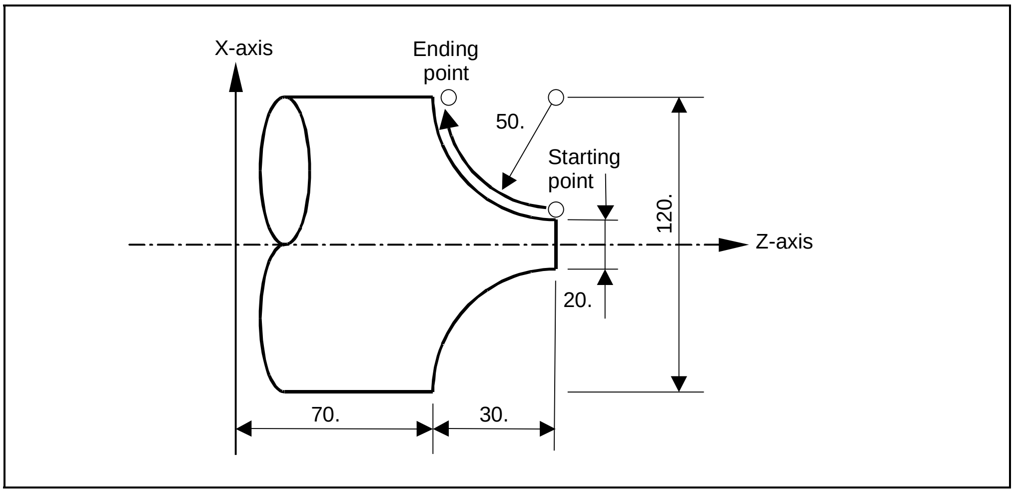
1. For positioning (G00), linear interpolation (G01) and coordinate system setting (G92)

   Designate the position in diameter values for the specified axis.

2. For circular interpolation (G02/G03)

   Only the position of the ending point is to be designated in a diameter value for the specified axis. The center, or radius, of the arc must always be designated in radius values (with I, K, or R). The example below refers to a turning program with the X-axis specified as the axis in question. The values with X and I denote the diameter data of the ending point and the radius data of the arc center (incremental to the starting point), respectively, for the X-axis.

   Absolute programming: `G90 G02 X120.Z70.I50.F200`
   Incremental programming: `G91 G02 X100.Z-30.I50.F200`



3. For fixed cycle of turning

   Designate the position in diameter values for the specified axis. The amount of taper (for turning fixed cycle) as well as the depth of cut and the finishing allowance (for compound cycle of turning), however, must always be designated in radius values.

4. For threading (G32/G33, G34, G1.1)

   Designate the position of the ending point in diameter values for the specified axis. The lead, however, must always be designated in radius values (with F or E).

- NOTE -

6 INTERPOLATION FUNCTIONS

6-1 Positioning (Rapid Feed) Command: G00

1. Function and purpose

Positioning command G00 involves use of a coordinate word. This command positions a tool by moving it linearly to the ending point specified by a coordinate word.

2. Programming format

G00 Xx/Uu Zz/Ww $\alpha\alpha$; (α denotes an additional axis, that is, B-, C- or Y-axis)

Where x, u, z, w and α denote a coordinate.
The command addresses are valid for all additional axis.

3. Detailed description

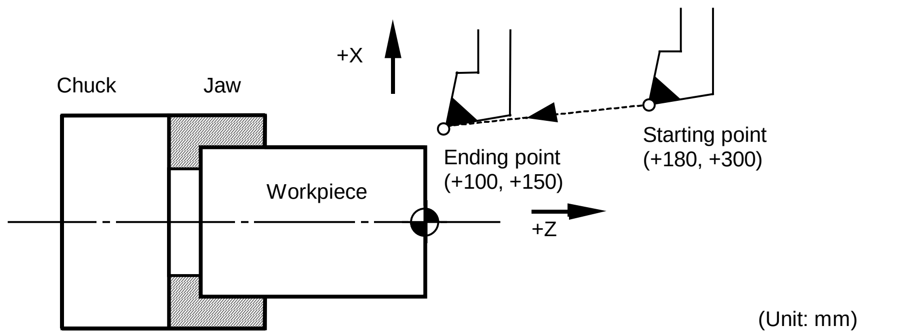
1. Once this command has been given, the G00 mode will be retained until any other G-code command that overrides this mode, that is, either G01, G02, G03, or G32 of command group 01 is given. Thus, a coordinate word will only need be given if the next command is also G00. This function is referred to the modal function of the command.
2. In the G00 mode, acceleration/deceleration always takes place at the starting/ending point of a block and the program proceeds to the next block after confirming that the pulse command in the present block is 0 and the tracking error of the acceleration/deceleration cycle is 0. The width of in-position can be changed using a parameter (**S13**).
3. The G-code functions (G83 to G89) of command group 09 are canceled by the G00 command (G80).
4. The tool path can be made either linear or nonlinear using a parameter (**F91** bit 6) but the positioning time remains unchanged.

   - Linear path
     As with linear interpolation (G01), the tool speed is limited according to the rapid feed rate of each axis.
   - Nonlinear path
     The tool is positioned according to the separate rapid feed rate of each axis.

5. When no number following G address, this is treated as G00.

4. Sample programs

Example:



TEP012'

The diagram above is for:

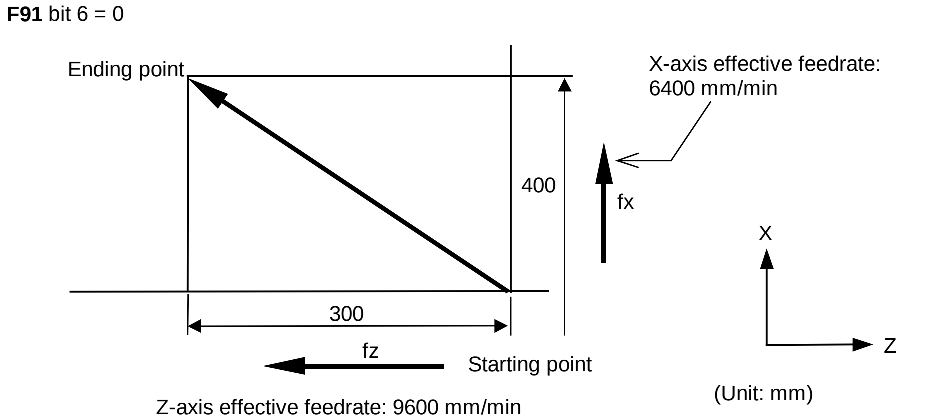
```
G00 X100.000 Z150.000;      Absolute data command
G00 U–80.000 W–150.000;     Incremental data command
```

5. Remarks

1. If bit 6 of user parameter **F91** is 0, the tool will take the shortest path connecting the starting and ending points. The positioning speed will be calculated automatically to give the shortest allocation time within the limits of the rapid feed rate of each axis.
   For example, if you set a rapid feed rate of 9600 mm/min for both X- and Z-axes and make the program:

   ```
   G00 Z–300.000 X400.000;
   ```

   then the tool will move as shown in the diagram below.



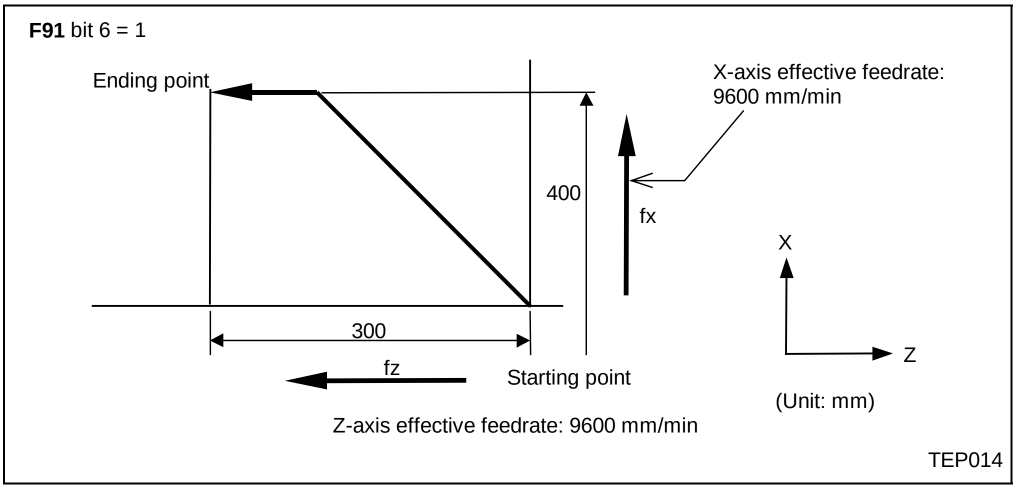
TEP013

   For inch-specification machines, the rapid feed rate of the C-axis is limited to 89 rpm (32000/360) even if item C of parameter **M1** is set to a value greater than 32000.

2. If bit 6 of user parameter **F91** is 1, the tool will move from the starting point to the ending point according to the rapid feed rate of each axis.
   For example, if you set a rapid feed rate of 9600 mm/min for both X- and Z-axes and make the program:

   ```
   G00 Z–300.000 X400.000;
   ```

then the tool will move as shown in the diagram below.



3. The rapid feed rate that you can set for each axis using the G00 command varies from machine to machine. Refer to the relevant machine specification for further details.

4. Rapid feed (G00) deceleration check
   When processing of rapid feed (G00) is completed, the next block will be executed after the deceleration check time (Td) has passed.
   The deceleration check time (Td) is calculated by following expressions depending on the acceleration/deceleration type.

   Linear acceleration/linear deceleration .... Td = Ts + a
   Exponential acceleration/linear deceleration .... Td = 2 × Ts + a
   Exponential acceleration/exponential deceleration........... Td = 2 × Ts + a
   (Where Ts is the acceleration time constant, a = 0 to 14 msec)

   The time required for the deceleration check during rapid feed is the longest among the rapid feed deceleration check times of each axis determined by the rapid feed acceleration/deceleration time constants and by the rapid feed acceleration/deceleration mode of the axes commanded simultaneously.

6-2 One-Way Positioning: G60

1. Function and purpose

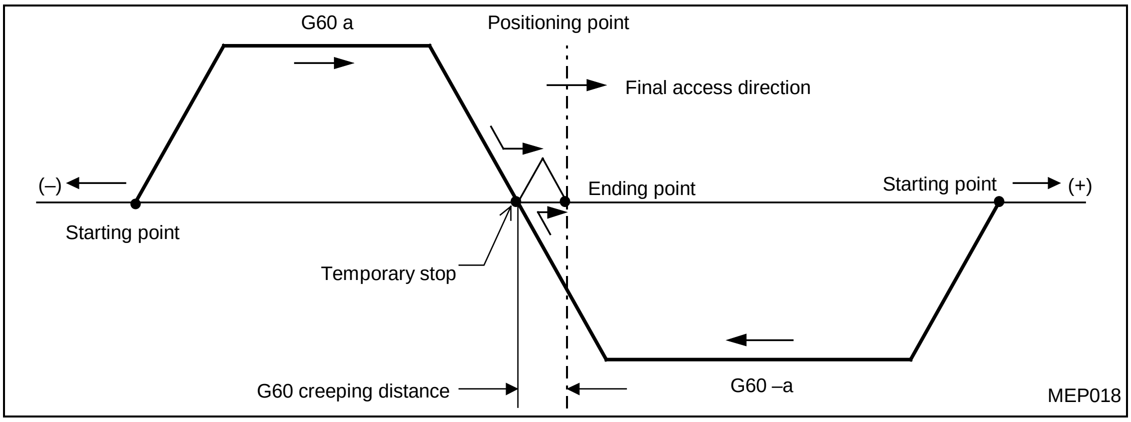
Highly accurate positioning free from any backlash error can be performed when the axis movement is controled by the G60 command so that the final access always takes place in one determined direction.

2. Programming format

G60 Xx/Uu Zz/Ww $\alpha\alpha$; (α: Additional axis)

3. Detailed description

1. The direction of final access and its creeping distance must be set in parameter **I1**.
2. After rapid approach to a position away from the ending point by the creeping distance, the final access is performed in the predetermined direction at a speed corresponding with the rapid feed.



3. The positioning pattern described above also applies during machine locking or for a Z-axis command with the Z-axis cancellation activated.
4. In the dry run mode (G00 mode), the whole positioning is carried out at the dry-running speed.
5. The creeping to the einding point can be halted with Reset, Emergency stop, Interlock, or Feed hold, or by setting the rapid feed override to 0 (zero).
   The creeping is performed according to the setting of the rapid feed, and the rapid feed override function is also effective for the creeping.
6. One-way positioning is automatically invalidated for the hole-drilling axis in hole-drilling fixed-cycle operations.
7. One-way positioning is automatically invalidated for shifting in fine-boring or back-boring fixed-cycle operations.
8. Usual positioning is performed for an axis not having a parameter-set creeping distance.
9. One-way positioning is always of non-interpolation type.
10. An axis movement command for the same position as the ending point of the preceding block (movement distance = 0) will cause reciprocation through the creeping distance so that the final access can be performed in the predetermined direction for an accurate positioning to the desired point.

6-3 Linear Interpolation Command: G01

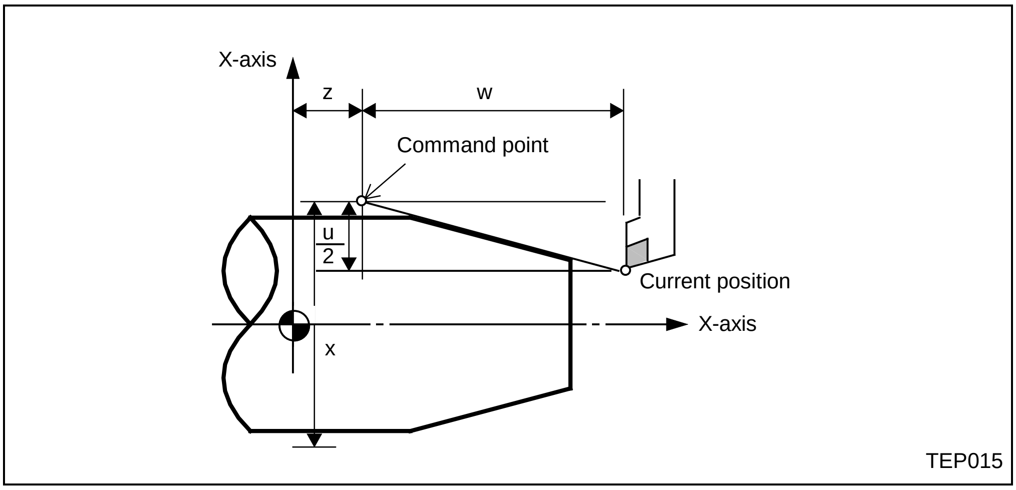
1. Function and purpose

Command G01 involves use of both a coordinate word and a feed rate command. This command moves (interpolates) linearly a tool from the current position to the ending point specified by a coordinate word, at the feed rate specified by address F. The feed rate specified by address F, however, acts as the linear velocity relative to the direction of movement of the tool center.

2. Programming format

G01 Xx/Uu Zz/Ww $\alpha\alpha$ Ff; (α: Additional axis)

where x, u, z, w and α each denote a coordinate.



3. Detailed description

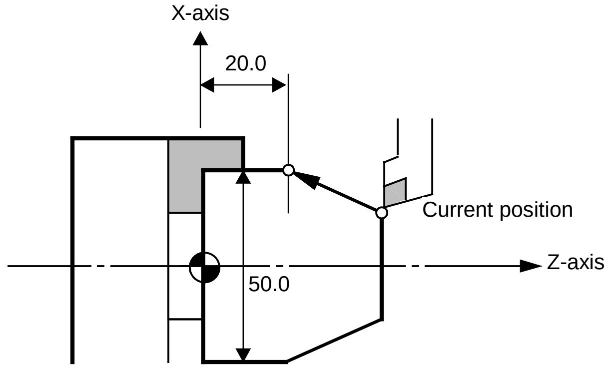
Once this command has been given, the G01 mode will be retained until any other G-code command that overrides this mode, that is, either G00, G02, G03 or G32 of command group 01 is given. Thus, a coordinate word will only need be given if the next command is also G01, that is, if the feed rate for the next block remains the same. A programming error will result if an F-code command is not given to the first G01 command.

The feed rates for rotational axes must be set in deg/min. (Example : F300 = 300 deg/min)

The G-code functions (G70 to G89) of command group 09 are cancelled by G01 (set to G80).

4. Sample program

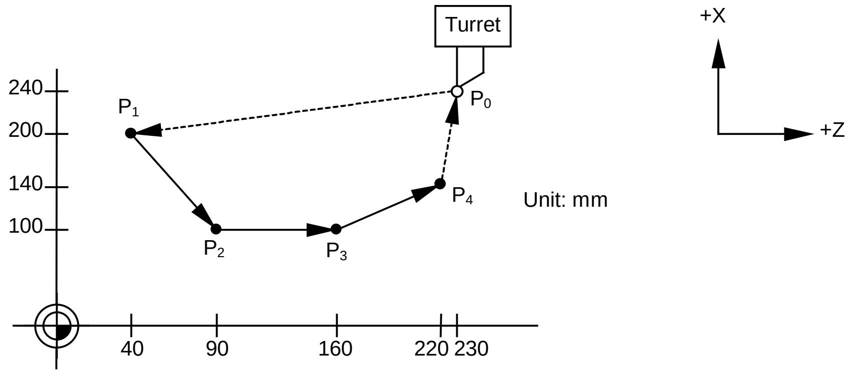
Example 1: Taper turning



TEP016

```
G01 X50.0 Z20.0 F300;
```

Example 2: Program for moving the tool at a cutting feed rate of 300 mm/min via the route of $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4$ (where the sections $P_0 \rightarrow P_1$ and $P_4 \rightarrow P_0$ form a positioning route for the tool):



TEP017

G00 X200.000 Z40.000;		$P_0 \rightarrow P_1$
G01 X100.000 Z90.000 F300;		$P_1 \rightarrow P_2$
Z160.000;		$P_2 \rightarrow P_3$
X140.000 Z220.000;		$P_3 \rightarrow P_4$
G00 X240.000 Z230.000;		$P_4 \rightarrow P_0$

6-4 Circular Interpolation Commands: G02, G03

1. Function and purpose

Commands G02 and G03 move the tool along an arc.

2. Programming format

G02 (G03) Xx/Uu Zz/Ww (Yy/Vv) Ii Kk (Jj) Ff ;

- Xx/Uu Zz/Ww (Yy/Vv): Coordinates of the ending point
- Ii Kk (Jj): Coordinates of the arc center
- Ff: Feedrate
- G03: Counterclockwise(CCW)
- G02: Clockwise (CW)

X/U: Arc ending point coordinates, X-axis (absolute value of workpiece coordinate system for X, incremental value from present position for U)

Z/W: Arc ending point coordinates, Z-axis (absolute value of workpiece coordinate system for Z, incremental value from present position for W)

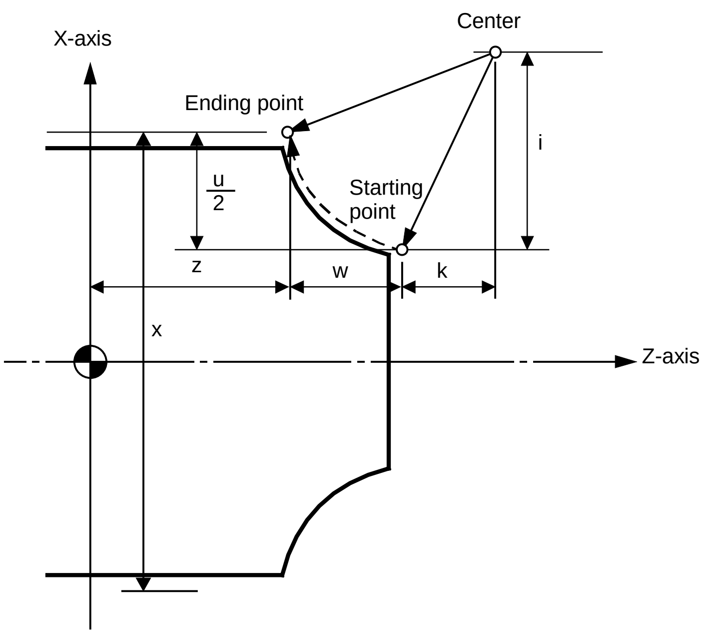
Y/V: Arc ending point coordinates, Y-axis (absolute value of workpiece coordinate system for Y, incremental value from present position for V)

I : Arc center, X-axis (radius command, incremental value from starting point)

K : Arc center, Z-axis (incremental value from starting point)

J : Arc center, Y-axis (incremental value from starting point)

F : Feed rate



TEP018

For machines with Y-axis control, arc interpolation is, additionally to Z-X plane, also available for X-Y and Y-Z planes.

X-Y plane   G17;
G02 (G03) X_Y_I_J_F_;   For milling on the face

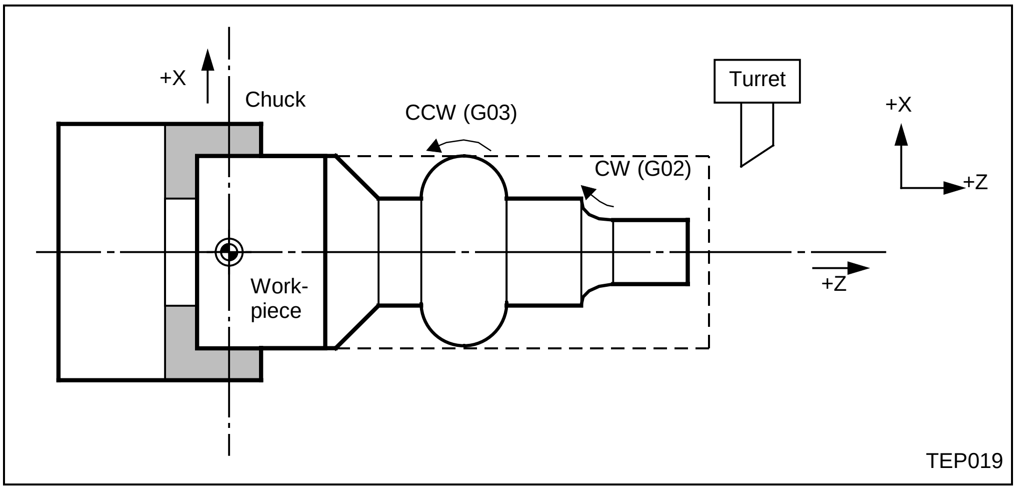
Z-X plane   G18;
G02 (G03) X_Z_I_K_F_;   For normal turning

Y-Z plane   G19;
G02 (G03) Y_Z_J_K_F_;   For Y-axis milling on OD surface

3. Detailed description

1. Once the G02 (or G03) command has been given, this command mode will be retained until any other G-code command used to override the G02 (or G03) command mode, that is, G00 or G01 of command group 01 is given.
2. The direction of circular movement is determined by G02/G03.

   G02: CW (Clockwise)
   G03: CCW (Counterclockwise)

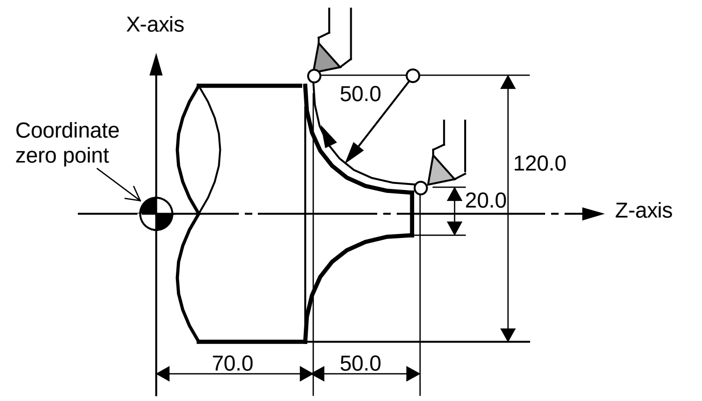


3. Interpolation of an arc that spans multiple quadrants can be defined with one block.
4. To perform circular interpolation, the following information is required:

   - Rotational direction .... CW (G02) or CCW (G03)
   - Arc ending point coordinates .... Given with address X, Z, Y, U, W, V.
   - Arc center coordinates .... Given with address I, K, J. (Incremental dimension)
   - Feed rate.................................... Given with address F.

5. If none of the addresses I, K, J and R is specified, a program error will occur.
6. Addresses I, K and J are used to specify the coordinates of the arc center in the X, Z and Y directions respectively as seen from the starting point, therefore, care must be taken for signs.

4. Sample programs

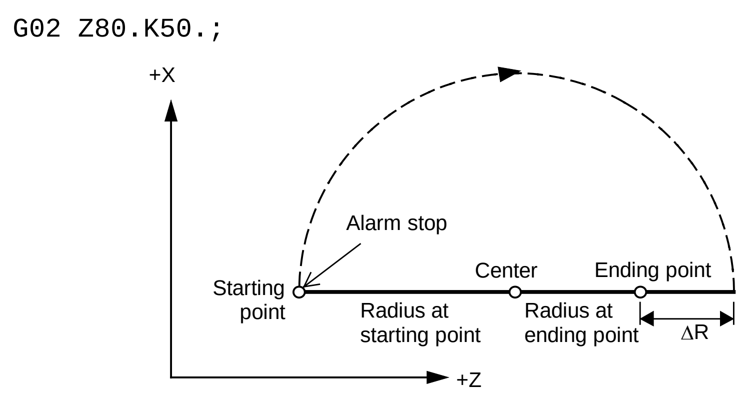


TEP020

```
G02 X120.0 Z70.0 I50.0 F200;     Absolute data setting
G02 U100.0 W–50.0 I50.0 F200;    Incremental data setting
```

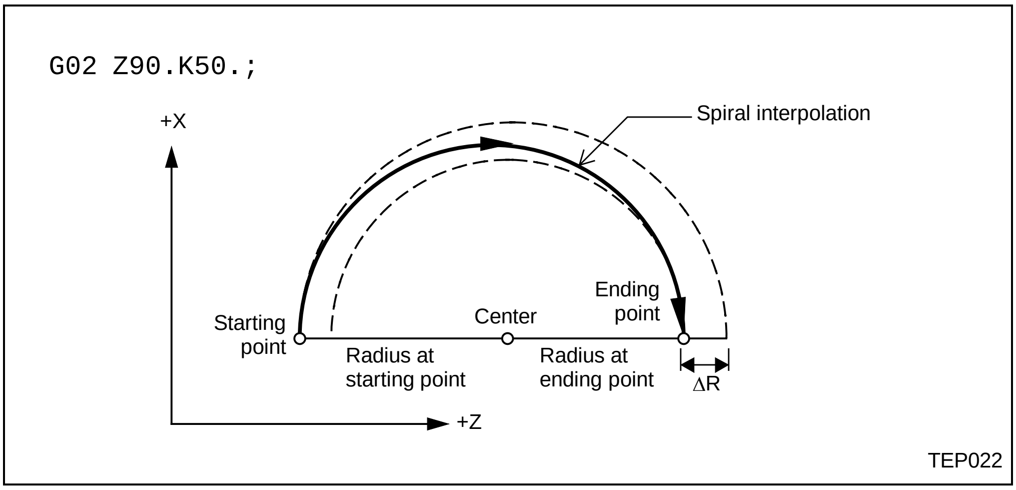
5. Notes on circular interpolation

1. Clockwise (G02) or Counterclockwise (G03) during circular interpolation refers to the rotational direction in the right-handed coordinate system when seen from the plus side toward the minus side of the coordinate axis perpendicular to the plane to be interpolated.
2. If the coordinates of the ending point are not set or if the starting and ending points are set at the same position, designating the center using address I, K or J will result in an arc of 360 degrees (true circle).
3. The following will result if the starting-point radius and the ending-point radius are not the same.
   - If error ΔR is larger than the parameter **F19** (tolerance for radial value difference at ending point), a program error (**817 INCORRECT ARC DATA**) will occur at the starting point of the arc.



TEP021

- If error ΔR is equal to or smaller than the parameter data, interpolation will take a spiral form heading for the programmed ending point of the arc.



The examples shown above assume that excessively large parameter data is given to facilitate your understanding.

6-5 Radius Designated Circular Interpolation Commands: G02, G03

1. Function and purpose

Circular interpolation can be performed by designating directly the arc radius R as well as using conventional arc center coordinates (I, K, J).

2. Programming format

G02 (G03) Xx/Uu Zz/Ww (Yy/Vv) Rr Ff ;

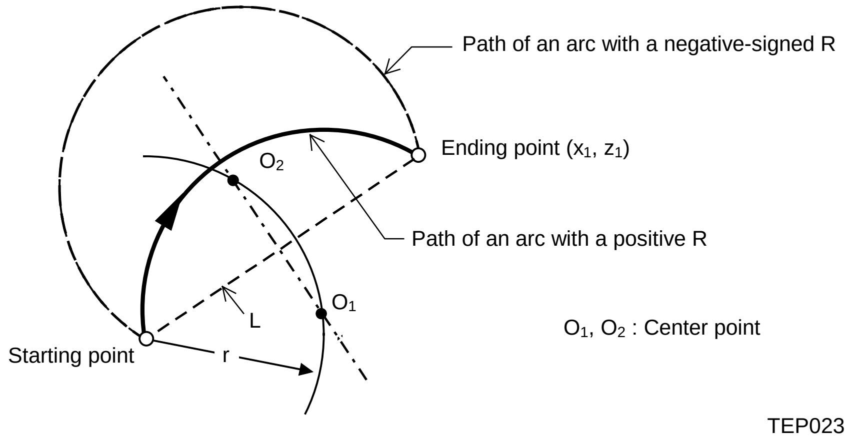
where	x/u:	X-axis coordinate of the ending point
	z/w:	Z-axis coordinate of the ending point
	y/v:	Y-axis coordinate of the ending point
	r:	Radius of the arc
	f:	Feed rate

3. Detailed description

The arc center is present on the mid-perpendicular to the segment which connects the starting point and the ending point. The crossing point of the mid-perpendicular and that circle of the designated radius r that has the center set at the starting point gives the center coordinates of the designated arc.

A semi-circle or smaller will be generated if R is a positive value.

An arc larger than the semi-circle will be generated if R is a negative value.



TEP023

To use the radius-designated arc interpolation commands, the following requirement must be met:

$$\frac{L}{2 \bullet r} \leq 1$$

where L denotes the length of the line from the starting point to the ending point.

If radius data and arc center data (I, J, K) are both set in the same block, the circular interpolation by radius designation will have priority in general.

For complete-circle interpolation (the ending point = the starting point), however, use center-designation method with addresses I, J and K, since the radius-specification command in this case will immediately be completed without any machine operation.

4. Sample programs

1. G02 Xx_1 Zz_1 Rr_1 Ff_1 ;
2. G02 Xx_1 Zz_1 Ii_1 Kk_1 Rr_1 Ff_1 ;

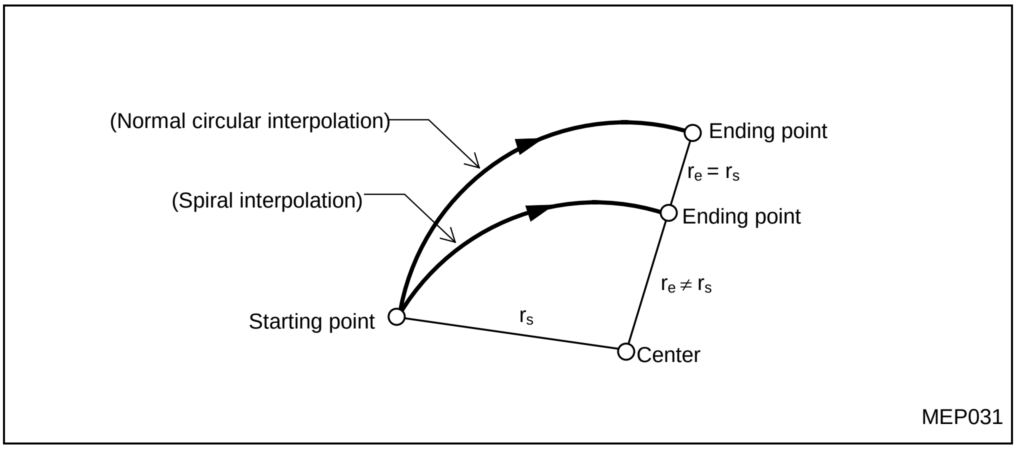
   (If radius data and center data (I, K, J) are set in the same block, circular interpolation by radius designation will have priority.)

Note: "I0", "K0" or "J0" can be omitted.

6-6 Spiral Interpolation: G2.1, G3.1 (Option)

1. Function and purpose

Commands G2.1 and G3.1 provide such an interpolation that the starting and ending points are connected smoothly for an arc command where the radii of the both points differ from each other.



2. Programming format

G17 G2.1 (or G3.1) $\underbrace{\text{Xp\_ Yp\_}}$ $\underbrace{\text{I\_ J\_}}$ (α_) F_ P_

Arc center coordinates
Arc ending point coordinates

G18 G2.1 (or G3.1) Zp_ Xp_ K_ I_ (α_) F_ P_

G19 G2.1 (or G3.1) Yp_ Zp_ J_ K_ (α_) F_ P_

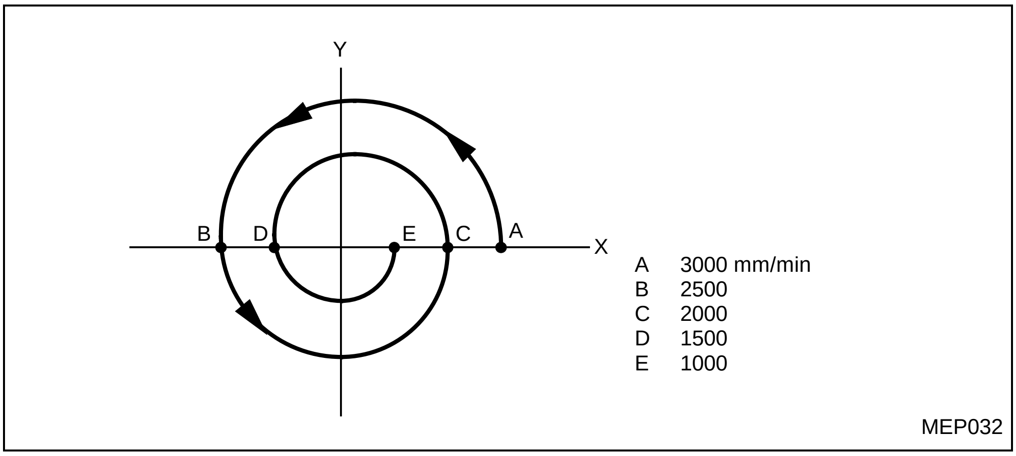
P : Number of pitches (revolutions) (P can be omitted if equal to 0.)
α : Any axis other than circular interpolation axes (For helical cutting only)
F : Rate of feed along the tool path

3. Detailed description

1. Circular movement directions of G2.1 and G3.1 correspond with those of G02 and G03, respectively.
2. Radius designation is not available for spiral interpolation. (The starting and ending points must lie on the same arc for a radius designation.)

   Note: When a radius is designated, this command will be regarded as a radius-designated circular interpolation.
3. Conical cutting or tapered threading can be done by changing the radii of the arc at its starting and ending points and designating a linear-interpolation axis at the same time.
4. Even for normal circular command G2 or G3, spiral interpolation will be performed if the difference between the radii of the starting point and the ending point is smaller than the setting of parameter **F19.**

Example: When the following program is executed, the feed rates for each of the points will be as shown in the diagram below.



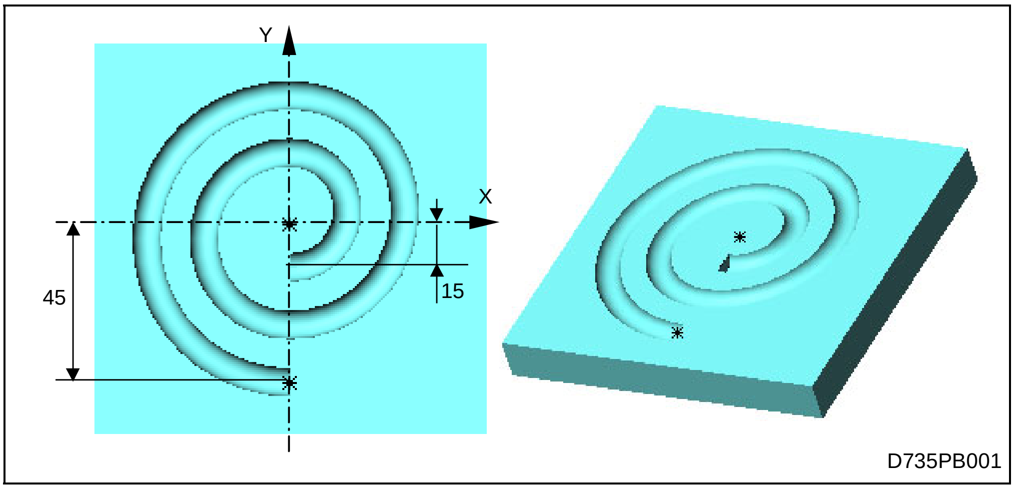

```
G28 X0 Y0
G00 Y–200.
G17 G3.1 X–100. Y0 I–150. J0 F3000 P2
M30
```

Return to Library

4. Sample programs

Example 1: Spiral cutting

Shown below is an example of programming for spiral contouring with incremental data input of the arc center (X = 0, Y = 45.0) and absolute data input of the arc ending point (X = 0, Y = –15.0).



`G28 W0`	Zero point return on the Z-axis
`G80 G40`	Fixed-cycle cancellation
`T001T000M06`	Tool change
`G54.1 P40`	Coordinate system setting
`G94 G00 X0 Y-45.0`	Approach in the XY-plane to the starting point (0, –45.0)
`G43 Z30.0 H01`	Positioning on the Z-axis to the initial point
`Z3.0`	
`S1500 M03`	Normal rotation of the spindle
`M50`	Air blast ON
`G01 Z-1.0 F150`	Infeed on the Z-axis
`G2.1 X0 Y-15.0 I0 J45.0 F450 P2`	Command for spiral interpolation with arc ending point = (0, –15.0), arc center = (0, 0)*, and pitch = 2. * I- and J-values refer to increments to the starting point.
`G00 Z3.0`	Return on the Z-axis
`M05 M09`	Spindle stop and Air blast OFF
`Z30.0`	
`M30`	End of machining

The rate of feed at the starting point is 450 mm/min, as specified in the block of G2.1, and the rate of feed at the ending point can be calculated as follows:

(Ending point's radius/Starting point's radius) × Command value of the rate of feed.

As the radius of the starting point = 45.0, that of the ending point = 15.0, and the command rate of feed (F) = 450, the rate of feed results in

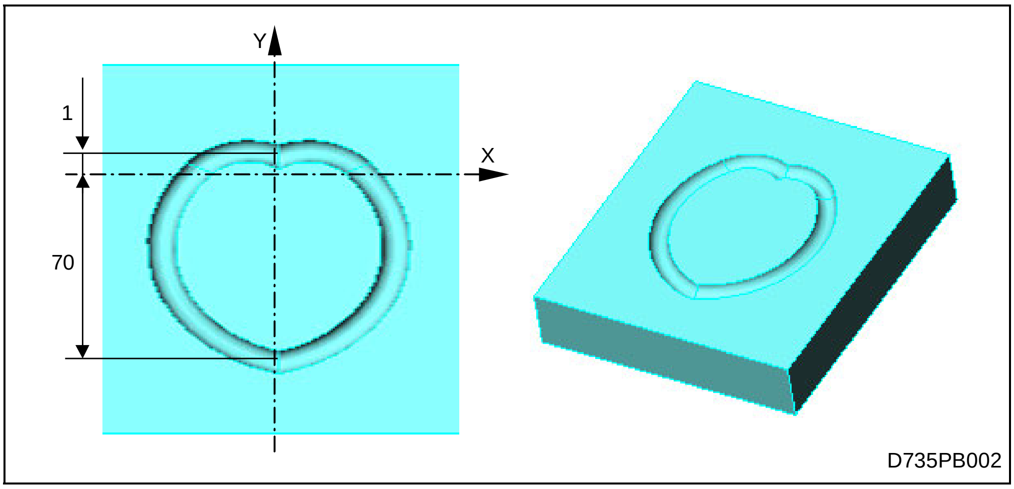
(15.0/45.0) × 450 = 150 mm/min

at the ending point.

Note 1: Take care not to use radius designation (argument R) for spiral interpolation; otherwise a normal circular interpolation (by G02 or G03) will be executed.

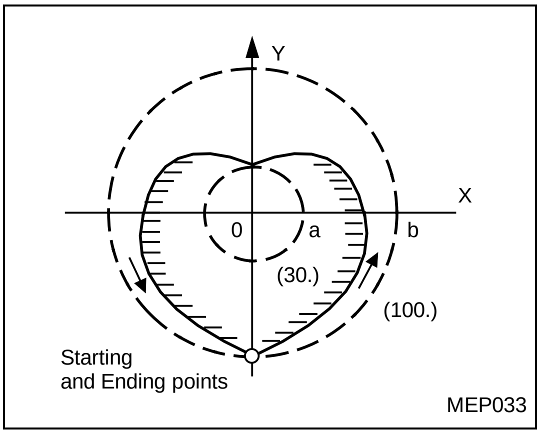
Note 2: It is not possible to give the command for a spiral interpolation the starting and ending points of which should have different centers specified.

Example 2: Heart-shaped cam (by absolute data input)



`G28 W0`	Zero point return on the Z-axis
`G80 G40`	Fixed-cycle cancellation
`T001T000M06`	Tool change
`G54.1 P40`	Coordinate system setting
`G94 G00 X0 Y-70.0`	Approach in the XY-plane to the starting point (0, –70.0)
`G43 Z30.0 H01`	Positioning on the Z-axis to the initial point
`S1500 M03`	Normal rotation of the spindle
`Z3.0`	
`M50`	Air blast ON
`G01 Z-1.0 F150`	Infeed on the Z-axis
`G2.1 X0 Y1.0 I0 J70.0 F450`	Command for the left-hand half curve
`X0 Y-70.0 I 0 J-1.0`	Command for the right-hand half curve
`G00 Z3.0`	Return on the Z-axis
`M05 M09`	Spindle stop and Air blast OFF
`Z30.0`	
`M30`	End of machining

Example 3: Heart-shaped cam (by incremental data input)



The difference (b–a) between the radii of the starting point and ending point denotes a displacement for heart shape.
Use two blocks for programming separately the right-half and the left-half shape.

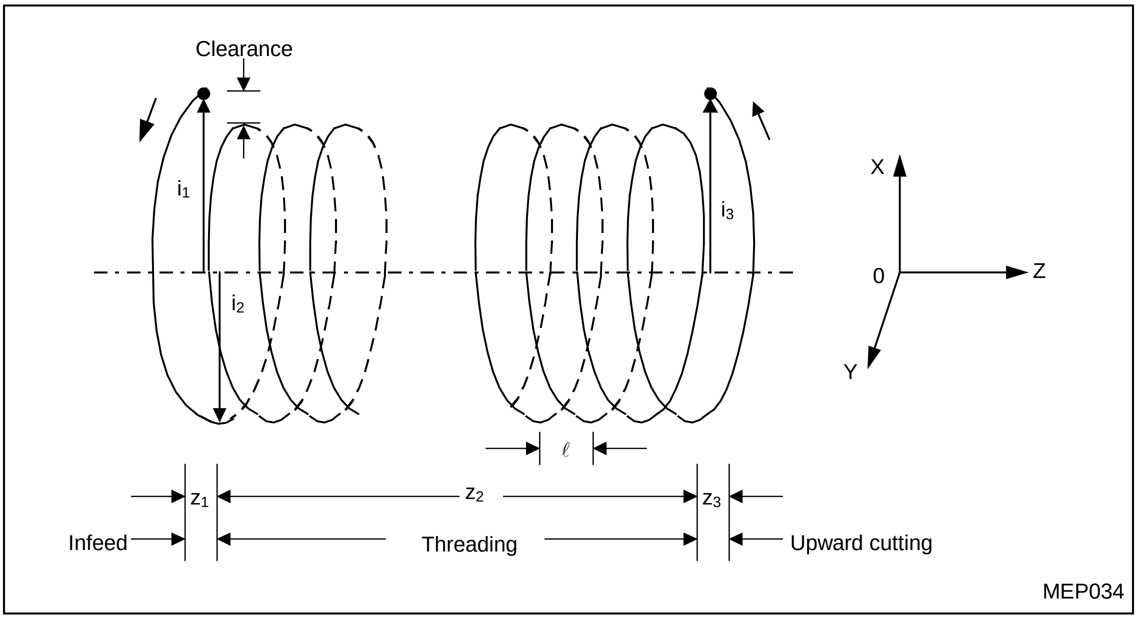
<u>A sample program in incremental data input:</u>

```
G3.1 Y130. J100. F1000.............. (Right half)
      a+b    b
G3.1 Y-130. J-30....................... (Left half)
      -a-b    -a
```

a = 30. (Minimum arc radius)
b = 100. (Maximum arc radius)
a + b = 130. (Ending-point coordinate of the right half-circle)
–a – b = –130. (Ending-point coordinate of the left half-circle)

Example 4: Large-size threading

To perform large-size threading, use three heilical-interpolation blocks for programming separately infeed section, threading section and upward-cutting section. Spiral interpolation is required to designate the amounts of diameter clearance for both the infeed block and the upward-cutting block. (The starting and ending points are shifted through the designated clearance amounts from the circumference of threading section.)

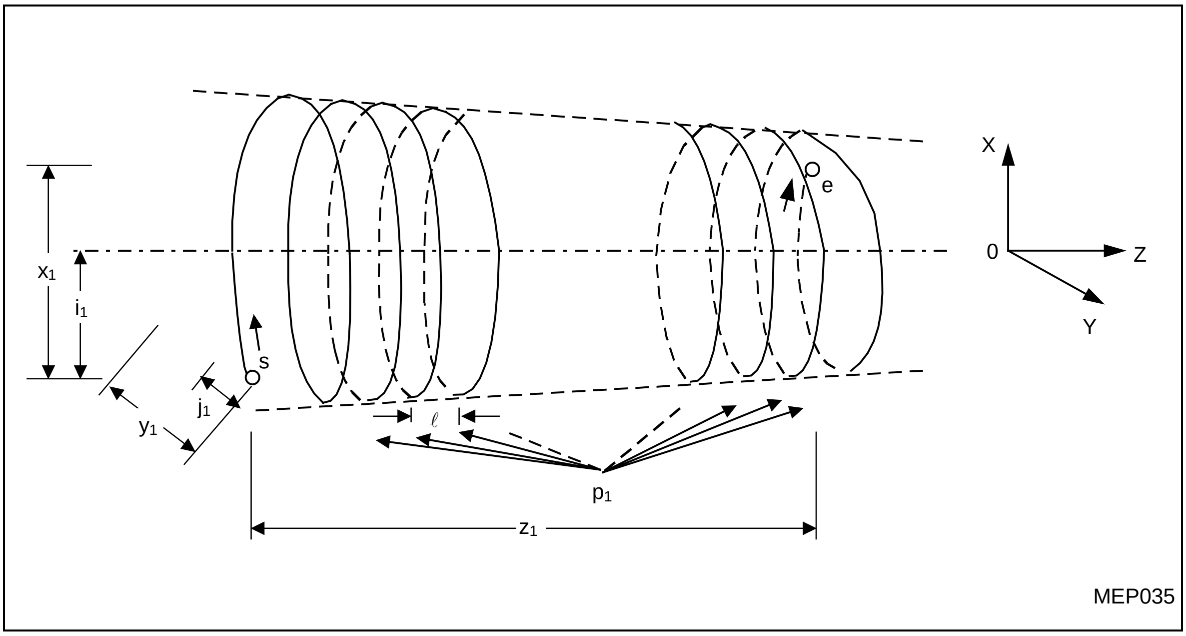


G3.1 X–i_1–i_2 Y0 Zz_1 I–i_1 J0 Ff_1	(Infeed block, half-circle)
G03 X0 Y0 Zz_2 Ii_2 J0 Pp_2	(Threading block, complete circle)
G3.1 Xi_2+i_3 Y0 Zz_3 Ii_2 J0	(Upward-cutting block, half-circle)

* The number of pitches, p_2, in the threading block is given by dividing the stroke z_2 by the pitch ℓ. Note that the value p_2 must be an integer.

Example 5: Tapered threading

As shown in the figure below, tapered helical cutting that begins at any angle can be performed.



Data with addresses X, Y and Z must be the increments x_1, y_1 and z_1 respectively, from the starting point s to the ending point e; data of I and J must be the increments i_1 and j_1 respectively, from the starting point s to the circular center, and data of P must be equal to the number of pitches p_1.

G3.1 Xx_1 Yy_1 Zz_1 Ii_1 Jj_1 Pp_1 Ff_1

The amount of taper t and the pitch ℓ are calculated as follows:

$$t = \frac{2(re - rs)}{x_1}$$

where $rs = \sqrt{i_1^2 + j_1^2}$, $re = \sqrt{(x_1 - i_1)^2 + (y_1 - j_1)^2}$;

$$\ell = \frac{z_1}{(2\pi \bullet \pi_1 + \theta) / 2\pi}$$

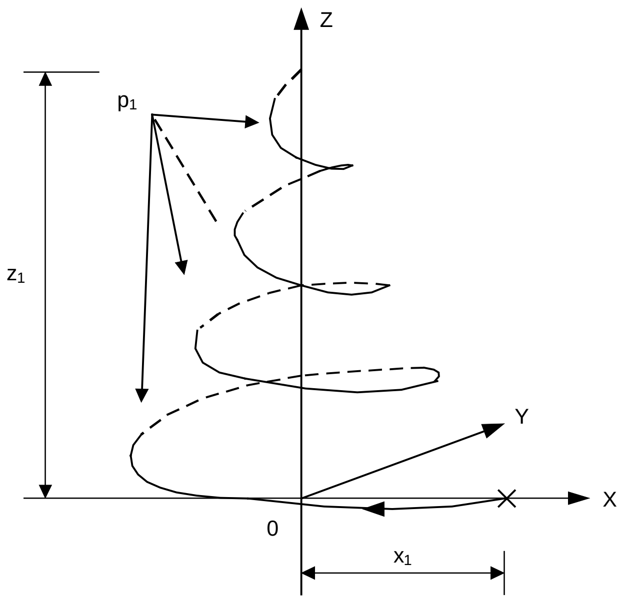
where $\theta = \theta e - \theta s = \tan^{-1} \frac{j_1 - y_1}{i_1 - x_1} - \tan^{-1} \frac{-j_1}{-i_1}$

where rs and re denote the radii at the starting point and the ending point respectively, and qs and qe denote the angles at the starting point and the ending point respectively.

Return to Library

Example 6: Conical cutting

Conical cutting is an application of tapered threading, and have its starting or ending point on the center line. Tapering results from gradually increasing or decreasing the arc diameter. The pitch is determined by z_1/p_1.




```
G2.1 X–x1 Y0 Zz1 I–x1 Pp1 Ff1
```

x_1 : Radius of the base

z_1 : Height

p_1 : Number of pitches

f_1 : Feed rate

Note: Use the **TRACE** display to check the tool path during spiral interpolation.

Return to Library

6-7 Plane Selection Commands: G17, G18, G19

6-7-1 Outline

1. Function and purpose

Commands G17, G18 and G19 are used to select a plane on which arc interpolation, tool nose radius compensation, etc. are to be done.
Registering the three fundamental axes as parameters allows you to select a plane generated by any two non-parallel axes.

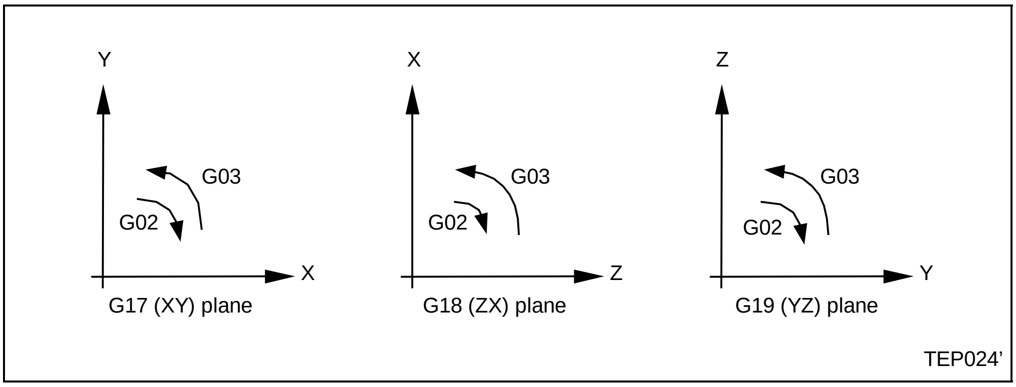
The available planes are the following three types:

- Plane for circular interpolation
- Plane for tool nose radius compensation
- Plane for polar coordinate interpolation

2. Programming format

G17; (X-Y plane selection)
G18; (Z-X plane selection)
G19; (Y-Z plane selection)

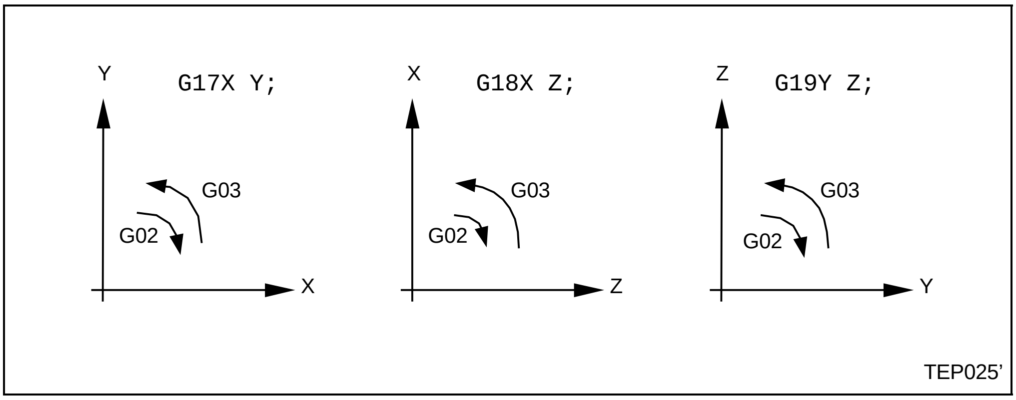
X, Y, and Z denote respective coordinate axes or their corresponding parellel axes.



6-7-2 Plane selection methods

Plane selection by parameter setting is explained in this section.

1. Which of the fundamental axes or their parallel axes are to form the plane you want to select is determined by the type of plane selection command (G17, G18 or G19) and the axis address specified in the same block.



2. Automatic plane selection does not occur for blocks that do not have an issued plane-selection command (G17, G18 or G19)

   G18 X_ Z_; Z-X plane
   Y_ Z_; Z-X plane (No plane change)

3. If axis addresses are not set for blocks having an issued plane-selection command (G17, G18 or G19), the fundamental three axes will be regarded as set.

   G18_; (Z-X plane = G18 XZ ;)

Note 1: Upon power on or resetting, G18 plane is selected.

Note 2: In turning mode, G17 or G19 plane selection is impossible and in milling mode, G18 plane selection respectively.
If such selection were attempted, alarm would be caused.

Note 3: The G-codes for plane selection (G17, G18 or G19) should be commanded in a block independently. If such a G-code is commanded in a block containing the axis move command, a movement independent from the selected plane can be caused.

6-8 Polar Coordinate Interpolation ON/OFF: G12.1/G13.1

1. Function and purpose

It is available for face helical grooving or cam shaft grinding on the lathe.
It is a function to convert a command programmed by the rectangular coordinate system into the linear axis movement (tool movement) and the rotational axis movement (workpiece rotation) to give contouring control.

2. Programming format

The polar coordinate interpolation is commanded by the following G-codes (group 26).

G12.1: Polar coordinate interpolation mode (Mode by which the polar coordinate is interpolated)

G13.1: Polar coordinate interpolation cancel mode (Mode by which the polar coordinate is not interpolated)

These G-codes should be commanded in an independent block.

3. Detailed description

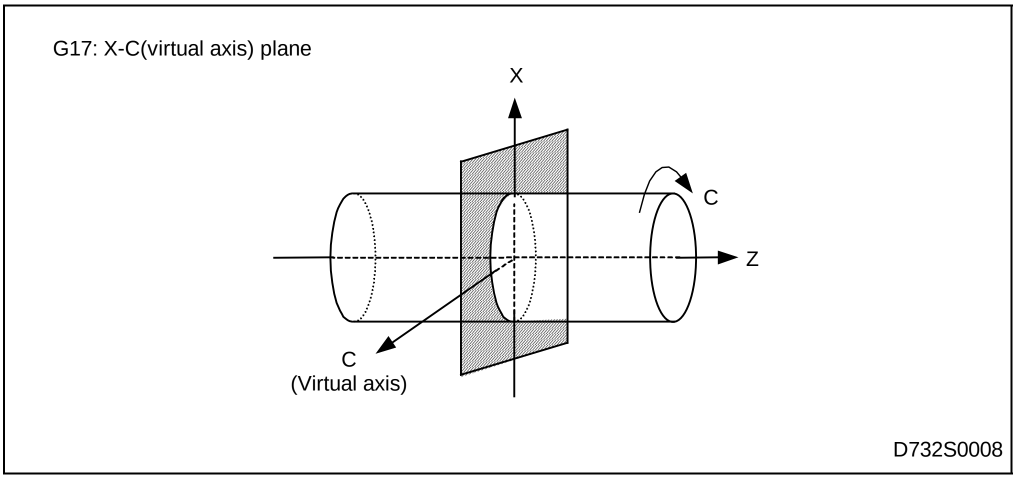
1. When turning on the power and resetting, the polar coordinate interpolation cancel mode (G13.1) is provided. Commanding G12.1 provides a plane selected by G17.

2. The polar coordinate interpolation uses the zero point of workpiece coordinate system as that of the coordinate system. A plane (hereinafter referred to as “polar coordinate interpolation plane”) is selected using the linear axis as the 1st axis of the plane and the virtual axis perpendicular to the linear axis as the 2nd axis of the plane. The polar coordinate interpolation is given on that plane.

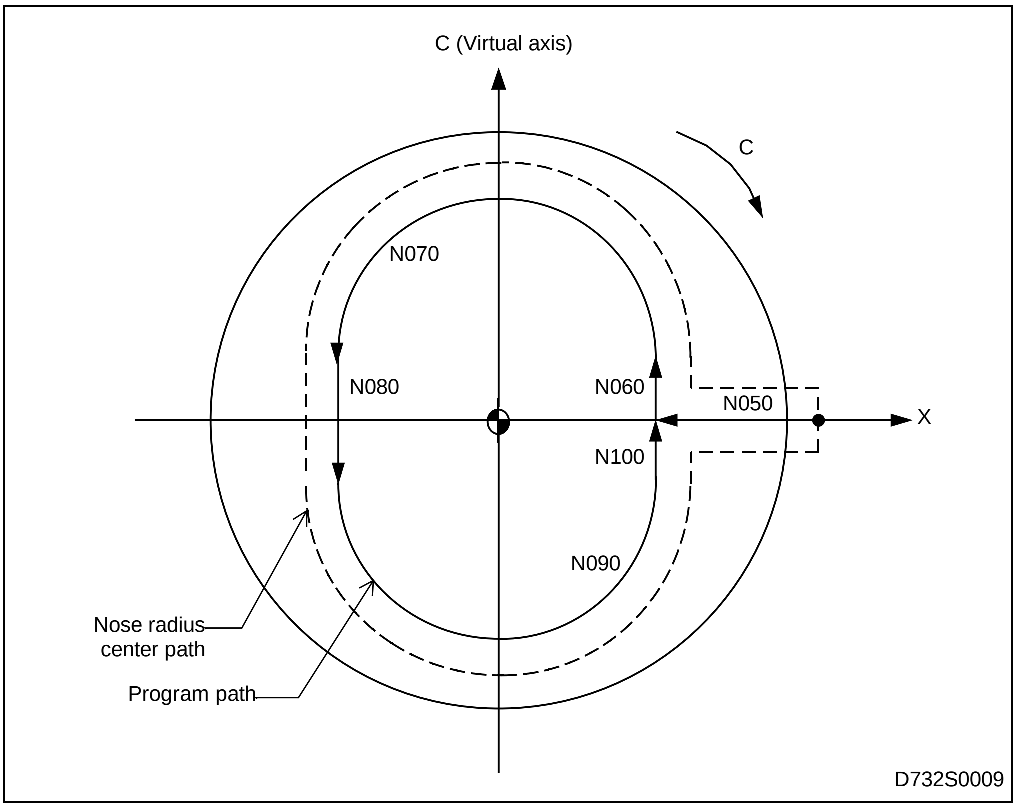
3. The program during polar coordinate interpolation mode is commanded by the rectangular coordinate value on the polar coordinate interpolation plane. The axis address of the rotational axis (C) is used for that of the command of the 2nd axis of the plane (virtual axis).

   A command is given in mm or inch as with the 1st axis of the plane (command by the axis address of the linear axis), and not in degrees. And whether designation is given by the diameter or by the radius is not determined by the 1st axis of the plane, but the designation is the same as the rotational axis.

4. Absolute command and incremental command for the linear interpolation (G01) and the circular interpolation (G02, G03) can be commanded during the polar coordinate interpolation mode.
   The nose radius compensation can also be made for the program command, and the polar coordinate interpolation is given to the path after the nose radius compensation. However, the polar coordinate interpolation mode (G12.1, G13.1) cannot be changed during the nose radius compensation mode (G41, G42). G12.1 and G13.1 must be commanded in G40 mode (Nose radius compensation cancel mode).
5. The feed rate is commanded using tangential speed (relative speed of the workpiece and a tool) on the polar coordinate interpolation plane (rectangular coordinate system) as F (mm/min or inch/min is used for a unit of F).
6. The coordinate value of the virtual axis when G12.1 is commanded provides “0”. That is, the polar coordinate interpolation is started taking the position where G12.1 is commanded as the angle = 0.



4. Sample programs




```
      ⋮
N001  G00 G97 G98;
N004  G28 U0 W0;
N008  M200;
N010  T001T000M06;
N020  G00 X100.0 Z10.0 C0.0;       Positioning to the start point
N030  G12.1;                       Polar coordinate interpolation start
N040  G42;
N050  G01 X50.0 F500;
N060      C10.0;
N070  G03 X-50.0 C10.0 I-25.0;
N080  G01 C-10.0;                  Shape program
N090  G03 X50.0 C-10.0 R25.0;      (Program with rectangular coordinate values
N100  G01 C0.0;                    on X-C plane)
N110  G00 X100.0;
N120  G40;
N130  G13.1;                       Polar coordinate interpolation cancel
N140  M202;
      ⋮
```

5. Notes

1. Before G12.1 is commanded, a workpiece coordinate system must be set using the center of rotational axis as the zero point of the coordinate system. The coordinate system must not be changed during G12.1 mode.
2. The plane before G12.1 is commanded (plane selected by G17, G18 or G19) is temporarily cancelled, and it is restored when G13.1 (polar coordinate interpolation cancel) is commanded. The polar coordinate interpolation mode is cancelled in resetting, and the G18 plane is provided.
3. The method of commanding the circular radius (which address of I, J and K is used) when the circular interpolation (G02, G03) is given on the polar coordinate interpolation plane depends on which axis of the basic coordinate system the 1st axis of the plane (linear axis) corresponds to.
   - Command is given by I and J taking the linear axis as the X-axis of X_p-Y_p plane.
   - Command is given by J and K taking the linear axis as the Y-axis of Y_p-Z_p plane.
   - Command is given by K and I taking the linear aixs as the Z-axis of Z_p-X_p plane.

   The circular radius can also be designated by R command.
4. G-codes capable of command during G12.1 mode are G04, G65, G66, G67, G00, G01, G02, G03, G98, G99, G40, G41 and G42.
5. Move command of an axis other than those on the selected plane during G12.1 mode is executed independently of the polar coordinate interpolation.
6. Tool offset must be commanded in the polar coordinate interpolation cancel mode before G12.1 is commanded. It cannot be commanded during the polar coordinate interpolation mode. Offset amount must not be changed during the polar coordinate interpolation mode.
7. Current position display during G12.1 mode
   Every current position during the polar coordinate interpolation mode is displayed with an actual coordinate value. However, only “residue moving distance” (**REMAIN**) is displayed with the residue moving distance on the polar coordinate command plane.
8. Program restart cannot be made for a block during G12.1 mode.

6-9 Virtual-Axis Interpolation: G07

1. Function and purpose

Specify with G07 code one of the two circular-interpolation axes for helical or spiral interpolation with synchronous linear interpolation as a virtual axis (a pulse-distributed axis without actual movement), and an interpolation on the plane defined by the remaining circular axis and the linear axis can be obtained along the sine curve which corresponds with the side view of the circular interpolation with synchronous linear interpolation.

2. Programming format

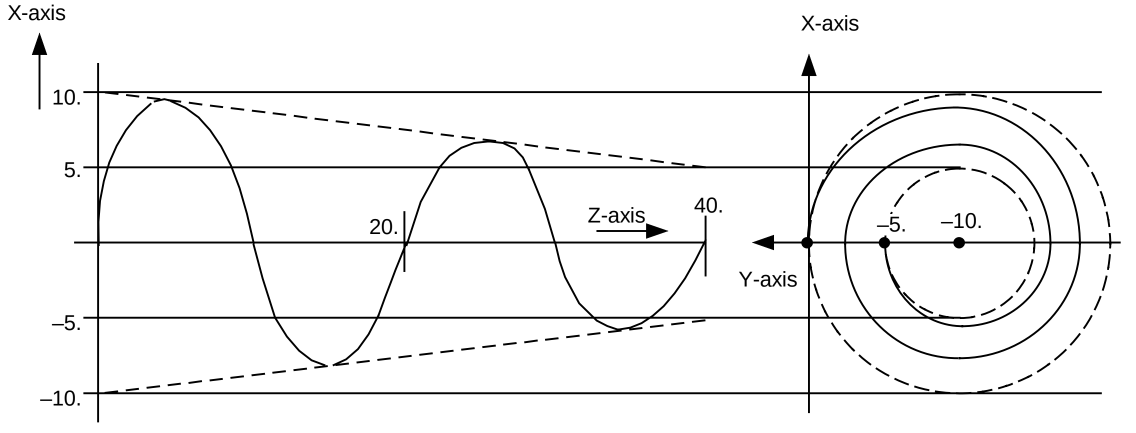
G07 α0	To set a virtual axis
⋮	} To interpolate with the virtual axis
G07 α1	To cancel the virtual axis

3. Detailed description

1. Only helical or spiral interpolation can be used for the virtual-axis interpolation.
2. In the program section from G07α0 to G07α1, the "alpha" axis is processed as a virtual axis. If, therefore, the alpha axis is included independently in this section, the machine will remain in dwell status until pulse distribution to the virtual axis is completed.
3. The virtual axis is valid only for automatic operation; it is invalid for manual operation.
4. Protective functions, such as interlock, stored stroke limit, etc., are valid even for the virtual axis.
5. Handle interruption is also valid for the virtual axis. That is, the virtual axis can be shifted through the amount of handle interruption.

4. Sample program

`G07 Y0`	Sets the Y-axis as a virtual axis.
`G17G2.1X0Y–5.I0J–10.Z40.P2F50`	Sine interpolation on X-Z plane
`G07 Y1`	Resets the Y-axis to an actual axis.



MEP037

Return to Library

6-10 Spline Interpolation: G06.1 (Option)

1. Function and purpose

The spline interpolation automatically creates a curve that smoothly traces specified points, and thus enables a high-speed and high-accuracy machining for free shapes along smoothly curved tool path.

2. Programming format

G06.1 Xx_1 Yy_1

3. Detailed description

A. Setting and cancellation of spline interpolation mode

The spline interpolation mode is set by the preparatory function G06.1, and cancelled by another Group 01 command (G00, G01, G02 or G03).

Example 1:

```
N100 G00    X_Y_    P1
N200 G06.1  X_Y_    P2
N201        X_Y_    P3
N202        X_Y_    P4
N203        X_Y_    P5
    :        :
N290        X_Y_    Pn
N300 G01    X_Y_    Pn+1
```

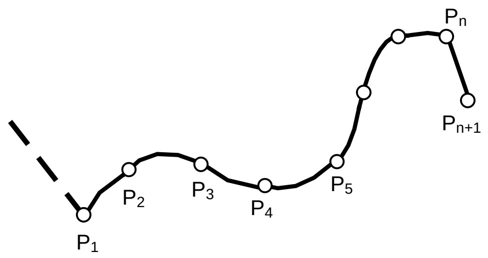


Fig. 6-1 Interpolated line by spline interpolation

In the above example, the spline interpolation is activated at N200 (block for movement from P_1 to P_2) and it is cancelled at N300. Therefore, a spline curve is created for a group of ending points from P_1 to P_n, and interpolation is applied along the created curve.

For creating a spline interpolation curve, it is generally required to specify two or more blocks (at least three points to be traced) in the mode. If the spline interpolation mode is set just for one block, the path to the ending point of the block is interpolated in a straight line.

Example 2:

```
N100  G01    X_Y_    P1
N200  G06.1  X_Y_    P2
N300  G01    X_Y_    P3
```

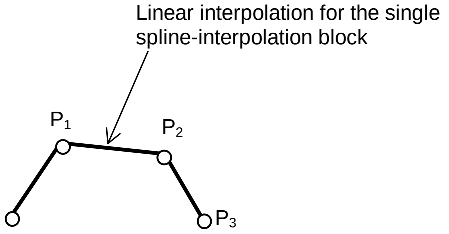


Fig. 6-2 Spline interpolation applied to a single block

B. Division of spline curve in spline-interpolation mode

The spline interpolation mode generally creates a continuous curve that smoothly connects all specified points from the beginning of the mode to the end of it. However, the spline curve is divided into two discontinuous curves as often as one of the following conditions is satisfied:

- When the angle between linear movement lines of two neighboring blocks is beyond the spline-cancel angle,
- When the movement distance of a block exceeds the spline-cancel distance, or
- When there is a block without any movement command in the spline-interpolation mode.

1. When the relative angle of two neighboring blocks is beyond the spline-cancel angle

Spline-cancel angle ········· Parameter **F101**

As to the sequence of points P_1, P_2, P_3, ··· P_n in a spline interpolation mode, when the angle θ_i made by two continuous vectors $\overrightarrow{P_{i-1}P_i}$ and $\overrightarrow{P_iP_{i+1}}$ is larger than **F101**, the point P_i is regarded as a corner. In that event, the point group is divided into two sections of P_1 to P_i and P_i to P_n at P_i, and spline curve is individually created for each section.
When the spline-cancel angle is not set (**F101** = 0), this dividing function is not available.

Example 1: **F101** = 80 deg

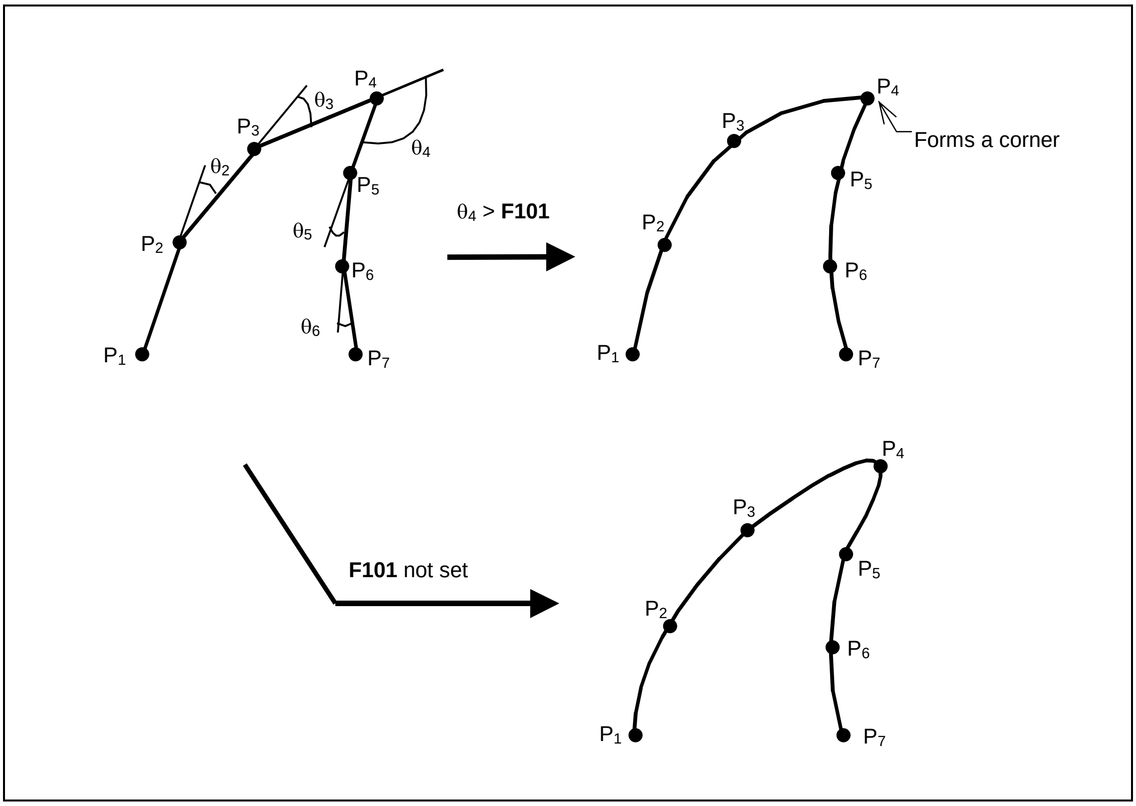


Fig. 6-3 Spline cancel depending on angle

When there are more than one point where θ_i > **F101**, such points are treated as corners to divide the point group and multiple spline curves are created for respective sections.

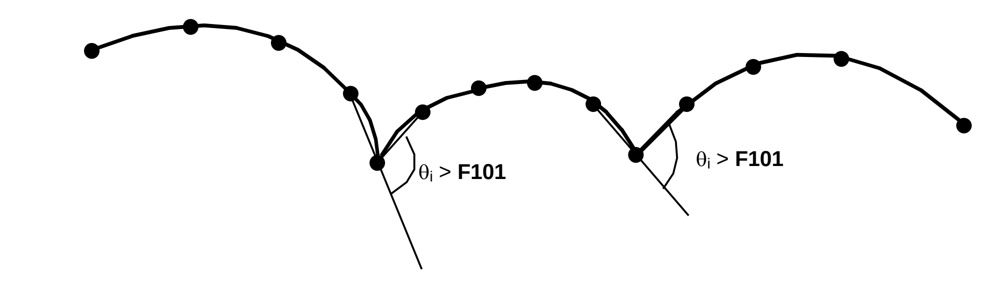


Fig. 6-4 Multiple-cornered spline curve depending on angle

When any two corner points (where θ_i > **F101**) successively exist, the block for the second point is automatically set under control of linear interpolation. Therefore, it can be omitted to specify G01 code in each intermediate block of pick feed, for example, during 2.5-dimensional machining, which considerably simplifies the programming.

Example 2: **F101** < 90 (deg)

In the following program (shown in Fig. 6-5), the angle of the Y-directional pick feed to the X-Z plane (of spline interpolation) is always 90°. If **F101** is set slightly smaller than 90°, spline interpolation is automatically cancelled in the pick-feed blocks (N310, N410, ···), which are then linearly interpolated each time. If no value is set for **F101**, it is required to specify G-codes parenthesized in the program below to change the mode of interpolation.

```
N100   G00      X_Y_Z_    P1
N200   G06.1    X_Z_      P2
N210            X_Z_      P3
  ⋮               ⋮
N300            X_Z_      Pi
N310   (G01)    Y_        Pi+1
N320   (G06.1)  X_Z_      Pi+2
  ⋮               ⋮
N400            X_Z_      Pj
N410   (G01)    Y_        Pj+1
N420   (G06.1)  X_Z_      Pj+2
  ⋮               ⋮
N700            X_Z_      Pn
N710   G01                ⋮
```

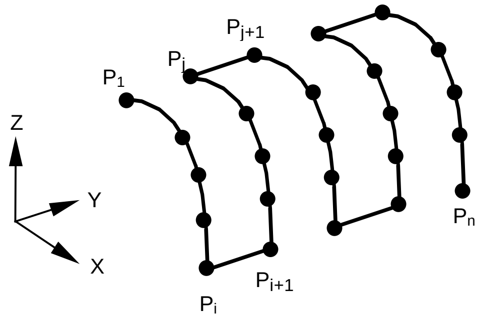


Fig. 6-5 Linear interpolation for pick feed in spline-interpolation mode

Return to Library

2. When the movement distance of a block exceeds the spline-cancel distance

Spline-cancel distance ········· Parameter **F100**

As to the sequence of points P_1, P_2, P_3, ··· P_n in a spline interpolation mode, when the length $\overline{P_iP_{i+1}}$ of the vector $\overrightarrow{P_iP_{i+1}}$ is longer than **F100**, the block for point P_{i+1} is automatically set under control of linear interpolation, while the preceding and succeeding sections P_1 to P_i and P_{i+1} to P_n are individually interpolated in spline curves.
In this case, the inclination of the tangent vector at P_i (at the end of spline P_1 to P_i) and the inclination of the tangent vector at P_{i+1} (at the beginning of spline P_{i+1} to P_n) do not correspond to that of the line segment $\overline{P_iP_{i+1}}$ in general.

When the spline-cancel distance is not set (**F100** = 0), this dividing function is not available.

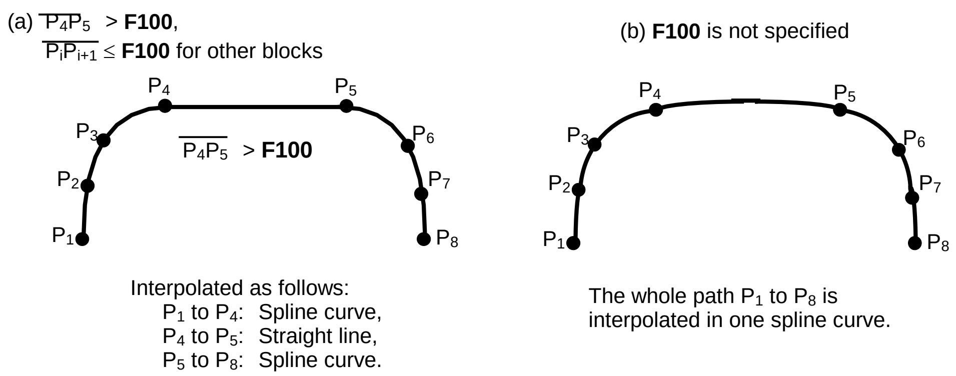


Fig. 6-6 Spline cancel depending on movement distance of a block

When there are more than one block where $\overline{P_iP_{i+1}}$ > **F100**, all those blocks will individually undergo the linear interpolation.

3. When there is a block without any movement command in the spline-interpolation mode
Any block without movement command temporarily cancels the spline interpolation, and the sections before and after such a block will independently be spline-interpolated.

```
N100  G01    X_Y_
N110  G06.1  X_Y_     P1
N120         X_Y_     P2
  ⋮             ⋮
N300         X_Y_     P5
N310         X0       P5 (Not moved)
N320         X_Y_     P6
  ⋮             ⋮
N500         X_Y_     P8
N510  G01    X_Y_
```

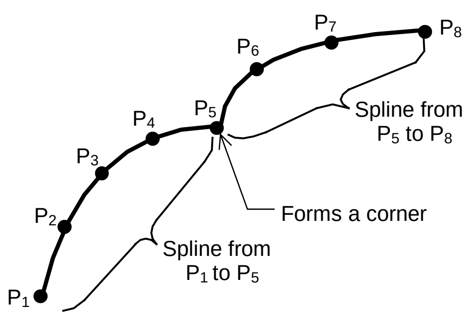


Fig. 6-7 Spline cancel by a block without movement command

C. Fine spline function (curved shape correction)

The fine spline function works with spline interpolation and automatically corrects the shape of a spline curve, as required, to make the path of the curve smoother.
More specifically, the fine spline function works in the following two cases:

- The case that the curve errors in blocks are significant
- The case that an unusually short block exists (automatic correction in this case is referred to as fairing.)

Automatic correction in the above cases is explained below.

1. Automatic correction for significant curve errors in blocks

When the curve data in CAD undergoes micro-segmentation with CAM, approximation using a polygonal line is usually executed with a curve tolerance (chord error) of about 10 microns. At this time, if any inflection points are included in the curve, the micro-segment block including the inflection points may increase in length (see $\overline{P_3 P_4}$. in the figure below)
Also, if the length of this block becomes unbalanced against those of the immediately preceding and succeeding blocks, the spline curve in this zone may have a significant error with respect to the original curve.

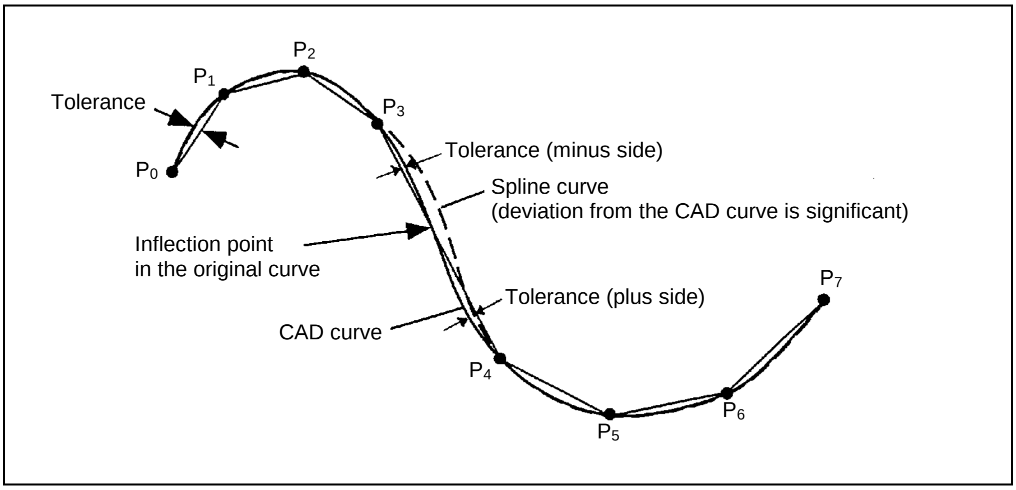


Fig. 6-8 Spline curve having a significant chord error (inflection points present)

This function detects the sections whose chord errors in the curve due to the presence of inflection points become significant, and corrects the shape of the spline curve in that zone automatically so that the chord errors in the curve fall within the data range of the specified parameter.

Curve error 1 ········· Parameter **F102**

If a block in the spline interpolation mode is judged to have inflection points in the spline curve and the maximum chord error of the spline curve from the segment is greater than the value of **F102**, the shape of that spline curve will be corrected for a maximum chord error not exceeding the value of **F102**.

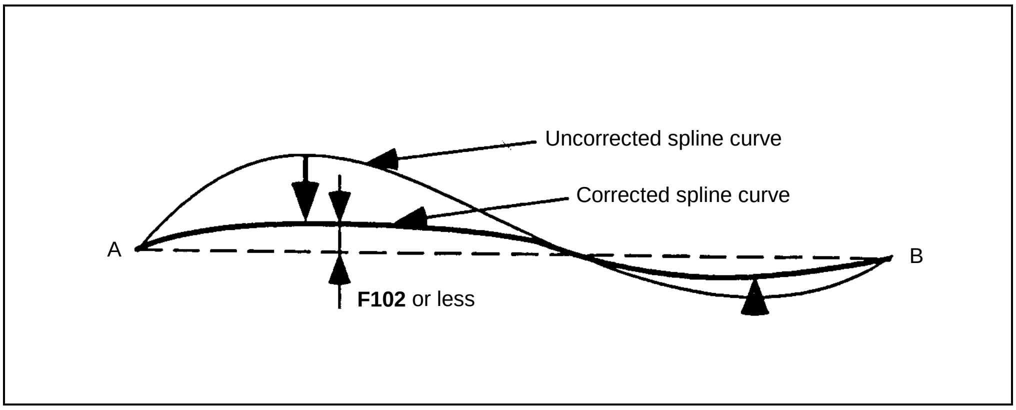


Fig. 6-9 Shape correction 1 for spline curve

The shape of a curve can also be corrected if the chord error in the spline curve increases due to an imbalance in the lengths of adjoining blocks occurs for any reasons other than the presence of inflection points or for other reasons.

Curve error 2 ········· Parameter **F104**

If a blocks in the spline interpolation mode is judged to have no inflection points in the spline curve and the maximum chord error in the spline curve and block is greater than the value of **F104**, the shape of that spline curve will be corrected for a maximum chord error not exceeding the value of **F104**.

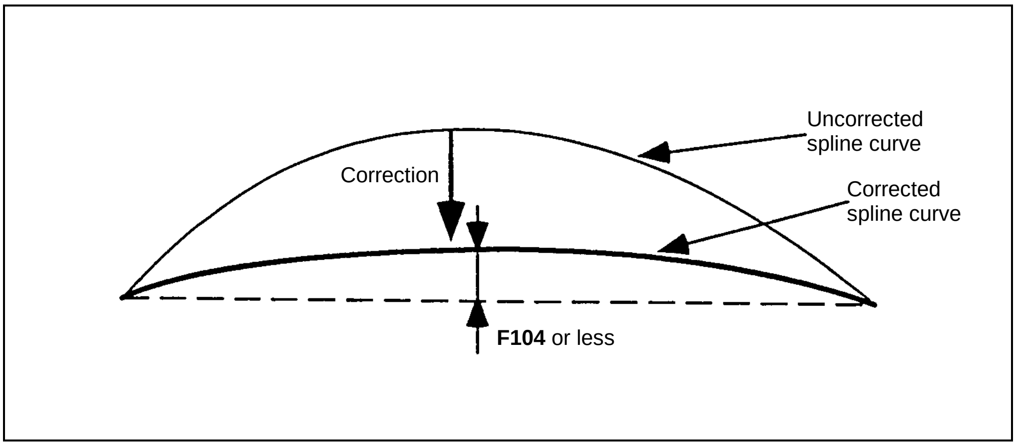


Fig. 6-10 Spline curve having a significant chord error (no inflection points)

Remark 1: In all types of spline curve correction, the curve correction function works only for the corresponding block. Therefore, the tangential vectors at the boundaries with the immediately preceding and succeeding blocks become discontinuous.

Remark 2: If parameter **F102** is set to 0, all blocks regarded as including inflection points will become linear. If parameter **F104** is set to 0, all blocks regarded as including no inflection points will become linear.

Remark 3: Curved-shape correction based on parameter **F102** or **F104** usually becomes necessary when adjoining blocks are unbalanced in length. If the ratio of the adjoining block lengths is very large, however, spline interpolation may be temporarily cancelled between the blocks prior to evaluation of the chord error.

2. Automatic correction of the spline curve in an unusually short block (Fairing)
   When CAD data is developed into micro-segments by CAM, a very small block may be created in the middle of the program because of internal calculation errors. Such a block is often created during creation of a tool diameter offset program which requires convergence calculation, in particular. Since this unusually small block usually occurs at almost right angles to the direction of the spline curve, this curve tends not to become smooth.

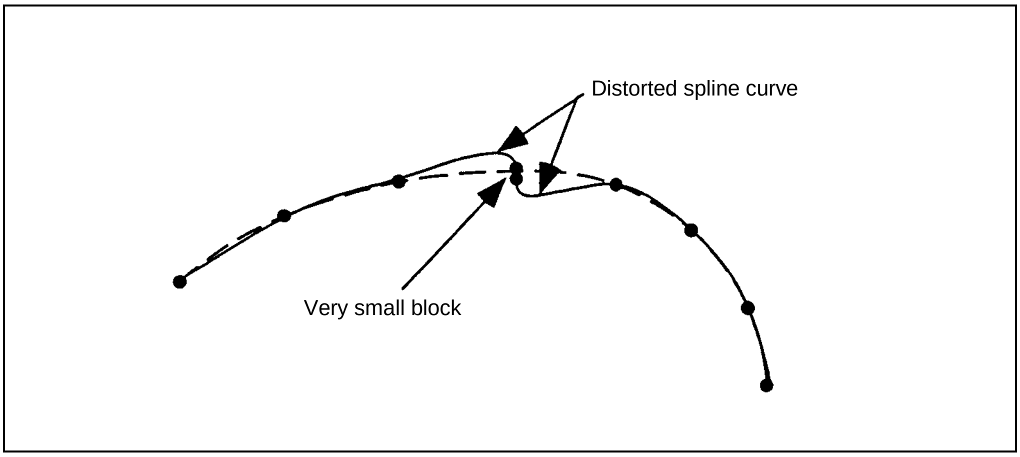


Fig. 6-11 Distortion of a spline curve due to the effects of a very small block

If it detects such an extremely small block during spline interpolation, the shape correction function will remove that block and then connect the preceding and succeeding blocks directly (this is referred to as fairing) to create a smooth spline curve free from distortion.

Block fairing length ········· Parameter **F103**

Assume that the length of the i-th block in spline interpolation mode is taken as li and that the following expressions hold:

$l_{i-1} >$ **F103** $\times 2$
$l_i \leq$ **F103**
$l_{i+1} >$ **F103** $\times 2$

In the above case, the ending point of the (i–1)-th block and the starting point of the i+1 block are moved to the mid-point of the ith block and as a result, the ith block is deleted. Spline interpolation is executed for the sequence of points that has thus been corrected.

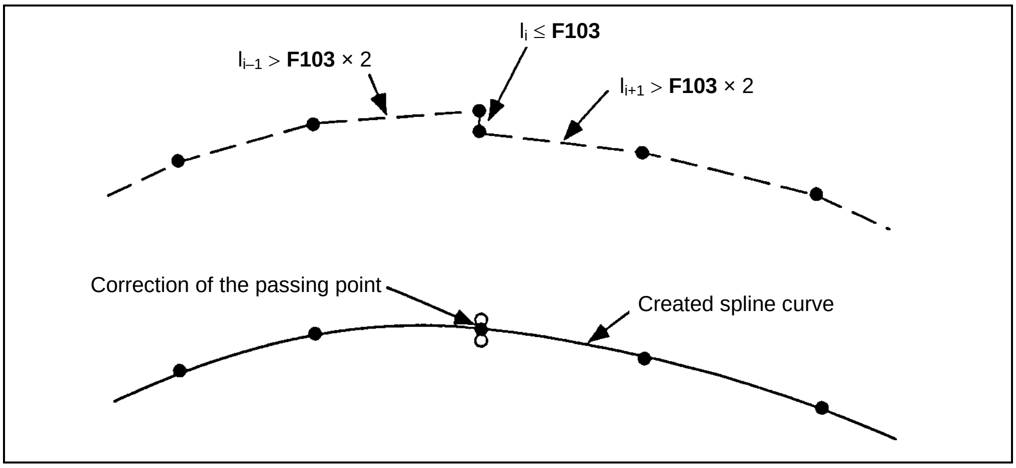


Fig. 6-12 Correction of spline curve passing points by fairing

Assume that the first block in spline interpolation mode is very small and that the following expressions hold:

$l_1 \leq$ **F103**

$l_2 >$ **F103** $\times 2$

In the above case, the starting point of the second block is changed to that of the first block and as a result, the first block is deleted.

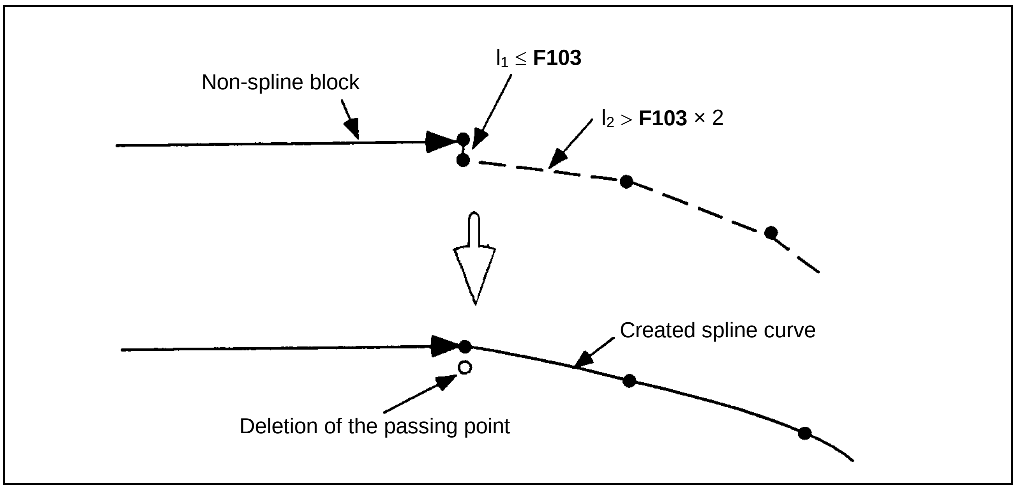


Fig. 6-13   Fairing at the starting point of a spline curve

Assume that the last block in spline interpolation mode is very small and that the following expressions hold:

$l_{n-1} >$ **F103** $\times 2$

$l_n \leq$ **F103**

In the above case, the ending point of the (n–1)-th block is changed to that of the nth block and as a result, the nth block is deleted.

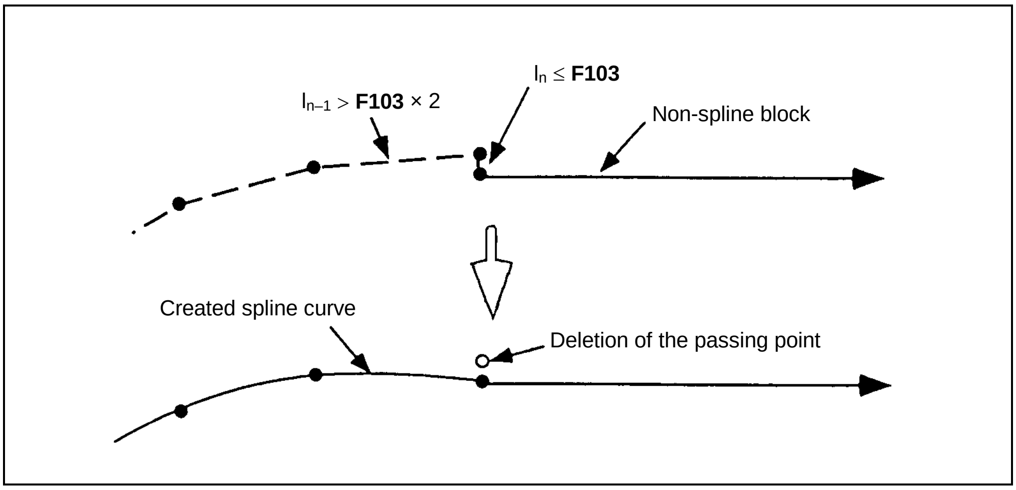


Fig. 6-14   Fairing at the ending point of a spline curve

This function is executed preferentially over the curve slitting function based on the angle of spline cancellation.

D. Feed-rate limitation in spline-interpolation mode

The modal cutting feed rate F remains valid in general for the spline interpolation; however, if the feed rate should be kept constant, it would yield excessively high acceleration at portions where the curvature is big (the curvature radius is small) as shown in Fig. 6-15.

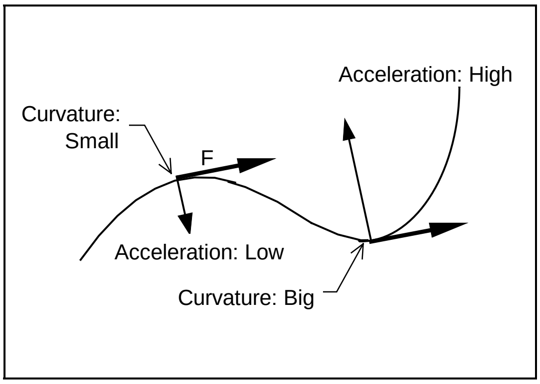


Fig. 6-15 Change of acceleration depending on curvature

In the spline-interpolation mode of our NC, the feed rate can be controlled so that it does not exceed the allowable limit, calculated from the related parameters, for pre-interpolation acceleration.

To obtain an appropriate feed rate for each block of spline interpolation, the limit feed rate F' is calculated by the equation [1] shown below where the smaller between two radii Rs (curvature radius at the starting point of the block) and Re (curvature radius at its ending point) will be regarded as the reference radius R for the block. The modal feed rate F will then be temporarily overridden by F' for the respective block if F>F', so that the whole spline curve can be interpolated block-by-block at the appropriate feed rate according to the curvature radius.

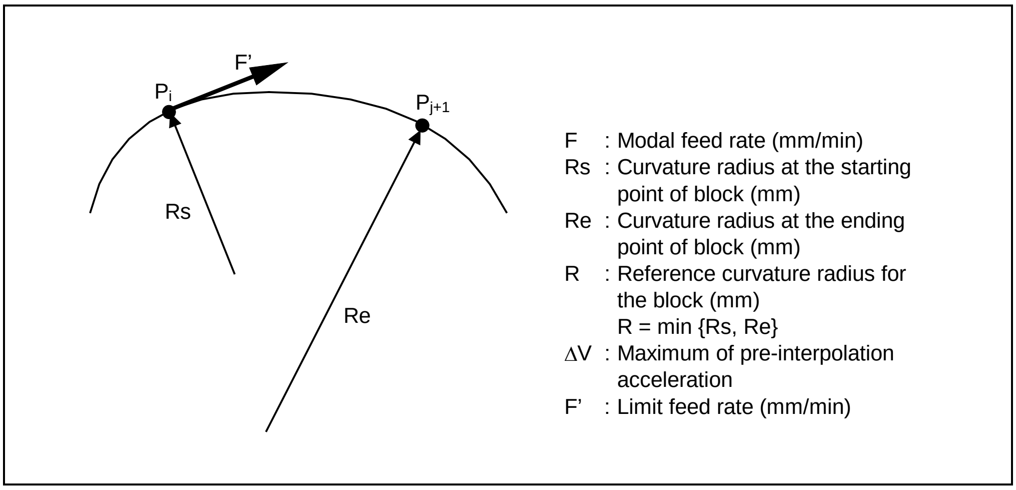


Fig. 6-16 Feed-rate limitation for spline interpolation

$$F' = \sqrt{R \times \Delta V \times 60 \times 1000} \quad \text{.........} [1]$$

$$\Delta V = \frac{\text{G1bF (mm/min)}}{\text{G1btL (msec)}}$$

E. Spline interpolation during tool-diameter offset

The spline interpolation can be performed during tool-diameter offset as follows.

1. Tool-diameter offset (2-dimensional)

   Shown in Fig. 6-17 is an example that the command route is straight in the section P_0P_1, polygonal line in the section $P_1P_2 \dots P_n$ that is the object of spline interpolation, and straight in the section P_nP_{n+1}. The interpolation route with tool-diameter offset is created by the following procedure.

   1) In the first step is created a polygonal line $P_0'P_1'P_2' \dots P_n'P_{n+1}'$ that is offset by the tool-diameter offset value r compared with the original polygonal line $P_0P_1P_2 \dots P_nP_{n+1}$.
   2) Next, a point P_i'' where $\overline{P_iP_i''} = r$ on the vector $\overrightarrow{P_iP_i'}$ is determined for all the pass points P_i (i = 2, 3, . . . n–1) other than the starting point P_1 and the ending point P_n of the spline curve.
   3) Spline interpolation is now conducted for the polygonal line $P_1'P_2''P_3'' \dots P_{n-1}''P_n'$ and the curve thus created will act an offset path of tool center for the commanded spline curve.

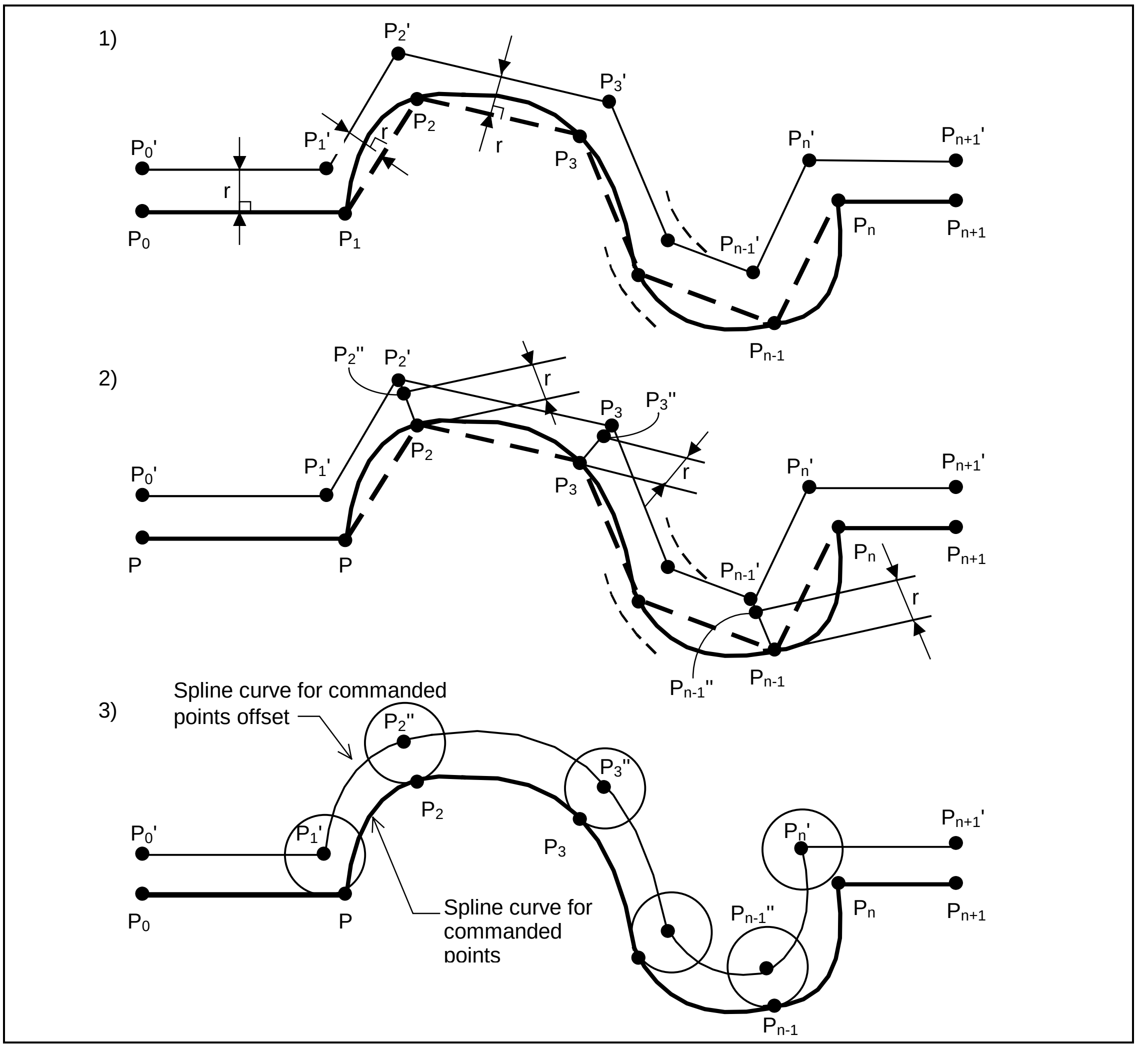


Fig. 6-17 Spline interpolation during tool-diameter offset

The spline curve created in the above-mentioned procedure is not the strict offset, indeed, of the commanded spline curve, but an approximation of it.

2. 3-dimensional tool-diameter offset

   In the 3-dimensional tool-diameter offset, each point defined with programmed coordinates is first offset through the tool radius “r” in the direction of the specified normal vector (i, j, k) and then, the serial points thus offset in the spline-interpolation section are connected in a smooth curve, which will act as the path of tool-radius center for the 3-dimensional spline interpolation.

F. Others

1. The spline interpolation targets the basic coordinate axes of X, Y and Z; however, it is not always required to specify objective axes on commanding the spline interpolation.
   Moreover, the spline-interpolation command code (G06.1) can be given in a block without any movement command.

   Example:

```
N100  G06.1  X_Y_Z0   →    N100    G06.1    X_Y_
N200         X_Y_Z_        N200             X_Y_Z_
N300         X_Y_Z_        N300             X_Y_Z_
 :             :            :                 :

N100  G06.1  F_  (← No movement commands)
N200         X_Y_Z_
N300         X_Y_Z_
 :             :
```

2. The spline-interpolation command (G06.1) falls under the G-code group 01.
3. In the single-block operation mode, the spline interpolation is cancelled and all the respective blocks will individually undergo the linear interpolation.
4. In tool-path check, the blocks of spline interpolation are not actually displayed in a spline curve but in a polygonal line that connects linearly the repective points, which, in case of tool-diameter offset, will have been offset in the same manner as described in the foregoing article E.
5. During spline interpolation, when feed hold is executed, the block for which the feed hold function has been executed will be interpolated, at the beginning of the restart operation along the spline curve existing before the feed hold function was executed, and then the spline curve in the next block onward will be re-created and interpolation executed.
6. Althouth spline interpolation can also be executed in the high-speed maching mode (G05P2 mode), curve shape correction by fairing becomes invalid in the G05P2 mode.

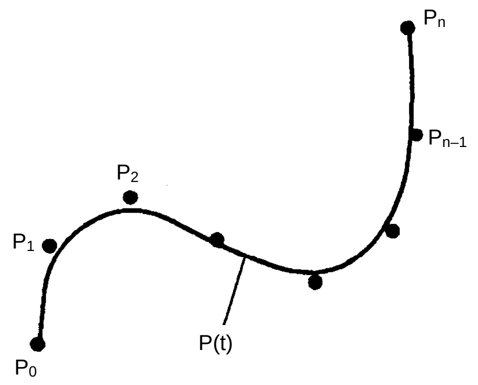
6-11 NURBS Interpolation: G06.2 (Option)

1. Function

The NURBS interpolation function provides interpolation by performing NURBS-defined CNC-internal computations on the command issued from the CAD/CAM system in the NURBS format. With this optional function, a very smooth interpolation path can be obtained since the interpolation process is performed directly without dividing a NURBS-formatted free-form curve into minute line segments.

2. Definition of the NURBS curve

NURBS, short for Non-Uniform Rational B-Spline, provides rationalization of the B-spline function.
The NURBS curve is defined as follows:



$$P(t) = \frac{\sum_{i=0}^{n} N_{i,m}(t) w_i P_i}{\sum_{i=0}^{n} N_{i,m}(t) w_i} \quad (x_{m-1} \le t \le x_{n+1})$$

$$N_{i,1}(t) = \begin{cases} 1 & (x_i \le t \le x_{i+1}) \\ 0 & (t < x_i,\ x_{i+1} < t) \end{cases}$$

$$N_{i,k}(t) = \frac{(t - x_i)\, N_{i,k-1}(t)}{x_{i+k-1} - x_i} + \frac{(x_{i+k} - t)\, N_{i+1,k-1}(t)}{x_{i+k} - x_{i+1}}$$

MEP300

Fig. 6-18 NURBS curve

- “P_i” and “w_i” denote respectively a control point and the weight on the control point.
- “m” denotes the rank, and the NURBS curve of rank “m” is a curve of the (m–1)-th order.
- “x_i” denotes a knot ($x_i \le x_{i+1}$), and an array of knots [x_0 x_1 x_2 ... x_{n+m}] is referred to as the knot vector.
- A variation in parameter “t” from x_{m-1} to x_{n+1} produces NURBS curve P(t).
- $N_{i,k}(t)$ is the B-spline basis function expressed by the above recurrence equation.

Thus the NURBS curve is uniquely defined from the weighted control points and the knot vector.

3. Programming format

```
G6.2[P]  K_X_Y_Z_[R_][F_]  ← NURBS interpolation ON
         K_X_Y_Z_[R_]
         K_X_Y_Z_[R_]
         K_X_Y_Z_[R_]
         :
         K_X_Y_Z_[R_]
         K_
         K_
         K_
         K_                ← NURBS interpolation OFF
```

P : Rank (omissible)
X, Y, Z : Coordinates of the control point
R : Weight on the control point (omissible)
K : Knot
F : Speed of interpolation (omissible)

4. Detailed description

Set the G6.2 code to select the NURBS interpolation mode. Subsequently, designate the rank, the coordinates and weights of the control points, and the knots to determine the shape of the NURBS curve.
The modal code G6.2, which belongs to group 1 of G-codes, is of temporary validity and the modal function relieved by a G6.2 code will automatically be retrieved upon cancellation (termination) of the NURBS interpolation. The G6.2 code can only be omitted for an immediately subsequent setting of the next NURBS curve.

Address P is used to set the rank, and the NURBS curve of rank “m” is of the (m–1)-th order, that is, set as the rank

- P2 for a straight line (curve of the first order),
- P3 for a quadratic curve (of the second order) or
- P4 for a cubic curve (of the third order).

Setting another value than 2, 3 and 4 will cause an alarm, and P4 will be used in default of argument P. The rank, moreover, should be specified in the first block (containing the G6.2 code).
Designate the control points in as many sequential blocks as required by specifying their respective coordinates and weights at addresses X, Y, Z and R. Argument R denotes the weight proper to each control point (R1.0 will be used in default), and the more the weight is applied, the closer will be drawn the NURBS curve to the control point.

Address K is assigned to knots, and the NURBS curve of rank “m” for an “n” number of control points requires an (n+m) number of knots. The required array of knots, referred to as knot vector, is to be designated in sequential blocks, namely: the first knot in the same block as the first control point, the second knot in the same block as the second control point, and so forth. Following the “n” blocks entered thus, designate the remaining “m” knots in single-command blocks. The leading single-command block of argument K also notifies the NC of the completion of entering the control points, and the NURBS interpolation function itself will be terminated with the last block for the “m” knots.

5. Remarks

1. Only the fundamental axes X, Y and Z can undergo the NURBS interpolation.
2. Do not fail to explicitly designate all the required axes X, Y and/or Z in the first block (containing G6.2). Designating a new axis in the second block onward will cause an alarm.
3. Since the first control point serves as the starting point of the NURBS curve, set in the first block (with G6.2) the same coordinates as the final point of the previous block. Otherwise, an alarm will be caused.
4. The setting range for the weight (R) is from 0.0001 to 99.9999. For a setting without decimal point, the least significant digit will be treated as units digit (for example, 1 = 1.0).
5. The knot (K) must be designated for each block. Omission results in an alarm.
6. Knots, as with the weight, can be set down to four decimal digits, and the least significant digit of a setting without decimal point will be regarded as units digit.
7. Knots must be monotonic increasing. Setting a knot smaller than that of the previous block will result in an alarm.
8. The order of addresses in a block can be arbitrary.

Return to Library

9. The shape of the NURBS curve can theoretically be modified very flexibly by changing the rank, the positions and weights of the control points, and the knot vector (the relative intervals of knots).
   In practice, however, manual editing is almost impossible, and a special CAD/CAM system should be used to edit the NURBS curve and create the program for the interpolation. Generally speaking, do not edit manually the program created by a CAD/CAM system for the NURBS interpolation.

6. Variation of curve according to knot vector

The NURBS curve, which in general passes by the control points, can be made to pass through a specific control point by setting a certain number of knots in succession with the same value. In particular, setting as many leading and trailing knots as the rank (value of P) with the respective identical values will cause the NURBS curve to start from the first control point (P_0) and to end in the last one (P_5).
The examples given below exhibit a variation of the NURBS curve according to the knot vector with the control points remaining identical.

Example 1: Rank : 4
Number of control points : 6
Knot vector : [0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0]

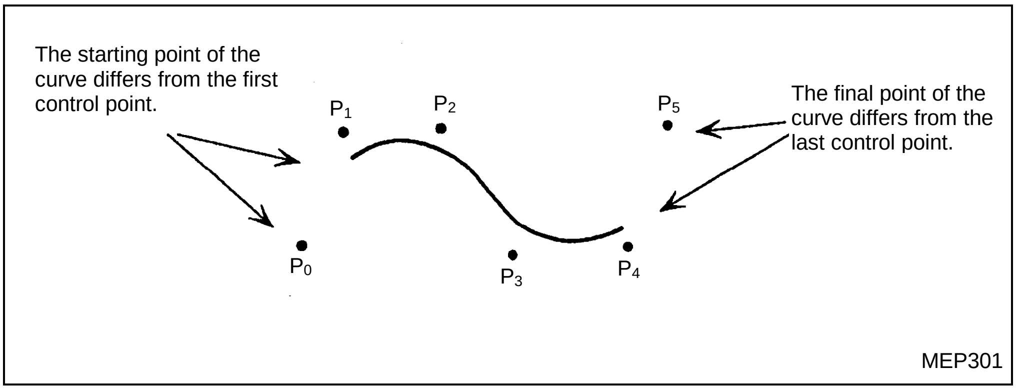


Fig. 6-19 NURBS curve for continuously increasing knots

Example 2: Rank : 4
Number of control points : 6
Knot vector : [<u>0.0 0.0 0.0 0.0</u> 1.0 2.0 <u>3.0 3.0 3.0 3.0</u>]
[1] [2]

Point [1]: The first four (=rank) knots have the same value assigned.
Point [2]: The last four (=rank) knots have the same value assigned.

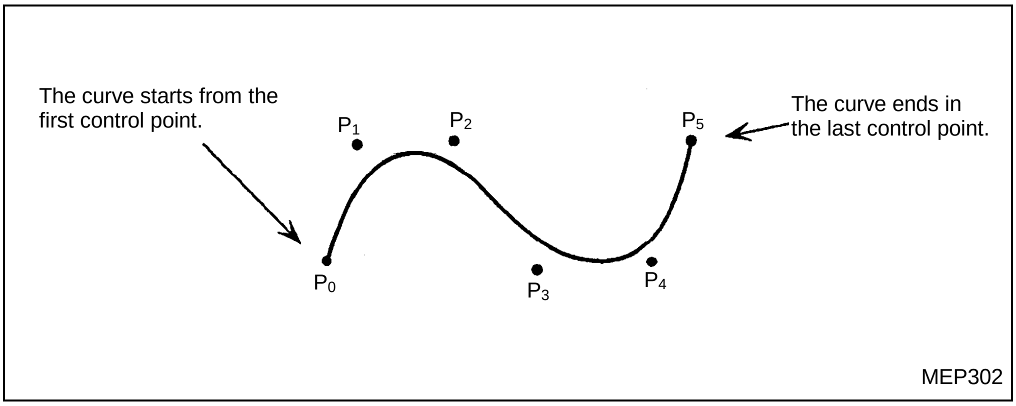


Fig. 6-20 NURBS curve for some identical knots

Return to Library

Note 1: The NURBS interpolation can be performed only for the NURBS curve that starts and ends from the first and in the last control point. Do not fail, therefore, to set as many leading and trailing knots as the rank with the respective identical values.

Note 2: The NURBS interpolation is executed at the designated feed rate (F-code). During the shape correction mode, however, the interpolation speed is controlled in order that the maximum available acceleration may not be exceeded in the section of a considerable curvature.

7. Compatibility with the other functions

The tables in this section specify the compatibility of the NURBS interpolation with the other functions. Pay attention to the incompatible functions, especially G-codes.

A. Preparatory, feed and auxiliary functions

The table below enumerates the G-codes, F-, M-, S-, T- and B-codes with regard to their availability before, with and after G6.2.

○: available ×: not available

Function	Code	before G6.2	with G6.2	after G6.2
G-codes of group 00	all	○	×	×
G-codes of group 01	all	○	○ (Note)	×
G-codes of group 02	G17	○	○	×
	G18			
	G19			
G-codes of group 04	G22	×	×	×
	G23	○	×	×
G-codes of group 05	G93	○	○	×
	G98			
	G99			
G-codes of group 06	G20	○	○	×
	G21			
G-codes of group 07	G40	○	×	×
	G41	×	×	×
	G42	×	×	×
G-codes of group 09	G80	○	×	×
	the others	×	×	×
G-codes of group 12	G54 - G59	○	○	×
G-codes of group 13	G61.1	○	×	×
	G61.2	○	×	×
	G61	×	×	×
	G62	×	×	×
	G63	×	×	×
	G64	○	×	×
G-codes of group 14	G66	×	×	×
	G66.1	×	×	×
	G66.2	×	×	×
	G67	○	×	×
G-codes of group 16	G68.5	×	×	×
	G69.5	○	×	×
High-speed machining mode	G5P0	○	×	×
	G5P2	×	×	×
Feed function	F	○	○	×
Auxiliary function	MSTB	○	×	×

B. Skip instructions

The table below enumerates the skip instructions with regard to their availability before, with and after G6.2.

○: available ×: not available

Instruction	before G6.2	with G6.2	after G6.2
Optional block skip	○	○	×
Control Out/In	○	○	×

Note: Designating another address than X, Y, Z, R and K in the mode of (i. e. after) G6.2 will cause an alarm.

C. Interruption and restart

The table below enumerates the functions for interrupting and restarting the program flow with regard to their availability before, with and after G6.2.

○: available ×: not available

Function	before G6.2	with G6.2	after G6.2
Single-block operation	○	×	○ (Note)
Feed hold	○	×	○
Reset	○	○	○
Program stop	○	×	×
Optional stop	○	×	×
Manual interruption (Pulse feed and MDI)	○	×	×
Restart	○	×	×
Comparison stop	○	×	×

Note: The single-block stop only occurs between blocks with different knots.

D. Tool path check

The tool path in a section of the NURBS interpolation can only be displayed as if the control points were linearly interpolated (in the mode of G01).

8. Sample program

The program section below refers to a NURBS interpolation of rank 4 (cubic curve) for seven control points.

Control points: P_0 P_1 P_2 P_3 P_4 P_5 P_6
Knot vector: [0.0 0.0 0.0 0.0 1.0 2.0 3.0 4.0 4.0 4.0 4.0]

```
        :
        :
G90 G01 X0 Y120.F3000
Y100.       ............. P0
G6.2 P4 X0 Y100.R1.K0... P0
X10.Y100.R1.K0.......... P1
X10.Y60.R1.K0........... P2
X60.Y50.R1.K0........... P3
X80.Y60.R1.K1........... P4
X100.Y40.R1.K2.......... P5
X100.Y0 R1.K3........... P6
K4.
K4.
K4.
K4.
G01 X120................ P7
        :
        :
```

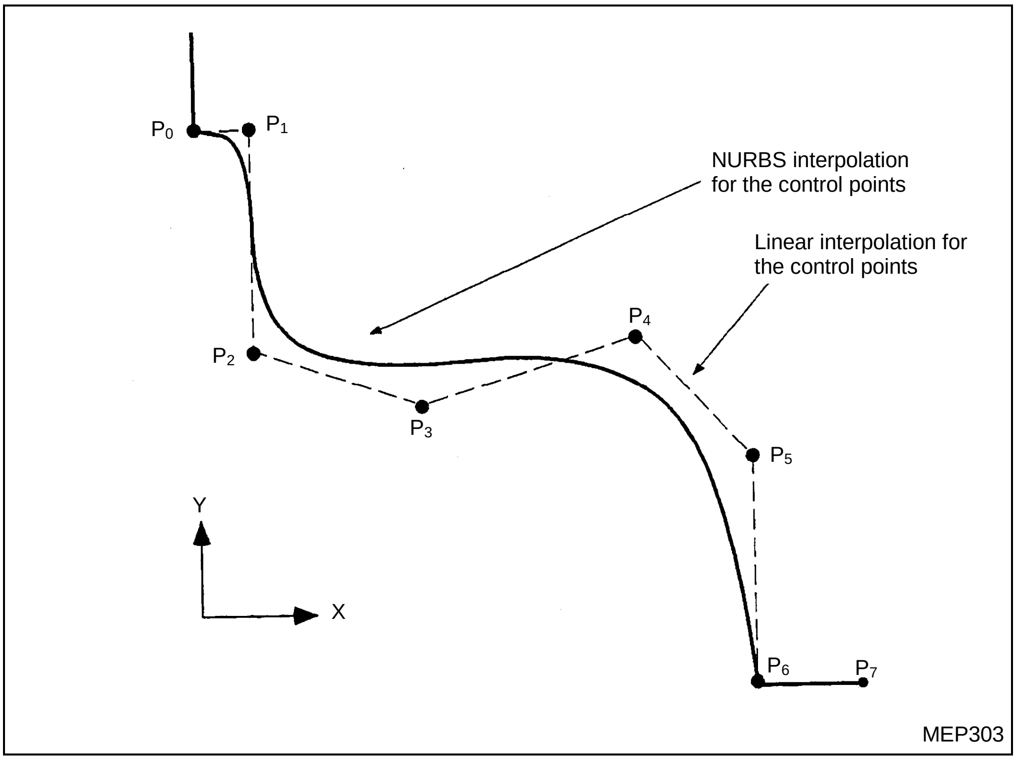


Fig. 6-21 NURBS interpolation and linear interpolation

9. Related alarms

The table below enumerates the alarms related to the NURBS interpolation.

Alarm list

Alarm No.	Alarm message	Cause	Remedy
806	ILLEGAL ADDRESS	Another address than those for the nominated axes (X, Y and/or Z), the weight (R) and the knot (K) is set in the G6.2 mode.	Clear the inadequate address.
807	ILLEGAL FORMAT	1. The modal condition is not appropriate to set G6.2.	1. Satisfy the modal condition with reference to item 7-A.
		2 A block in the G6.2 mode is set without knot (K).	2. Do not fail to set a knot in each block in the G6.2 mode.
		3. The number of blocks with the same knot in succession does not reach the rank.	3. Set an appropriate knot vector with reference to example 2 given in item 6.
809	ILLEGAL NUMBER INPUT	1. The number of digits exceeds the specification of axis commands (X, Y or Z).	1. Specify the axis command within eight digits.
		2. The rank (P) is not admissible.	2. Set 2, 3 or 4 at address P.
		3. The value of a knot is not admissible.	3. Set a value in a range of 0.0001 to 99.9999.
		4. The knot vector is not monotonic increasing.	4. Check the blocks for a decreasing knot.
816	FEEDRATE ZERO	The feed rate (F-code) has not yet been designated.	Set an F-code before or in the same block as the G6.2 code.
936	OPTION NOT FOUND	The system is not equipped with the optional function of the NURBS interpolation.	Purchase and install the optional function.
955	START AND END POINT NOT AGREE	The axis coordinates designated in the block of G6.2 do not correspond to the final point of the previous block.	Designate in the first block of the NURBS interpolation the same position as the final point of the previous block.
956	RESTART OPERATION NOT ALLOWED	The designated restart block falls within the mode of G6.2.	Restart operation is not allowed from the midst of the NURBS interpolation.
957	MANUAL INTERRUPT NOT ALLOWED	An interruption by pulse handle or MDI operation is commanded in the midst of the G6.2 mode.	Manual interruption is not allowed in the midst of the NURBS interpolation.

6-12 Cylindrical Interpolation Command: G07.1

1. Function and purpose

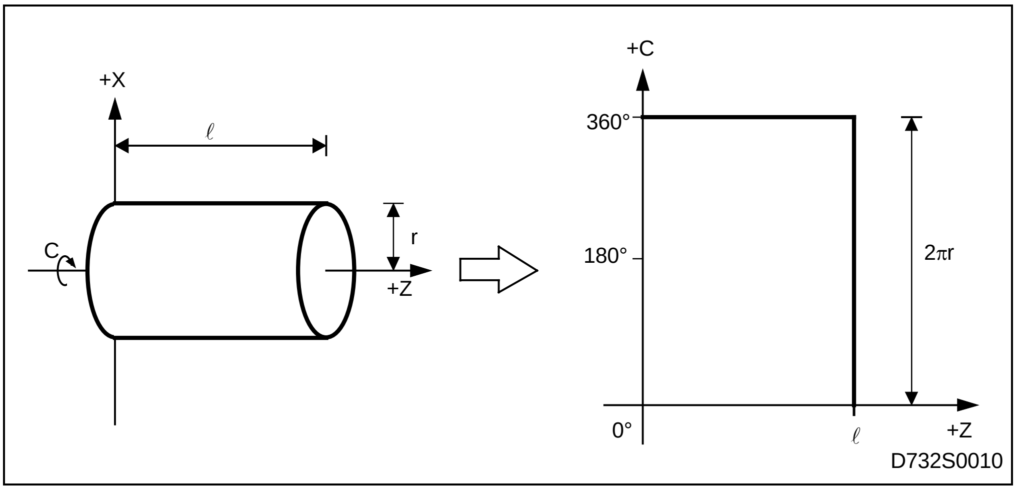
Cylindrical interpolation function refers to a function by which the sides of a cylindrical workpiece are machined. The cylindrical interpolation function capable of programming in the form in which the sides of a cylinder are spread can very easily prepare programs including cylindrical cam-grooving.

2. Programming format

G07.1 C_; Cylindrical interpolation mode (C: cylindrical radius)
G07.1 C0; Cylindrical interpolation cancel mode
(These G-codes should be commanded in an independent block.)

* When the cylindrical radius (address C) is not commanded, a cylinder is defined taking as radius current value of X-axis (treated as radius value) when G07.1 is commanded.

3. Operation



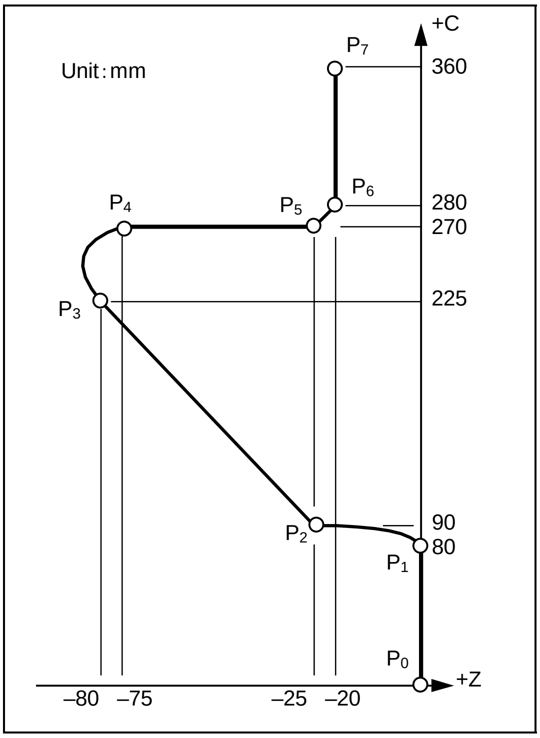
The moving distance of rotational axis commanded with an angle is converted to the linear distance on the circumference in CNC. After the conversion, linear interpolation or circular interpolation is given with the other axis. After the interpolation, the calculated movement is converted again to the moving distance of rotational axis.

Return to Library

4. Sample programs

In case of the figure on the right: $P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5 \rightarrow P_6 \rightarrow P_7$ (r = 50 mm)

```
G00 G98;
G28 U0 W0;
T001T000M06;
M200;
G18    W0 H0;
X52.   M203 S1000;
G01    X40.F100;
G07.1 C50.;
G01    C80.F100;          P0 → P1
G03    Z-25.C90.R50.;     P1 → P2
G01    Z-80.C225.;        P2 → P3
G02    Z-75.C270.R55.;    P3 → P4
G01    Z-25.;             P4 → P5
G03    Z-20.C280.R80.;    P5 → P6
G01    C360.;             P6 → P7
G07.1 C0.;
G28    U0;
G28    W0 H0;
M202;
M30;
```



5. Supplement

Relation of cylindrical interpolation mode to other functions

A. Feed rate designation

The feed rate commanded during cylindrical interpolation mode provides a speed on the plane where cylindrical sides are spread.

Note: The example of programming shown on the right (F10) realizes a C-axis feed of 143°/min [approximation of 10/(4 × 2π) × 360].

```
⋮
G98;
G07.1 C4;
G01 C_ F10;
⋮
```

B. Circular interpolation (G02, G03)

1. Plane selection
   Giving the circular interpolation between the rotational axis and other linear axis during cylindrical interpolation mode requires the command of plane selection (G17, G18, G19).

   Example: When the circular interpolation is given between Z- and C-axes, the circular interpolation command is

   G18 Z_ C_ ;
   G02/G03 Z_ C_ R_ ;

2. Radius designation
   The circular radius by word address I, J or K cannot be commanded during cylindrical interpolation mode. The circular radius is commanded by address R. The radius must be commanded not with angle, but with mm or inch.

C. Tool nose radius compensation

Giving the tool nose radius compensation during cylindrical interpolation mode requires the command of plane selection as with the circular interpolation. However, giving the tool nose radius compensation requires start-up and cancel during cylindrical interpolation mode. Establishing a cylindrical interpolation mode with the tool nose radius compensation given does not provide proper compensation.

D. Positioning

Positioning (including commands producing the cycle of rapid feed such as G28 and G80 to G89) cannot be accomplished during cylindrical interpolation mode. Positioning requires establishing a cylindrcal interpolation cancel mode.

E. Coordinate system setting

The workpiece coordinate system (G50) cannot be commanded during cylindrical interpolation mode.

6. Notes

1. The cylindrical interpolation mode cannot be re-established during cylindrical interpolation mode. Re-establishment requires the cancel of cylindrical interpolation mode.
2. The cylindrical interpolation (G07.1) cannot be commanded during positioning mode (G00).
3. Accuracy

   - Automatic operation
     During cylindrical interpolation mode, the moving distance of rotational axis commanded with an angle is once internally converted to the distance on the circumference. And after arithmetic operation is performed on linear interpolation or circular interpolation with the other axis, the calculated movement is again converted to the angle.
     As a result, where the cylindrical radius is small, the actual moving distance may differ from the commanded value. However, the error produced then is not accumulated.

     $$\text{Actual moving distance} = \left(\frac{\text{MOVE}}{2 \times 2\pi r} \times \left(\text{Command value} \times \frac{2 \times 2\pi r}{\text{MOVE}}\right)\right)$$

     MOVE : Moving distance per rotation of rotational axis (Parameter)
     r : Workpiece radius
     (   ) : Rounding to the least input increment

   - Manual operation
     Performing manual operation during cylindrical interpolation mode in manual absolute ON status may cause an error for the above reason.

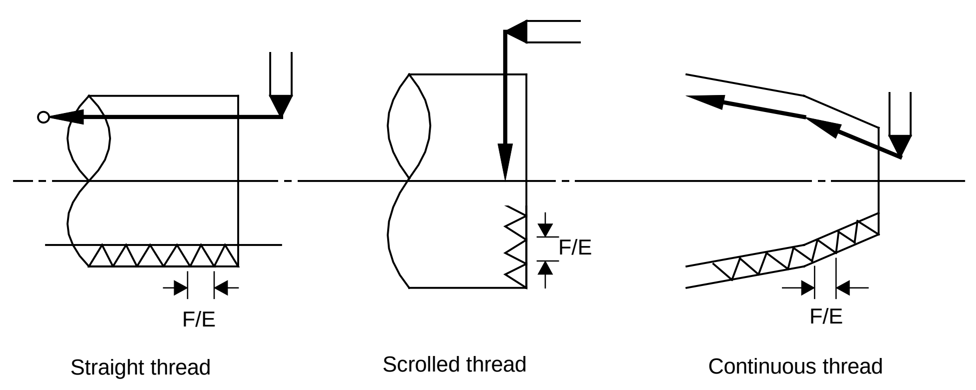
4. The hole machining fixed cycle (G83 to G89) cannot be commanded during cylindrical interpolation mode.

6-13 Threading

6-13-1 Constant lead threading: G32 [Series M: G33]

1. Function and purpose

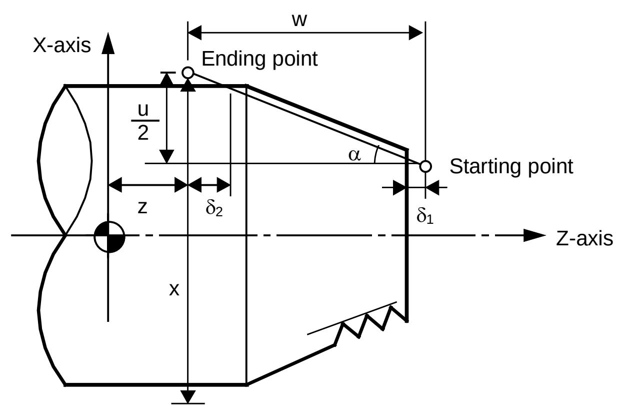
The G32 command controls the feedrate of the tool in synchronization with the spindle rotation and so this enables both the straight and scrolled thread cutting of constant leads and the continuous thread cutting.



TEP026

2. Programming format

G32 Zz/Ww Xx/Uu Ff;		(Normal lead thread cutting commands)
Where	Zz, Ww, Xx, Uu:	Thread ending point addresses and coordinates
	Ff:	Lead of long axis (axis of which moving distance is the longest) direction
G32 Zz/Ww Xx/Uu Ee;		(Precision lead threading commands)
Where	Zz, Ww, Xx, Uu:	Thread ending point addresses and coordinates
	Ee:	Lead of long axis (axis of which moving distance is the longest) direction

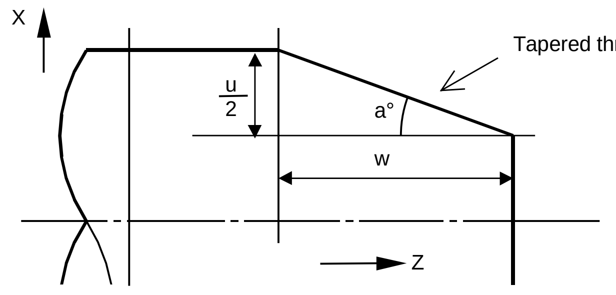


TEP027

3. Detailed description

1. The E command is also used for the number of threads in inch threading, and whether the thread number or precision lead is to be designated can be selected by parameter setting. (Bit 7 of address **F91** is set to 0 for precision lead designation.)

2. The lead in the long axis direction is commanded for the taper thread lead.



When a < 45° lead is in Z-axis direction.
When a > 45° lead is in X-axis direction.
When a = 45° lead can be in either Z- or X-axis direction.

TEP028

Refer to Section 7-5 for details of lead setting range.

Note: It is not possible to designate a lead where the feed rate as converted into per-minute feed exceeds the maximum cutting feed rate.

3. The constant peripheral speed control function should not be used here.

4. The spindle speed should be kept constant throughout from the roughing until the finishing.

5. If the feed hold function is employed to stop the feed during thread cutting, the thread height will lose their shape. For this reason, feed hold does not function during thread cutting.
If the feed hold button is pressed during threading, block stop will result at the ending point of the block following the block in which threading is completed (no longer in G32 mode).

6. The converted cutting feed rate is compared with the cutting feed clamp rate when threading starts, and if it is found to exceed the clamp rate, an alarm will result. (See the Note in item 2 above.)

7. In order to protect the lead during threading, a converted cutting feed rate may sometimes exceed the cutting feed clamp rate.

8. An illegal lead is produced at the start and at the end of the thread cutting because of servo system delay and other factors.
Therefore, it is necessary to command a thread length obtained by adding the illegal lead lengths δ_1 and δ_2 to the required thread length.

9. The spindle speed is subject to the following restriction:

$1 \leq R \leq$ Maximum feed rate/Thread lead

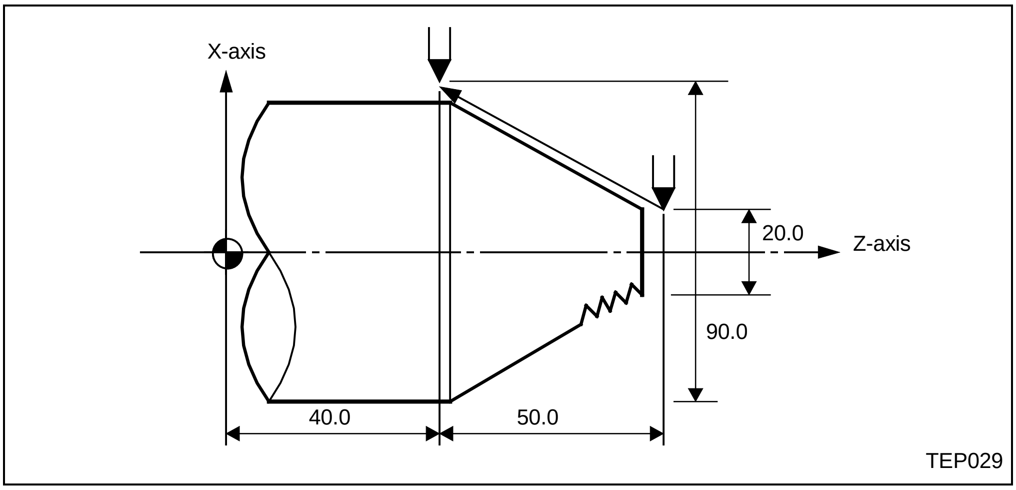
where R : Spindle speed (rpm) $\leq$ Permissible speed of encoder (rpm)
Thread lead = mm or inches
Maximum feed rate = mm/min or inch/min (this is subject to the restrictions imposed by the machine specifications).

10. During threading, use or disuses of dry run can be specified by setting parameter **F111** bit 1.

11. Synchronous feed applies for the threading commands even with an asynchronous feed mode (G98).

12. Spindle override is valid even during threading. But the override value will not be changed during threading.
13. When a threading command is programmed during tool nose R compensation, the compensation is temporarily cancelled and the threading is executed.
14. When the mode is switched to another automatic operation mode while G32 is executed, the following block which does not contain a threading command is first executed and then the automatic operation stops.
15. When the mode is switched to manual operation mode while G32 is executed, the following block which does not contain a threading command is first executed and then the automatic operation stops. In the case of the single block operation, the following block which does not contain a threading command is first executed and then the automatic operation stops.
16. The threading command waits for the single rotation synchronization signal of the rotary encoder and starts movement.
    With this NC unit, however, movement starts without waiting for this signal when another system issues a threading command during threading by one system.
    Therefore, threading commands should not be issued by a multiple number of systems.

4. Sample programs




```
G32 X90.0 Z40.0 E12.34567;     Absolute data command
G32 U70.0 W–50.0 E12.34567;    Incremental data command
```

6-13-2 Inch threading: G32 [Series M: G33]

1. Function and purpose

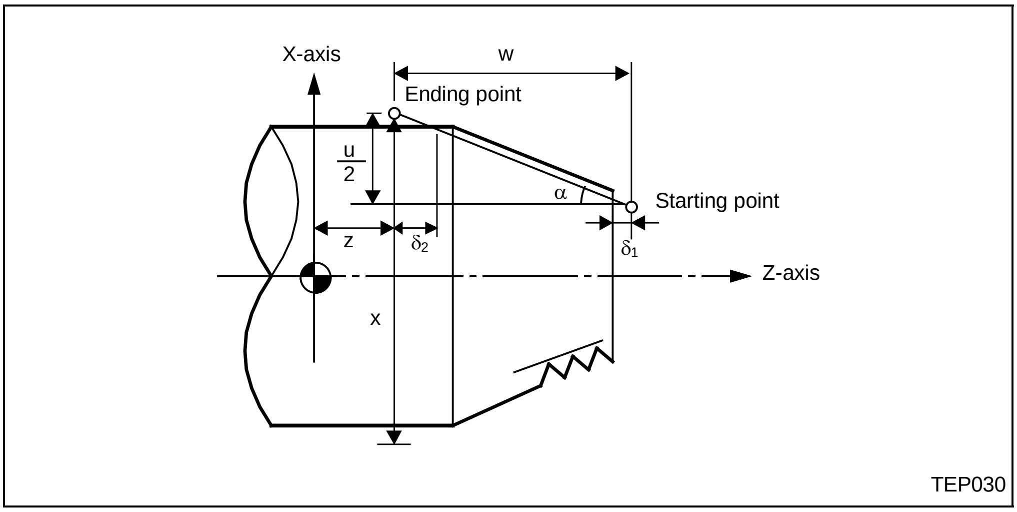
If the number of threads per inch in the long axis direction is designated in the G32 command, the feed of the tool will be controlled to synchronize with the spindle rotation. That is, constant lead straight threading, taper threading and continuous threading can be performed.

2. Programming format

G32 Zz/Ww Xx/Uu Ee;

Where Zz, Ww, Xx, Uu: Thread ending point addresses and coordinates

Ee: Number of threads per inch in direction of long axis (axis of which the moving distance is the longest)
(Decimal point command can also be assigned.)

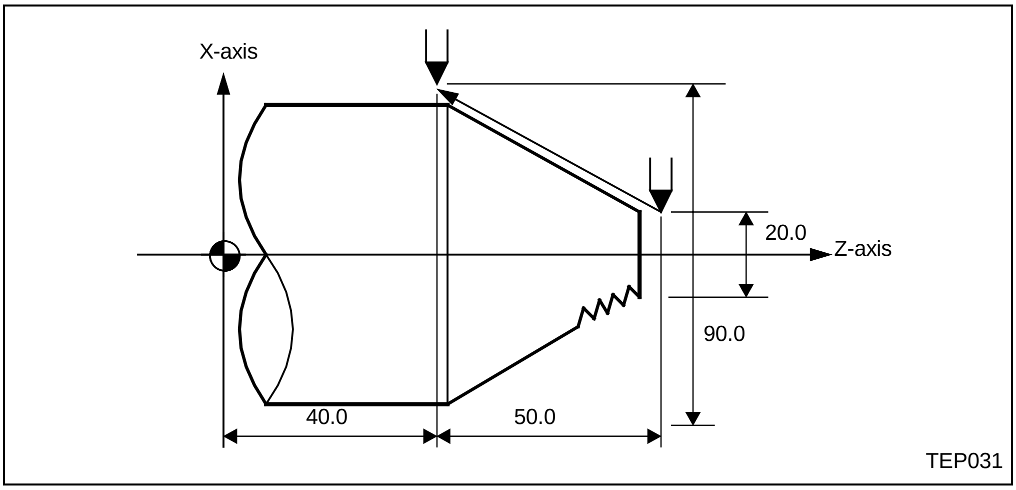


3. Detailed description

1. The number of threads in the long axis direction is assigned as the number of threads per inch.
2. The E code is also used to assign the precision lead length, and whether the thread number or precision lead length is to be designated can be selected by parameter setting (allowed by parameter **F91** bit 7).
3. The E command value should be set within the lead value range when converted to the lead.
4. See Subsection 6-13-1 on “Constant lead threading” for further details.

Return to Library

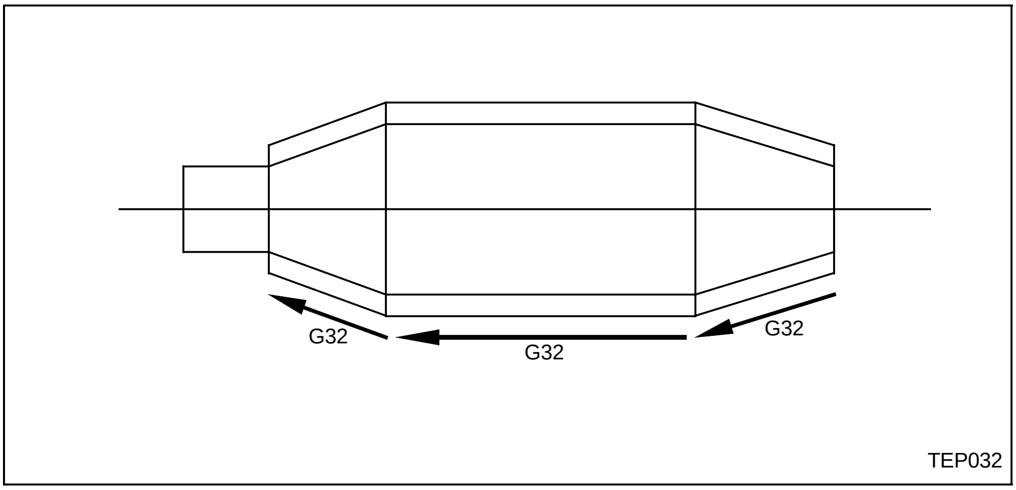
4. Sample programs




```
G32 X90.0 Z40.0 E12.0;       Absolute data command
G32 U70.0 W–50.0 E12.0;      Incremental data command
```

6-13-3 Continuous threading

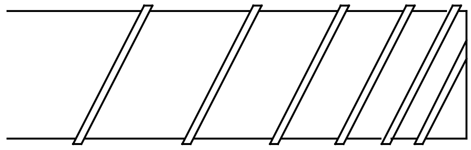
Continous threading is possible by designating threading commands continuously. In this way, it is possible to cut special threads whose lead or shape changes.



6-13-4 Variable lead threading: G34

1. Function and purpose

Variable lead threading is possible by commanding the increase or decrease of a lead per screw rotation.



D732S0012

2. Programming format

G34 Xx/Uu Zz/Ww Ff/Ee Kk;

It is the same as the case of straight and taper threading of G32 except an address K.
A value commanded with K gives the increase or decrease of a lead per screw rotation.

Values which K can take are as follows:

Metric input: ±0.00001 to ±999.99999 mm/rev
Inch input: ±0.000001 to ±99.999999 in./rev

3. Notes

1. As a result of the increase or decrease of a lead, when exceeding the range of the command value of screw lead or when cutting feed gets excessively high, the feed rate is clamped at rapid feed rate.
2. “Feed hold during threading” function is invalid for G34.

6-13-5 Threading with C-axis interpolation: G01.1

1. Function and purpose

The G01.1 command in the milling mode enables a simultaneous interpolation on the C-axis and the X- and/or the Z-axis for straight, tapered or scrolled thread cutting of constant leads.

2. Programming format

G01.1 Zz/Ww Xx/Uu Ff Ss;

Where Zz, Ww, Xx, Uu: Thread ending point addresses and coordinates (mm or in.)

Ff: Lead of long axis (axis of which moving distance is the longest) direction

Ss: Rotational speed of C-axis (rpm)

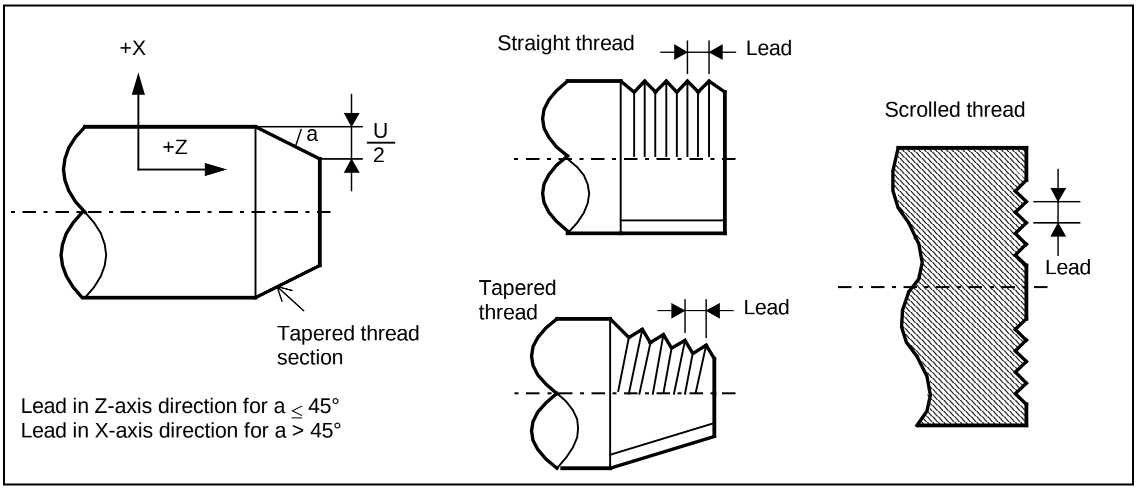
Set parameter **F111** bit 3 to select the direction of C-axis rotation:

F111 bit 3 = 0 : Normal rotation of C-axis

= 1 : Reversed rotation of C-axis

3. Detailed description

1. For tapered thread cutting, specify the lead in the long axis direction.



2. Range of specification of lead (address F)

   - For data input in mm : 0.0001 to 500.0000 mm
   - For data input in in. : 0.000001 to 9.999999 in.

3. Specification range of rotational speed (address S)

   1 ≤ S ≤ Max. speed of C-axis rotation

   - The maximum speed of C-axis rotation (1/360 of value "C" of parameter **M3**) depends on the respective machine model.
   - Do not create a program nor operate the overriding keys in such a manner that the maximum speed of C-axis rotation should be exceeded.

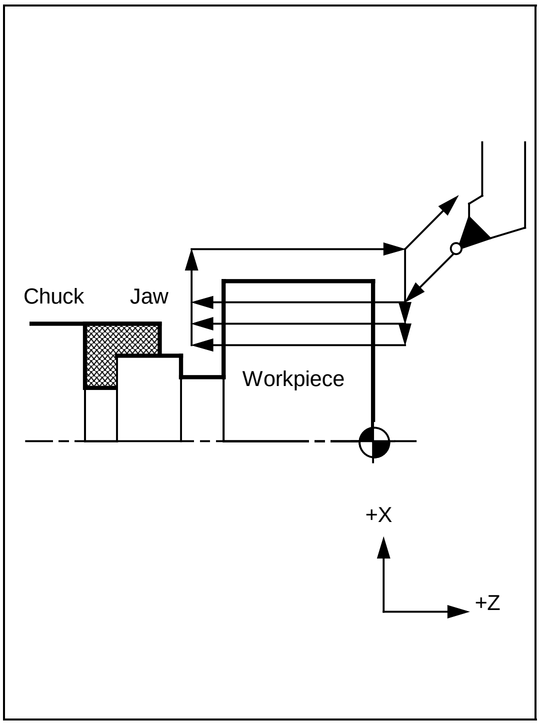
4. During execution of G01.1 command, it is possible, indeed, but not advisable at all to apply feed hold or to change the override value for fear of deformation of the thread.

5. The speed of C-axis rotation should be kept constant throughout from roughing till finishing.

6. The number of C-axis revolutions for execution of one G01.1 command must not exceed 2982.

4. Sample programs

```
G98 G97;
G28 U0 W0;
T001T000M06;
G50 X300.Z100.;
M200;
G00 X100.Z2.C0.;
G01.1 W-100.F2.S400;(*)
G00 U10.;
W100.C0.;
U-11.;
G01.1 W-100.F2.S400;(*)
G00 U11.;
W100.C0.;
G00 U-12.;
G01.1 W-100.F2.S400;(*)
G00 U12.;
W100.;
G28 U0 W0.;
M202;
M30;
```

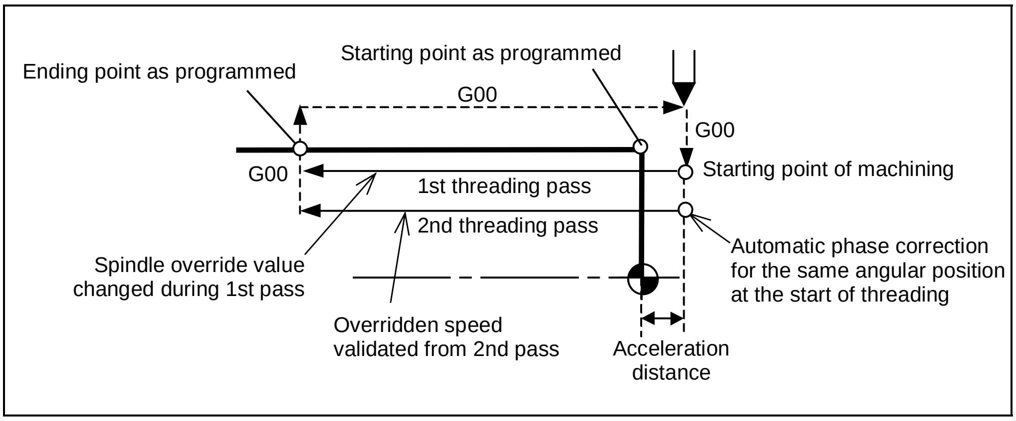


(*) Command for threading with C-axis control, 2 mm lead and 400 rpm

6-13-6 Automatic correction of threading start position (for overriding in a threading cycle)

1. Function and purpose

The phase of the spindle is automatically corrected at the start of each threading pass to prevent the threading position from deviating even when the spindle override value is updated in the middle of a threading cycle.
The use of this option allows the thread cutting conditions to be changed even in the flow of a threading cycle.



2. Related G-codes

The automatic correction function is applicable to the following G-codes of threading:

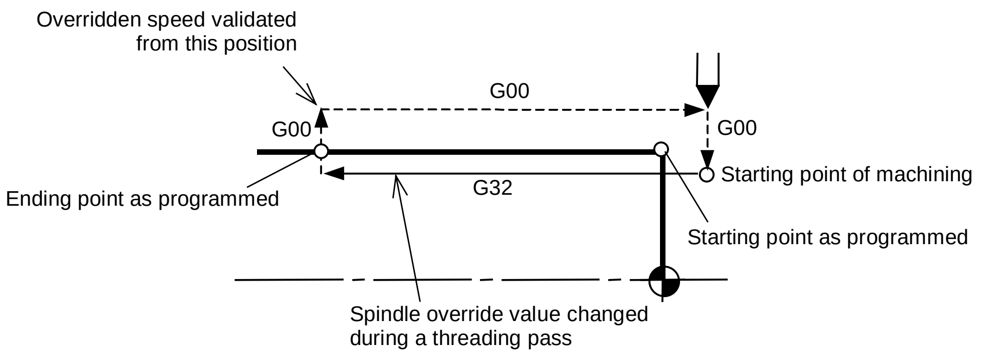
Function	G-code series T
Thread cutting (straight, taper)	G32
Turning fixed cycle for threading	G92
Compound fixed cycle for threading	G76

Note: Variable-lead threading (G34), or continuous threading for different-pitch sections, requires continuous or transitional acceleration between blocks, as well as different distances of acceleration. The automatic correction function cannot guarantee correct thread forming for a speed overriding in the middle of these threading cycles.

3. Detailed description

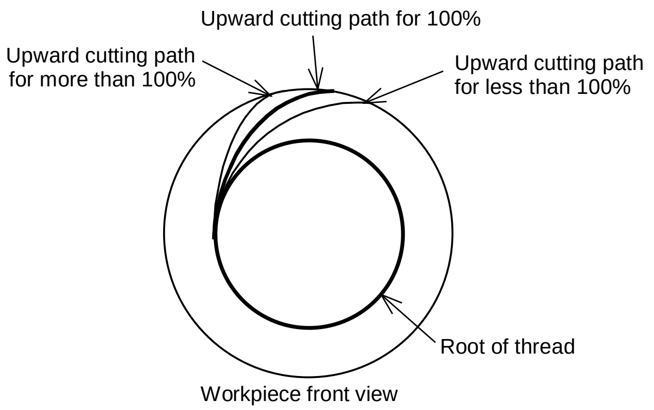
1. The automatic correction function is an option.

2. Even in the middle of a threading pass, operating the turning/milling spindle speed overriding keys immediately changes the speed indication in percentages, indeed, but the actual speed will not accordingly change till completion of the threading block (or a series of the threading blocks in the case of “continuous threading”).



3. The function for automatic correction of threading start position does not include corresponding adjustment of the acceleration distance for threading. To use an overriding value above 100%, therefore, specify in the machining program such an acceleration distance as to allow for the maximum spindle speed.

4. As for the end of thread, the length of the upward cutting path on the workpiece will become shorter, or greater, for a spindle override value below, or above, 100%.



Changes in the upward cutting path according to the spindle override value

4. Notes

1. This function is not valid for a threading by simultaneous cutting with both turrets.
2. This function is not valid for a threading by synchronization of both turning spindles.
3. This function is only valid for a longitudinal threading (by cutting feed on the Z-axis).
4. After changing the spindle override value the execution of a threading block should not be started until spindle rotation has been stabilized; otherwise the starting section will only be cut to an incomplete thread.
5. Do not allow a threading block to be executed with the spindle override value set to 0%; otherwise the machine operation will be stopped at the beginning of that block.

6-14 Helical Interpolation: G17, G18, G19 and G02, G03

1. Function and purpose

Command G02 or G03 with a designation for the third axis allows synchronous circular interpolation on the plane specified by plane-selection command G17, G18 or G19 with the linear interpolation on the third axis.

2. Programming format

G17 G02 Xx_1 Yy_1 Zz_1 Ii_1 Jj_1 Pp_1 Ff_1 ;
(G03)

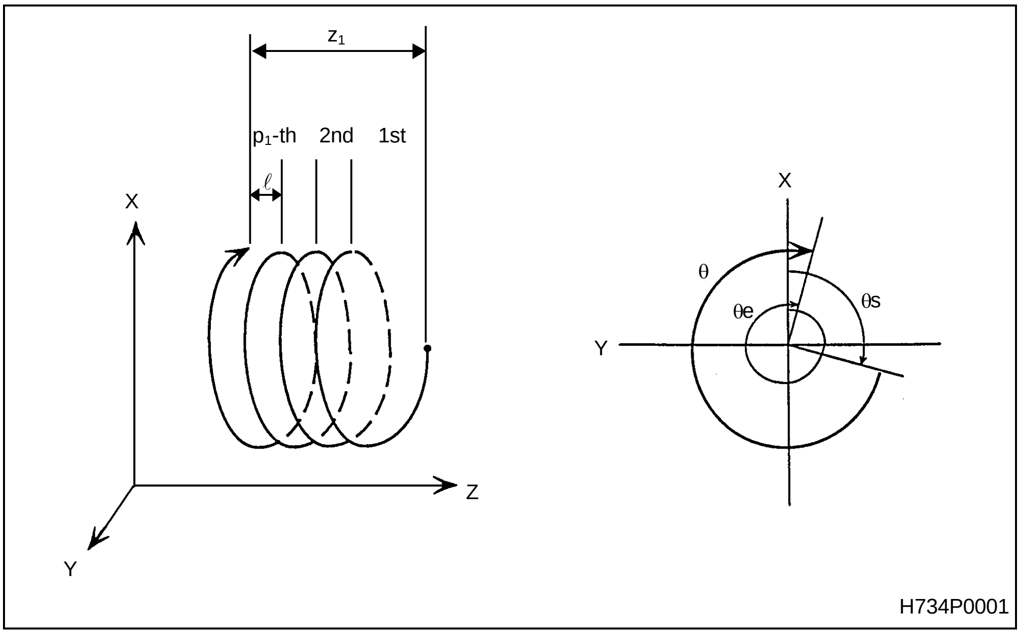
- Ff_1: Feed rate
- Pp_1: Number of pitches
- Ii_1 Jj_1: Arc center coordinates
- Zz_1: Linear axis ending point coordinate
- Xx_1 Yy_1: Arc ending point coordinates

or

G17 G02 Xx_2 Yy_2 Zz_2 Rr_2 Pp_2 Ff_2 ;
(G03)

- Ff_2: Feed rate
- Pp_2: Number of pitches
- Rr_2: Arc radius
- Zz_2: Linear axis ending point
- Xx_2 Yy_2: Arc ending point coordinates

3. Detailed description



1. For helical interpolation, movement designation is additionally required for one to two linear axes not forming the plane for circular interpolation.
2. The velocity in the tangential direction must be designated as the feed rate F.
3. The pitch ℓ is calculated as follows:

$$\ell = \frac{z_1}{(2\pi \bullet p_1 + \theta)/2\pi}$$

$$\theta = \theta e - \theta s = \tan^{-1}\frac{ye}{xe} - \tan^{-1}\frac{ys}{xs} \quad (0 \leq \theta < 2\pi)$$

where (xs, ys): relative coordinates of starting point with respect to the arc center
(xe, ye): relative coordinates of ending point with respect to the arc center

Return to Library

4. Address P can be omitted if the number of pitches is 1.
5. Plane selection
   As with circular interpolation, the circular-interpolation plane for helical interpolation is determined by the plane-selection code and axis addresses. The basic programming procedure for helical interpolation is: selecting a circular-interpolation plane using a plane-selection command (G17, G18 or G19), and then designating the two axis addresses for circular interpolation and the address of one axis (perpendicular to the circular-interpolation plane) for linear interpolation.

   - X-Y plane circular, Z-axis linear

     After setting G02 (or G03) and G17 (plane-selection command), set the axis addresses X, Y and Z.

   - Z-X plane circular, Y-axis linear

     After setting G02 (or G03) and G18 (plane-selection command), set the axis addresses Z, X and Y.

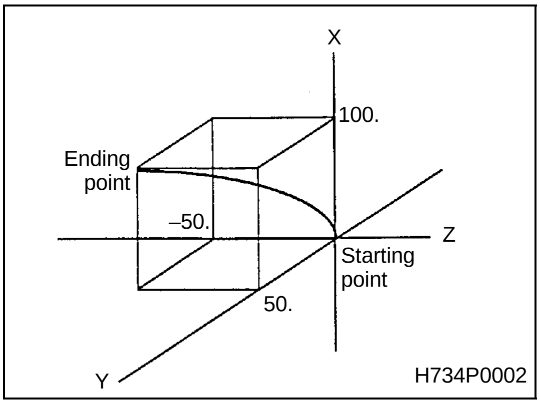
   - Y-Z plane circular, X-axis linear

     After setting G02 (or G03) and G19 (plane-selection command), set the axis addresses Y, Z and X.

4. Sample programs

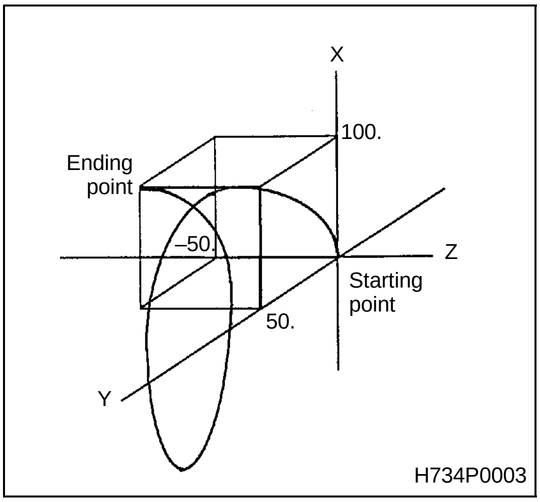
Example 1:

```
G28 U0 W0 Y0;
G50 X0 Z0 Y0;
G17 G03 X100. Y50. Z-50. R50. F1000;
```



Example 2:

```
G28 U0 W0 Y0;
G50 X0 Z0 Y0;
G17 G03 X100. Y50. Z-50. R50. P2 F1000;
```



7 FEED FUNCTIONS

7-1 Rapid Traverse Rates

A separate rapid traverse rate can be set for each axis. The maximum rate of rapid traverse, however, is limited according to the particular machine specifications.

Refer to the Operating manual for the machine for rapid traverse rates.

Two types of tool paths are available for positioning: an interpolation type, which uses a line to perform interpolation from the starting point through the ending point, and a non-interpolation type, which moves the tool at the maximum speed of each axis.

Use a parameter to select the interpolation type or the non-interpolation type. The positioning time is the same for both types.

7-2 Cutting Feed Rates

A cutting feed rate must be designated using address F and an eight-digit number (F8-digit direct designation).

The F8 digits must consist of five integral digits and three decimal digits, with the decimal point.

Cutting feed rates become valid for commands G01, G02, G03, G32 and G34.

Example:

Asynchronous feed	Feed rate
G01 X100. Z100. F200*;	200.0 mm/min
G01 X100. Z100. F123.4;	123.4 mm/min
G01 X100. Z100. F56.789;	56.789 mm/min

* It means the same if F200. or F200.000 is set in stead of F200.

Note: An alarm (No. 713) will result if a feed rate command is not set for the first cutting command (G01, G02, G03, G32 or G34) that is read firstly after power-on.

7-3 Asynchronous/Synchronous Feed: G98/G99 [Series M: G94/G95]

1. Function and purpose

Command G99 allows a feed rate per revolution to be set using an F-code.

To use this command, a rotational encoder must be mounted on the spindle.

2. Programming format

G98: Feed per minute (/min) [Asynchronous feed]

G99: Feed per revolution (/rev) [Synchronous feed]

Since the command G99 is modal command, it will remain valid until the command G98 is issued.

3. Detailed description

1. Feed rates that can be set using F-codes are listed in the table below.
   The table below also lists synchronous feed rates, which are to be set in millimeters (or inches) per spindle revolution using F-codes.

	G98F_ (Feed per minute)	G99F_ (Feed per revolution)
Input in mm	1 to 240000 mm/min (F1 to F240000)	0.0001 to 500.0000 mm/rev (F1 to F5000000)
Input in inches	0.01 to 9600.00 in./min (F1 to F960000)	0.000001 to 9.999999 in./rev (F1 to F9999999)

2. The effective feed rate per revolution, that is, the actual moving speed of the machine, can be calculated as follows:

   FC = F × N × OVR (Expression 1)

   where FC: Effective feed rate (mm/min or inches/min)
   F: Designated feed rate (mm/rev or inches/rev)
   N: Spindle speed (rpm)
   OVR: Cutting feed override

   If multiple axes are selected at the same time, effective feed rate FC given by expression 1 above will become valid for the corresponding vectorial direction.

4. Remarks

1. An effective feed rate that is expressed in a feed rate per minute (mm/min or inches/min) is displayed on the **POSITION** display.
2. If the effective feed rate is larger than the cutting feed clamping speed, that clamping speed will become valid.
3. During machine lock high-speed processing, the feed rate is 60000 mm/min (or 2362 inches/min, 60000 deg/min) regardless of the commanded speed and spindle speed. When high-speed processing is not undertaken, the feed rate is the same as for non-machine lock conditions.
4. In the dry run mode, feed will become asynchronous and the machine will operate at an externally preset feed rate (mm/min or inches/min).
5. According to the setting of bit 1 of parameter **F93**, synchronous or asynchronous feed mode (G99 or G98) is automatically made valid upon power-on or by execution of M02 or M30.

7-4 Selecting a Feed Rate and Effects on Each Control Axis

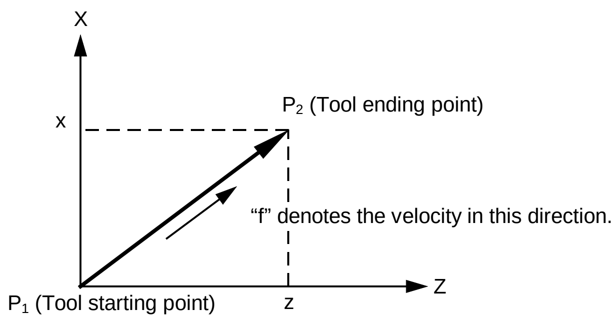
As mentioned earlier, the machine has various control axes. These control axes can be broadly divided into linear axes, which control linear motions, and rotational axes, which control rotational motions. Feed rates for control axes have different effects on the tool speed, which is of great importance for machining quality, according to the particular type of axis controlled.

The amount of displacement must be designated for each axis, whereas the feed rate is to be designated as a single value for the intended tool movement. Before letting the machine control two or more axes at the same time, therefore, you must understand how the feed rate designated will act on each axis. In terms of this, selection of a feed rate is described below.

1. Controlling linear axes

The feed rate that has been selected using an F-code acts as a linear velocity in the moving direction of the tool, irrespective of whether only one axis is to be controlled or multiple axes simultaneously.

Example: If linear axes (X- and Z-axes) are to be controlled using a feed rate of f:



When only linear axes are to be controlled, setting of a cutting feed rate itself is only required. The feed rate for each axis refers to that component of the specified feed rate which corresponds with the ratio of movement stroke on the respective axis to the actual movement distance.

In the example shown above:

$$\text{X-axis feed rate} = f \times \frac{x}{\sqrt{x^2 + z^2}}$$

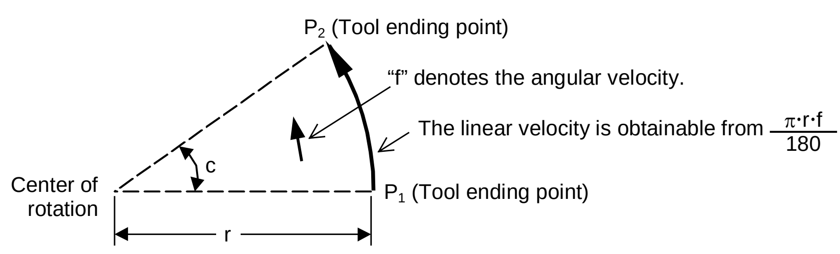
$$\text{Z-axis feed rate} = f \times \frac{z}{\sqrt{x^2 + z^2}}$$

2. Controlling a rotational axis

When a rotational axis is to be controlled, the selected feed rate acts as the rotating speed of the rotational axis, that is, as an angular velocity.

Thus, the cutting speed in the moving direction of the tool, that is, a linear velocity varies according to the distance from the rotational center to the tool. This distance must be considered when setting a feed rate in the program.

Example 1: If a rotational axis (C-axis) is to be controlled using a feed rate of f (deg/min):



TEP034

In this case, the cutting speed in the moving direction of the tool (linear velocity) "fc" is calculated by:

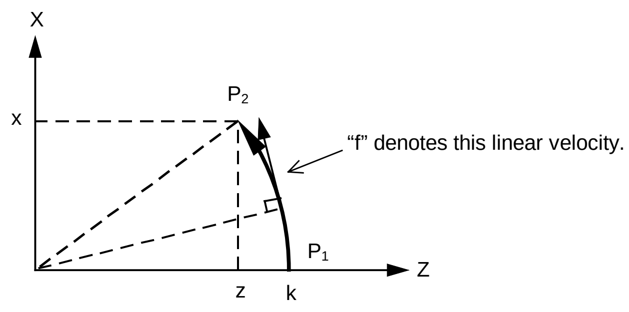
$$fc = f \times \frac{\pi \bullet r}{180}$$

Hence, the feed rate to be programmed for the required value fc is:

$$f = fc \times \frac{180}{\pi \bullet r}$$

Note: If the tool is to be moved by controlling linear axes along the circumference using the circular interpolation function, the feed rate programmed is the velocity acting in the moving direction of the tool, that is, in the tangential direction.

Example 2: If linear axes (X- and Z-axes) are to be controlled at a feed rate of f using the circular interpolation function:



TEP036

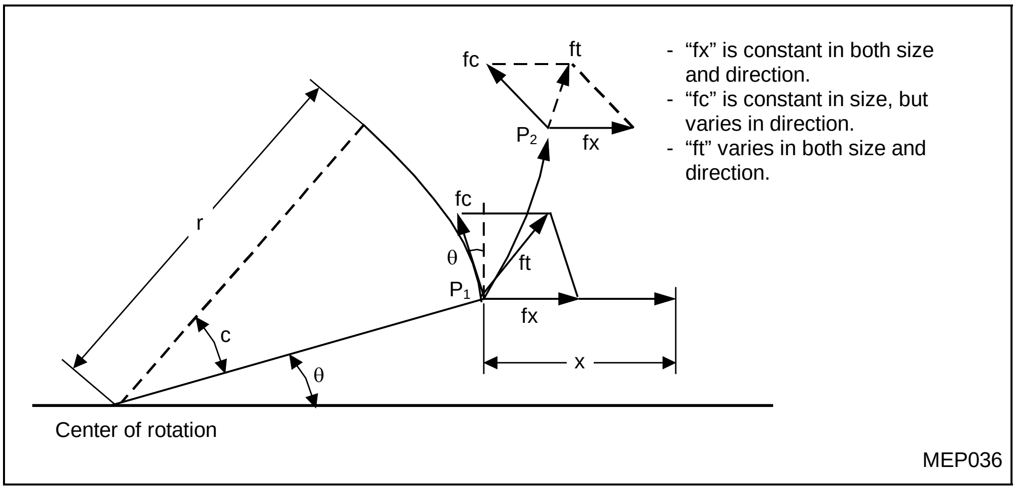
In this case, the X- and Z-axis feed rates will change with the movement of the tool. The resultant velocity, however, will be kept at the constant value, f.

Return to Library

3. Controlling a linear axis and a rotational axis at the same time

The NC unit controls linear axes and rotational axes in exactly the same manner.
For control of rotational axes, data given as a coordinate word (C or H) is handled as an angle, and data given as a feed rate (F) is handled as a linear velocity. In other words, an angle of one degree for a rotational axis is handled as equivalent to a moving distance of 1 mm for a linear axis. Thus, for simultaneous control of a linear axis and a rotational axis, the magnitudes of the individual axis components of the data that has been given by F are the same as those existing during linear axis control described previously in Subparagraph 1. above. In this case, however, the velocity components during linear axis control remain constant in both magnitude and direction, whereas those of rotational axis control change in direction according to the movement of the tool. Therefore, the resulting feed rate in the moving direction of the tool changes as the tool moves.

Example: If a linear axis (X-axis) and a rotational axis (C-axis) are to be controlled at the same time at a feed rate of f:



X-axis incremental command data is expressed here as x, and that of C-axis as c.
The X-axis feed rate (linear velocity), fx, and the C-axis feed rate (angular velocity), ω, can be calculated as follows:

$$fx = f \times \frac{x}{\sqrt{x^2 + c^2}} \cdots\cdots [1] \qquad \omega = f \times \frac{c}{\sqrt{x^2 + c^2}} \cdots\cdots [2]$$

The linear velocity "fc" that relates to C-axis control is expressed as:

$$fc = \omega \bullet \frac{\pi \bullet r}{180} \cdots\cdots [3]$$

If the velocity in the moving direction of the tool at starting point P_1 is taken as "ft", and its X- and Y-axis components as "ftx" and "fty" respectively, then one can express "ftx" and "fty" as follows:

$$ftx = -r \sin\left(\frac{\pi}{180}\theta\right) \times \frac{\pi}{180}\omega + fx \cdots\cdots [4]$$

$$fty = -r \cos\left(\frac{\pi}{180}\theta\right) \times \frac{\pi}{180}\omega \cdots\cdots [5]$$

where r denotes the distance (in millimeters) from the rotational center to the tool, and q denotes the angle (in degrees) of starting point P_1 to the X-axis at the rotational center.

From expressions [1] through [5] above, the resultant velocity “ft” is:

$$ft = \sqrt{ftx^2 + fty^2}$$

$$= f \times \frac{\sqrt{x^2 - x \bullet c \bullet r \sin\left(\frac{\pi}{180}\theta\right)\frac{\pi}{90} + \left(\frac{\pi \bullet r \bullet c}{180}\right)^2}}{\sqrt{x^2 + c^2}} \quad \cdots\cdots\cdot [6]$$

The feed rate f that is to be set in the program must be therefore:

$$f = ft \times \frac{\sqrt{x^2 + c^2}}{\sqrt{x^2 - x \bullet c \bullet r \sin\left(\frac{\pi}{180}\theta\right)\frac{\pi}{90} + \left(\frac{\pi \bullet r \bullet c}{180}\right)^2}} \quad \cdots\cdots\cdot [7]$$

In expression [6], “ft” is the velocity at starting point P_1 and thus the value of ft changes with that of θ which changes according to the rotational angle of the C-axis. To keep cutting speed “ft” as constant as possible, the rotational angle of the C-axis in one block must be minimized to ensure a minimum rate of change of θ.

7-5 Threading Leads

The thread lead in the threading mode (G32, G34, G76 or G92) can be designated using a seven-digit value preceded by address F or eight-digit value preceded by address E.
The thread lead command range is 0.0001 to 999.9999 mm/rev (F with 7 digits) or 0.0001 to 999.99999 mm/rev (E8-digit) (with unit of data setting of microns).

Thread cutting (metric input)

Unit of program data input	0.0001 mm		
Command address	F (mm/rev)	E (mm/rev)	E (Number of threads per inch)
Unit of minimum data setting		1 (=0.0001) (1.=1.0000)	1 (=1) (1.=1.00)
Range of command data	0.0001 to 500.0000	0.0001 to 999.9999	0.01 to 9999999.9

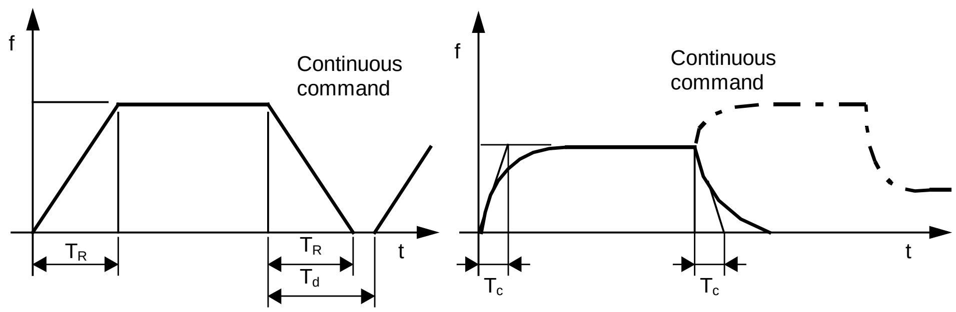
Thread cutting (inch input)

Unit of program data input	0.000001 inch		
Command address	F (in./rev)	E (in./rev)	E (Number of threads per inch)
Unit of minimum data setting		1 (=0.000001) (1.=1.000000)	1 (=1) (1.=1.0000)
Range of command data	0.000001 to 9.999999	0.000001 to 99.999999	0.0001 to 9999.9999

Return to Library

7-6 Automatic Acceleration/Deceleration

The rapid traverse and manual feed acceleration/deceleration pattern is linear acceleration and linear deceleration. Time constant T_R can be set independently for each axis using parameters in 1 msec steps within a range from 1 to 500 msec. The cutting feed (not manual feed) acceleration/deceleration pattern is exponential acceleration/deceleration. Time constant T_C can be set independently for each axis using parameters in 1 msec steps within a range from 1 to 500 msec. (Normally, the same time constant is set for each axis.)



Rapid feed acceleration/deceleration pattern
(T_R = Rapid feed time constant)
(T_d = Deceleration check time)

Cutting feed acceleration/deceleration pattern
(T_c = Cutting feed time constant)

TEP037

During rapid traverse and manual feed, the following block is executed after the command pulse of the current block has become "0" and the tracking error of the acceleration/deceleration circuit has become "0". During cutting feed, the following block is executed as soon as the command pulse of the current block becomes "0" and also the following block can be executed when an external signal (error detection) can detect that the tracking error of the acceleration/deceleration circuit has reached "0". When the in-position check has been made valid (selected by machine parameter) during the deceleration check, it is first confirmed that the tracking error of the acceleration/deceleration circuit has reached "0", then it is checked that the position deviation is less than the parameter setting, and finally the following block is executed.

7-7 Speed Clamp

This function exercises control over the actual cutting feed rate in which override has been applied to the cutting feed rate command so that the speed clamp value preset independently for each axis is not exceeded.

Note: Speed clamping is not applied to synchronous feed and threading.

Return to Library

7-8 Exact-Stop Check Command: G09

1. Function and purpose

Only after the in-position status has been checked following machine deceleration and stop or after deceleration checking time has been passed, may you want to start the next block command in order to reduce possible machine shocks due to abrupt changes in tool feed rate and to minimize any rounding of workpieces during corner cutting. An exact-stop check function is provided for these purposes.

2. Programming format

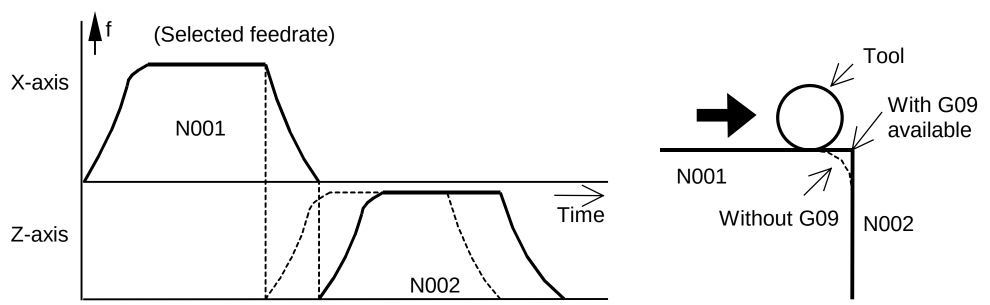
G09 G01 (G02, G03) ;

Exact-stop check command G09 is valid only for the cutting command code (G01, G02, or G03) that has been set in that block.

3. Sample program

```
N001 G09 G01  X100.000 F150;
N002          Z100.000 ;
```

N001: The next block is executed after an in-position status check following machine deceleration and stop.



The solid line indicates a feedrate pattern with the G09 available.
The dotted line indicates a feedrate pattern without the G09.

TEP038

Fig. 7-1 Validity of exact-stop check

Return to Library

4. Detailed description

A. Continuous cutting feed commands

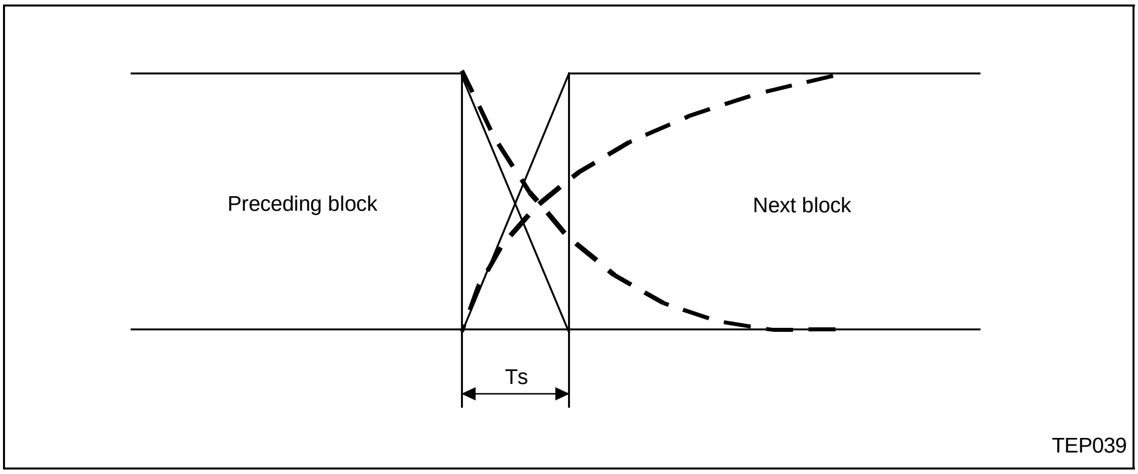


Fig. 7-2 Continuous cutting feed commands

B. Cutting feed commands with in-position status check

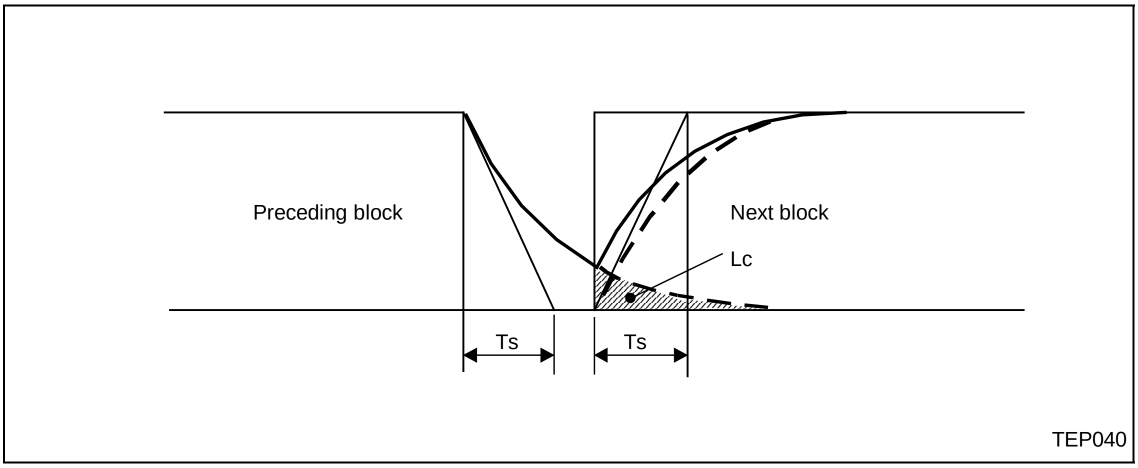


Fig. 7-3 Block-to-block connection in cutting feed in-position status check mode

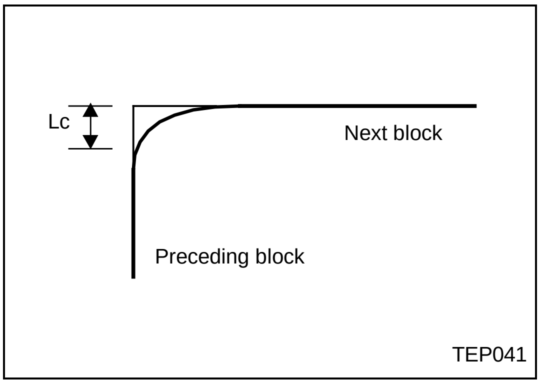
In Fig. 7-2 and 7-3 above,

Ts: Cutting feed acceleration/deceleration time constant
Lc: In-position width

As shown in Fig. 7-3, in-position width Lc represents the remaining distance within the block immediately preceding the next block to be executed.

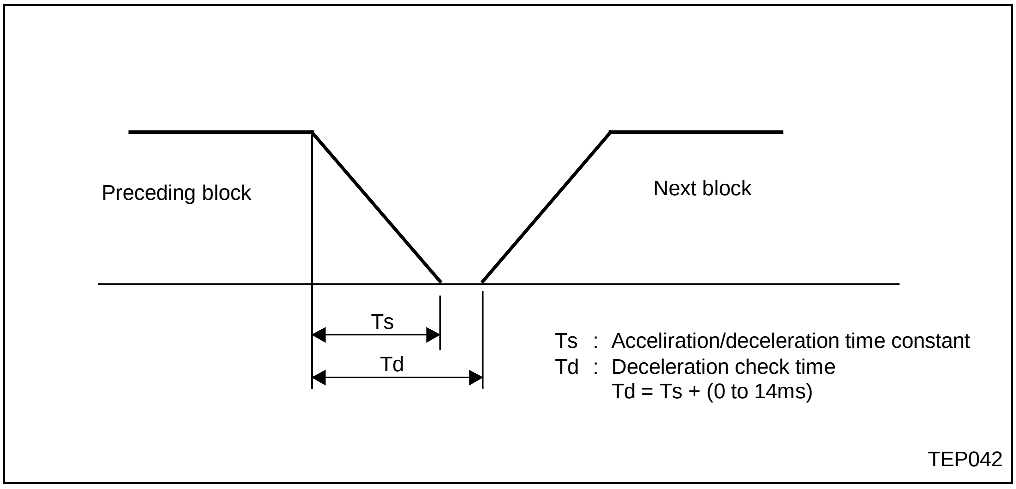
The in-position width helps keep any rounding of workpieces during corner cutting within a fixed level.

If rounding of workpieces at corners is to be completely suppressed, include dwell command G04 between cutting blocks.

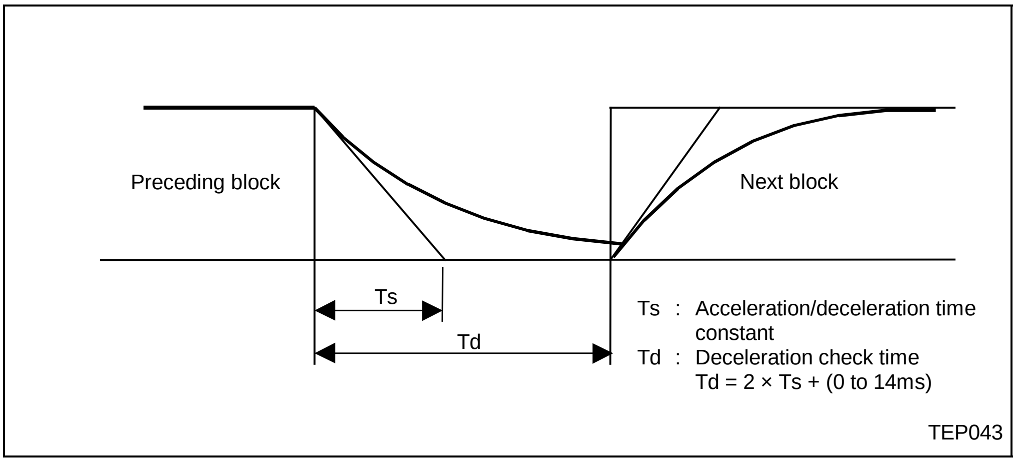


C. With deceleration check

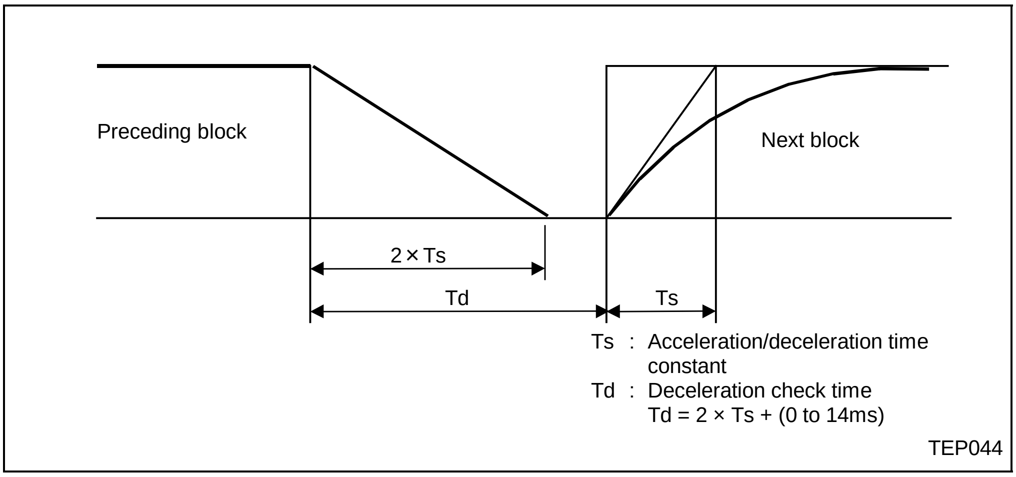
- With linear acceleration/deceleration



- With exponential acceleration/deceleration



- With exponential acceleration/linear deceleration



The time required for the deceleration check during cutting feed is the longest among the cutting feed deceleration check times of each axis determined by the cutting feed acceleration/deceleration time constants and by the cutting feed acceleration/ deceleration mode of the axes commanded simultaneously.

7-9 Exact-Stop Check Mode Command: G61

1. Function and purpose

Unlike exact-stop check command G09 which performs an in-position status check on that block only, command G61 functions as a modal command. That is, this command acts on all its succeeding cutting commands (G01, G02, and G03) so that deceleration occurs at the end of each block, followed by an in-position status check. This command is cleared by automatic corner override command G62 or cutting mode command G64.

2. Programming format

G61;

7-10 Automatic Corner Override Command: G62

1. Function and purpose

Command G62 automatically overrides in the tool-diameter offset mode the selected feed rate to reduce the tool load during inner-corner cutting or automatic inner-corner rounding.
Once this command has been issued, the automatic corner override function will remain valid until it is cancelled by tool-diameter offsetting cancellation command G40, exact-stop check mode command G61, or cutting mode command G64.

2. Programming format

G62 ;

3. Detailed description

A. Inner-corner cutting

When inner corner of a workpiece is cut as shown in the figure below, the load on the tool increases because of large amount of cutting. Using G62 in such a case allows the cutting feed rate to be automatically overriden within the preset zone, and thus the tool load to be reduced to accomplish appropriate cutting.
This function, however, is valid only for programming the as-finished shape of a workpiece.

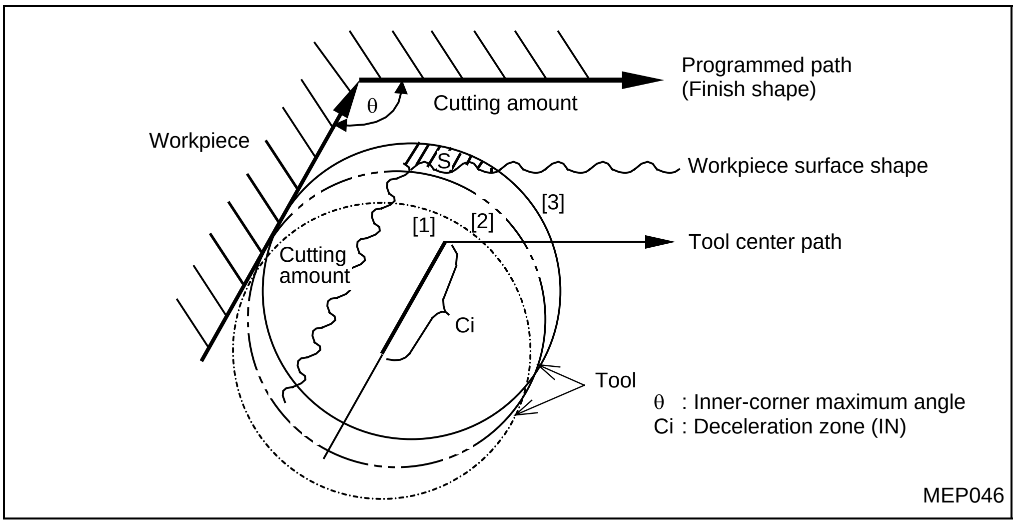


Fig. 7-4 Inner-corner cutting

<Machine operation>

- When the automatic corner override function is not used:
  In the figure above, as the tool is moving in order of positions [1]→[2]→[3], the load on the tool increases because the cutting amount at position [3] is larger than that of position [2] by the area of hatched section S.
- When the automatic corner override function is used:
  In the figure above, if maximum angle q of the inner corners is smaller than that preset in the appropriate parameter, the feed rate is automatically overriden with the preset value for movement through deceleration zone Ci.

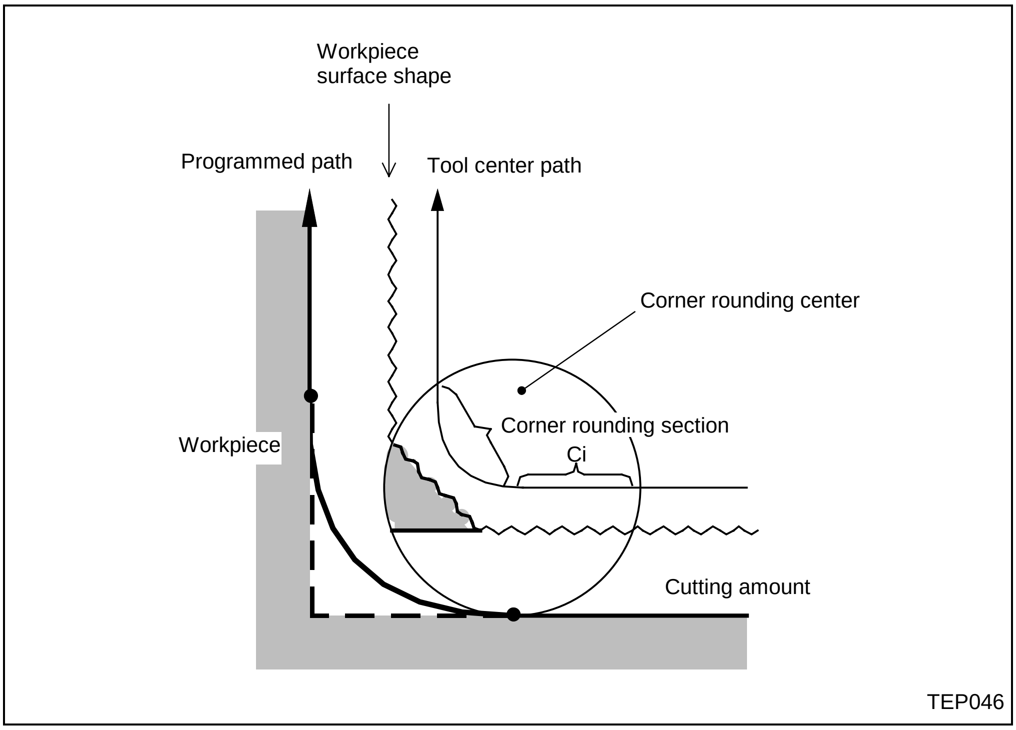
<Setting parameters>

Set the following parameters as user parameters:

- **E22**: Override	0 to 100 (%)
- **F21**: Inner-corner maximum angle θ	0 to 180 (deg)
- **F22**: Deceleration zone Ci data	0 to 99999.999 (mm) or to 3937.000 (inches)

For further details of parameter setting, refer to the description in the Operating manual and the Parameter list.

B. Automatic corner rounding



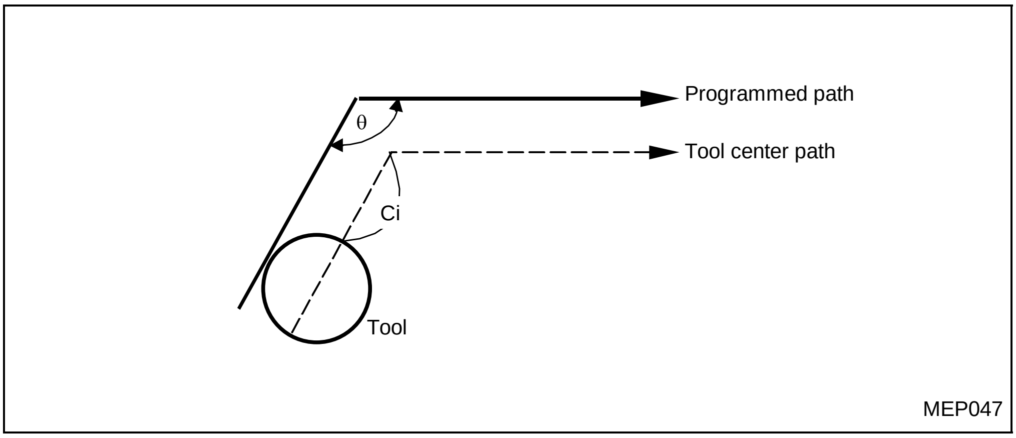
<Operation>

For inner corner cutting with automatic corner rounding, override will be effected as set in parameter through the deceleration zone Ci and corner rounding section (No check made about angle).

Return to Library

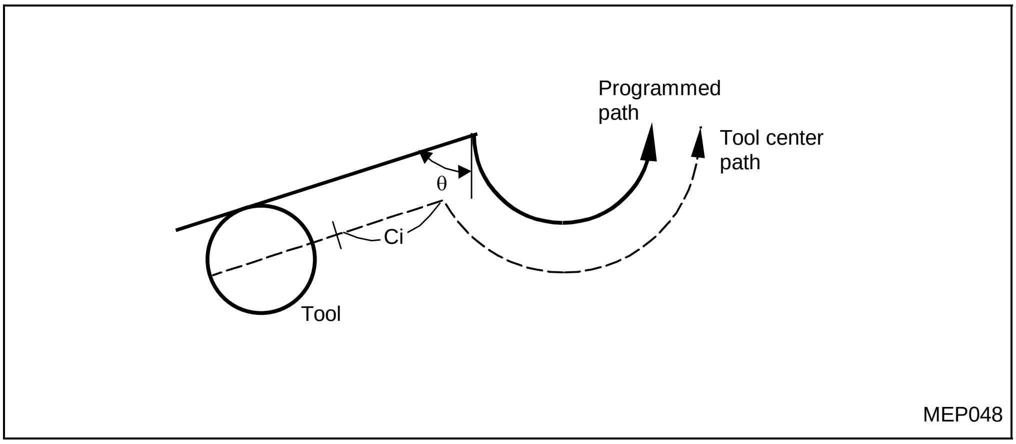
4. Operation examples

- Line-to-line corner



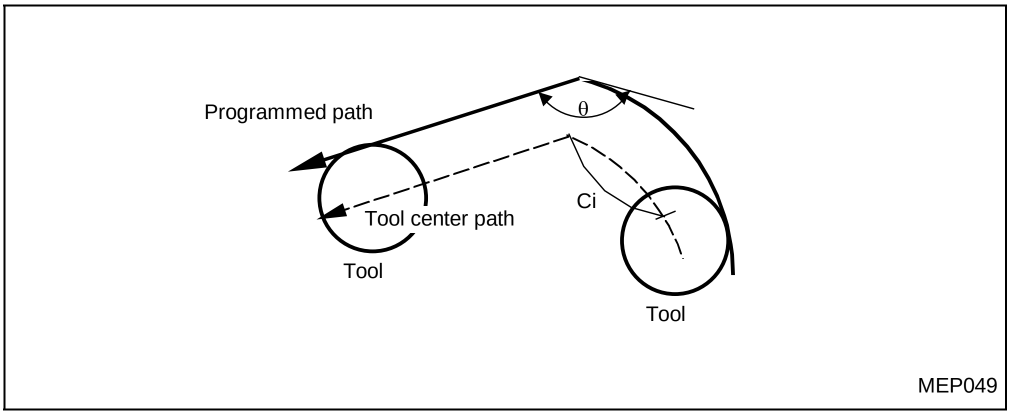
The feed rate is automatically overridden with the preset value by the parameter **E22** through deceleration zone Ci.

- Line-to-circular (outside offsetting) corner



The feed rate is automatically overridden with the preset value by the parameter **E22** through deceleration zone Ci.

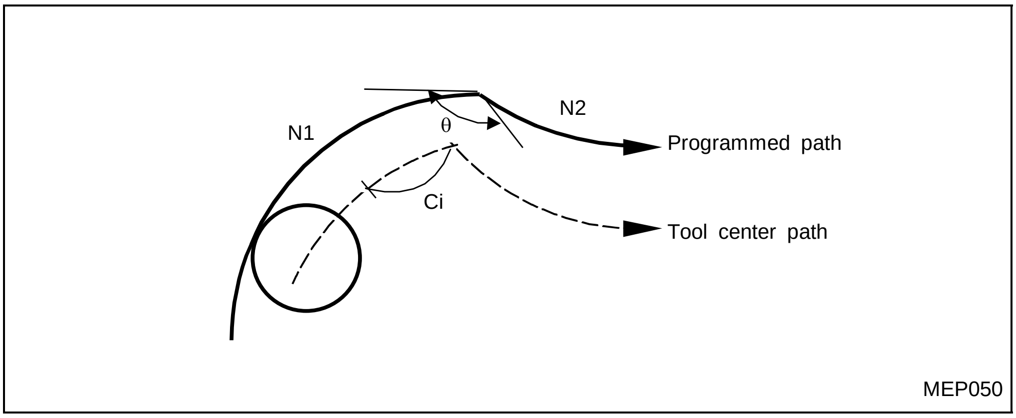
- Arc(internal compensation)-to-line corner



The feed rate is automatically overridden with the preset value by the parameter **E22** through deceleration zone Ci.

Note: Data of deceleration zone Ci at which automatic overriding occurs represents the length of the arc for a circular interpolation command.

- Arc(internal compensation)-to-arc (external compensation) corner



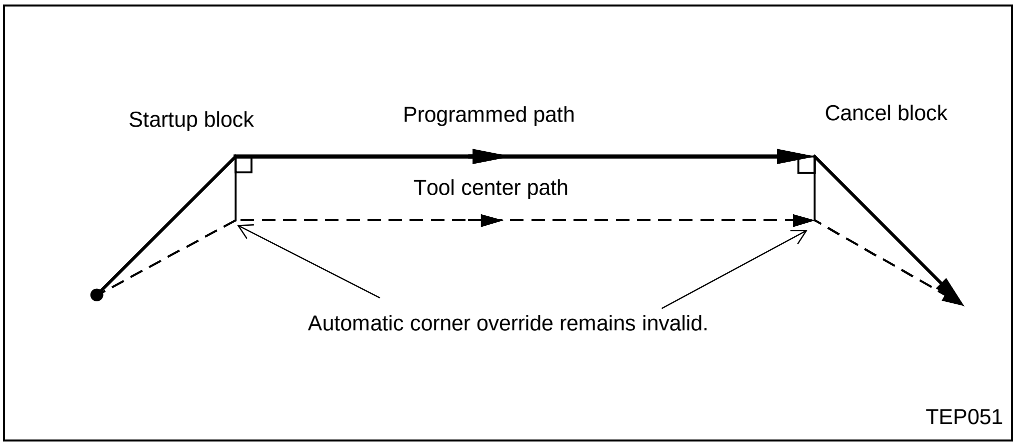
The feed rate is automatically overridden with the preset value by the parameter **E22** through deceleration zone Ci.

5. Correlationships to other command functions

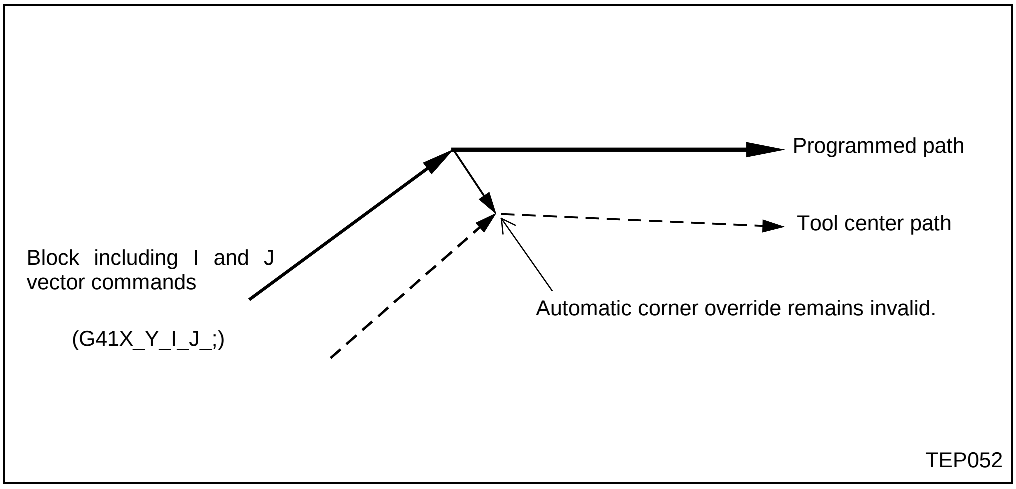
Function	Override at corners
Cutting feedrate override	Automatic corner override is applied after cutting feed override.
Override cancel	Automatic corner override is not cancelled by override cancel.
Feed rate clamp	Valid (for the feed rate after automatic corner override)
Dry run	Automatic corner override is invalid.
Synchronous feed	A synchronous feed rate is automatically corner-overridden.
Skip (G31)	During tool-diameter offset, G31 will result in a program error.
Machine lock	Valid
G00	Invalid
G01	Valid
G02, G03	Valid

6. Precautions

1. Automatic corner override is valid only during the G01, G02 or G03 modes; it is invalid during the G00 mode. Also, when the command mode is changed over from G00 to G01, G02, or G03 (or vice versa) at a corner, automatic corner override is not performed on the G00-containing block at that corner.
2. Even in the automatic corner override mode, automatic corner override is not performed until the tool diameter compensation mode has been set.
3. Automatic corner override does not occur at corners where tool diameter compensation is to start or to be cancelled.



4. Automatic corner override does not occur at corners where tool diameter compensation I, J and K vector commands are to be executed.



5. Automatic corner override occurs only when crossing points can be calculated. Crossing points can not be calculated in the following case:

   - Four or more blocks that do not include move command appear in succession.

6. For circular interpolation, the deceleration zone is represented as the length of the arc.
7. The parameter-set angle of an inner corner is applied to the angle existing on the programmed path.
8. Setting the maximum angle to 0 or 180 degrees in the angle parameter results in an automatic corner override failure.
9. Setting the override to 0 or 100 in the override parameter results in an automatic corner override failure.

7-11 Cutting Mode Command: G64

1. Function and purpose

Command G64 enters the NC unit into a control mode proper to obtain smoothly cut surfaces. Unlike the exact-stop check mode (G61 command mode), the cutting mode allows the next block to be executed without decelerating/stopping the machine between cutting feed blocks.
The G64 command mode is cleared by exact-stop check mode command G61 or automatic corner override command G62.
In the initial state of the NC unit, the cutting mode is selected.

2. Programming format

G64 ;

7-12 Geometry Compensation/Accuracy Coefficient: G61.1/,K

7-12-1 Geometry compensation function: G61.1

1. Function and purpose

The geometry compensation function (G61.1) is provided to reduce conventional geometry errors caused by delayed follow-up of smoothing circuits and servo systems.
The geometry compensation function is canceled, or replaced, by the functions of exact stop mode (G61), automatic corner override (G62) and cutting mode (G64).

The geometry compensation function is composed of the following four functions:

1. Pre-interpolation acceleration/deceleration
2. Feed forward control
3. Optimum corner deceleration
4. Precise vector compensation

Refer to Section 11-2 “Geometry Compensation Function” in Chapter 3 of the Operating Manual for the description of the above functions.

2. Programming format

G61.1;

3. Sample program

```
N001 G0X100.Z100.
     G61.1G01F2000        Selection of the geometry compensation function
     U10.W30.
     U5.W30.
     U-5.W30.
     U-10.W10.
     U-30.W5.
     G64                  Cancellation of the geometry compensation function
```

4. Remarks

1. The geometry compensation function cannot be selected or canceled for EIA/ISO programs by the setting of the parameter **F72** (which is only effective for MAZATROL programs).
2. The geometry compensation is an optional function. On machines without corresponding option the code G61.1 can only lead to an alarm (**808 MIS-SET G CODE**).
3. The geometry compensation function is suspended during execution of the following operations:
   Rapid traverse of non-interpolation type (according to bit 6 of parameter **F91**), Synchronous tapping, Measurement (skipping), Constant peripheral speed control, Threading.
4. The pre-interpolation acceleration/deceleration is effective from the block of G61.1 onward.

7-12-2 Accuracy coefficient (,K)

1. Function and purpose

In the mode of geometry compensation (G61.1) the feed of the tool is automatically decelerated at relevant corners and for circular motions by the optimal corner deceleration and the circular feed limitation, respectively, in order to enhance the machining accuracy. Specifying an accuracy coefficient in the machining program can further improve the accuracy by additionally decelerating the feed for the sections concerned.

2. Programming format

,K_; Specify the rate of reduction of the corner deceleration speed and the circular feed rate limitation in percentage terms.

The accuracy coefficient is canceled in the following cases:

- Resetting is performed,
- The geometry compensation function is canceled (by G64),
- A command of ",K0" is given.

3. Sample program

<Example 1>

```
N001  G61.1
N200  G1U_W_,K30  ←——  The rate of feed for a corner deceleration or circular motion in the section from
                       this block onward will be reduced to 70% of the value applied in default of the
                       accuracy coefficient command.
N300    U_W_
N400  ···
```

<Example 2>

```
N001  G61.1
N200  G2I-10.,K30←——  Deceleration to 70% occurs for this block only.
N300  G1U10.,K0  ←——  The accuracy coefficient is canceled from this block onward.
N400  ···
```

4. Remarks

1. The accuracy coefficient cannot be specified in a MAZATROL program.
2. Specifying an accuracy coefficient 1 to 99 at address ",K" increases the machining time according to the additional deceleration at relevant corners and for circular motions.

Return to Library

- NOTE -

8 DWELL FUNCTIONS

The start of execution of the next block can be delayed using a G04 command.

8-1 Dwell Command in Time: (G98) G04 [Series M: (G94) G04]

1. Function and purpose

Setting command G04 in the feed-per-second mode (command G98) delays the start of execution of the next block for the specified time.

2. Programming format

G98 G04 X/U_;
or
G98 G04 P_;

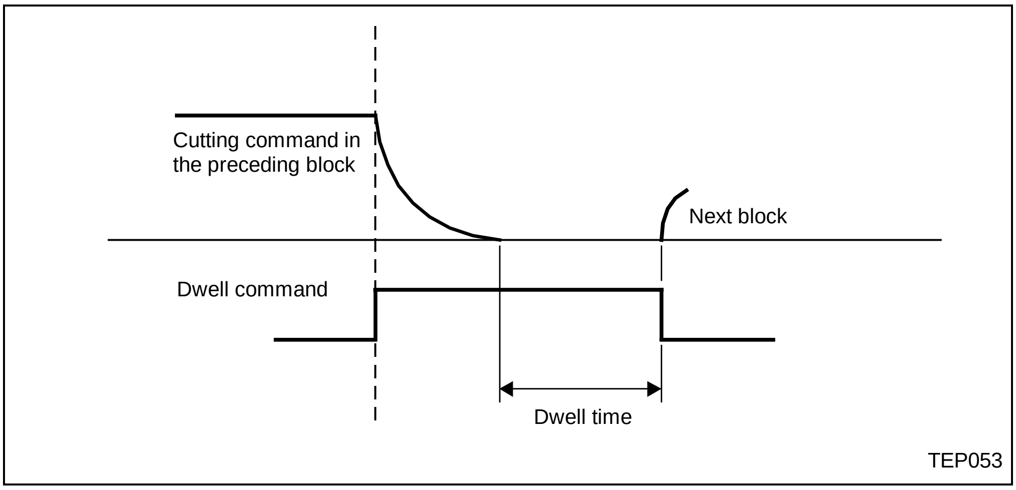
Data must be set in 0.001 seconds.
For address P, the decimal point is not available. Setting a decimal point will cause an alarm.

3. Detailed description

1. The setting range for dwell time is as follows:

Unit of data setting	Range for address X or U	Range for address P
0.001 mm, 0.0001 inches	0.001 to 99999.999 (sec)	1 to 99999999 (× 0.001 sec)

2. The count for the dwell command which is preceded by a block with cutting-feed command is not started until the movement of the preceding block has been brought to a complete stop.



If the dwell command is given in one block together with an M-, S- T- or B-code, the dwell count and the execution of the respective code will be started at the same time.

3. If the bit 2 of parameter **F92** is set to 1, dwell command value is always processed in time specification irrespective of G98 and G99 modes.

4. Sample programs

- When data is to be set in 0.01 mm, 0.001 mm or 0.0001 inches:

  G04 X 500 ;...Dwell time = 0.5 sec
  G04 X 5000 ;.......................................Dwell time = 5.0 sec
  G04 X 5. ;..Dwell time = 5.0 sec
  G04 P 5000 ;.......................................Dwell time = 5.0 sec
  G04 P 12.345 ; ....*Alarm*

- When data is to be set in 0.0001 inches and dwell time is included before G04:

  X5. G04 ;...Dwell time = 50 sec (Equivalent to X50000G04.)

8-2 Dwell Command in Number of Revolutions: (G99) G04 [Series M: (G95) G04]

1. Function and purpose

Setting command G04 in the feed-per-revolution mode (command G99) suspends the start of execution of the next block until the spindle has rotated the specified number of revolutions.

2. Programming format

G99 G04 X/U_ ;
or
G99 G04 P_ ;

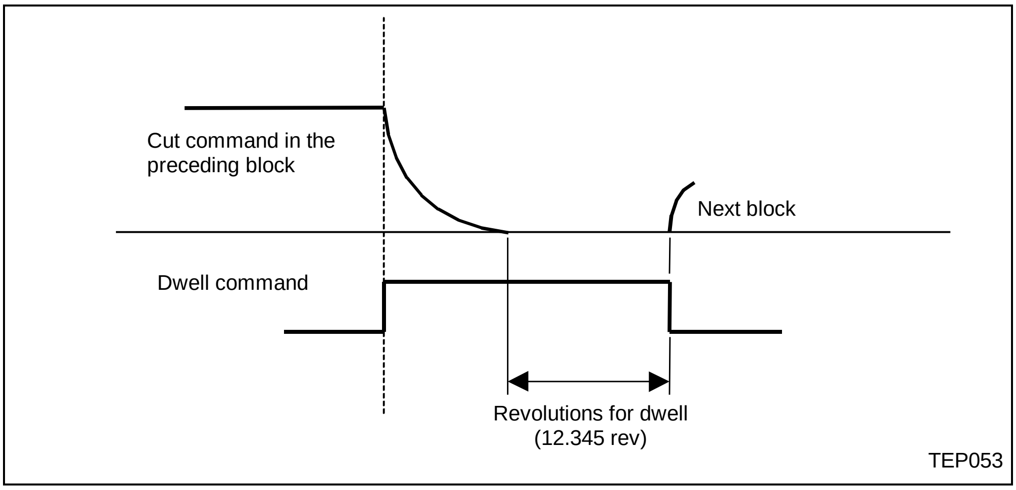
Data must be set in 0.001 revolutions.
For address P, the decimal point is not available. Setting a decimal point will cause an alarm.

3. Detailed description

1. The setting range for number of dwell revolutions is as follows:

Unit of data setting	Range for address X or U	Range for address P
0.001 mm, 0.0001 inches	0.001 to 99999.999 (rev)	1 to 99999999 (× 0.001 rev)

2. The count for the dwell command which is preceded by a block with cutting-feed command is not started until the movement of the preceding block has been brought to a complete stop.



If the dwell command is given in one block together with an M-, S- T- or B-code, the dwell count and the execution of the respective code will be started at the same time.

3. The dwell function is also valid during the machine lock mode.

4. During rest of the spindle, dwell count is also halted. When the spindle restarts rotating, dwell count will also restart.
5. If the bit 2 of parameter **F92** is set to 1, dwell command value is alway processed in time specification.
6. This function cannot be used unless the position detecting encoder is provided to the spindle.

- NOTE -

9 MISCELLANEOUS FUNCTIONS

9-1 Miscellaneous Functions (M3-Digit)

Miscellaneous functions, which are also referred to as M-code functions, give spindle forward/ backward rotation and stop commands, coolant on/off commands, and other auxiliary commands to the NC machine.
For the NC unit, these functions must be selected using M3-digit data (three-digit data preceded by address M). Up to four sets of M3-digit data can be included in one block.

Example: G00 Xx_1 Mm_1 Mm_2 Mm_3 Mm_4;

If five or more sets of M3-digit data are set, only the last four sets will become valid.
Refer to the machine specification for more specific relationships between available data and functions.
For M-codes M00, M01, M02, M30, M98, M99, M998 and M999, the next block of data is not read into the input buffer since pre-reading is disabled automatically.
The M-codes can be included in any block that contains other command codes. If, however, the M-codes are included in a block that contains move commands, then the execution priority will be either

- the M-code functions are executed after completion of movement, or
- the M-code functions are executed together with movement.

It depends on the machine specifications which type of processing is applied.
Processing and completion sequences are required in each case for all M commands except M98 and M99.
The following lists six types of special M-code functions:

1. Program Stop: M00
   When this M-code is read, the tape reader will stop reading subsequent block. Whether the machine function such as spindle rotation and coolant will also stop depends on the machine specifications.
   The machine operation is restarted by pressing the cycle start button on the operation panel.
   Whether resetting can be initiated by M00 or not also depends on the machine specifications.

2. Optional Stop: M01
   When the M01 code is read with the **[OPTIONAL STOP]** menu function set to ON, the tape reader will stop operating to perform the same function as M00.
   The M01 command will be ignored if the **[OPTIONAL STOP]** menu function is set to OFF.

Example:

```
 ⋮
N10  G00 X1000;
N11  M01;
N12  G01 X2000 Z3000 F600;
 ⋮
```

<**[OPTIONAL STOP]** menu function status and operation>
If the menu function is on, operation stops at N11.
If the menu function is off, operation does not stop at N11 and N12 is executed.

3. Program End: M02 or M30
   Usually, the program end command is given in the final block of machining program. Use this command mainly for reading data back to the head of the program during memory operation, or rewinding the tape. The NC unit is automatically reset after tape rewinding and

execution of other command codes included in that block.
Automatic resetting by this command cancels both modal commands and offsetting data, but the designated-position display counter is not cleared to zero.
The NC unit will stop operating when tape rewinding is completed (the automatic run mode lamp goes out). To restart the NC unit, the cycle start button must be pressed.
Beware that if, during the restart of the NC unit following completion of M02 or M30 execution, the first movement command has been set in a coordinate word only, the valid mode will be the interpolation mode existing when the program ended. It is recommended, therefore, that the first movement command be given with an appropriate G-code.

4. Subprogram Call/End: M98, M99
   Use M98 or M99 to branch the control into a subprogram or to recall it back to the calling program.
   As M99 and M99 are internaly processed by the NC M-code signals ans strobe signals are not output.

<Internal processing by the NC unit when M00, M01, M02 or M30 is used>
After M00, M01, M02 or M30 has been read, data pre-reading is automatically aborted. Other tape rewinding operations and the initialization of modals by resetting differ according to the machine specification.

Note 1: M00, M01, M02 and M30 output independent signals, which will be cancelled by pressing the RESET key.

Note 2: Tape rewinding is performed only when the tape reader has a rewinding function.

9-2 No. 2 Miscellaneous Functions (A8/B8/C8-Digit)

The No. 2 miscellaneous functions are used for positioning an index table. For the NC unit, these functions must be designated using an eight-digit value (form 0 to 99999999) preceded by address A, B or C.
The output signals are BCD signals of command data and start signals.
A, B or C codes can be included in any block that contains other command codes. If, however, the A, B or C codes can be included in a block that contains move commands, then the execution priority will be either

- the A, B or C code functions are performed after completion of movement, or
- the A, B or C code functions are performed together with movement.

It depends on the machine specifications which type of processing is applied.
Processing and completion sequences are required in each case for all No. 2 miscellaneous functions.

Address combinations are shown below. The same address for both additional axis and the No. 2 miscellaneous functions cannot be used.

Additional axis / No. 2 miscellaneous functions	A	B	C
A	×	○	○
B	○	×	○
C	○	○	×

Note: When A has been designated as the No. 2 miscellaneous function address, linear angle commands cannot be used.

10 SPINDLE FUNCTIONS

10-1 Spindle Function (S5-Digit Analog)

When the S5-digit function is added, this function must be set using the numerical command of five digits preceding an S code (0 to 99999) and for other case, two digits preceding by an S code is used.
S command binary outputs must be selected at this time.
By designating a 5-digit number following the S code, this function enables the appropriate gear signals, voltages corresponding to the commanded spindle speed (rpm) and start signals to be output.
Processing and completion sequences are required for all S commands.
The analog signal specifications are given below.

- Output voltage .... 0 to 10V or –8 to +8V
- Resolution... 1/4096 (2 to the power of –12)
- Load conditions .... 10 kiloohms
- Output impedance .... 220 ohms

If the parameters for up to 4 gear range steps are set in advance, the gear range corresponding to the S command will be selected by the NC unit and the gear signal will be output. The analog voltage is calculated in accordance with the input gear signal.

- Parameters corresponding to individual gears... Limit speed, maximum speed, gear shift speed and maximum speed during tapping.
- Parameters corresponding to all gears .... Orient speed, minimum speed

10-2 Constant Peripheral Speed Control ON/OFF: G96/G97

1. Function and purpose

This function controls automatically the spindle speed as the coordinates are changed during cutting in diametral direction so as to execute cutting by keeping constant the relative speed between tool tip and workpiece.

2. Programming format

G96 Ss Pp Rr; ..... Constant peripheral speed control ON

s: Axis for constant peripheral speed control
p: Peripheral speed
r: Spindle for constant peripheral speed control

G97; .... Constant peripheral speed control OFF

3. Detailed description

1. Axis for constant peripheral speed control is to be set by address P.

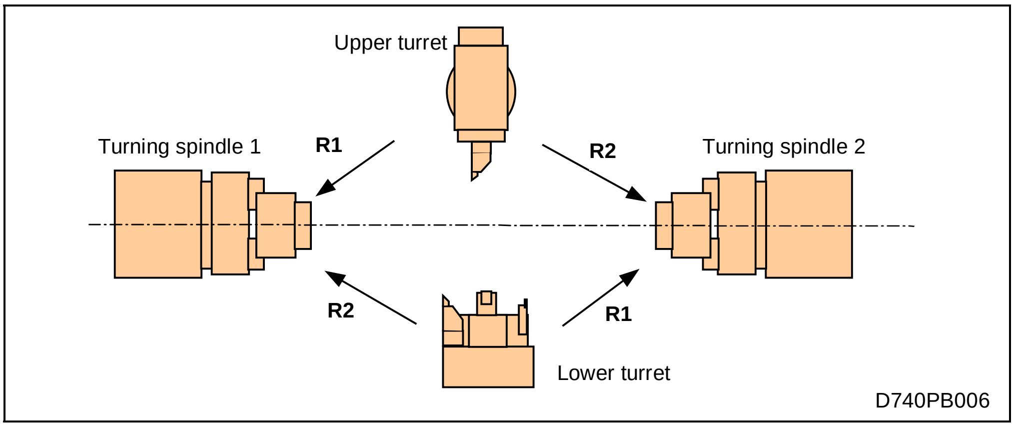
   P1: First axis
   P2: Second axis

   X-axis (the first axis) is automatically selected if argument P is omitted.

2. Spindle for constant peripheral speed control is to be set by address R.

   R1: Turning spindle (see the figure below)

   R2: Turning spindle (see the figure below)



   The default value is “R1” (automatically set if argument R is omitted).

3. Control change program and actual movement

`G90 G96 G01 X50. Z100. S200;` ⋮	Spindle speed is controlled for a peripheral speed of 200 m/min.
`G97 G01 X50. Z100. F300 S500;` ⋮	Spindle speed is controlled for 500 rpm.
`M02;`	The initial modal state will be resumed.

4. Remarks

1. The initial modal state (G96 or G97) can be selecyed by parameter **F93** bit 0.

   F93 bit 0 = 0: G97 (Constant peripheral speed control OFF)

   = 1: G96 (Constant peripheral speed control ON)

2. The function is not effective for blocks of rapid motion (G00).
   The spindle speed calculated for the peripheral velocity at the ending point is applied to the entire motion of a block of G00.

3. The last value of S in the control mode of G96 is stored during cancellation of the control (G97) and automatically made valid upon resumption of the control mode (G96).

   Example:

`G96 S50;`	50 m/min or 50 ft/min
`G97 S1000;`	1000 rpm
`G96 X3000;`	50 m/min or 50 ft/min

4. The constant peripheral speed control is effective even during machine lock.

5. Cancellation of the control mode (G96) by a command of G97 without specification of S (revs/min) retains the spindle speed which has resulted at the end of the last spindle control in the G96 mode.

   Example:

`G97 S800;`	800 rpm
`G96 S100;`	100 m/min or 100 ft/min
`G97;`	**x** rpm

   The speed **x** denotes the spindle speed of G96 mode at the end of the preceding block.

6. The peripheral speed constant control does not apply to the milling spindle.

10-3 Spindle Clamp Speed Setting: G50 [Series M: G92]

1. Function and purpose

The code G50 can be used to set the maximum and minimum spindle speeds at addresses S and Q, respectively.

2. Programming format

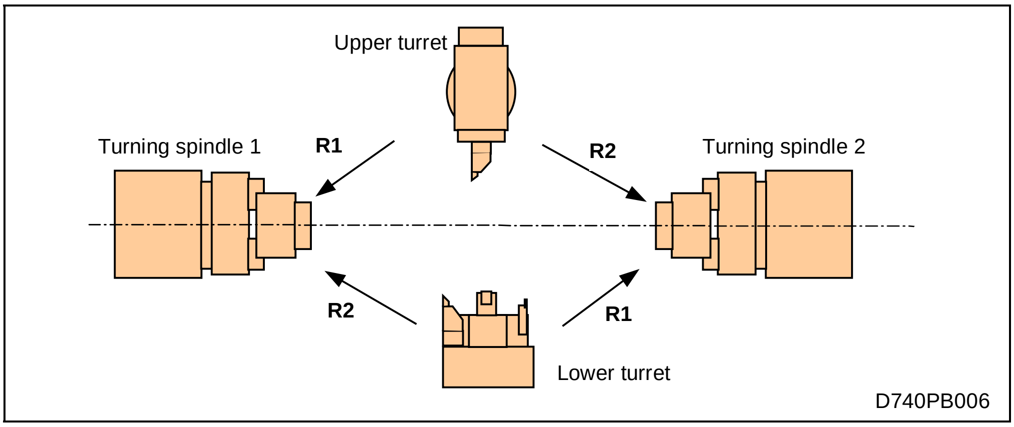
G50 Ss Qq Rr; Constant peripheral speed control ON

s: Maximum spindle speed
q: Minimum spindle speed
r: Spindle for speed clamping

3. Detailed description

1. For gear change between the spindle and spindle motor, four steps of gear range can be set by the related parameters in steps of 1 min^{-1} (rpm). In range defined by two ways, parameter setting and S50 SsQq setting, the smaller data will be used for the upper limit and the larger data for the lower limit.
2. Spindle for speed clamping is to be set by address R.

   R1: Turning spindle (see the figure below)
   R2: Turning spindle (see the figure below)
   R3: Milling spindle



The default value is “R1” (automatically set if argument R is omitted).

- NOTE -

11 TOOL FUNCTIONS

11-1 Tool Function [for ATC systems]

A next tool and tool offset number can be designated for the machine provided with ATC function by commanding T-code in the format shown below. The next tool refers to a tool used for the next machining, which can be assigned when it is currently accomodated in the magazine. The next tool in the magazine can be indexed at ATC position beforehand by commanding the next tool, thus permitting reduced ATC time.

T ○○○.◇◇ T△△△.◇◇ M6 D□□□;

○○○: Number of the tool to be changed for
◇◇: Tool ID code
○○○: Number of the tool to be used next
□□□: Tool offset number (only for Series T)

Use two digits after the decimal point as follows to designate the tool ID code with reference to the settings on the **TOOL DATA** display:

<Normal tools>

ID code	w/o	A	B	C	D	E	F	G	H	J	K	L	M
◇◇	00	01	02	03	04	05	06	07	08	09	11	12	13
ID code	N	P	Q	R	S	T	U	V	W	X	Y	Z	
◇◇	14	15	16	17	18	19	21	22	23	24	25	26	

<Heavy tools>

ID code		A	B	C	D	E	F	G	H	J	K	L	M
◇◇		61	62	63	64	65	66	67	68	69	71	72	73
ID code	N	P	Q	R	S	T	U	V	W	X	Y	Z	
◇◇	74	75	76	77	78	79	81	82	83	84	85	86	

11-2 Tool Function [4-Digit T-Code for Turret-Indexing Systems] (Series T)

Tool function, also referred to as T-code function, is used to designate the tool number and offset number. Of a four-digit integer at address T, upper and lower two digits are respectively used to specify the tool number and offset number.
Use bit 4 of parameter **F162** to select the number of digits for the tool function (0 or 1 for 4- or 6-digit T-code).

T○○□□.◇◇ ;

Tool ID code
Tool offset number
Tool number

Only one T-code can be included in a block, and the available range of T-codes depends on the machine specifications. For further details, especially on how to number the actual tools to be used, refer to the operating manual of the relevant machine.

The T-code can be given with any other commands in one block, and the T-code given together with an axis motion command is executed, depending upon the machine specifications, in one of the following two timings:

- The T-code is not executed till completion of the motion command, or
- The T-code is executed simultaneously with the motion command.

11-3 Tool Function [6-Digit T-Code for Turret-Indexing Systems] (Series T)

This function is also used to designate the tool number and offset number. Of a six-digit integer at address T, upper and lower three digits are respectively used to specify the tool number and offset number. See the above description of the 4-digit T-code for the meaning of the decimal fractions.

The available range of T-codes depends on the machine specifications. For further details, refer to the operating manual of the relevant machine.

Only one T-code can be included in a block.

Use bit 4 of parameter **F162** to select the number of digits for the tool function (0 or 1 for 4- or 6-digit T-code).

T ○○○□□□.◇◇ ;

- ◇◇ — Tool ID code
- □□□ — Tool offset number
- ○○○ — Tool number

11-4 Tool Function [8-digit T-code]

This function allows you to select a tool number (from 0 to 99999999) using eight-digit command data preceded by address T. Only one T-code can be included in a block.

Set bit 4 of parameter **F94** to 0 to select the group-number designation for T-code funciton, or set this bit to 1 to select the tool-number designation.

12 TOOL OFFSET FUNCTIONS (FOR SERIES T)

12-1 Tool Offset

1. Outline

Tool offset must be set either for the upper turret with a three-digit number following address D, or for the lower turret with the lower two or three digits of a four-digit or six-digit number folowing address T (where the higher two or three digits are used to designate the tool number). Whether the offset number is set by lower two or three digits is selected by parameter **F162** bit 4.
One set of T command can be included in the same block.
The tool offset amount differs according to the combination of G53.5/G52.5 (MAZATROL coordinate system selection/cancel) and parameter **F111** bit 5 (MAZATROL tool wear offset data valid/invalid) as in the following table.

		G53.5 (MAZATROL coordinate system)		G52.5 (Cancellation of MAZATROL coord. sys.)	
Program	Upper turet	T001 T000 M6 D000 [1] [2]	T001 T000 M6 D001 [1] [2]’	T001 T000 M6 D000 [1] [2]	T001 T000 M6 D001 [1] [2]’
	Lower turret	T001 000 [1] [2]	T001 001 [1] [2]’	T001 000 [1] [2]	T001 001 [1] [2]’
Parameter	**F111** bit 5 = 1 (Validation of MAZATROL tool wear offset data)	[1] - Tool of TNo. 1 indexed - **TOOL SET** data (on **TOOL DATA** display) of TNo. 1 validated [2] - Tool offset cancel	[1] - Tool of TNo. 1 indexed - **TOOL SET**, **WEAR COMP.** and **TL EYE CM** data (on **TOOL DATA** display) of TNo. 1 validated [2]’ - Data of No. 1 on **TOOL OFFSET** display validated	[1] - Tool of TNo. 1 indexed [2] - Tool offset cancel	[1] - Tool of TNo. 1 indexed [2]’ - Data of No. 1 on **TOOL OFFSET** display validated
	F111 bit 5 = 0 (Invalidation of MAZATROL tool wear offset data)	*See above.*	[1] - Tool of TNo. 1 indexed - **TOOL SET** data (on **TOOL DATA** display) of TNo. 1 validated [2]’ - Data of No. 1 on **TOOL OFFSET** display validated	*See above.*	*See above.*

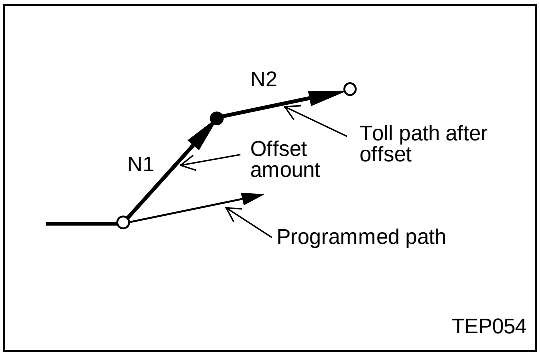
2. Tool offset start

There are two ways to execute tool offset and these can be selected by parameter **K95** bit 2: executing offset when the T command is executed and executing offset not in T command execution but in the block containing move commands.

A. Offset in T command execution

```
N1 T001T000M6D001;
N2 X100.Z200.;
```

Tool length offset and tool nose wear offset are executed simultaneously.



Note 1: The movement when offsetting with the T command is rapid feed in a G00 modal and cutting feed with other modals.

Note 2: When performing offset in T command execution, the path is made by linear interpolation in an arc modal.

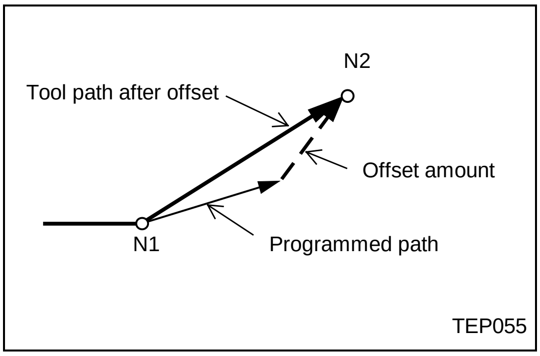
Note 3: When performing offset in T command execution, offset will not function until the arrival of any command G except those listed below when the T command is included in the same block as those commands G.

G04: Dwell
G10: Data setting
G50: Coordinate system setting

B. Offset with move command

```
N1 T001T000M6D001;
N2 X100.Z200.;
```

Tool offset is executed simultaneously.



Note: When performing offset with a move command, offset is applied if the offset amount is lower than the parameter value of “tolerance for radial value difference at starting and ending points in arc command” when offset is performed for the first time with an arc command. If the amount is higher, a program error will occur. (This also applies when the arc command and T command are in the same block for offsetting with T command.)

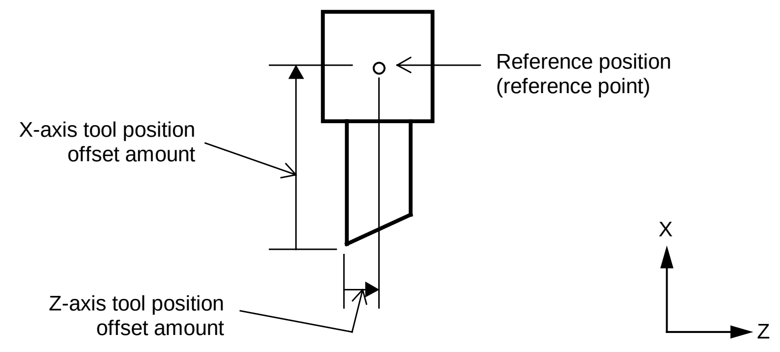
Return to Library

12-2 Tool Position Offset

1. Tool position offset amount setting

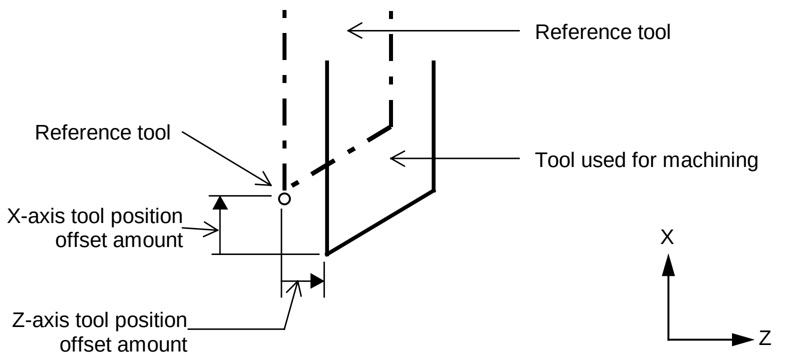
This function offsets tool position with respect to the program reference position. This position may generally be set to either the center position of the turret or the tool nose position of the reference tool.

A. Setting to the center position of turret



TEP056

B. Setting to the tool nose position of reference tool



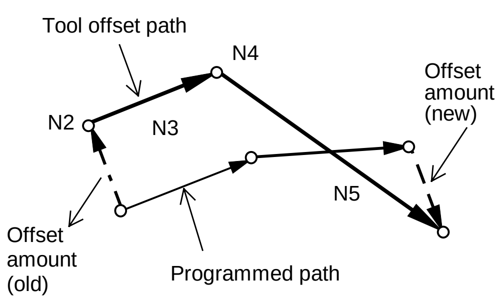
TEP057

2. Tool position offset number change

When tool number is changed, the tool position offset for the new tool number is added to the movement amount in the machining program.

```
N1  T001T000M6D001;
N2  G1 X10.0 Z10.0 F100;
N3  G1 X13.0 Z15.0 F20.0;
N4  T001T000M6D002;
N5  G1 X13.0 Z20.0 F25.0;
```

In this example, the tool position is offset with the tool number and offset is performed in the block including the move command.



TEP058

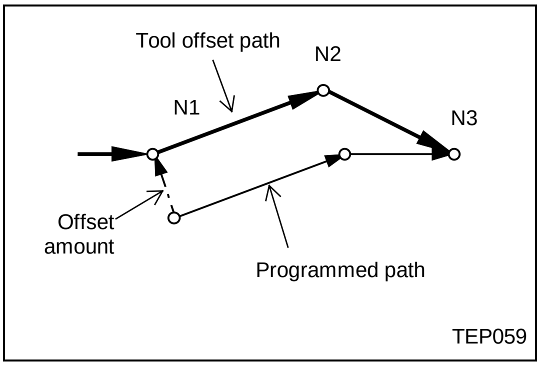
3. Tool position offset cancel

A. When an offset number of zero is set

Offset is cancelled when 0 as the tool position offset number preceded by T-code is executed.

```
N1  X10.0 Z10.0 F10;
N2  T001T000M6D000;
N3  G1 X10.0 Z20.0;
```

In this case, offset is performed by the block with the move command.

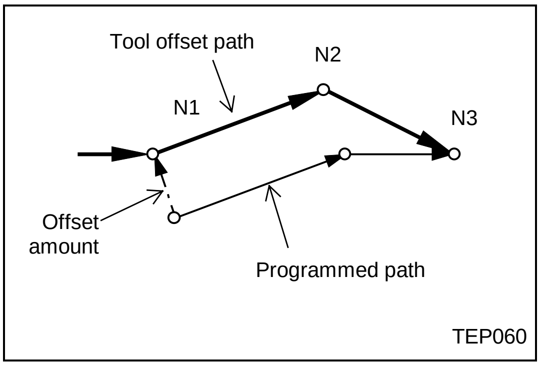


B. When 0 is set as the offset amount

Offset is cancelled when 0 is set as the offset amount of the tool position offset number.

```
N1  G1 X10.0 Z10.0 F10;
N2  T001T000M6D000;
N3  G1 X10.0 Z20.0;
```

In this case, offset is performed by the block with the move command.



4. Remarks

- When G28, G29 or G30 is commanded, the movement is performed to the position where offset is cancelled. But as offset amount remains stored in the memory, the positioning for the succeeding move command is executed with the offset operation.
- The tool position offset is cleared by resetting and by emergency stop.

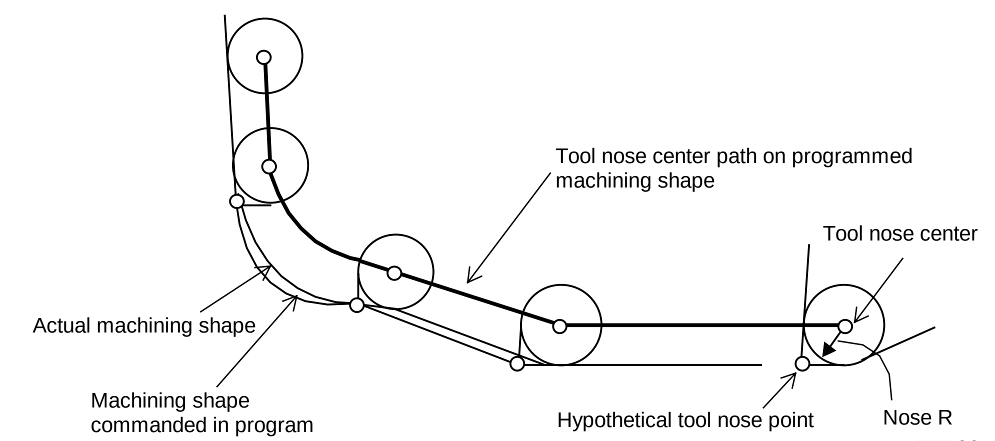
12-3 Nose R/Tool Radius Compensation: G40, G41, G42

12-3-1 Outline

1. Function and purpose

The tool nose is generally rounded and so a hypothetical tool nose point is treated as the tool nose for programming. With such a programming, an error caused by the tool nose rounding arises during taper cutting or arc interpolation between the actually programmed shape and the cutting shape. Nose R or tool radius compensation is a function for automatically calculating and offsetting this error by setting the nose radius or tool radius value.

The command codes enable the offset direction to be fixed or automatically identified.



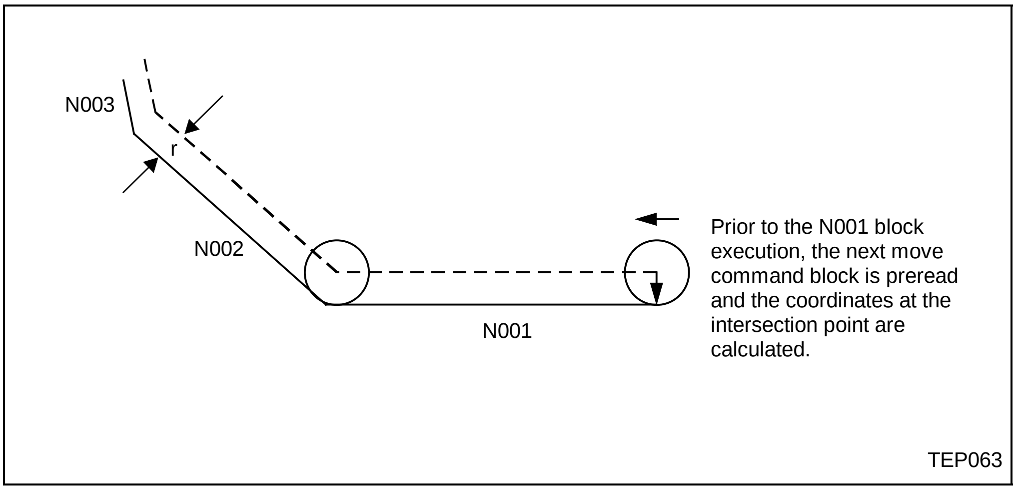
TEP061

2. Programming format

Code	Function	Programming format
G40	Nose R/Tool radius compensation mode cancel	G40 Xx/Uu Zz/Ww Ii Kk ;
G41	Nose R/Tool radius compensation left mode ON	G41 Xx/Uu Zz/Ww ;
G42	Nose R/Tool radius compensation right mode ON	G42 Xx/Uu Zz/Ww ;

3. Detailed description

1. G40 serves to cancel the tool nose radius compensation mode.
2. Tool nose radius compensation function prereads the data in the following two move command blocks (up to 5 blocks when there are no move function commands) and controls the tool nose radius center path by the intersection point calculation method so that it is offset from the programmed path by an amount equivalent to the nose radius.



In the above figure, "r" is the tool nose radius compensation amount (nose radius).

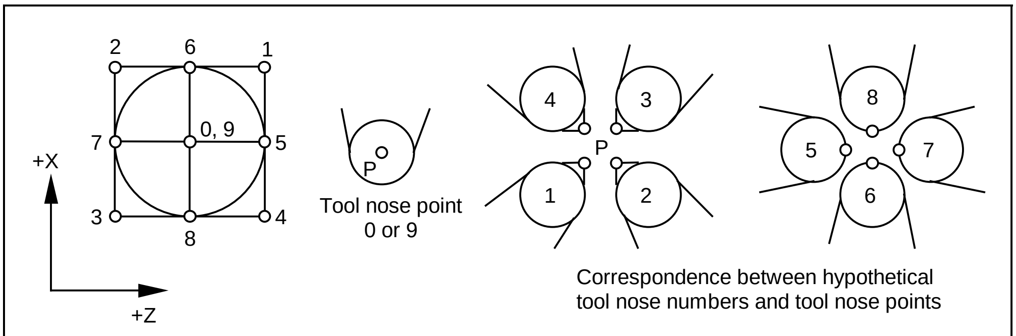
3. The tool nose radius compensation amount corresponds to the tool length number and it should be preset with the tool nose point.
4. If four or more blocks without move commands exist in five continuous blocks, overcutting or undercutting will result. However, blocks in which optional block skip is valid are ignored.
5. Tool nose radius compensation function is also valid for fixed cycles (G77 to G79) and for roughing cycles (G70, G71, G72 and G73).
   However, in the roughing cycles, the tool nose radius compensation function applied for finish shape is cancelled and upon completion of the roughing, NC unit will re-enter the compensation mode.
6. With threading commands, compensation is temporarily cancelled in one block before.
7. The compensation plane, move axes and next advance direction vector follow the plane selection command designated by G17, G18 or G19.

   G17............... XY plane X, Y; I, J
   G18............... ZX plane Z, X; K, I
   G19............... YZ plane Y, Z; J, K

12-3-2 Tool nose point and compensation directions

1. Tool nose point

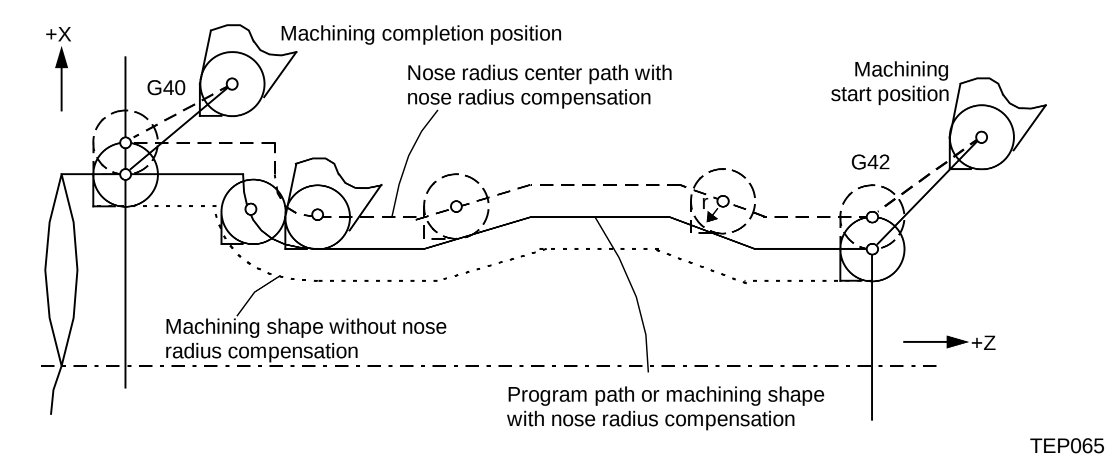
Since the tool nose is generally rounded, the programmed tool nose position is aligned with point P shown in the examples of the figures below.
For tool nose radius compensation, select one point among those in the figures below for each tool length number and preset. (Selection from 0 to 9 in the G41/G42 mode.)



TEP064

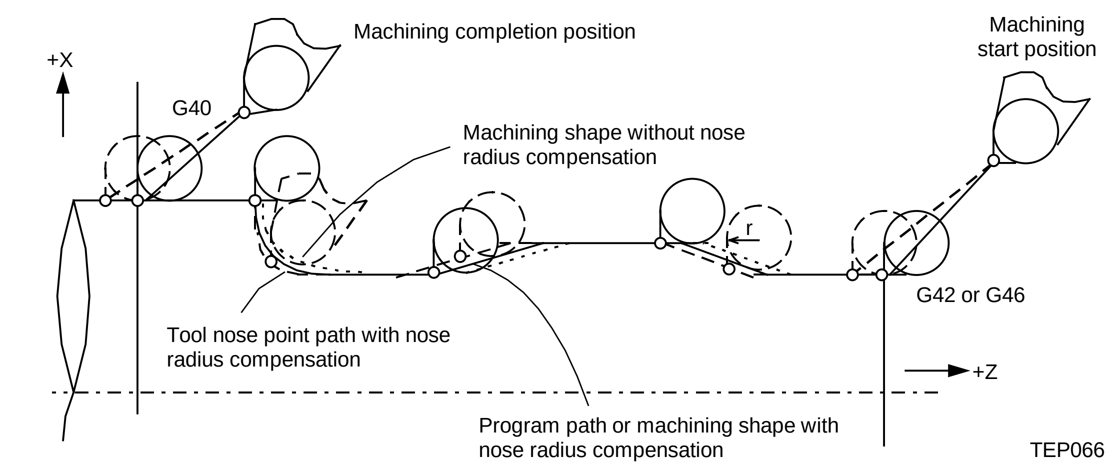
2. Tool nose point and compensation operation

A. When the nose radius center has been aligned with the machining start position



TEP065

B. When the tool nose point has been aligned with the machining start position



TEP066

12-3-3 Operations of nose R/tool radius compensation

1. Cancellation of nose R/tool radius compensation

Nose R/Tool radius compensation is automatically cancelled in the following cases:

- After power has been turned on
- After the reset key on the NC operation panel has been pressed
- After M02 or M30 has been executed (if these two codes have a reset function)
- After G40 (tool nose radius compensation cancellation command) has been executed
- After tool number 0 has been selected (T00 has been executed)

In the compensation cancellation mode, the offset vector becomes zero and the tool nose point path agrees with the programmed path.
Programs containing the tool nose radius compensation function must be terminated during the compensation cancellation mode.

2. Startup of nose R/tool radius compensation

Nose R/Tool radius compensation will begin when all the following conditions are met:

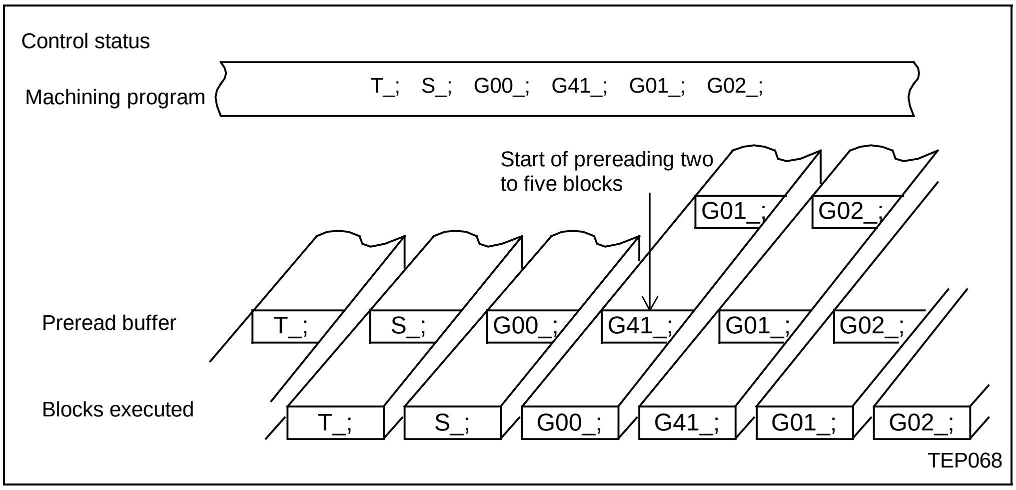
- Command G41 or G42 has been executed.
- The command used with the offsetting command is a move command other than those used for arc interpolation.

Offsetting will be performed only when reading of two through five blocks in succession is completed, irrespective of whether the continuous operation or the single-block operation mode is used. (Two blocks are preread if move command is present and five blocks are preread if such command is not present.)
During offsetting, maximal five blocks are preread and then calculation for offsetting is performed.
Some G-codes may not allow prereading. If startup compensation vector cannot be provided owing to inability of prereading, program error will occur.
(**Example:** G41 T0101; G28 X10. Z20. ; ···)

Prereading is not allowed for the following G-codes:

G10, G27, G28, G29, G30, G30.1, G36, G37

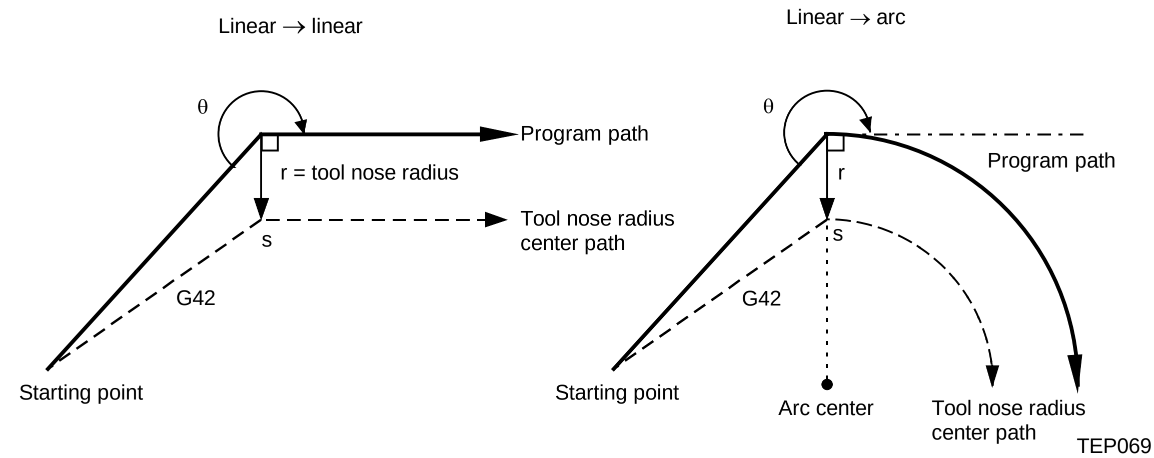
If error is caused because of the reason above, provide serveral blocks including move commands after G41, G42 or T command.



3. **Start operation for nose R/tool radius compensation**

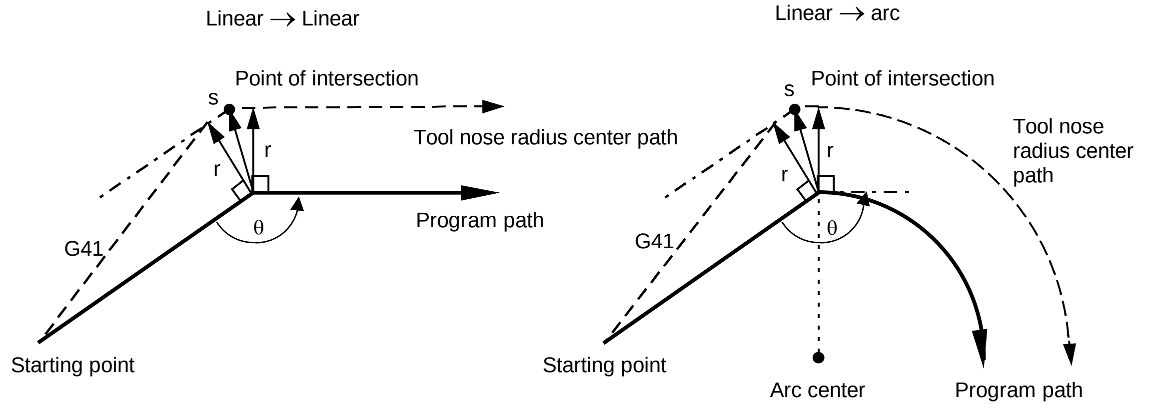
In the following figures, “s” denotes the single block stop point.

A. **For the corner interior**



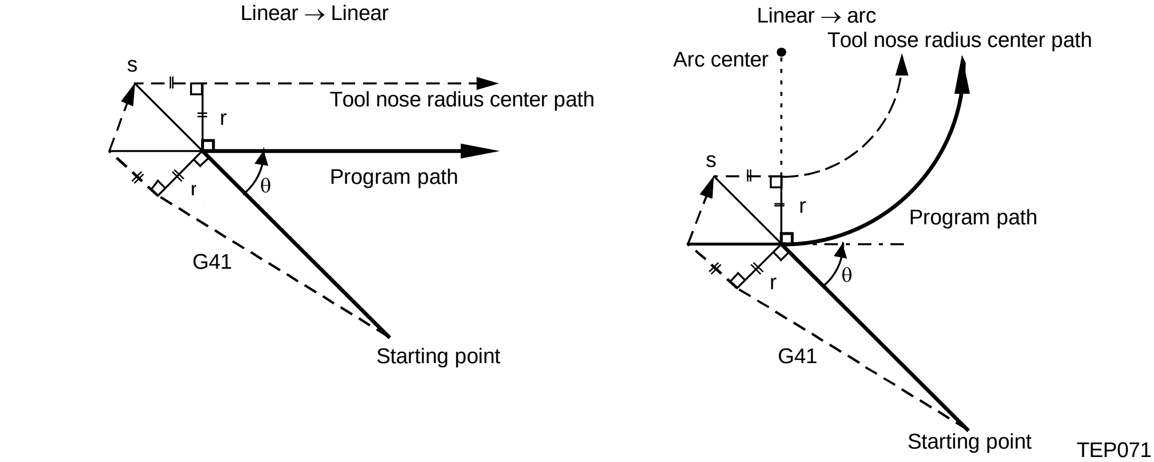
TEP069

B. **For the corner exterior (obtuse angle) (90° ≤ θ < 180°)**



TEP070

C. **For the corner exterior (acute angle) (θ < 90°)**



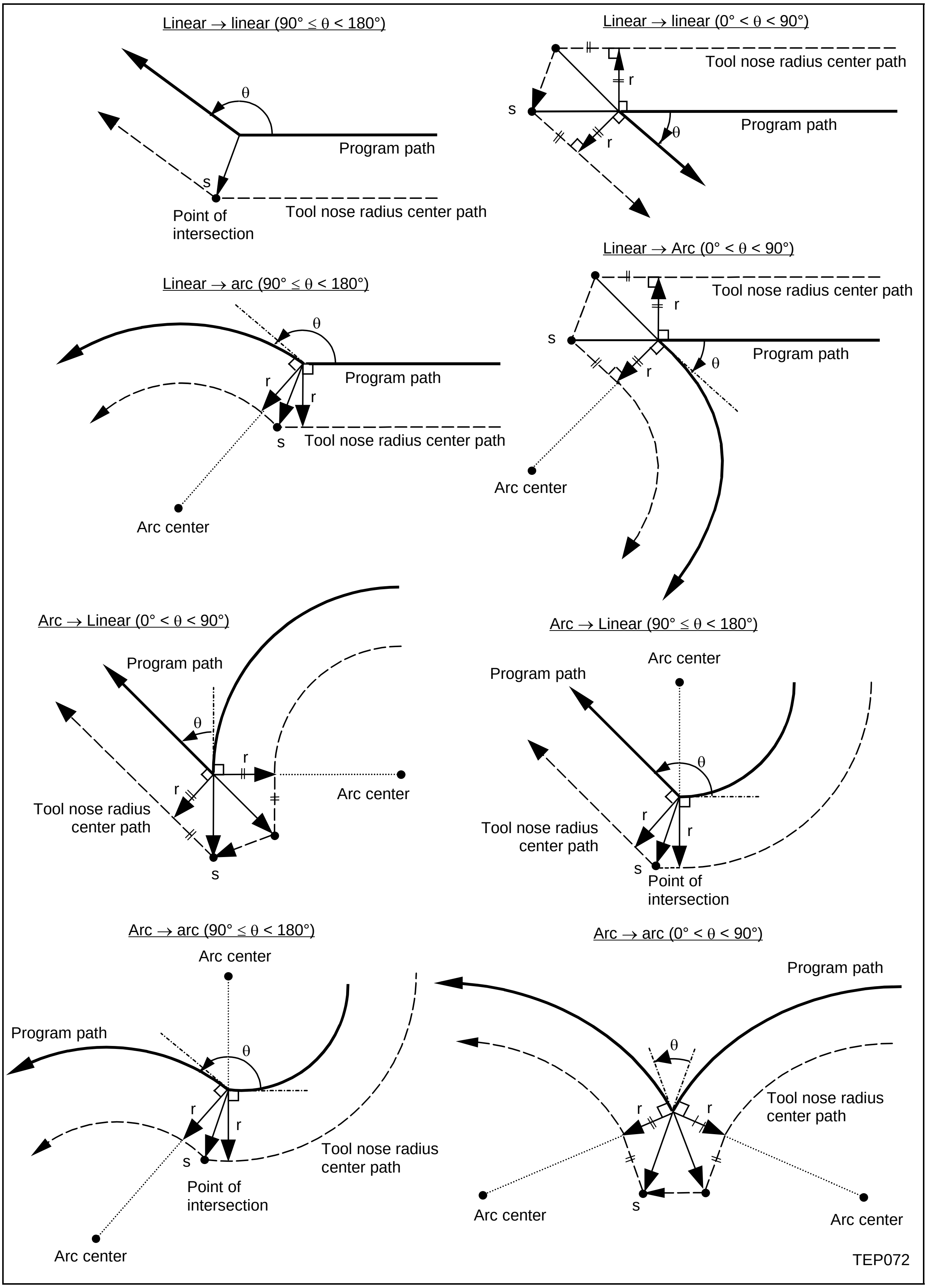
TEP071

Note: When there is no axis move command in the same block, compensation is performed perpendicularly to the movement direction of the next block direction.

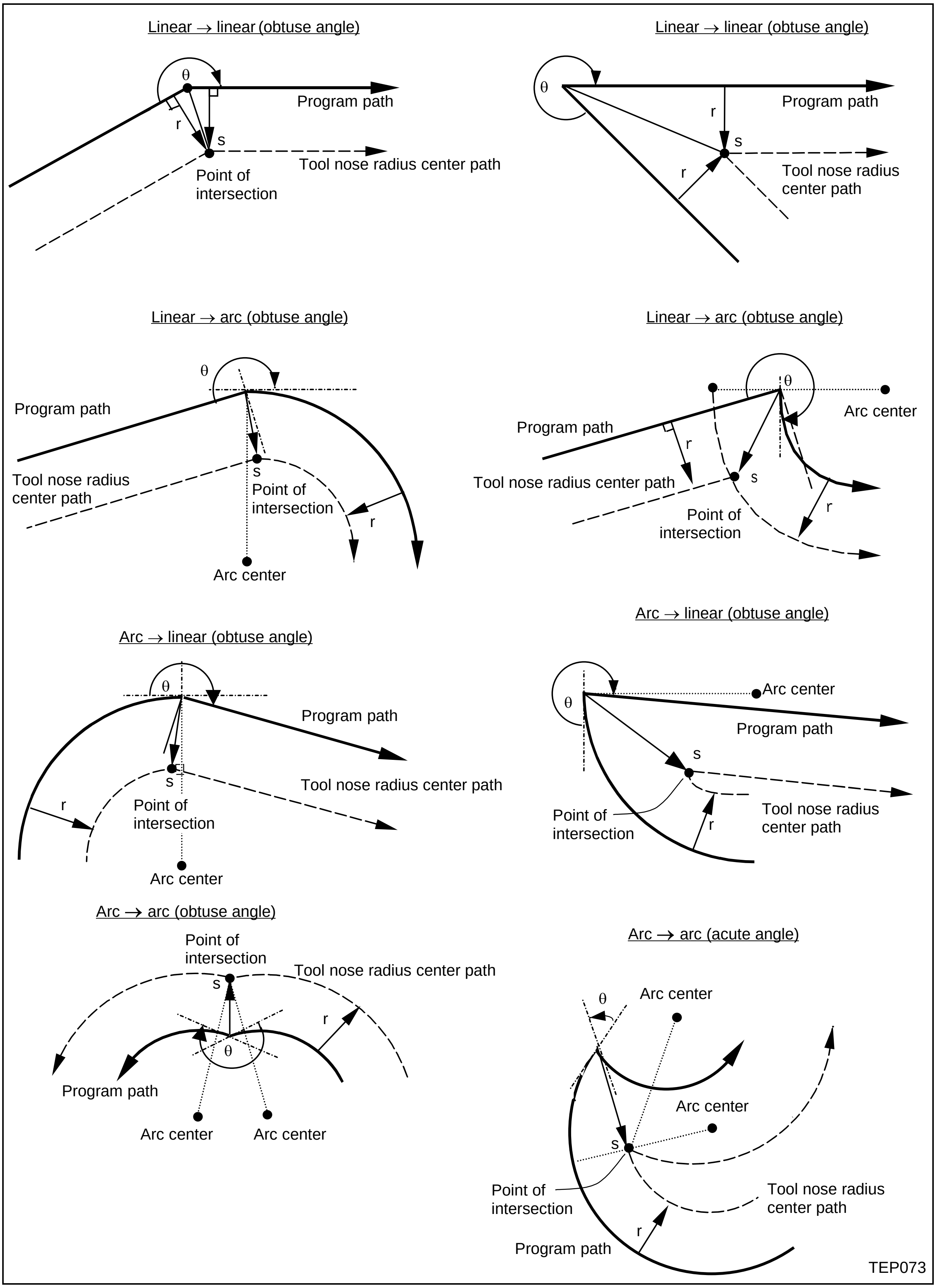
4. Movement in compensation mode

Compensation is valid both for positioning and for interpolation commands such as arc and linear interpolation.
Even if the same compensation command G41/G42 is set in a nose R/tool radius compensation mode (G41/G42), the command will be ignored.
When four or more blocks not including move command are commanded in the compensation mode, overcutting or undercutting will result.
When the M00 command has been set during nose R/tool radius compensation, pre-reading is prohibited.

A. For the corner exterior

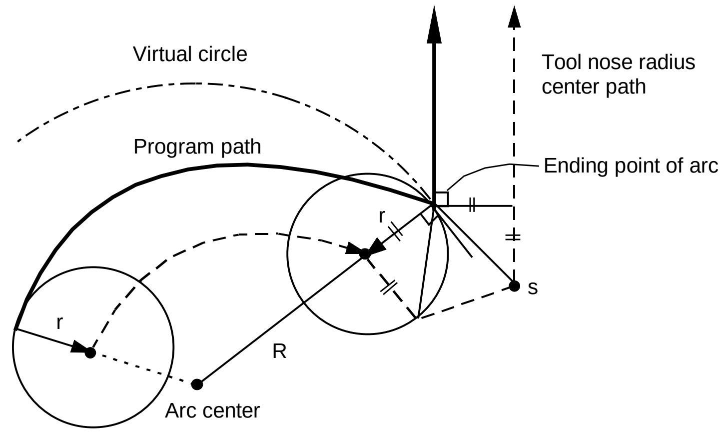


B. For the corner interior



C. For the arc on which the ending point is not found

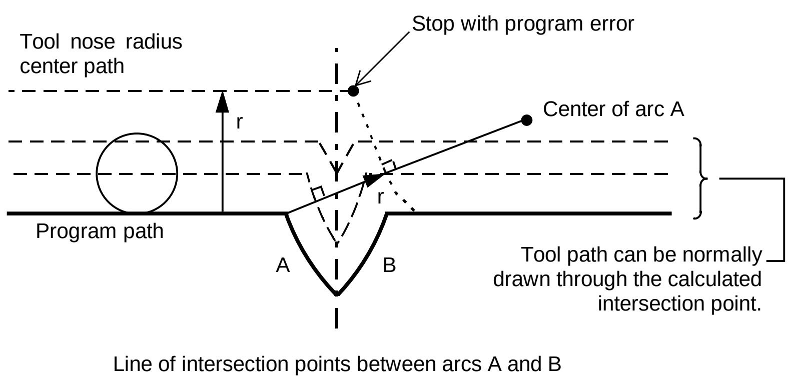
If the error applied after compensation is within the “arc error” set by parameter, the area from the arc starting point to the ending point is interpolated as a spiral arc.



TEP074

D. In cases that no inner intersection point exist inside the corner

In cases such as those shown in the figure below, there may or may not be an intersection point of arcs A and B, depending on the particular offset data. In latter cases, a program error occurs and the tool stops at the ending point of the previous block.



TEP075

5. Cancellation of nose R/tool radius compensation

If either of the following conditions is met in the nose R/tool radius compensation mode, the compensation will be cancelled.

- Command G40 has been executed.
- Tool number T00 has been executed.

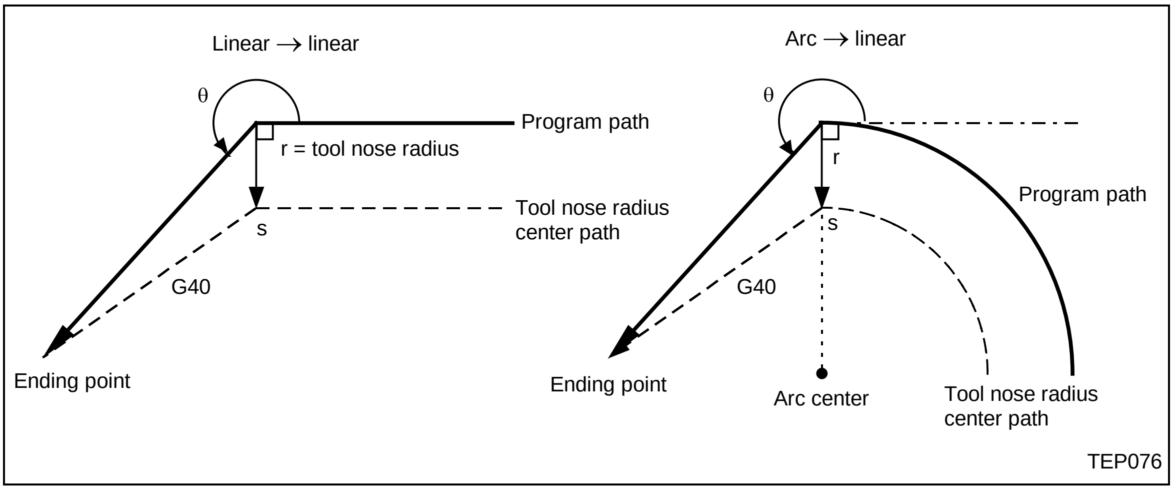
However, the move command executed must be one other than those used for arc interpolation. A program error will occur if an attempt is made to cancel compensation using an arc command.

The cancel mode is established once the compensation cancel command has been read, five-blocks prereading is suspended and one-block pre-reading is made operational.

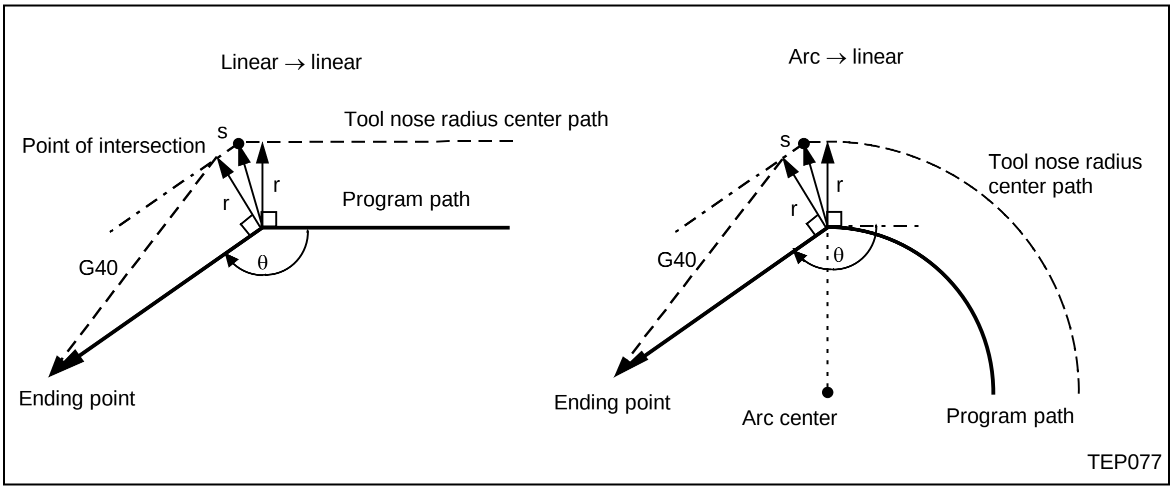
Return to Library

6. Cancel operation for nose R/tool radius compensation

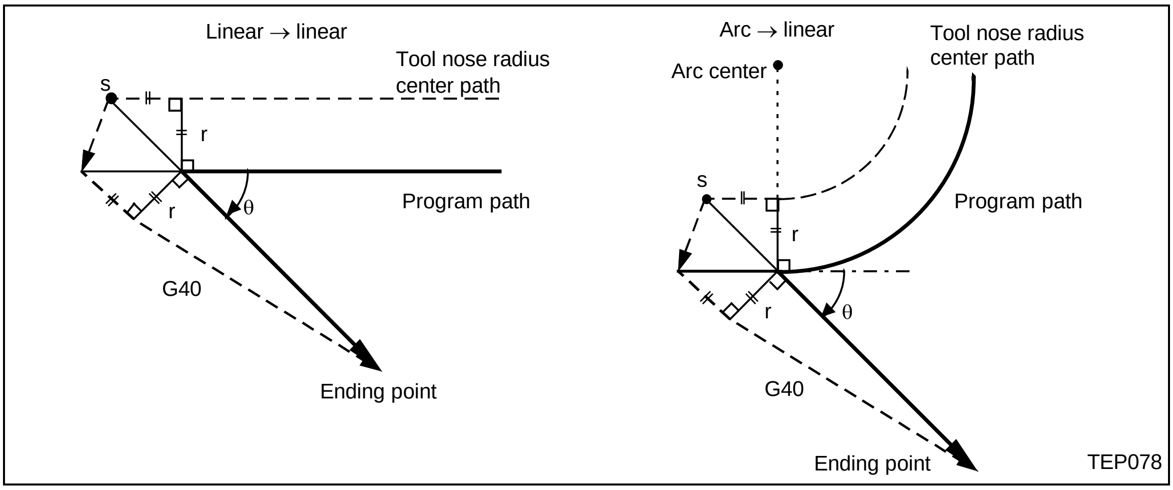
A. For the corner interior



B. For the corner exterior (obtuse angle)



C. For the corner exteiror (acute angle)



Return to Library

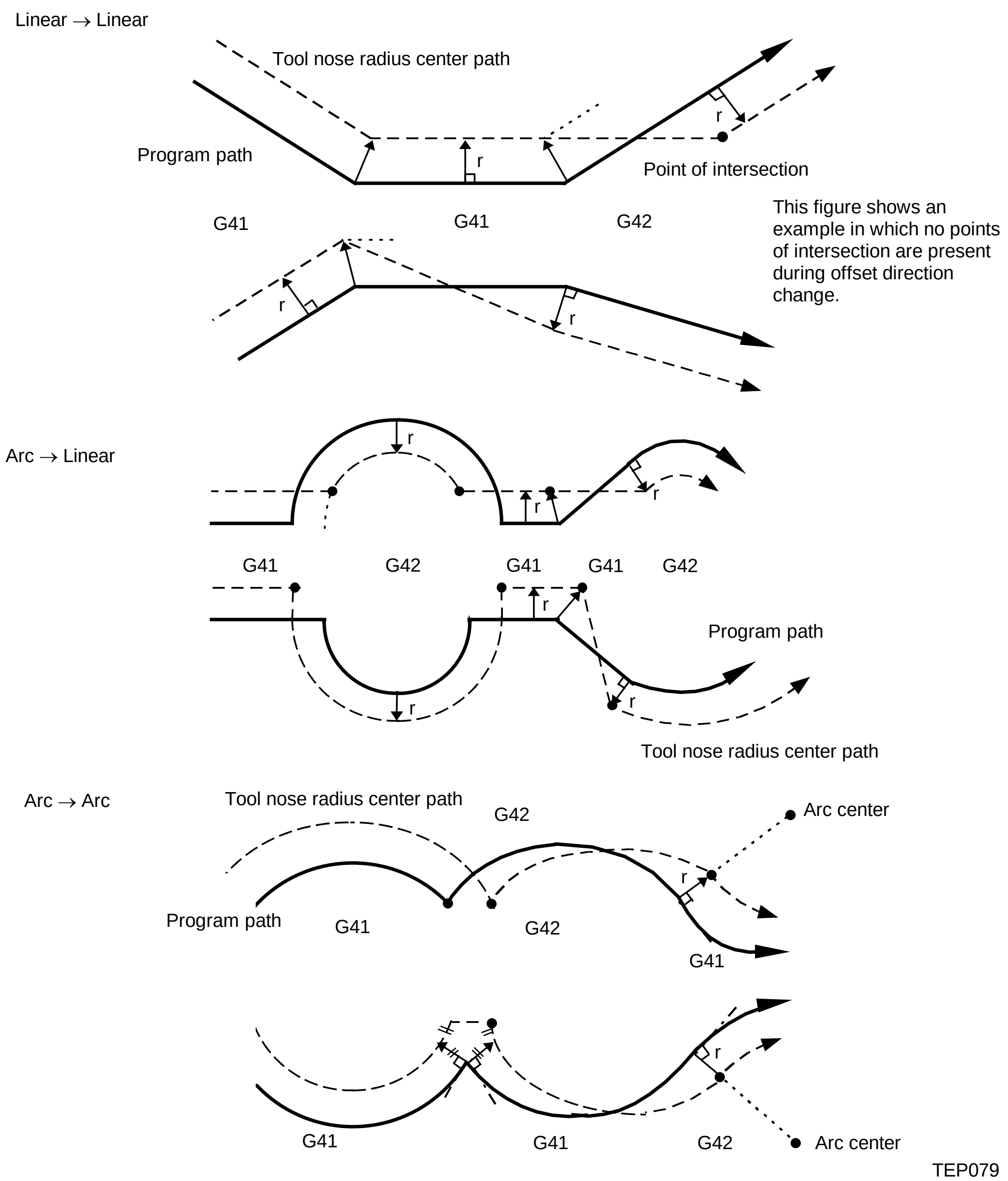
12-3-4 Other operations during nose R/tool radius compensation

1. Changing the compensation direction during nose R/tool radius compensation

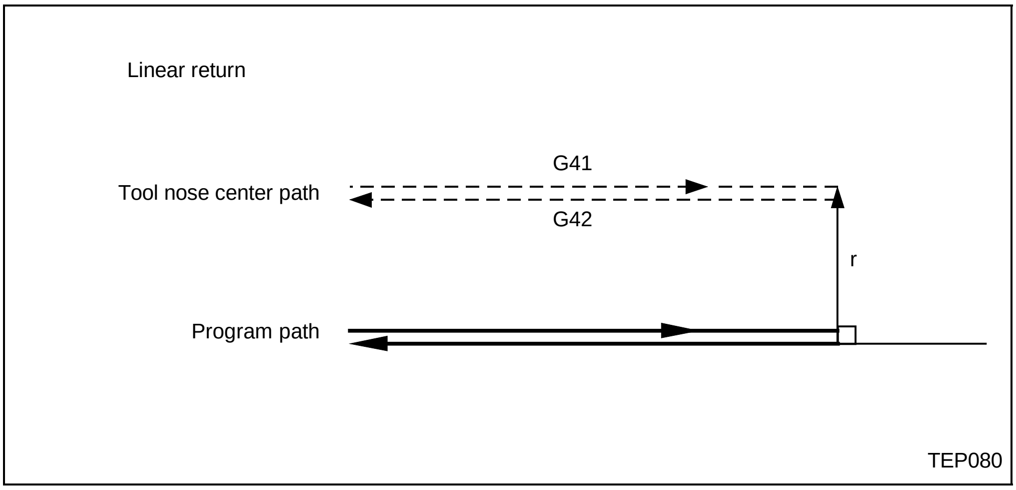
The compensation direction is determined by the nose R/tool radius compensation commands (G41, G42).

G41	Lef-hand compensation
G42	Right-hand compensation

The compensation direction can be changed by changing the compensation command without commanding compensation cancel in the compensation mode. However, no change is possible in the compensation start block and the following block.



TEP079

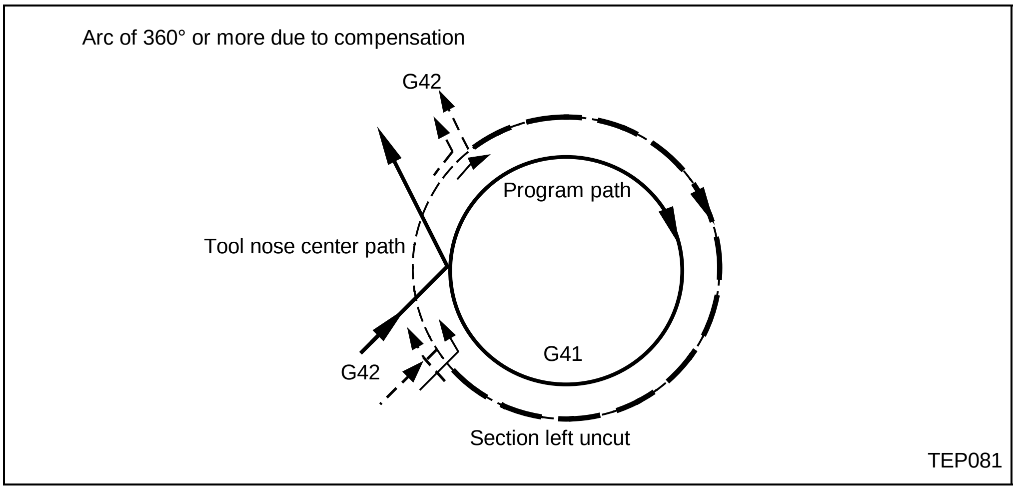


In the following cases, it is possible that the arc may exceed 360°.

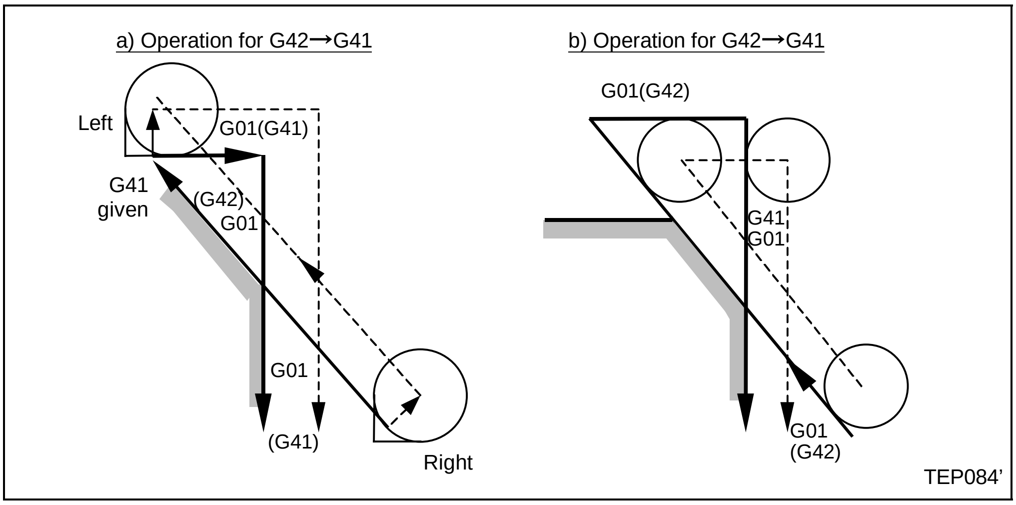
- Compensation direction is changed by the selection of G41 or G42.

- I, J, K are commanded with G40.

In such cases, compensation is provided as shown above and a section will be left uncut.



2. Nose R/Tool radius compensation by G41/G42 for closed path

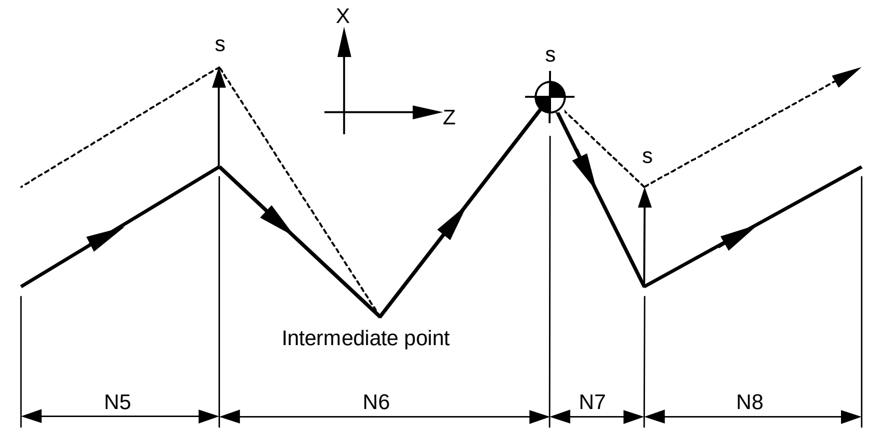


Return to Library

3. Command for temporarily canceling offset vectors

When the following command is set in the compensation mode, the current offset vectors are lost temporarily and then the NC unit will re-enter the compensation mode.
In this case, the compensation is not cancelled, and the program control will be transferred from one intersection point vector directly to the vectorless point, that is, to the programmed point. Control will also be transferred directly to the next intersection point when the offset mode is re-entered.

A. Reference point return command



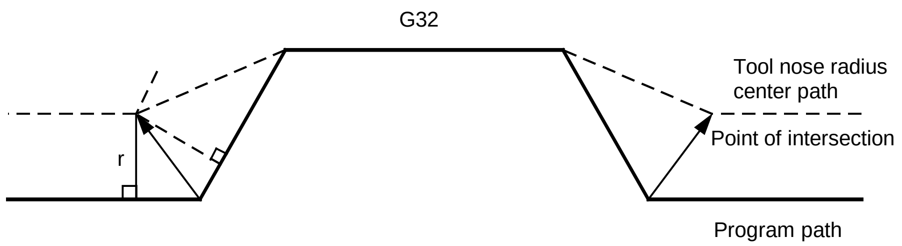

```
(G41)      ⋮
N5 G01 U+30. W+60.;
N6 G28 U-40. W+50.; ←  Temporarily vector 0 as compensation at intermediate
                       point (Reference point when the intermediate point is
                       not available)
N7     U-60. W+30.;
N8     U+40. W+70.;
           ⋮
```

TEP083

Note: The offset vectors do not change with the coordinate system setting command G52.

B. G32 thread cutting command

Tool nose radius compensation does not apply to the G32 block.



TEP084

C. Compound fixed cycles

When a compound fixed cycle I command (G70, G71, G72, G73) is assigned, the tool nose radius compensation is temporarily cancelled, the finishing shape to which tool nose radius compensation has been applied is cut in turning mode with the compensation cancelled and, upon completion, the compensation mode is automatically re-entered.

4. Blocks that do not include move command

The following blocks are referred to as those which do not include movement.

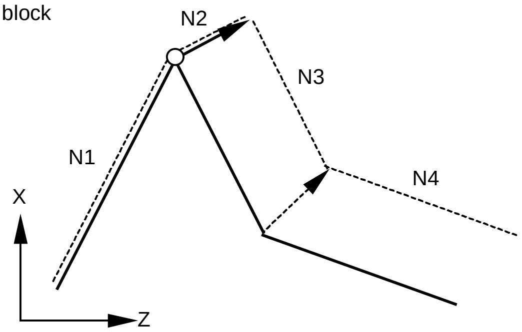
```
M03;................................. M command
S12;................................. S command
T001T000M6D001;.......... T command
G04X500;......................... Dwell
G10P01R50;..................... Offset stroke setting
G50X600.Z500.;............ Coordinate system setting
Y40.; ............................... Movement not on offset plane
G00;................................. G-code only
U0;.................................. Moving stroke 0 ———————— Movement stroke is zero
```

Move-free (M03; through G00;)

A. When a block that does not include movement is set during the start of compensation

Vertical compensation will be performed on the next move block.

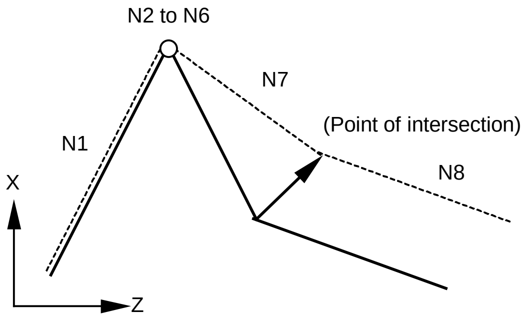
```
N1  U60.W30.T001T000M6D001;
N2  G41;———————— Move-free block
N3  U-50.W20.;
N4  U-20.W50.;
```



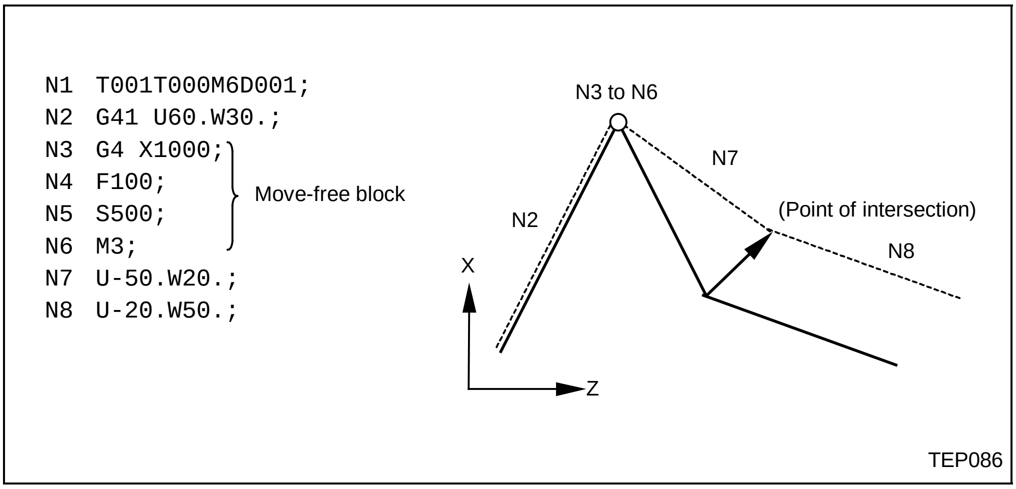
TEP085

Compensation vectors, however, will not be generated if four or more blocks that do not include move commands appear in succession.

```
N1  U60.W30.T001T000M6D001;
N2  G41;
N3  G4 X1000;
N4  F100;        Move-free block (N2 to N6)
N5  S500;
N6  M3;
N7  U-50.W20.;
N8  U-20.W50.;
```

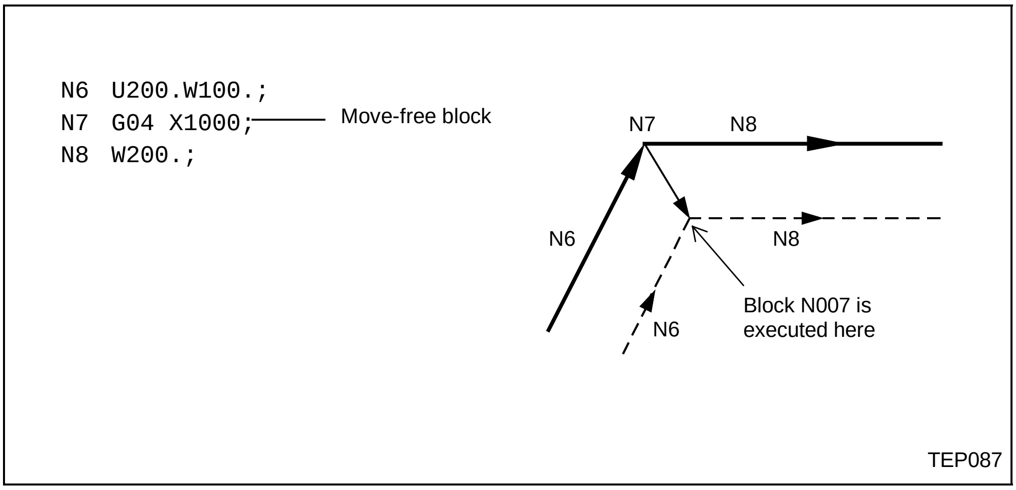


TEP086

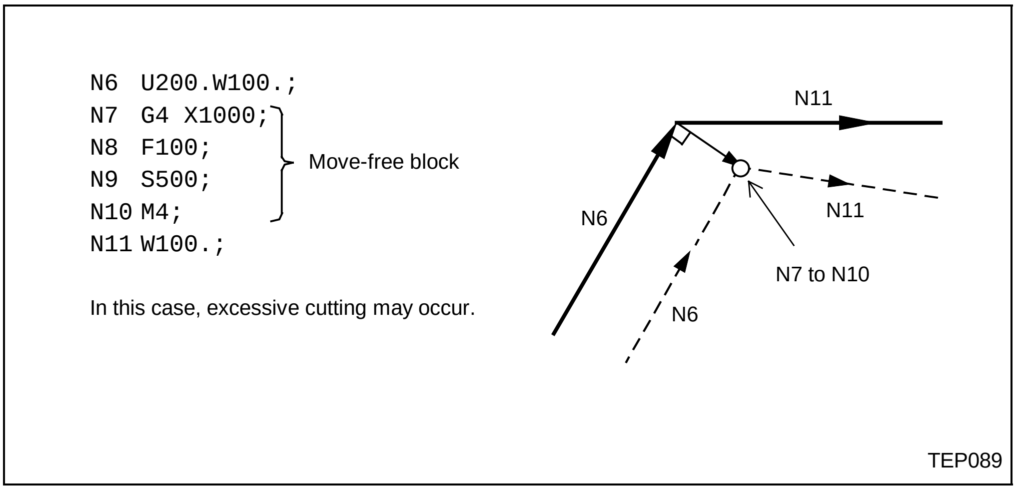


B. When a block that does not include movement is set during the compensation mode

Usual intersection point vectors will be generated unless four or more blocks that do not include movement appear in succession.

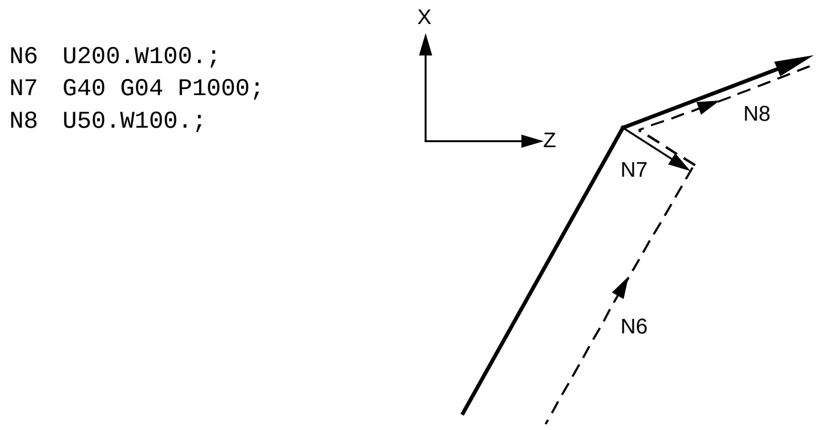


Vertical compensation vectors will be generated at the end point of preceding block if four or more blocks that do not include movement appear in succession.



C. **When a block that does not include movement is set together with compensation cancellation**

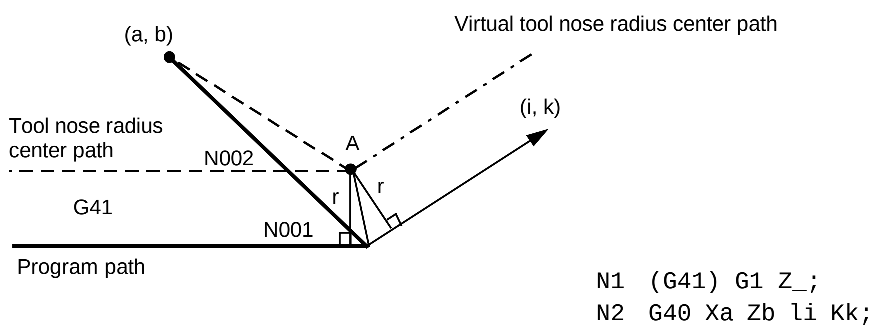
Only offset vectors will be cancelled if the block that does not include movement contains G40.



TEP089

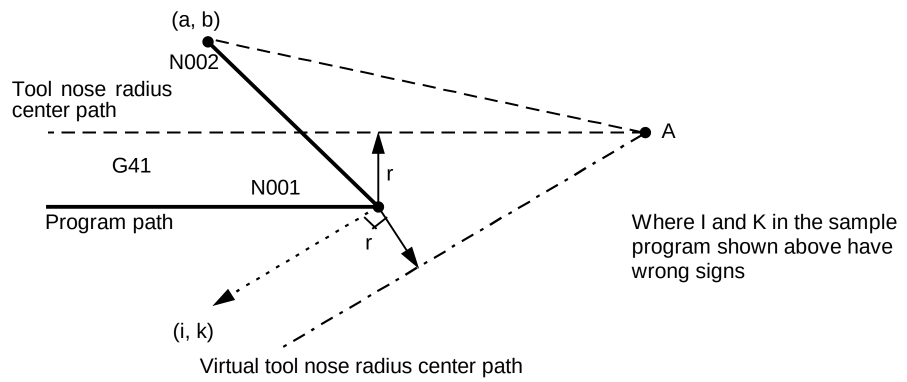
5. **If I, J and K are set with G40**

When the last move command of the four blocks which immediately precede the G40 command block is G41 or G42, movement will be handled as if it had been programmed to occur in the vectorial direction of I, J, and K from the ending point of that last move command. That is, the area up to the intersection point with the virtual tool center path will be interpolated and then compensation will be cancelled. The compensation direction will remain unchanged.



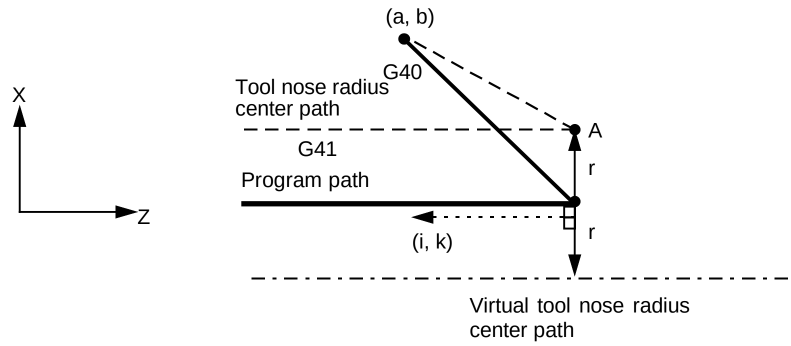
TEP090

In this case, pay attention that, irrespective of the compensation direction, the coordinates of the intersection point will be calculated even if wrong vectors are set as shown in the diagram below.



TEP091

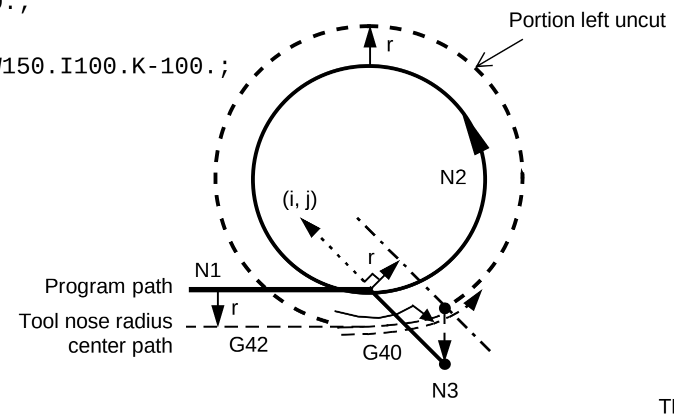
Also, pay attention that a vertical vector will be generated on the block before that of G40 if the compensation vector cannot be obtained by intersection point calculation.



TEP091

Note: Part of the workpiece will be left uncut if the I/J/K command data in G40 preceded by an arc command generates an arc of more than 360 degrees.

```
N1 (G42) G01 W200.;
N2 G03 I150.;
N3 G40 G1 U-150.W150.I100.K-100.;
```

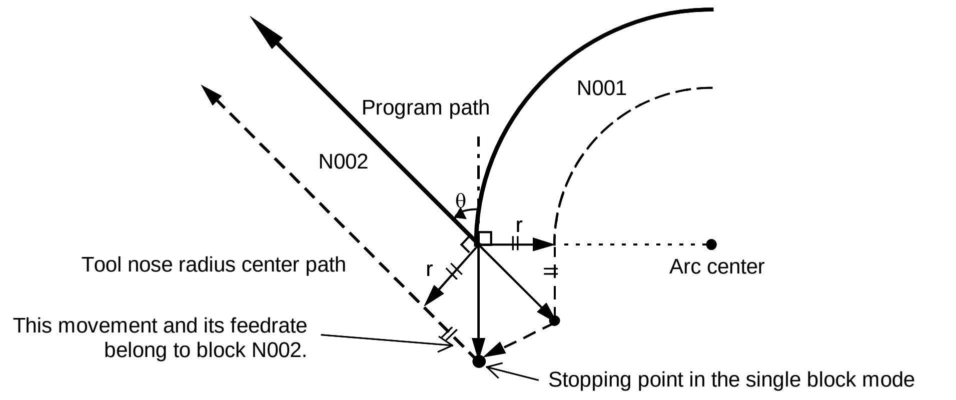


TEP093

Return to Library

6. Corner movement

If multiple offset vectors are generated at connections between move command blocks, the tool will move linearly between those vectors. This action is referred to as corner movement.
If the multiple vectors do not agree, the tool will move around the corresponding corners (but this movement belongs to the connection block). During single-block operation, the section of (Preceding block + Corner movement) is executed as one block and the remaining section of (Remaining corner movement + Next block) is executed during next movement as another block.



TEP094

12-3-5 Commands G41/G42 and I, J, K designation

The compensation direction can be intentionally changed by setting the command G41/G42 and I, J, K in the same block.

1. Programming format

G18 (Z-X plane) G41/G42 X_ Z_ I_ K_ ;

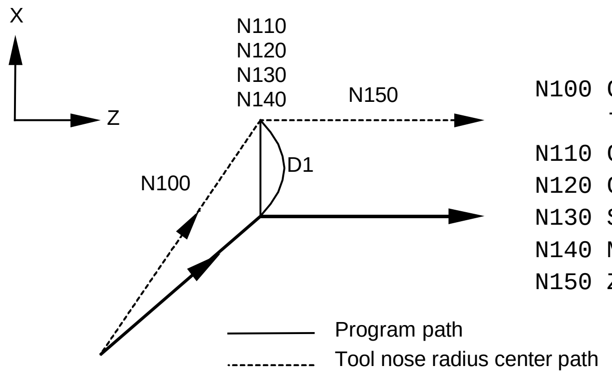
Set a linear interpolation command (G00, G01) as move command.

2. I, K type vector (G18 Z-X plane selection)

The new I, K type vector (G18 plane) created by this command is described here. (Similar descriptions apply to I, J vector for the G17 plane and to J, K vector for the G19 plane.)
Being different from the vector on the intersection point of the programmed path, the I, K type compensation vector is the vectors equivalent to the offset value, perpendicular to the direction designated by I, K. The I, K vector can be commanded even in the tool nose radius compensation mode (G41/G42 mode in the preceding block) and even at the compensation start (G40 mode in the preceding block).

Return to Library

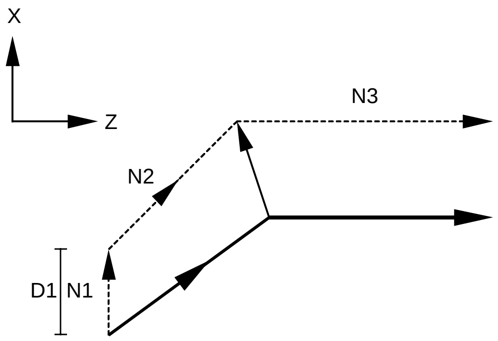
A. When I, K is commanded at compensation start:




```
        (G40)
          :
N100 G41 U100.W100.K150.
     T001T000M6D001;
N110 G04 X1000;
N120 G01 F1000;
N130 S500;
N140 M03;
N150 Z150.;
          :
```

TEP095

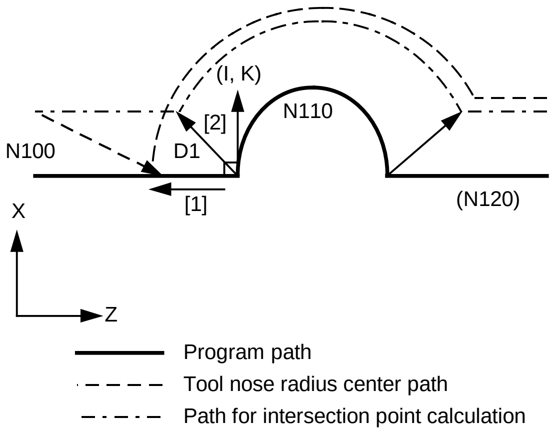
When there are no move commands at the compensation start




```
   (G40)
     :
N1 G41 K150. T001T000M6D001;
N2 U100.W100.;
N3 W150.;
     :
```

TEP096

B. When I, K has been commanded in the tool nose radius compensation mode (G18 plane)

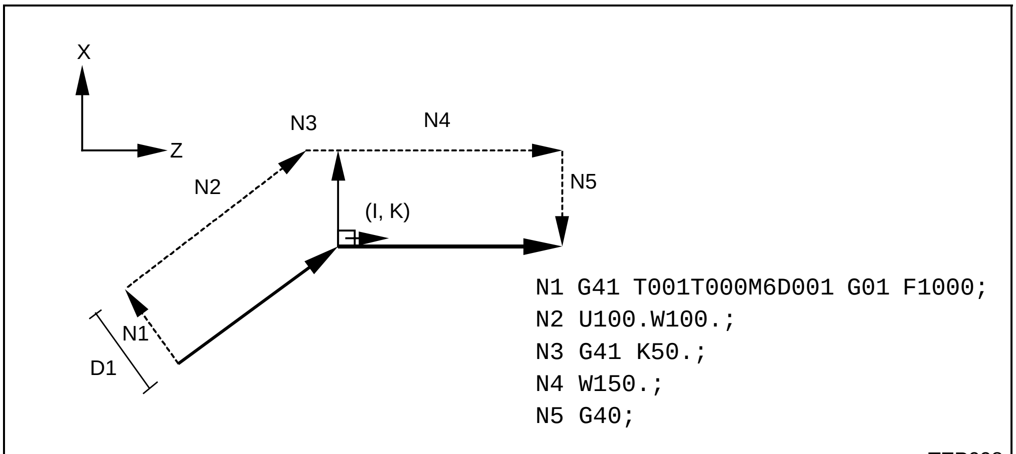



```
          :
    (G18 G41)
N100 G41 G00 W150.I50.;
N110 G02 W100.K50.;
N120 G00 W150.;
          :
```

[1] I,K type vector
[2] Intersection point calculation type vector

TEP097

When I, K has been commanded in a block without move command

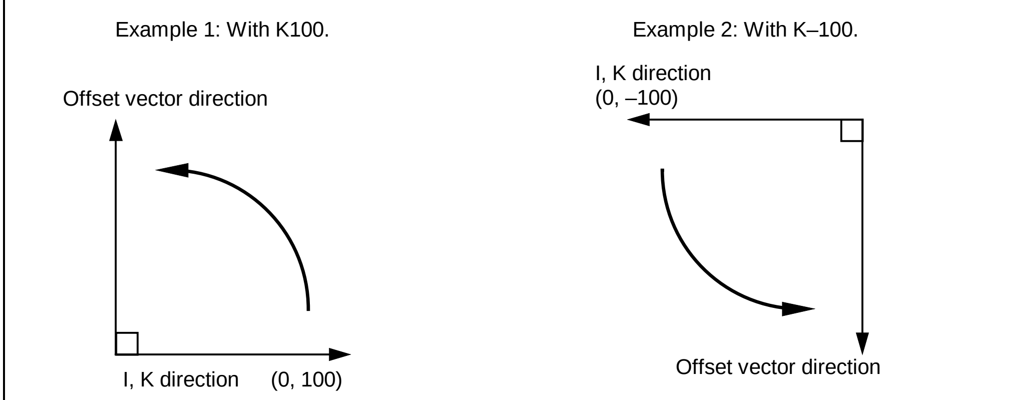


TEP098

3. Direction of offset vectors

A. In G41 mode

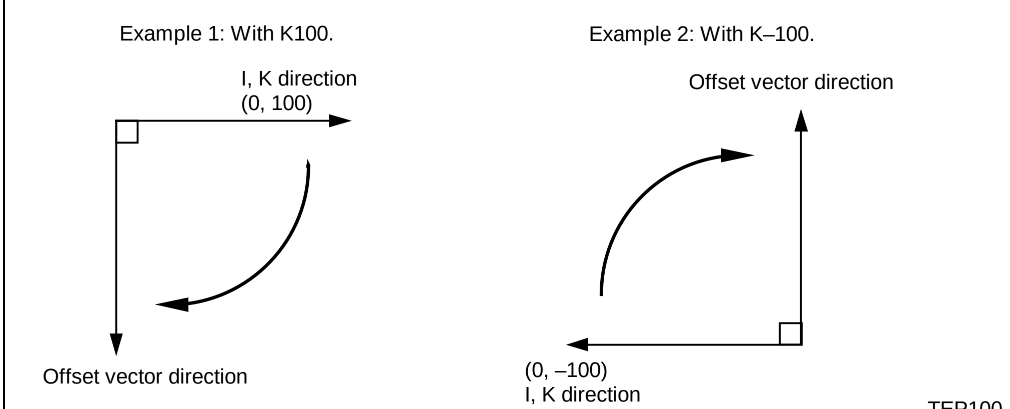
Direction produced by rotating the direction commanded by I, K vector through 90° to the left as seen from the forward direction of the Y-axis (third axis) to the zero point.



TEP099

B. In G42 mode

Direction produced by rotating the direction commanded by I, K vector through 90° to the right as seen from the forward direction of the Y-axis (third axis) to the zero point

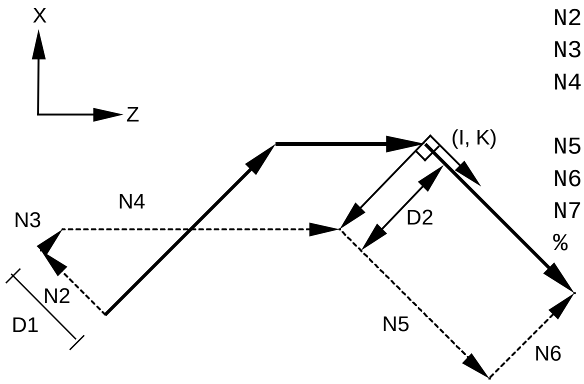


TEP100

Return to Library

4. Selection of offset modal

The G41 or G42 modal can be selected at any time.

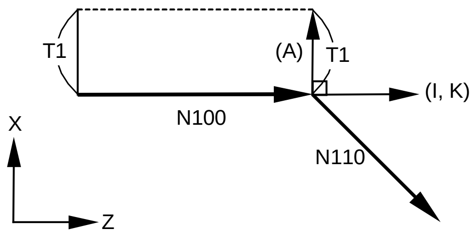



```
N1 G28 X0Z0;
N2 G41 T001T000M6D001 F1000;
N3 G01 U100.W100.;
N4 G42 W100.I-100.K100.
    T001T000M6D002;
N5 U-100.W100.;
N6 G40;
N7 M02;
%
```

TEP101

5. Offset stroke of offset vectors

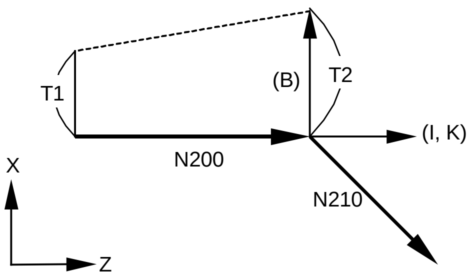
The offset stroke is determined by the offset number (modal) in the block including the I, K designation.




```
  (G41 T001T000M6D001)
     :
N100 G41 W150.K50.;
N110 U-100.W100.;
     :
```

TEP102

Vector (A) is the offset stroke entered in tool offset number modal 1 in the N100 block.



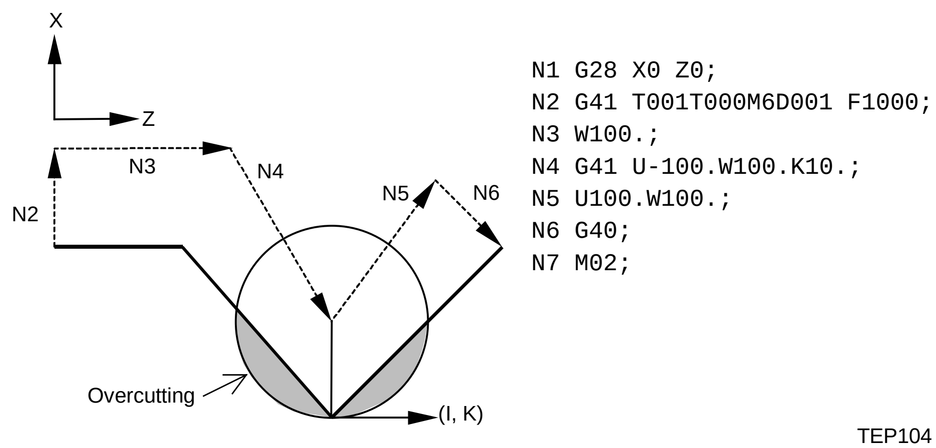

```
  (G41 T001T000M6D001)
     :
N200 G41 W150.K50.
     T001T000M6D002;
N210 U-100.W100.;
     :
```

TEP103

Vector (B) is the offset stroke entered in tool offset number modal 2 in the N200 block.

6. Notes

- Set the I, K type vector in a linear interpolation mode (G0, G1). If it is set in an arc interpolation mode at the start of compensation, program error will result.
  An I, K designation in an arc interpolation during the compensation mode functions as an arc center designation.
- When the I, K type vector has been designated, it is not deleted (avoidance of interference) even if there is interference. Consequently, overcutting may occur in such a case.



TEP104

7. Supplementary notes

Refer to the following table for the compensation methods based on the presence and/or absence of the G41 and G42 commands and I, K (J) command data.

G41/G42	I, K, (J)	Compensation method
No	No	Intersection point calculation type vector
Yes	No	Intersection point calculation type vector
Yes	Yes	I, K type vector, No insertion block

12-3-6 Interruptions during nose R/tool radius compensation

1. Interruption by MDI

Nose R/tool radius compensation is valid during automatic operation, whether it is based on the tape, memory, or MDI operation mode.

The following diagrams show what will occur if tape or memory operation is interrupted using the MDI function following termination of the program at a block:

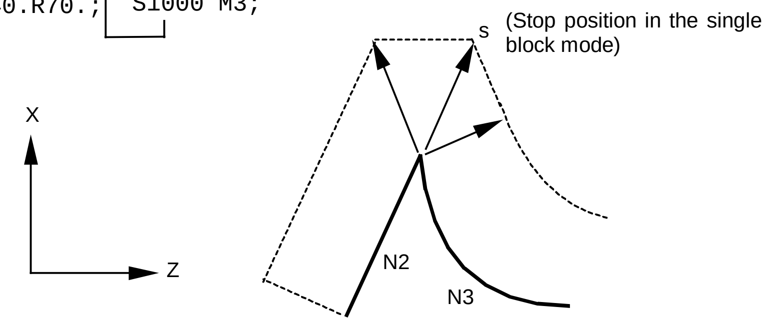
A. Interruption without movement

Tool path is not affected at all.

```
N1 G41 T001T000M6D001;
N2 U50.W20.;
N3 G3 U-40.W40.R70.;
```

MDI interruption

```
S1000 M3;
```



TEP105

B. Interruption with movement

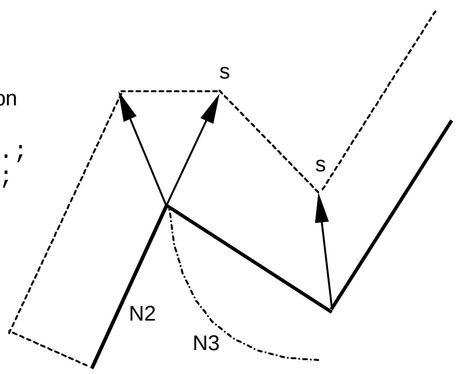
The offset vectors are recalculated automatically at the move command block after interruption.

With linear interruption

```
N1 G41 T001T000M6D001;
N2 U50.W20.;
N3 G3 U-40.W40. R70.;
```

MDI interruption

```
U-30.W50.;
U50.W30.;
```

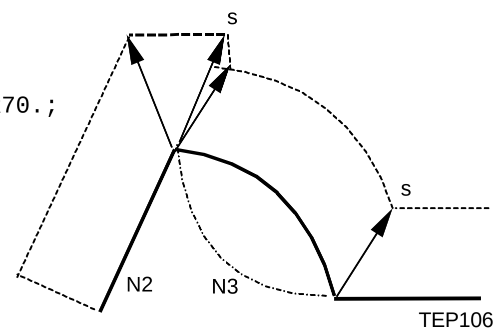


With arc interruption

```
N1 G40 T001T000M6D001;
N2 U50. W20.;
N3 G3 U-40.W40. R70.;
```

MDI interruption

```
G2 U-40.W40.R70.;
G1 W40.;
```

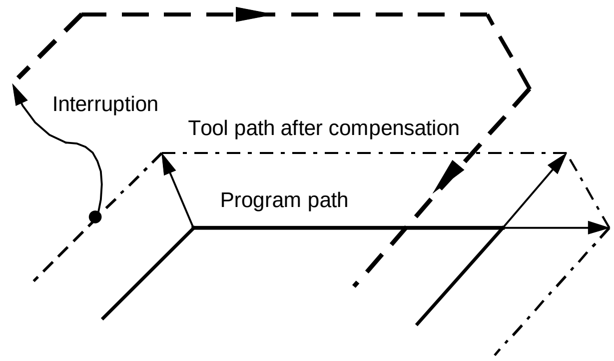


TEP106

2. Manual interruption

A. Interruption with manual absolute OFF

The tool path is shifted by an interruption amount.

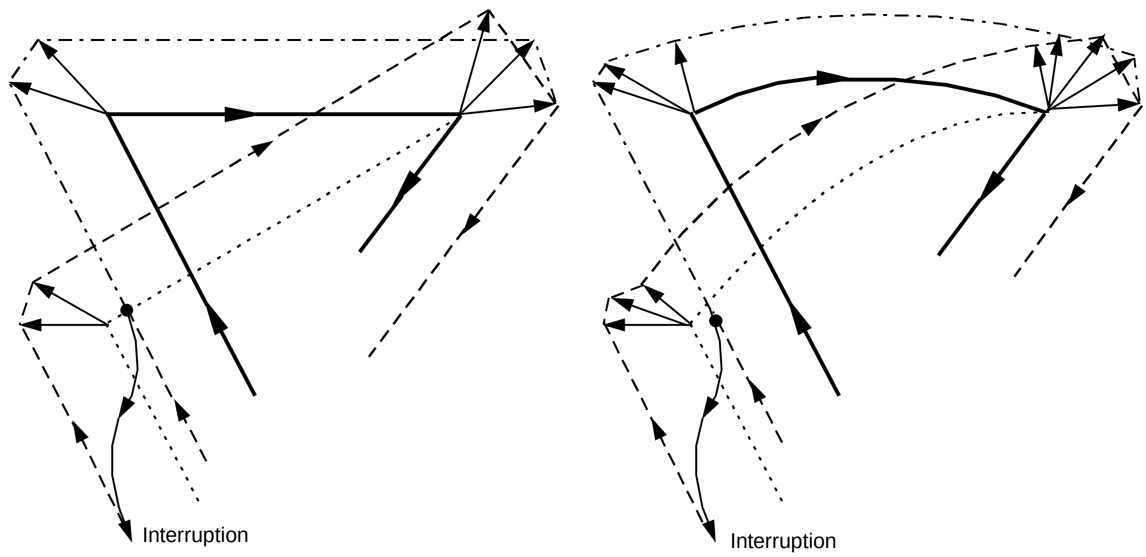


TEP107

B. Interruption with manual aboslute ON

In the incremental value command mode, the same operation results as with manual absolute OFF.

In the absolute value command mode, however, the tool returns to its original path at the ending point of the block following the interrupted block, as shown in the figure.



TEP108

Return to Library

12-3-7 General precautions on nose R/tool radius compensation

1. Selecting the amounts of compensation

The amounts of compensation are selected by specifying an offset number using a last one or two digits of the T-code. Depending on the machine specifications, the first digits may be used. Once a T-code has been set, it will remain valid until a new T-code is set.
T-codes are also used to select tool position offset data.

2. Updating the selected amounts of compensation

Updating of the selected amounts of compensation is usually to be done after a different tool has been selected during the compensation cancellation mode. If such updating is done during the compensation mode, vectors at the ending point of a block will be calculated using the offset data selected for that block.

3. Errors during tool nose radius compensation

1. An error results when any of the following commands are programmed during tool nose radius compensation.

   G17, G18, G19 (when a plane different from that selected during the compensation has been commanded)
   G31
   G74, G75, G76
   G81 to G89
2. An error results when an arc command is set in the first or last block of the tool nose radius compensation.
3. A programming error results during tool nose radius compensation when the intersection point is not determined with single block skip in the interference block processing.
4. A programming error results when an error occurs in one of the preread blocks during tool nose radius compensation.
5. A programming error results when interference can arise without no interference avoidance function during tool nose radius compensation.

12-3-8 Interference check

1. Overview

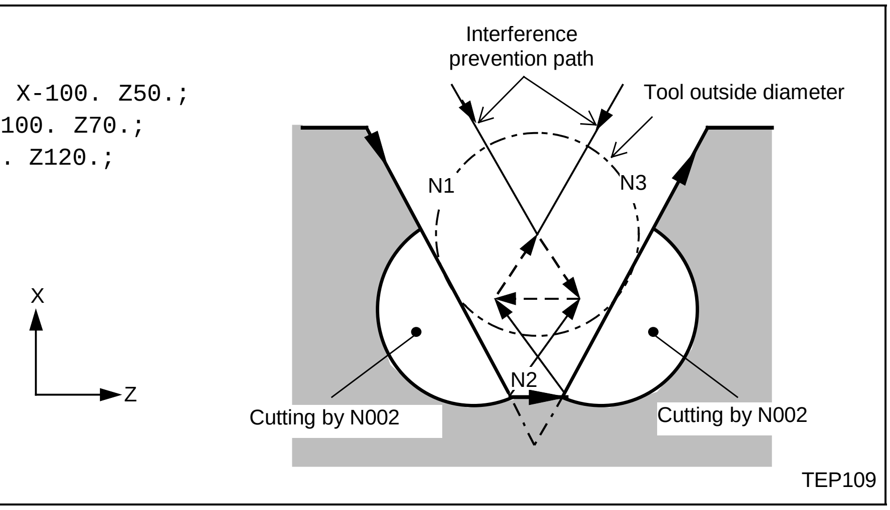
Even a tool whose nose radius has been compensated by usual tool nose R compensation based on two-block prereading may move into the workpiece to cut it. This status is referred to as interference, and a function for the prevention of such interference is referred to as interference check.

The following two types of interference check are provided and their selection is to be made using the parameter.

Function	Parameter (**F92** bit 5)	Operation
Interference check/alarm	Interference check/prevention OFF (**F92** bit 5 = 0)	The system will stop, with a program error resulting before executing the cutting block.
Interference check/prevention	Interference check/prevention ON (**F92** bit 5 = 1)	The path is changed to prevent cutting from taking place.

Example:

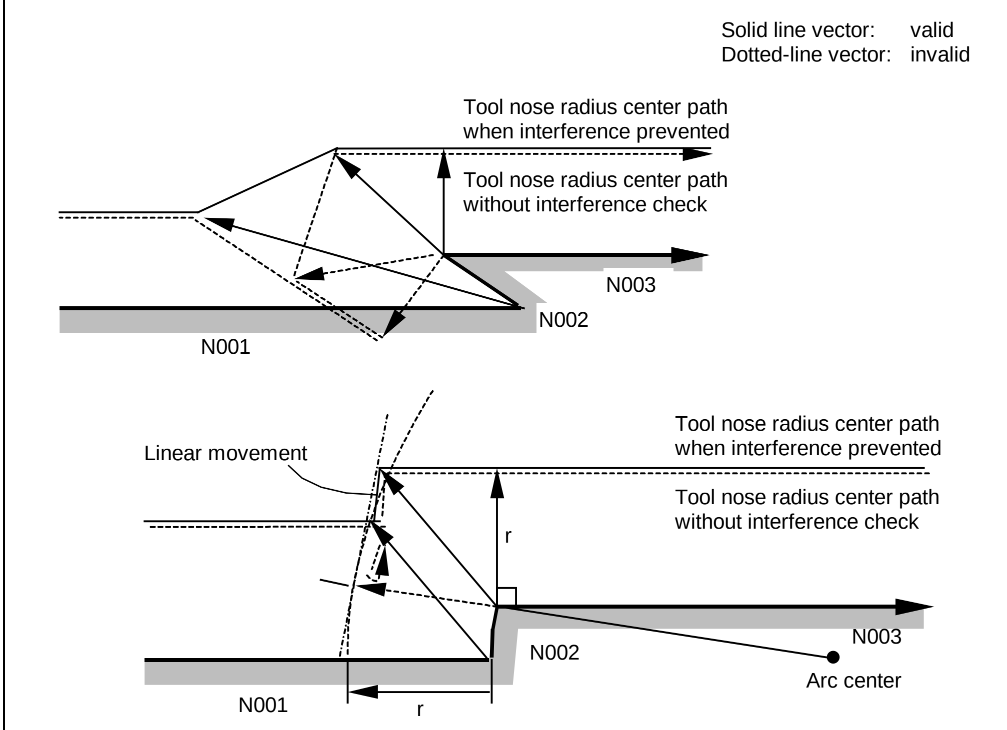
```
(G41)
N1  G1 X-100. Z50.;
N2  X-100. Z70.;
N3  X0. Z120.;
```



- For the alarm function
  An alarm occurs before N001 is executed. Machining can be continued by updating the program into, for example,
  N001 G1 X–100. Z–20.;
- Using the buffer correction function.
  For the interference check/prevention function
  Interference prevention vectors are generated by N001 and N003 intersection point calculation.

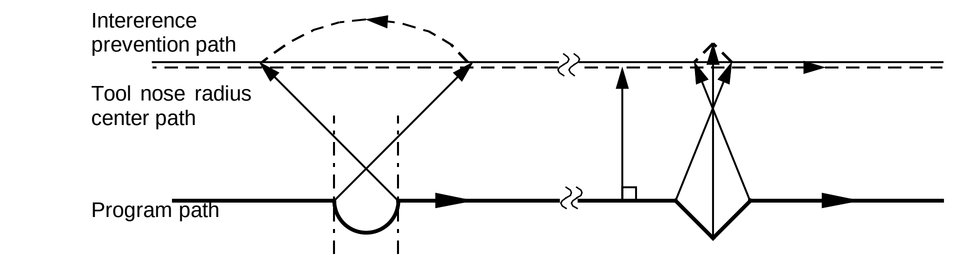
Return to Library

2. Operation during interference prevention



TEP110

In the case of the figure below, the groove will be left uncut.



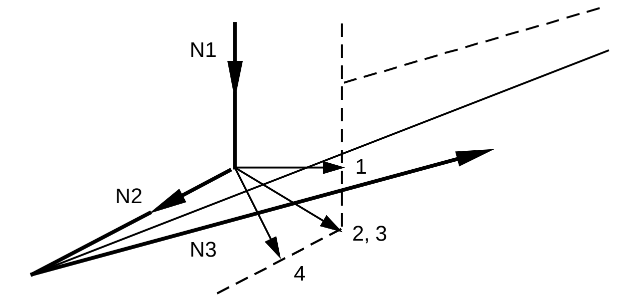
TEP111

3. Interference check/alarm

Cases that an interference check/alarm occurs are listed below.

A. When interference check/alarm is selected

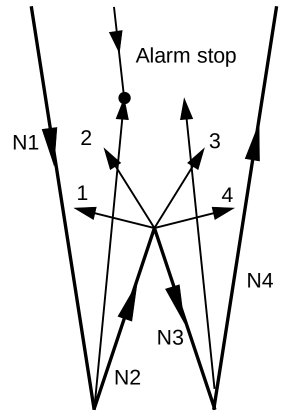
1. If all vectors at the ending point of the current block are erased:
   Prior to execution of N001, a program error will result if vectors 1 through 4 at the ending point of the N001 block are all erased as shown in the diagram below.



TEP112

B. When interference check/prevention is selected

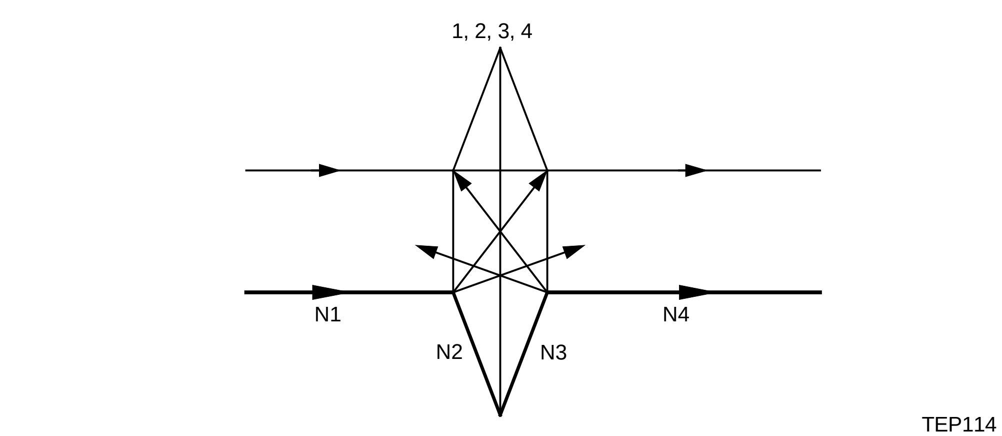
1. If all vectors at the ending point of the current block are erased but an effective vector(s) remains at the ending point of the next block:
   - For the diagram shown below, interference checking at N002 will erase all vectors existing at the ending point of N002, but leave the vectors at the ending point of N003 effective.



TEP113

At this time, a program error will occur at the ending point of N001.

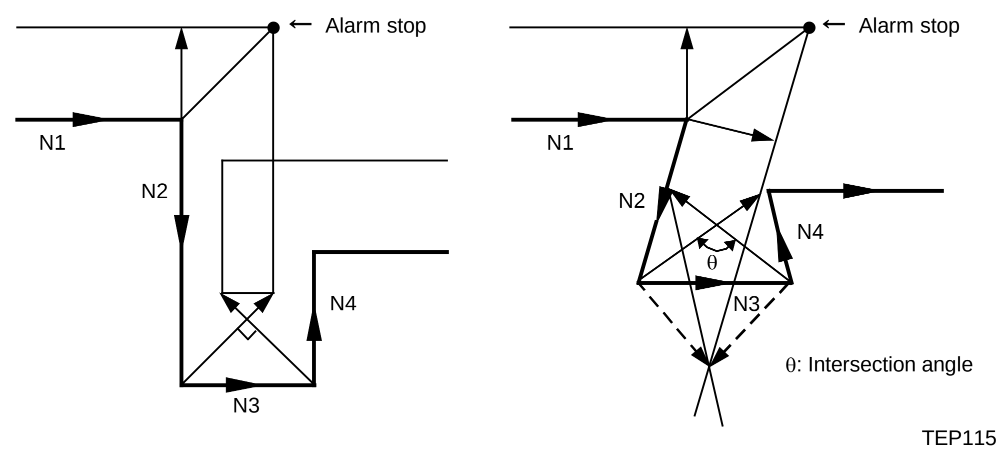
- For the diagram shown below, the direction of movement becomes opposite at N002. At this time, a program error will occur before execution of N001.



TEP114

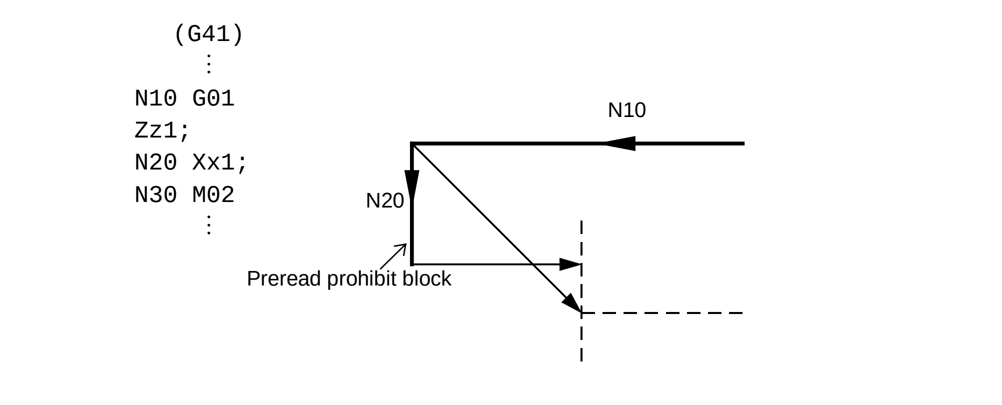
2. When prevention vectors cannot be generated:

- Prevention vectors may not be generated even when the conditions for generating them are satisfied. Or even after generation, the prevention vectors may interfere with N003.
  A program error will therefore occur at the ending point of N001 if those vectors intersect at angles of 90 degrees or more.



TEP115

- Prevention vectors may not be generated when preread prohibit blocks are interfered with and so program error occurs.

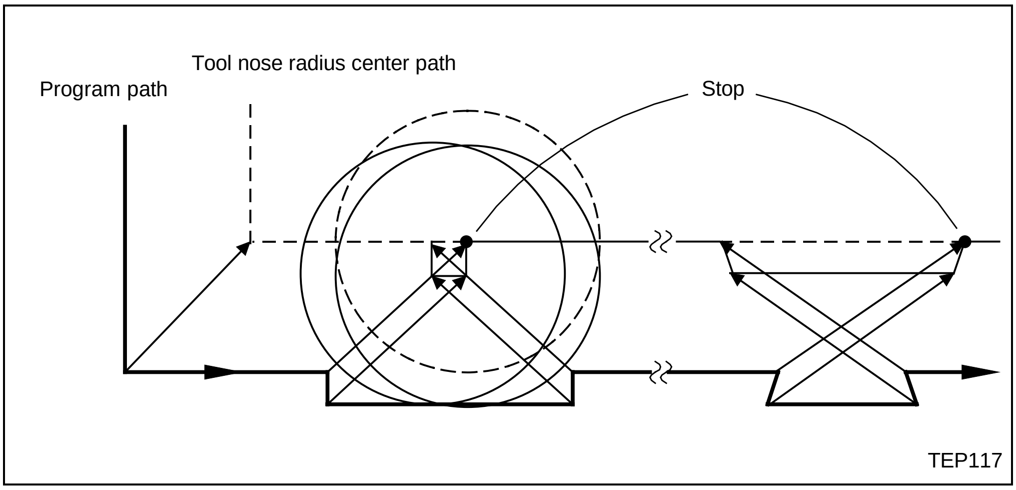


TEP116

Return to Library

3. When the after compensating moving direction of the tool is opposite to that of the program:

- For a program for the machining of parallel or downwardly extending grooves narrower than the tool diameter, interference may be regarded as occurring even if it is not actually occurring.



12-4 Programmed Data Setting: G10

1. Function and purpose

The G10 command allows tool offset data, work offset data and parameter data to be set or modified in the flow of program.

2. Programming formats

A. Programming workpiece offsets

- Programming format for the workpiece origin data

G10 L2 P_ X_ Y_ Z_α_ (α: Additional axis)

P: 0.....Coordinate shift (Added feature)
1.....G54
2.....G55
3.....G56
4.....G57
5.....G58
6.....G59

Data of P-commands other than those listed above are handled as P = 1.
If P-command setting is omitted, the workpiece offsets will be handled as currently effective ones.

- Programming format for the additional workpiece origin data (option)

G10 L20 P_ X_ Y_ Z_α_ (α: Additional axis)

P1: G54.1 P1
P2: G54.1 P2
⋮
P47: G54.1 P47
P48: G54.1 P48

The setting ranges of the data at axial addresses are as follows:

	Metric	Inch
Linear axis	±99999.9999 mm	±9999.99999 in.
Rotational axis	±99999.9999°	±99999.9999°

B. Programming tool offsets

- Programming format for the tool offset data of Type A

G10 L10 P_R_

P: Offset number
R: Offset amount

- Programming format for the tool offset data of Type B

G10 L10 P_R_ Geometric offset concerning the length
G10 L11 P_R_ Wear compensation concerning the length
G10 L12 P_R_ Geometric offset concerning the radius
G10 L13 P_R_ Wear compensation concerning the radius

- Programming format for the tool offset data of Type C

G10 L10 P_R_	Length offset; Geometric Z
G10 L11 P_R_	Length offset; Wear comp. Z
G10 L12 P_R_	Tool radius/Nose R offset (Geometric)
G10 L13 P_R_	Tool radius/Nose R offset (Wear comp.)
G10 L14 P_R_	Length offset; Geometric X
G10 L15 P_R_	Length offset; Wear comp. X
G10 L16 P_R_	Length offset; Geometric Y
G10 L17 P_R_	Length offset; Wear comp. Y
G10 L18 P_R_	Nose-R offset; Direction

The setting ranges for programming tool offset data are as follows:

Offset number (P):

1 to 128 or 512 (according to the number of available data sets)

Offset amount (R):

	Metric	Inch
TOOL OFFSET Type A	±1999.9999 mm	±84.50000 in.
TOOL OFFSET Type B Length Geom.	±1999.9999 mm	±84.50000 in.
TOOL OFFSET Type B Length Wear	±99.9999 mm	±9.99999 in.
TOOL OFFSET Type B Radius Geom.	±999.9999 mm	±84.50000 in.
TOOL OFFSET Type B Radius Wear	±9.9999 mm	±0.99999 in.
TOOL OFFSET Type C Geom. XYZ	±1999.9999 mm	±84.50000 in.
TOOL OFFSET Type C Geom. Nose R	±999.9999 mm	±84.50000 in.
TOOL OFFSET Type C Wear XYZ	±99.9999 mm	±9.99999 in.
TOOL OFFSET Type C Wear Nose R	±9.9999 mm	±0.99999 in.
TOOL OFFSET Type C Direction	0 - 9	0 - 9

C. Programming parameter data

G10 L50................ Parameter input mode ON
N_P_R_
N_R_
G11 .... Parameter input mode OFF

N: Parameter number
P: Axis number (for axis type parameter)
R: Data of parameter

Specify the parameters with address N as indicated below:

Parameter		N: Number	P: Axis No.
A	1 to 108	1001 to 1108	—
B	1 to 108	2001 to 2108	—
C	1 to 108	3001 to 3108	—
D	1 to 108	4001 to 4108	—
E	1 to 108	5001 to 5108	—
F	1 to 154 (47 to 66 excluded)	6001 to 6154	—
I	1 to 18	9001 to 9018	1 to 14
J	1 to 108	10001 to 10108	—
K	1 to 108	11001 to 11108	—
L	1 to 108	12001 to 12108	—
M	1 to 22	13001 to 13022	1 to 14
N	1 to 22	14001 to 14022	1 to 14
P	1 to 5	150001 to 150005	1 to 14
#	0 to 4095	150100 to 154195	1 to 14
S	1 to 22	16001 to 16022	1 to 14
SV	1 to 96	17001 to 17096	1 to 14
SP	1 to 384	18001 to 18384	1 to 4
SA	1 to 88	19001 to 19088	1 to 4
BA	1 to 132	20001 to 20132	—
TC	1 to 154	21001 to 21154	—

Note: As for the setting ranges of parameter data, refer to the Parameter List.

3. Detailed description

A. Workpiece origin data input

1. The G10 command is not associated with movement. However, do not use this command in the same block with a G-code command other than: G21, G22, G54 to G59.
2. Do not use the G10 command in the same block with a fixed cycle command or a sub-program call command. This will cause a malfunctioning or a program error.
3. Irrespective of workpiece offset type (G54 - G59 and G54.1), the data to the axial addresses have to refer to the origin of the fundamental machine coordinate system.
4. L-code and P-code commands can be omitted, indeed, but take notice of the following when omitting them:
   1) Omit both L-code and P-code commands only when

      The axial data should refer to the coordinate system that was last selected.
   2) The L-code command only may be omitted when the intended axial data refer to a coordinate system of the same type (in terms of L-code: L2 or L20) as the last selected one; give a P-command in such a case as follows:

      - Set an integer from 0 to 6 with address P to specify the coordinate shift data or one of the coordinate systems from G54 to G59.
      - Set an integer from 1 to 48 with address P to specify one of the additional workpiece coordinate systems of G54.1.
   3) If the P-code command only is omitted:

      An alarm will result if the value of L mismatches the coordinate system last selected.
5. Axis data without a decimal point can be entered in the range from –99999999 to +99999999. The data settings at that time depend upon the data input unit.

   Example: `G10 L2 P1 X–100. Y–1000 Z–100 B–1000`

   The above command sets the following data:

Metric system	X –100.	Y –1.	Z –0.1	B –1.
Metric system (up to 4 dec. places)	X –100.	Y –0.1	Z –0.01	B –0.1
Inch system	X –100.	Y –0.1	Z –0.01	B –1.
Inch system (up to 5 dec. places)	X –100.	Y –0.01	Z –0.001	B –0.1
6. The origin data updated by a G10 command are not indicated as they are on the **WORK OFFSET** display until that display has been selected anew.
7. Setting an illegal L-code value causes an alarm.
8. Setting an illegal P-code value causes an alarm.
9. Setting an illegal axial value causes an alarm.
10. The G10 command is invalid (or skipped) during tool path check.

B. Tool offset data input

1. The G10 command is not associated with movement. However, do not use this command in the same block with a G-code command other than: G21, G22, G54 to G59.
2. Do not use the G10 command in the same block with a fixed cycle command or a sub-program call command. This will cause a malfunctioning or a program error.
3. Offset data (R) without a decimal point can be entered in the range from –999999 to +999999 for geometric offset, or in the range from –99999 to +99999 for wear compensation. The data settings at that time depend upon the data input unit.

   Example: `G10 L10 P1 R1000`

   The above command sets the following data:

Metric system	1.
Metric system (up to 4 dec. places)	0.1
Inch system	0.1
Inch system (up to 5 dec. places)	0.01

4. The offset data updated by a G10 command are not indicated as they are on the **TOOL OFFSET** display until that display has been selected anew.
5. Setting an illegal L-code value causes an alarm.
6. A command of “G10 P_ R_” without an L-code is also available for tool offset data input.
7. Setting an illegal P-code value causes an alarm.
8. Setting an illegal offset value (R) causes an alarm.
9. The G10 command is invalid (or skipped) during tool path check.

C. Parameter data input

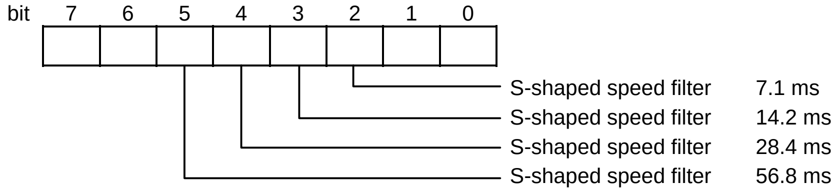
1. The G10 command is not associated with movement. However, do not use this command in the same block with a G-code command other than: G21, G22, G54 to G59.
2. Do not use the G10 command in the same block with a fixed cycle command or a sub-program call command. This will cause a malfunctioning or a program error.
3. Other NC statements must not be given in the parameter input mode.
4. No sequence number must be designated with address N in the parameter input mode.
5. Irrespective of the data input mode — absolute (G90) or incremental (G91) — the designated data will overwrite the existing parameter. Moreover, describe all the data in decimal numbers (hexadecimal and bit type data, therefore, must be converted).

   Example: For changing a bit type data of 0011011<u>0</u> to 0011011<u>1</u>:

   Since $(00110111)_2 = (55)_{10}$ [a binary number of 00110111 corresponds to “55” in decimal notation], set 55 with address R.
6. All decimal places, even if inputted, are ignored.

7. Some specific bit-type parameters require selection of one of multiple bits. For the parameter shown as an example below, set data that turns on only one of bits 2 to 5.

   Example: Parameter **K107**



   Setting “1” for bits 2 and 3, for example, could not make valid a speed filter of 21.3 msec (= 7.1 + 14.2).

8. The parameter data updated by a G10 L50 command are not made valid till the execution of a G11 command.
9. The parameter data updated by a G10 L50 command are not indicated as they are on the **PARAMETER** display until that display has been selected anew.
10. Setting an illegal L-code value causes an alarm.
11. Setting an illegal N-code value (parameter No.) causes an alarm.
12. Omission of P-code for an axis type parameter causes an alarm.
13. Setting an illegal parameter value with address R causes an alarm.
14. The G10 command is invalid (or skipped) during tool path check.

4. Sample programs

A. Entering tool offset data from tape

```
··· G10L10P10R–12345 G10L10P05R98765 G10L10P40R2468 ···
```

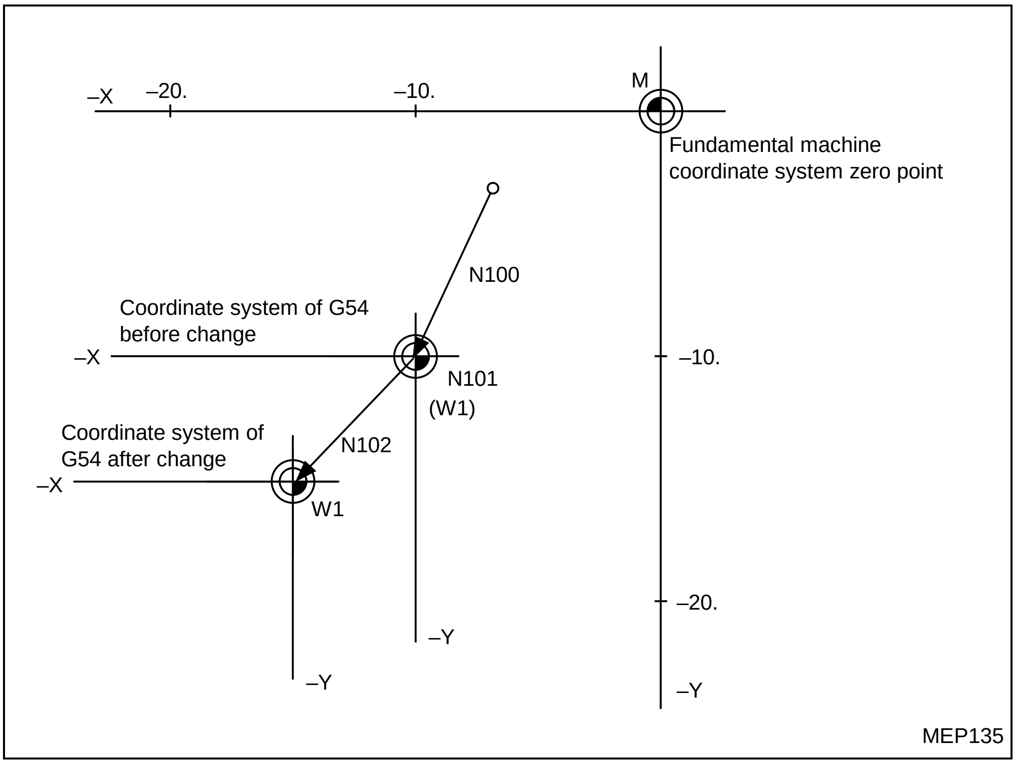
H10 = –12345 H05 = 98765 H40 = 2468

B. Updating the workpiece coordinate system offset data

Assume that the previous workpiece coordinate system offset data is as follows:

X = –10.000   Y = –10.000

```
  ⋮
N100  G00  G90  G54  X0  Y0
N101  G10  L2  P1  X–15.000  Y–15.000
N102   X0  Y0
  ⋮
M02
```



Note 1: Changes in the display of the workpiece position at N101
At N101, the display of tool position in the G54 coordinate system changes before and after workpiece coordinate system updating with G10.

X = 0 ⇨ X = +5.000
Y = 0   Y = +5.000

Note 2: Prepare the following program to set workpiece coordinate system offset data in G54 to G59:

```
G10L2P1X–10.000  Y–10.000
G10L2P2X–20.000  Y–20.000
G10L2P3X–30.000  Y–30.000
G10L2P4X–40.000  Y–40.000
G10L2P5X–50.000  Y–50.000
G10L2P6X–60.000  Y–60.000
```

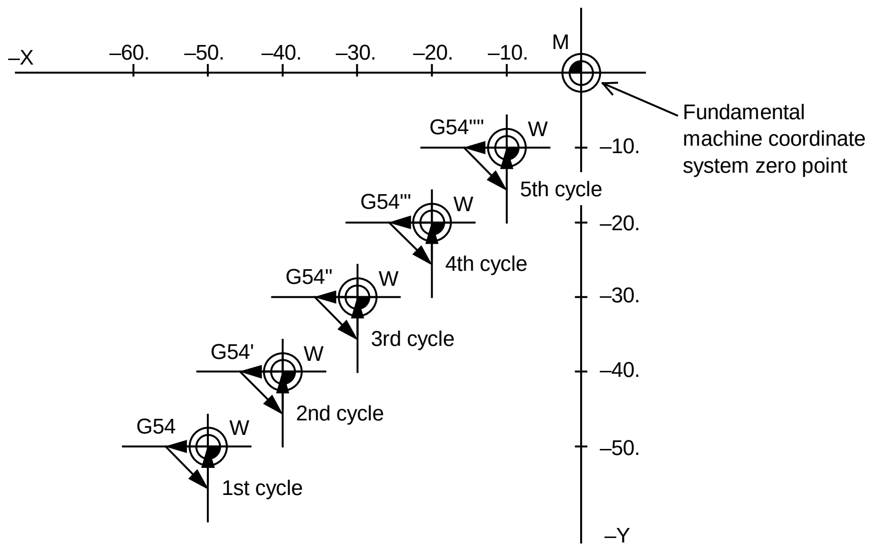
C. Programming for using one workpiece coordinate system as multiple workpiece coordinate systems

Main program

```
  :
#1=–50.  #2=10.
M98  P200  L5
  :
M02
%
```

Subprogram (O200)

```
N1 G90  G54  G10  L2  P1  X#1  Y#1
N2 G00  X0   Y0
N3 X–5. F100
N4 X0   Y–5.
N5 Y0
N6 #1=#1+#2
N7 M99
%
```



MEP136

D. Programming for parameter data input

`G10L50`	Parameter input mode ON
`N4017R10`	**D17** is set to “10”.
`N6088R96`	**F88** is set to “01100000”. [$(01100000)_2$ = $(96)_{10}$]
`N12067R–1000`	**L67** is set to “–1000”.
`N12072R67`	**L72** is set to “0x43”. [$(43)_{16}$ = $(67)_{10}$]
`N150004P1R50`	**P4** data for the 1st axis (X-axis) is set to “50”.
`G11`	Parameter input mode OFF

5. Related alarms

Alarm No.	Alarm message	Cause	Remedy
807	ILLEGAL FORMAT	Work offset setting: P-command is omitted in a block of G10 L20 (or L2) although the last selected coordinate system is one of the systems from G54 to G59 (or of the G54.1 systems). Parameter setting: An illegal parameter number is set.	Review the program data.
809	ILLEGAL NUMBER INPUT	Work offset setting: The setting range of the coordinate system number or the offset data is overstepped. Tool offset setting: The setting range of the offset data is overstepped. Parameter setting: The axis number is not specified for an axis type parameter. The setting range of the axis number or the parameter data is overstepped.	Review the program data.
839	ILLEGAL OFFSET No.	Tool offset setting: The specified offset number is greater than the number of available data sets.	Correct the offset number according to the number of available data sets.
903	ILLEGAL G10 L NUMBER	Work offset setting: A command of G10 L20 is set although the corresponding function for the G54.1 coordinate systems is not provided.	Give an available L-code command.

12-5 Tool Offsetting Based on MAZATROL Tool Data

Tool length and diameter offset can be performed on the basis of the MAZATROL tool data (diameter and length data) by particular parameter setting.

12-5-1 Selection parameters

Using the following parameters, select whether or not MAZATROL tool data is to be used:
User parameters

F92 bit 7 = 1: Tool diameter offsetting uses the MAZATROL tool data **ACT-ϕ** (tool diameter data).

F93 bit 3 = 1: Tool length offsetting uses the MAZATROL tool data **LENGTH** (tool length data).

F94 bit 2 = 1: Tool length offsetting using the MAZATROL tool data is prevented from being cancelled by a reference-point return command.

F94 bit 7 = 1: Tool offsetting uses the MAZATROL tool data **ACT-ϕ CO.** (or **No.**) and **LENG CO.** (or **No.**).
(Set **F94** bit 7 to 0 to use the data stored on the **TOOL OFFSET** display.)

1. Tool length offsetting

<table>
<tr><th colspan="2" rowspan="2">Data items used</th><th colspan="2">Parameter</th><th rowspan="2">Programming format</th><th rowspan="2">Remarks</th></tr>
<tr><th>F93 bit 3</th><th>F94 bit 7</th></tr>
<tr><td>TOOL OFFSET</td><td>Tool offset No.</td><td>0</td><td>0</td><td>G43/G44 H_ (P_)</td><td></td></tr>
<tr><td rowspan="3">TOOL DATA (MAZATROL)</td><td>LENGTH [1]</td><td rowspan="2">1</td><td rowspan="2">1</td><td>T_</td><td></td></tr>
<tr><td>LENGTH [1] + OFFSET No.
or
LENGTH + LENG CO. [2]</td><td>T_ + H_</td><td>- Length offset cancellation not required for tool change.
- G43 not required.</td></tr>
<tr><td>OFFSET No. or
LENG CO. [2]</td><td>0</td><td>1</td><td>G43/G44 H_</td><td>Length offset cancellation required for tool change. [3]</td></tr>
<tr><td>TOOL OFFSET +
TOOL DATA</td><td>Tool offset No. + LENGTH [1]</td><td>1</td><td>0</td><td>(G43/G44 H_)
+ (T_) (P_)</td><td>Length offset cancellation required for tool change. [3]</td></tr>
</table>

[1] TOOL LENGTH data for milling tools, and LENGTH A and LENGTH B for turning tools.
[2] LENG CO. data are only used for milling tools.
[3] Canceling method - Set G49 before tool change command.
- Set G28/G30 before tool change command (when **F94** bit 2 = 0).

2. Tool diameter offsetting

<table>
<tr><th colspan="2" rowspan="2">Data items used</th><th colspan="2">Parameter</th><th rowspan="2">Programming format</th></tr>
<tr><th>F92 bit 7</th><th>F94 bit 7</th></tr>
<tr><td>TOOL OFFSET</td><td>Tool offset No.</td><td>0</td><td>0</td><td>G41/G42 D_</td></tr>
<tr><td rowspan="2">TOOL DATA (MAZATROL)</td><td>ACT-ϕ + ACT-ϕ CO. or
ACT-ϕ + OFFSET No.</td><td>1</td><td>1</td><td>G41/G42 T_</td></tr>
<tr><td>ACT-ϕ CO. or
OFFSET No.</td><td>0</td><td>1</td><td>G41/G42 T_</td></tr>
<tr><td>TOOL OFFSET +
TOOL DATA</td><td>Tool offset No. + ACT-ϕ</td><td>1</td><td>0</td><td>G41/G42 D_ + T_</td></tr>
</table>

Return to Library

3. Nose-R compensation

Data items used		Parameter F92 bit 7	Parameter F94 bit 7	Programming format
TOOL OFFSET	Tool offset No.	0	0	G41/G42 D_
TOOL DATA (MAZATROL)	NOSE-R + OFFSET No.	1	1	G41/G42 T_
	OFFSET No.	0	1	G41/G42 T_
TOOL OFFSET + TOOL DATA	Tool offset No. + NOSE-R	1	0	G41/G42 D_ + T_

12-5-2 Tool diameter offsetting

1. Function and purpose

Tool diameter offsetting by a G41 or G42 command uses MAZATROL tool data **ACT-ϕ** as the offset amounts.

2. Parameter setting

Set bit 7 of parameter **F92** to 1.

3. Detailed description

- Tool diameter offsetting uses as its offset amounts the diameter data of the tool which is mounted in the spindle at the issuance of G41/G42.
- Tool diameter offsetting is cancelled by G40.
- If the tool diameter offset function is used with a D-command, the sum total of the data indicated by the corresponding offset number (D) and the radius of the tool will be used as the offset data.

Note 1: The tool used must be mounted in the spindle before restarting the program.

Note 2: Offsetting based on tool diameter data will not occur if registered MAZATROL tool diameter data is not present or if a tool for which tool diameter data cannot be entered is to be used.

Note 3: To carry out for an EIA/ISO program the radius compensation operations using the tool diameter data included in MAZATROL tool data, it is necessary to insert tool change command blocks, as is the case with tool length offsetting (refer to Note 5 in Subsection 13-7-2).

12-5-3 Tool data update (during automatic operation)

1. Function and purpose

Tool Data Update allows MAZATROL tool data to be updated during automatic operation based on an EIA/ISO program.

2. Parameter setting

Set parameter **L57** to 1.

3. Detailed description

This function allows the entire tool data, except for spindle tools, to be updated during automatic operation based on an EIA/ISO program.

Parameter	TOOL	NOM-ϕ	ACT-ϕ	LENGTH	COMP.	THR/HP	LIFE	TIME	MAT.	REV.
L57 = 0	No	No	No	No	No	No	Yes	Yes	No	Yes
L57 = 1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Note 1: In the table given above, “Yes” indicates that you can update the data, and “No” indicates that you cannot update the data.
Identification between MAZATROL programs and EIA/ISO programs is automatically made by whether the program currently being executed, is MAZATROL or EIA/ISO, irrespective of whether it is a main program or subprogram.
If, however, the main program is MAZATROL and its subprograms are EIA/ISO, then the currently active set of programs is regarded as a MAZATROL program.

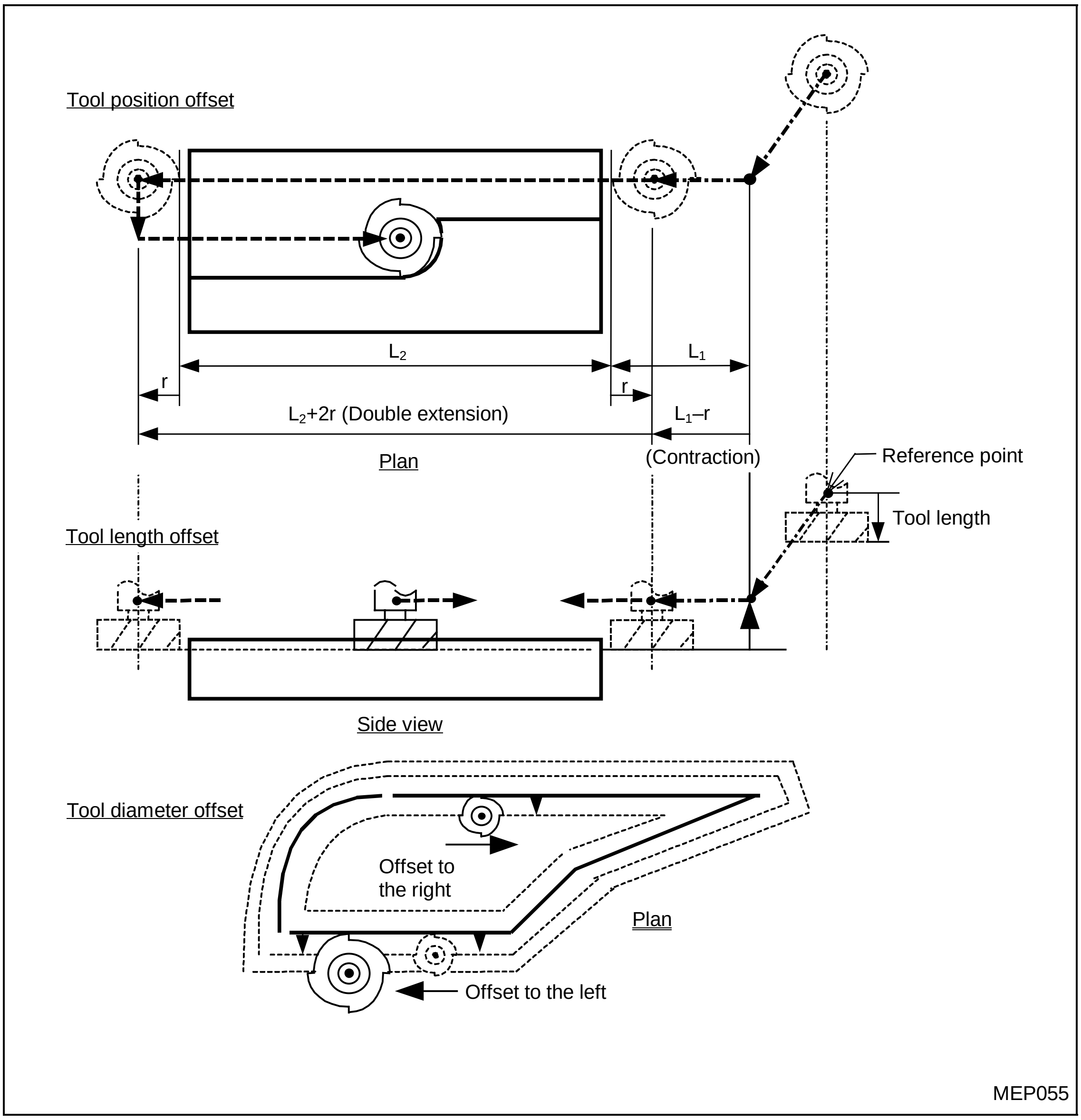
Note 2: An alarm **428 MEMORY PROTECT (AUTO OPERATION)** will occur if the spindle tool data is modified during automatic operation based on an EIA/ISO program.

13 TOOL OFFSET FUNCTIONS (FOR SERIES M)

13-1 Tool Offset

1. Overview

As shown in the diagram below, three types of basic tool offset functions are available: tool position offset, tool length offset, and tool diameter offset.
These three types of offset functions use offset numbers for designation of offset amount. Set the amount of offset directly using the operation panel or by applying the function of programmed parameter input. MAZATROL tool data can also be used for tool length offset or tool diameter offset operations according to the parameter setting.



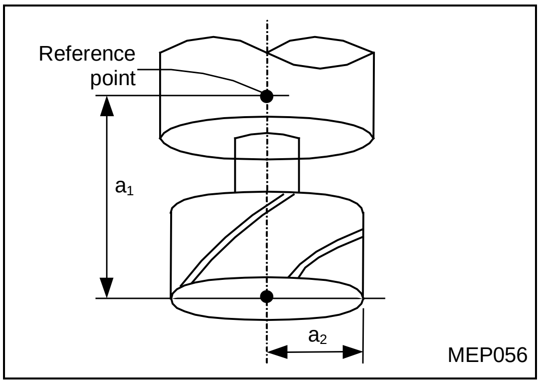
2. Selecting the amounts of tool offset

The amounts of tool offset corresponding to the offset numbers must be prestored on the **TOOL OFFSET** display by manual data input method or programmed data setting function (G10).

The mounts of tool offset can be selected using one of the following three types:

A. Type A

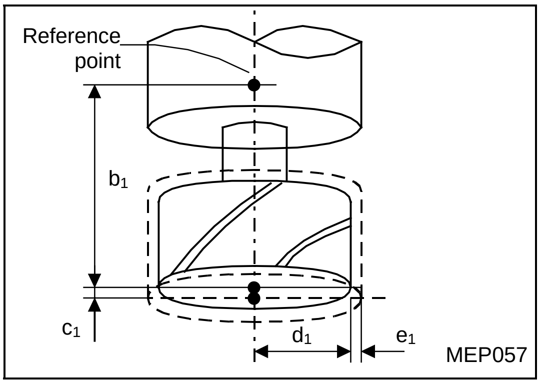
The same amount of offset will be set if identical offset numbers are selected using commands D and H.



(Dn) = a_n
(Hn) = a_n

B. Type B

Set an H-code and D-code, respectively, to use the total sum of the geometric offset amount and the wear compensation amount for tool length offset and tool diameter offset.



(Hn) = $b_n + c_n$
(Dn) = $d_n + e_n$

C. Type C

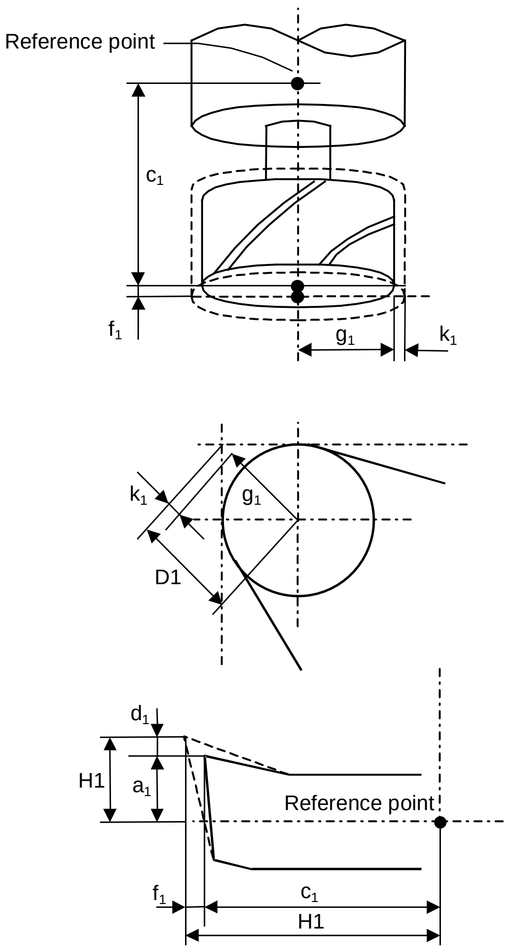
Data items used for turning tools are as follows:

X, Y, Z, and Nose-R of Geometric Offset,
X, Y, Z, and Nose-R of Wear Compensation, and
Direction.

Data items used for milling tools are as follows:

Z and Nose-R of Geometric Offset, and
Z and Nose-R of Wear Compensation.

Set an H-code and D-code, respectively, to use the total sum of the geometric offset amount and the wear compensation amount for tool length offset and tool diameter offset.



$(Hn) = c_n+f_n$
$(Dn) = g_n+k_n$

$(Hn)\ (x,y,z) = (a_n+d_n,\ b_n+e_n,\ c_n+f_n)$
$(Dn) = g_n+k_n$

Return to Library

3. TOOL OFFSET display types

As a data storage area for tool offsetting functions, three types of the **TOOL OFFSET** display are provided: Type A, Type B, and Type C.

Type	Length/Diameter distinguished	Geometric/Wear distinguished	Geometric/Wear for each axis distinguished	Milling	Turning
A	No	No	No	Used	Not used
B	Yes	Yes	No	Used	Not used
C	Yes	Yes	Yes	Used	Used

A. Type A

As listed in the table below, one offset data is given for one offset number. No distinction is drawn between length, diameter, geometric and wear compensation amounts. That is, one set of offset data comprises all these four factors.

(D1) = a_1, (H1) = a_1
(D2) = a_2, (H2) = a_2
⋮ ⋮
(Dn) = a_n, (Hn) = a_n

Offset No.	Offset amount
1	a_1
2	a_2
3	a_3
⋮	⋮
⋮	⋮
n	a_n

B. Type B

As listed in the table below, two types of offset data can be set for one offset number. That is, different amounts of geometric offset and wear compensation can be set for each of the selected tool length and the selected tool diameter.
Use command H to select offset data concerning the tool length, and use command D to select offset data concerning the tool diameter.

(H1) = $b_1 + c_1$, (D1) = $d_1 + e_1$
(H2) = $b_2 + c_2$, (D2) = $d_2 + e_2$
⋮ ⋮
(Hn) = $b_n + c_n$, (Dn) = $d_n + e_n$

Offset No.	Tool length (H)		Tool diameter (D) / (Position offset)	
	Geometric offset	Wear compensation	Geometric offset	Wear compensation
1	b_1	c_1	d_1	e_1
2	b_2	c_2	d_2	e_2
3	b_3	c_3	d_3	e_3
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
n	b_n	c_n	d_n	e_n

C. Type C (for turning tools and milling tools)

As tabulated below, various types of offset data can be set for one offset number: Geometric offset and Wear compensation data (X, Y, Z) for tool length, Geometric offset and Wear compensation data for tool diameter, and Direction.

Data items used for milling tools are: Geometric offset Z and Wear compensation Z (Length) and Geometric offset and Wear compensation (Diameter).
Data items used for turning tools are: Geometric offset X, Y, Z and Wear compensation X, Y, Z (Length), Geometric offset and Wear compensation (Diameter), and Direction.

For milling tools

(H1) = $c_1 + f_1$, (D1) = $g_1 + k_1$
(H2) = $c_2 + f_2$, (D2) = $g_2 + k_2$
⋮ ⋮
(Hn) = $c_n + f_n$, (Dn) = $g_n + k_n$

For turning tools

(H1) (x, y, z) = (a_1+d_1, b_1+e_1, c_1+f_1), (D1) = g_1+k_1
(H2) (x, y, z) = (a_2+d_2, b_2+e_2, c_2+f_2), (D2) = g_2+k_2
⋮ ⋮
(Hn) (x, y, z) = (a_n+d_n, b_n+e_n, c_n+f_n), (Dn) = g_n+k_n

Tool offset No.	Length (H)						Diameter (D)/(Position offset) Nose-R (D)		Direction
	Geometric offset			Wear comp.			Geometric offset	Wear comp.	
	X	Y	Z	X	Y	Z	R	R	
1	a_1	b_1	c_1	d_1	e_1	f_1	g_1	k_1	l_1
2	a_2	b_2	c_2	d_2	e_2	f_2	g_2	k_2	l_2
3	a_3	b_3	c_3	d_3	e_3	f_3	g_3	k_3	l_3
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	a_n	b_n	c_n	d_n	e_n	f_n	g_n	k_n	l_n

4. Tool offset numbers (H/D)

Tool offset numbers can be selected using address H or D.

- Use address H to offset the selected tool length. Use address D to offset the selected tool position or the selected tool diameter.
- Once a tool offset number has been selected, it will remain unchanged until a new H or D is used.
- Offset numbers can be set only once for one block. If offset numbers are set more than once for one block, only the last offset number will be used.
- The maximum available number of sets of offset numbers is as follows:
  Standard: 128 sets: H01 to H128 (D01 to D128)
  Optional: 512 sets: H01 to H512 (D01 to D512)
- The alarm **839 ILLEGAL OFFSET No.** will result if these limits are exceeded.

- The offset data range is as listed in the table below.
  Offset data for each offset number must be set beforehand on the **TOOL OFFSET** display.

	Micron system		Sub-micron for rotational axes		Sub-micron for all axes	
	Metric	Inch	Metric	Inch	Metric	Inch
TOOL OFFSET Type A	±9999.9999 mm	±845.0000 in.	±1999.999 mm	±84.5000 in.	±1999.9999 mm	±84.50000 in.
TOOL OFFSET Type B Length Geom.	±9999.9999 mm	±845.0000 in.	±1999.999 mm	±84.5000 in.	±1999.9999 mm	±84.50000 in.
TOOL OFFSET Type B Length Wear	±99.9999 mm	±9.9999 in.	±99.999 mm	±9.9999 in.	±99.9999 mm	±9.99999 in.
TOOL OFFSET Type B Dia. Geom.	±999.9999 mm	±99.9999 in.	±999.999 mm	±84.5000 in.	±999.9999 mm	±84.50000 in.
TOOL OFFSET Type B Dia. Wear	±9.9999 mm	±0.9999 in.	±9.999 mm	±0.9999 in.	±9.9999 mm	±0.99999 in.
TOOL OFFSET Type C Geom. XYZ	±9999.9999 mm	±845.0000 in.	±1999.999 mm	±84.5000 in.	±1999.9999 mm	±84.50000 in.
TOOL OFFSET Type C Geom. Nose-R	±999.9999 mm	±99.9999 in.	±999.999 mm	±84.5000 in.	±999.9999 mm	±84.50000 in.
TOOL OFFSET Type C Wear XYZ	±99.9999 mm	±9.9999 in.	±99.999 mm	±9.9999 in.	±99.9999 mm	±9.99999 in.
TOOL OFFSET Type C Wear Nose-R	±9.9999 mm	±0.9999 in.	±9.999 mm	±0.9999 in.	±9.9999 mm	±0.99999 in.

Note: The tool offset number (H- or D-code) is not be made effective if it is not designated in the corresponding offset mode.

5. Number of sets of tool offset numbers

The maximum available number of sets of tool offset numbers depends on the particular machine specifications.

	Number of tool offset combinations (max.)
Standard specifications	128
Optional specifications	512

Note: The maximum available number of sets of tool offset numbers under optional machine specifications refers to the total number of sets of tool offset numbers including those available under the standard machine specifications.

13-2 Tool Length Offset/Cancellation: G43, G44, or T-code/G49

1. Function and purpose

Commands G43 and G44 allow the ending point of execution of move commands to be shifted through the previously set offset amount for each axis. Any deviations between programmed tool lengths/diameters and actual lengths/diameters can be set as offset data using these commands to make the program more flexible.

2. Programming format

G43 Zz Hh (Pp)	Tool length offset +
G44 Zz Hh (Pp)	Tool length offset –
G49 Zz	Cancellation of tool length offset

There are two types of tool length offset: for milling tools and for turning tools.

For milling tools: Length offsetting is executed on the axis specified in the G43 or G44 block (unless the length offset axis is fixed to “Z” by a parameter setting [**F92** bit 3 = 1]).
For turning tools: Length offsetting is executed on all axes for which offset amounts are registered (and G49 cancels all offset amounts concerned).

Add an argument P as follows to designate the tool type. Note that the offset type for turning tools is to be selected in a measurement program using a touch sensor.

Tool type	Designation
Milling tool	The value of P is 0 (P0), or P is omitted.
Turning tool	The value of P is 1 (P1).

3. Detailed description

The maximum available number of sets of offset numbers is as follows:

Standard: 128 sets : H1 to H128
Optional: 512 sets : H1 to H512

where the maximum available number of sets of offset numbers refers to the total number of sets of offset numbers including those concerning the tool length, the tool position, and the tool diameter.

The following represents the relationship between the programming format and the stroke of movement after offsetting.

A. Tool length offsetting for milling tools

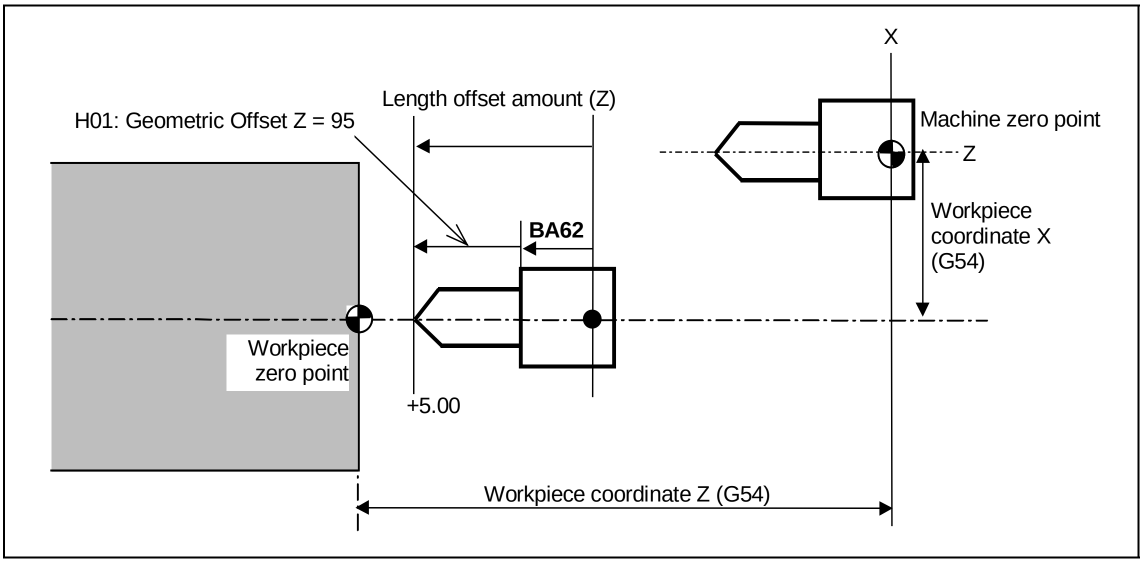
1. Z-axis motion distance

$G43Z\pm zHh_1$	$\pm z + \{\pm\ell h_1 - (\pm\ell h_0)\}$	Positive-direction offset by length offset amount
$G44Z\pm zHh_1$	$\pm z + \{\pm\ell h_1 - (\pm\ell h_0)\}$	Negative-direction offset by length offset amount
$G49Z\pm z$	$\pm z - (\pm\ell h_1)$	Cancellation of the offset amount

ℓh_1: **BA62** + Value of offset No. h_1
ℓh_0: Offset amount existing before the G43 or G44 block

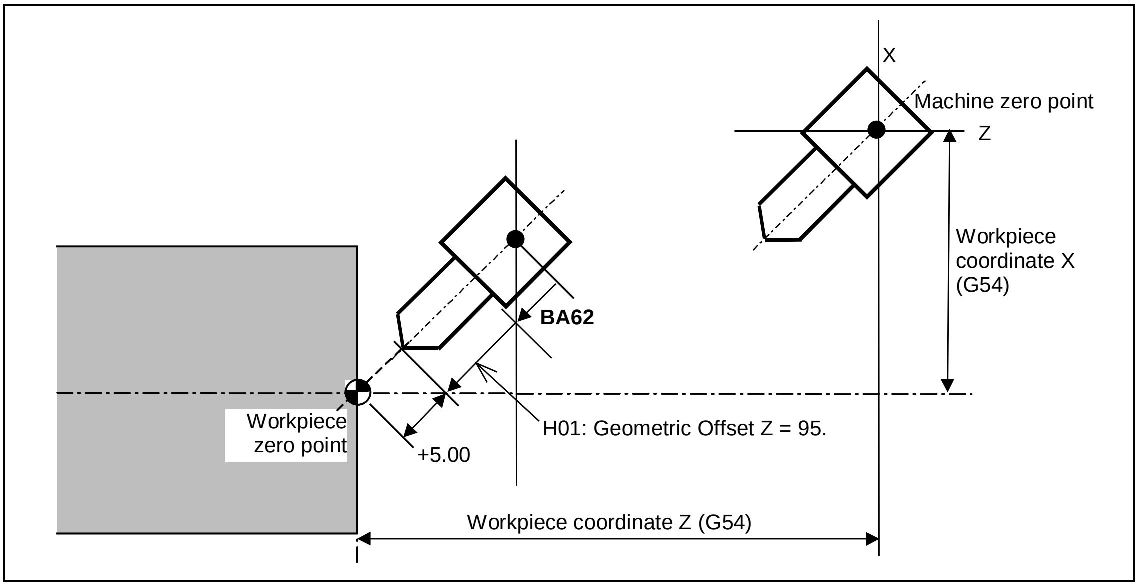
Irrespective of whether absolute or incremental programming method is used, the actual ending point coordinates are calculated by offsetting the programmed end point coordinate through the offset amount.
The initial state (upon turning-on or after M02) is of G49 (tool length offset cancellation).

2. Sample programs



For absolute data input
(H01: Z = 95.)

```
N001  G90  G94  G00  G40  G80
N002  G91  G28  Z0 X0
N003  T01  T00  M06
N004  G90  G54  X0 Y0
N005  G43  Z5.  H01
N006  G01  Z-50. F100
```



For absolute data input
(H01: Z = 95.)

```
N001  G90  G94  G00  G40  G80
N002  G91  G28  Z0 X0 B0
N003  T01  T00  M06
N004  G90  G54  G00 B45.
N005  G68 X0 Y0 Z0 I0 J1 K0 R45.
N006  G00 X0 Y0
N007  G43  Z5.  H01
N008  G01  Z-50. F100
```

Return to Library

3. Supplement

1) Tool length offset data can be set for the X-axis, the Y-axis, and additional axes, as well as the Z-axis. Whether the offset data is to be used for the Z-axis only or for the axis specified in the G43 or G44 block can be selected using bit 3 of parameter **F92**.

2) Even if multiple axis addresses are programmed in one block, offsetting will be performed on only one of the axes and the priority in this case is as follows:

C > Z > B > Y > X > A

Example:

G43 Xx_1 Hh_1
⋮ — Positive-direction offset on the X-axis, and cancellation
G49 Xx_2

G44 Yy_3 Hh_3
⋮ — Negative-direction offset on the Y-axis, and cancellation
G49 Yy_4

G43 $\alpha\alpha_5$ Hh_5
⋮ — Pos.-direct. offset on the additional axis, and cancellation
G49 $\alpha\alpha_6$

G43 Xx_7 Yy_7 Zz_7 Hh_7.............. Positive-direction offset on the Z-axis

3) Offsetting is always performed on the Z-axis if no axis addresses are programmed in the G43 or G44 block.

Example:

G43 Hh_1
⋮ — Offsetting on the Z-axis, and cancellation
G49

4) If reference point (zero point) return is performed in the offsetting mode, the mode is cancelled after completion of the returning operation.

Example:

G43 Hh_1
⋮ — Upon completion of return to the reference point (zero point), the offset stroke is cleared.
G28 Zz_2

G43 Hh_1
G49 G28 Zz_2 — Reference point return after a Z-axis motion at the current position for clearing the offset amount

5) If command G49 or H00 is executed, length offsetting will be immediately cancelled (the corresponding axis will move to clear the offset amount to zero).
When using MAZATROL tool data, do not use G49 as a cancellation command code; otherwise interference with the workpiece may result since automatic cancellation moves the tool on the Z-axis in minus direction through the distance equivalent to the tool length.
Use an H00 command, rather than a G49 command, if G43/G44 mode is to be cancelled temporarily.

6) The alarm **839 ILLEGAL OFFSET No.** will occur if an offset number exceeding the machine specifications is set.

7) When tool offset data and MAZATROL tool data are both validated, offsetting is executed by the sum of the two data items concerned.

8) In order to apply length offset for a milling tool in its axial direction, give the corresponding G68 command (Coordinate Rotation) following a B-axis angular motion command.

B. Tool length offsetting for turning tools

1. Z- and X-axis motion distance

$G43X\pm xZ\pm zHh_1P1$	$\pm z + \{\pm \ell h_{1z} - (\pm \ell h_{0z})\}$	Positive-direction length offset
	$\pm x + \{\pm \ell h_{1x} - (\pm \ell h_{0x})\}$	Positive-direction length offset
$G44X\pm xZ\pm zHh_1P1$	$\pm z + \{\pm \ell h_{1z} - (\pm \ell h_{0z})\}$	Negative-direction length offset
	$\pm x + \{\pm \ell h_{1x} - (\pm \ell h_{0x})\}$	Negative-direction length offset
$G49X\pm xZ\pm z$	$\pm z - (\pm \ell h_{1z})$	Cancellation of the offset amount
	$\pm x - (\pm \ell h_{1x})$	Cancellation of the offset amount

ℓh_{1x}: **BA62** + X-axis value of offset No. h_1

ℓh_{1z}: Z-axis value of offset No. h_1

ℓh_{0x}: X-axis offset amount existing before the G43 or G44 block

ℓh_{0z}: Z-axis offset amount existing before the G43 or G44 block

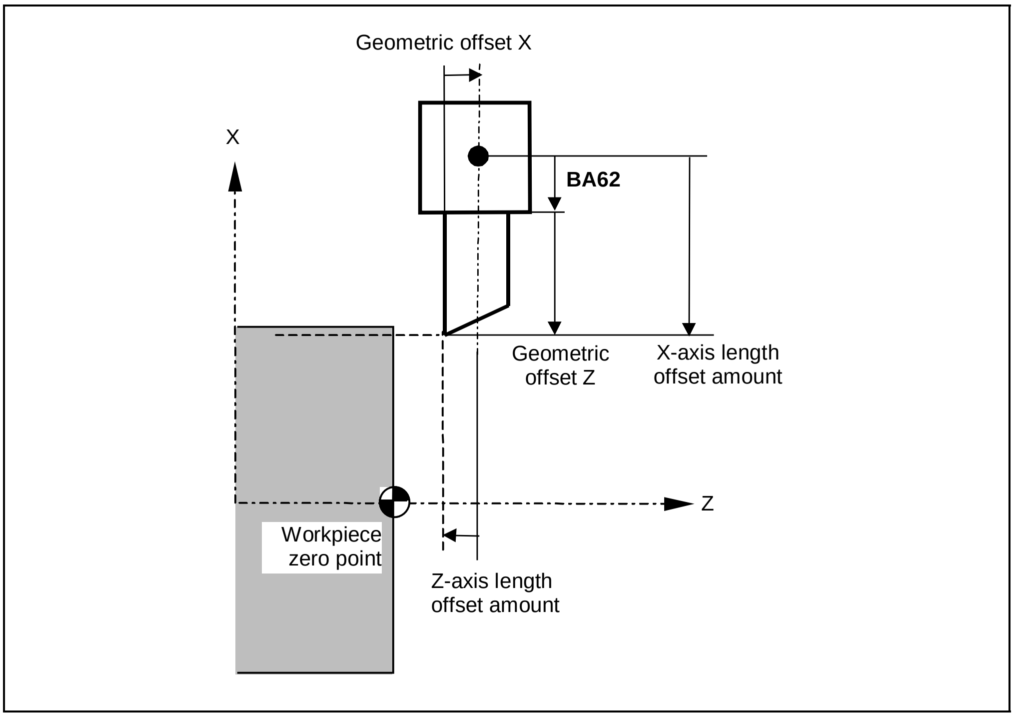
P1: Selection of the length offsetting type for turning tools

Irrespective of whether absolute or incremental programming method is used, the actual ending point coordinates are calculated by offsetting the programmed end point coordinates through the offset amount. Offsetting for a turning tool is executed on all axes for which offset amounts are registered.
The initial state (upon turning-on or after M02) is of G49 (tool length offset cancellation).

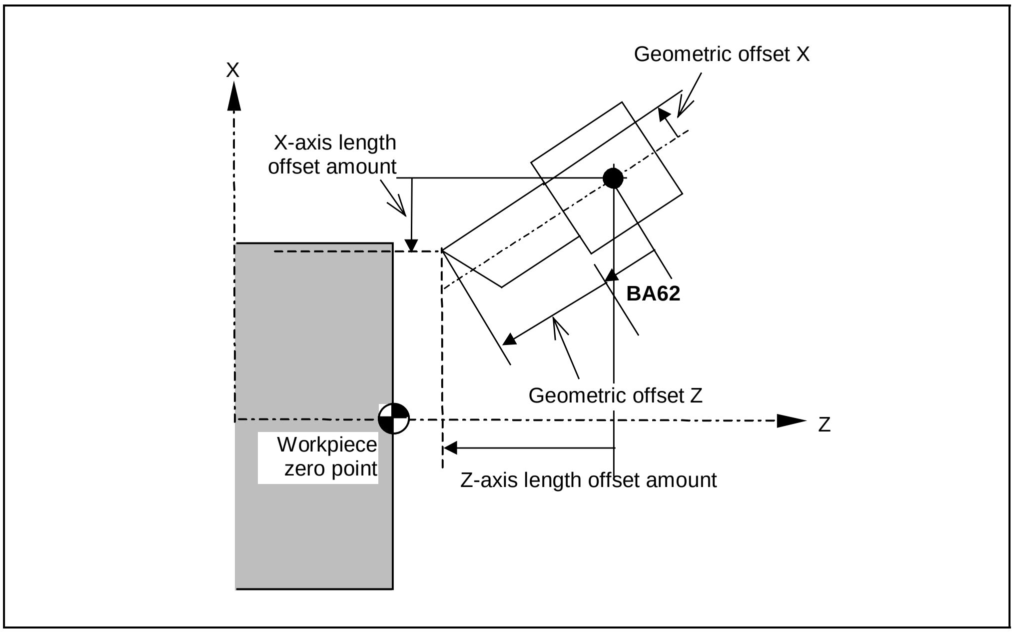
As for an angular application of the tool, the X- and Z-axis component vectors for length offsetting are automatically computed for the particular application angle, as shown below:

X-axis offset amount = (“Geometric” Z + **BA62**) sin θ + (“Geometric” X) cos θ
Z-axis offset amount = (“Geometric” Z + **BA62**) cos θ – (“Geometric” X) sin θ

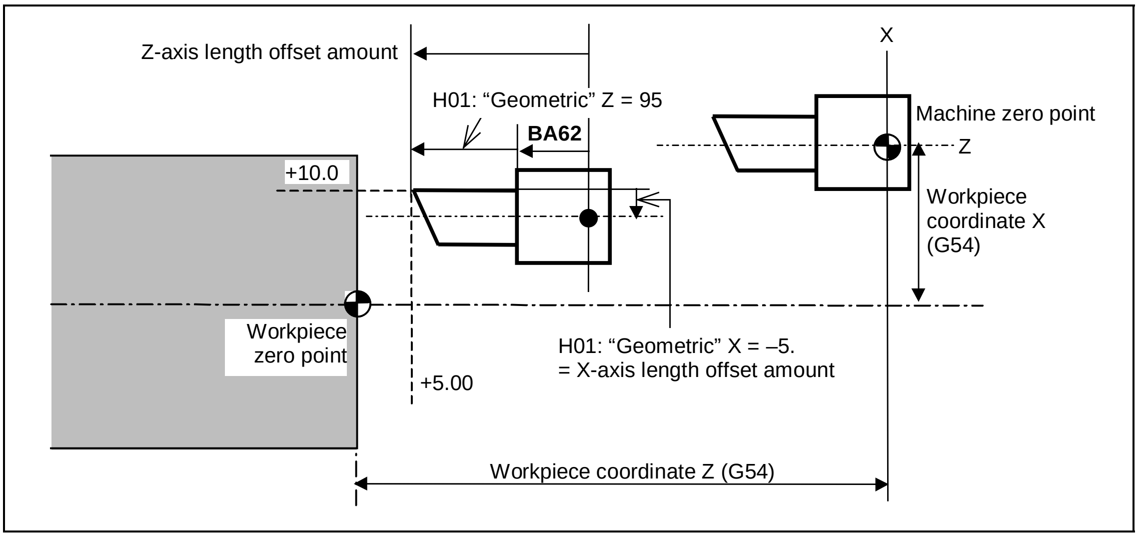
Example 1: B-axis position = 90°



Example 2: B-axis = 45°



2. Sample programs

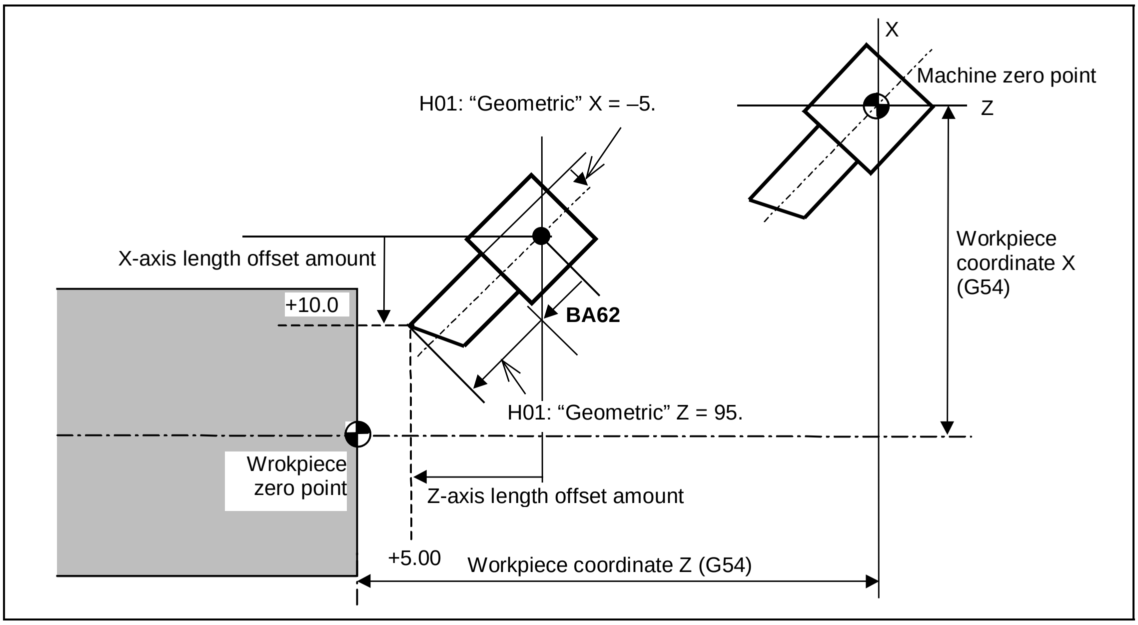


For absolute data input
(H01: Z = 95. X = –5.)

```
N001  G90  G94  G00  G40  G80
N002  G91  G28  Z0
N003  T01  T00  M06
N004  G90  G54  X100. Y0
N005  G43  X10. Z5.  H01 P1
N006  G01  Z-50.  F100
```

For incremental data input
(H01: Z = 95. X = –5.)

```
N001  G90  G94  G00  G40  G80
N002  G91  G28  Z0
N003  T01  T00  M06
N004  G90  G54  X100. Y0
N005  G91  G43  X-90. Z-195. H01 P1
N006  G01  Z-55.  F100
```



For absolute data input
(H01: Z = 95. X = –5.)

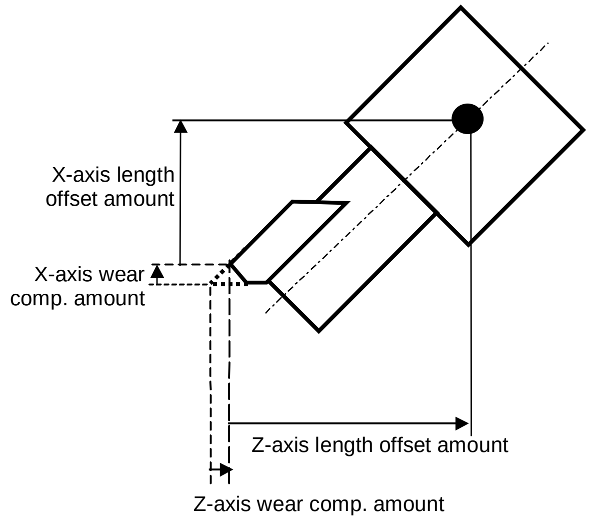
```
N001  G90  G94  G00  G40   G80
N002  G91  G28  Z0 B0
N003  T01  T00  M06
N004  G90  G0   B45.
N005  G54
N006  G43  X10. Z5.  H01 P1
N007  G01  Z-50. F100
```

For incremental data input
(H01: Z = 95. X = –5.)

```
N001  G90  G94  G00  G40  G80
N002  G91  G28  X0   Z0  B0
N003  T01  T00  M06
N004  G90  G0   B45.
N005  G54
N006  G91  G43  X-90. Z-195. H01 P1
N007  G01  Z-55. F100
```

3. MAZATROL "Wear Compensation" data for turning tools

Of MAZATROL tool data items, "Length A" and "Length B" correspond to the length and width of the tool, respectively, and "Wear Comp" values are used for tool compensation on the relevant controlled axes.



Set the following parameter to "1" to use the MAZATROL wear compensation data.

F111 bit 5	MAZATROL wear comp. valid/invalid
0	Invalid (Not used for EIA/ISO programs)
1	Valid (Used also for EIA/ISO programs)

4. Supplement

1) For turning tools, length offsetting is executed on all axes for which offset amounts are registered (and G49 cancels all offset amounts concerned).
Set "P1" in the block of G43 or G44 to select the length offsetting type for turning tools.

Example:

G43 Xx_1 Zz_1 Hh_1 P1 — Positive-direction offset on X and Z (and Y)
⋮
G49 Xx_2 — Cancellation of offsetting on X and Z (and Y)

2) Offsetting is always performed on all the axes concerned even if no axis addresses are programmed in the G43 or G44 block.

Example:

G43 Hh_1 P1
⋮
G49

} Offsetting on X and Z (and Y), and cancellation

3) If reference point (zero point) return is performed in the offsetting mode, the mode is cancelled after completion of the returning operation (if **F94** bit 2 = 0).

Example:

G43 Hh_1
⋮
G28 Zz_2

} Upon completion of return to the reference point (zero point), the offset stroke is cleared.

G43 Hh_1
G49 G28 Zz_2

} Reference point return after a Z-axis motion at the current position for clearing the offset amount

4) If command G49 or H00 is executed, offsetting will be immediately cancelled (the corresponding axis will move to clear the offset amount to zero).
When using MAZATROL tool data, do not use G49 as a cancellation command code; otherwise interference with the workpiece may result since automatic cancellation moves the tool on the Z-axis in minus direction through the distance equivalent to the tool length.
Use an H00 command, rather than a G49 command, if G43/G44 mode is to be cancelled temporarily.

5) The alarm **839 ILLEGAL OFFSET No.** will occur if an offset number exceeding the machine specifications is set.

6) As for offsetting by T-codes, offset amount is not actually made valid until a movement command is executed.

Example:

G28 Xx_3	
G28 Zz_3	
T01 M6	Offset amount of T01 validated, but no axis movement.
G00 Xx_3	Motion on the X-axis only with offsetting.
Zz_3	Motion on the Z-axis with offsetting.

7) Length offset is automatically executed in the axial direction of a turning tool for any angle of the B-axis. There is no need to give a command of G68 (Coordinate Rotation), which is required in the case of milling tools.

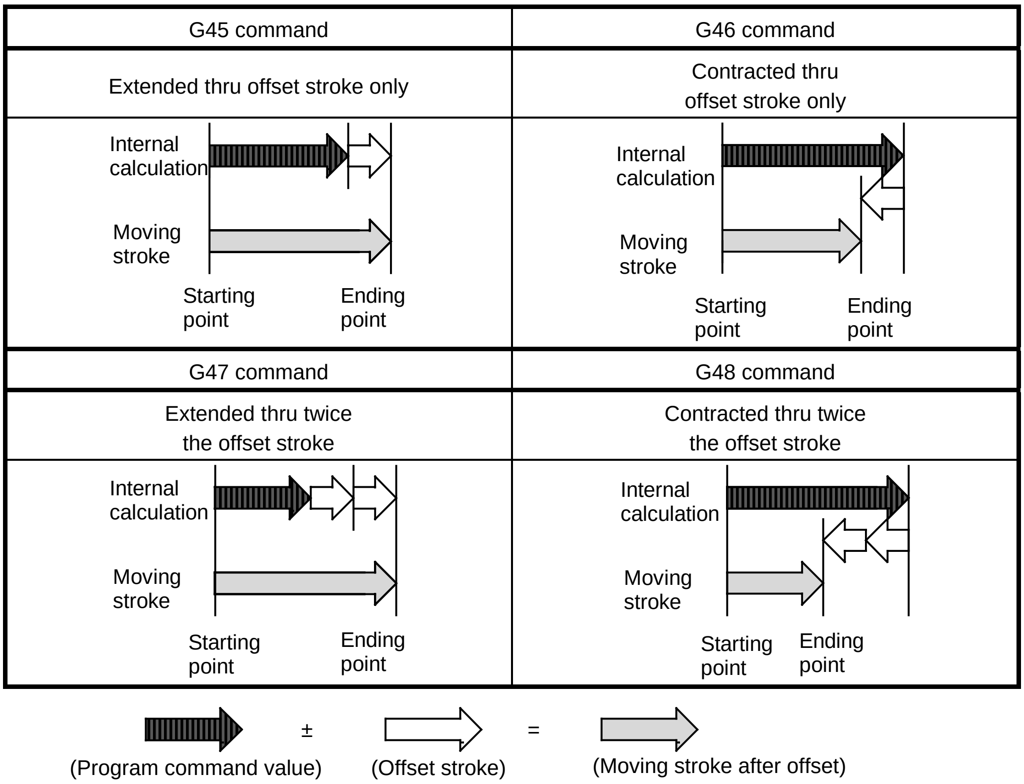
13-3 Tool Position Offset: G45 to G48

1. Function and purpose

Command G45 or G46 allows the axis movement distance set previously in that block to be increased or decreased, respectively, according to the offset data. Likewise, command G47 or G48 extends or contracts the previously set distance by twice the offset stroke, respectively.
The maximum available number of sets of offset numbers is as follows:

Standard: 128 sets: D1 to D128
Optional: 512 sets: D1 to D512

where the maximum available number of sets of offset numbers refers to the total number of sets of offset numbers including those concerning the tool length, the tool position, and the tool diameter.



2. Programming format

Command format	Function
G45 Xx Dd	To extend a moving stroke by the offset stroke which has been set in the offset memory.
G46 Xx Dd	To contract a moving stroke by the offset stroke which has been set in the offset memory.
G47 Xx Dd	To extend a moving stroke by twice the offset stroke which has been set in the offset memory.
G48 Xx Dd	To contract a moving stroke by twice the offset stroke which has been set in the offset memory.

Return to Library

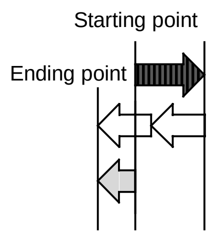
3. Detailed description

- Programming based on incremental data is shown below.

Tape command	Stroke of movement by equivalent tape command (selected offset stroke = ℓ)	Example (with x = 1000)
G45 Xx Dd	X {x + ℓ}	ℓ = 10 X = 1010 ℓ = –10 X = 990
G45 X–x Dd	X – {x + ℓ}	ℓ = 10 X = –1010 ℓ = –10 X = –990
G46 Xx Dd	X {x – ℓ}	ℓ = 10 X = 990 ℓ = –10 X = 1010
G46 X–x Dd	X – {x – ℓ}	ℓ = 10 X = –990 ℓ = –10 X = –1010
G47 Xx Dd	X {x + 2·ℓ}	ℓ = 10 X = 1020 ℓ = –10 X = 980
G47 X–x Dd	X – {x + 2·ℓ}	ℓ = 10 X = –1020 ℓ = –10 X = –980
G48 Xx Dd	X {x – 2·ℓ}	ℓ = 10 X = 980 ℓ = –10 X = 1020
G48 X–x Dd	X – {x – 2·ℓ}	ℓ = 10 X = –980 ℓ = –10 X = –1020

- Even if no offset numbers are set in the same block as that which contains commands G45 to G48, offsetting will be performed, based on previously stored tool position offset numbers.
- An alarm **839 ILLEGAL OFFSET No.** will occur if the designated offset number is an unavailable one.
- These G-code commands are not modal ones, and thus they are valid only for the designated block.
- These commands must be used in modes other than the fixed-cycle mode. They will be ignored if used in the fixed-cycle mode.
- The axis will move in reverse if internal calculation for changing the movement distance results in inversion of the direction of movement.

Program command: `G48 X20.000`
Offset stroke: + 15.000
Real move: X–10.000



MEP060

- The following lists how the machine operates if a movement distance of 0 using the incremental data command mode (G91) is programmed:

NC command	G45 X0 D01	G45 X–0 D01	G46 X0 D01	G46 X–0 D01
Equivalent command	X1234	X–1234	X–1234	X1234

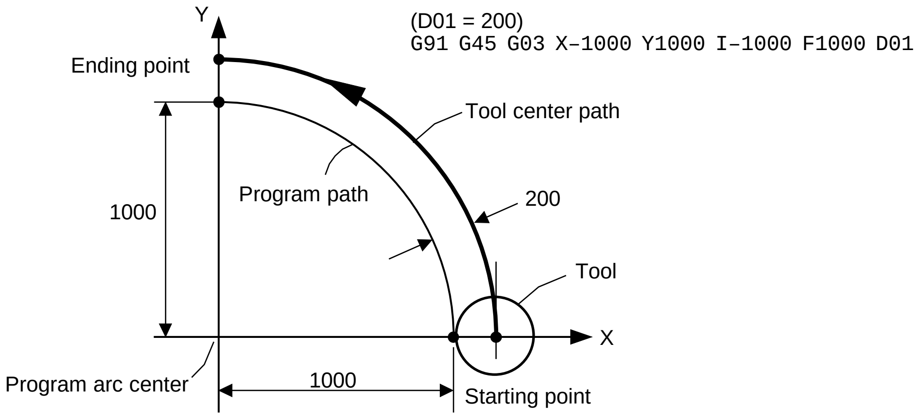
D01: Offset number
1234: Offset amount for D01

For absolute data commands, if the movement distance is set equal to 0, the block will be immediately completed and no movement through the offset distance will occur.

- When absolute data commands are used, each axis will also move from the ending point preset in the preceding block to the position set in the block that contains commands G45 through G48.
  That is, when absolute data commands are used, offsetting will be performed according to the movement distance (increments in distance) set in that block.

4. Sample programs

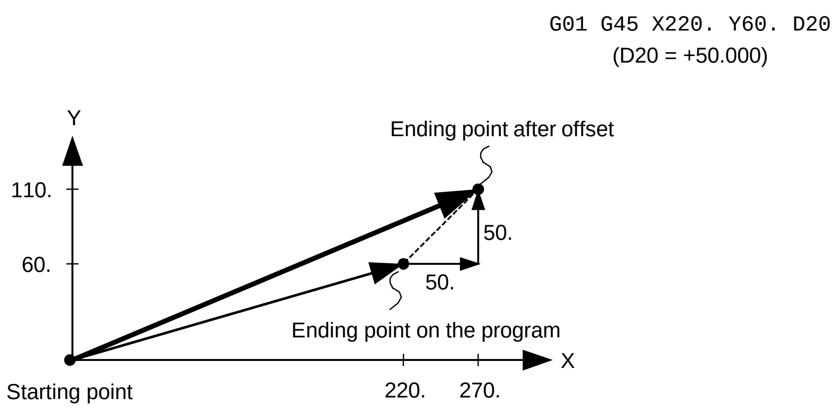
1. During arc interpolation, tool diameter offsetting using commands G45 to G48 can be done only for a 1/4, 1/2, or 3/4 circle whose starting and ending points are present on a coordinate axis passing through the arc center.



Tool position offset, with 1/4 arc command given

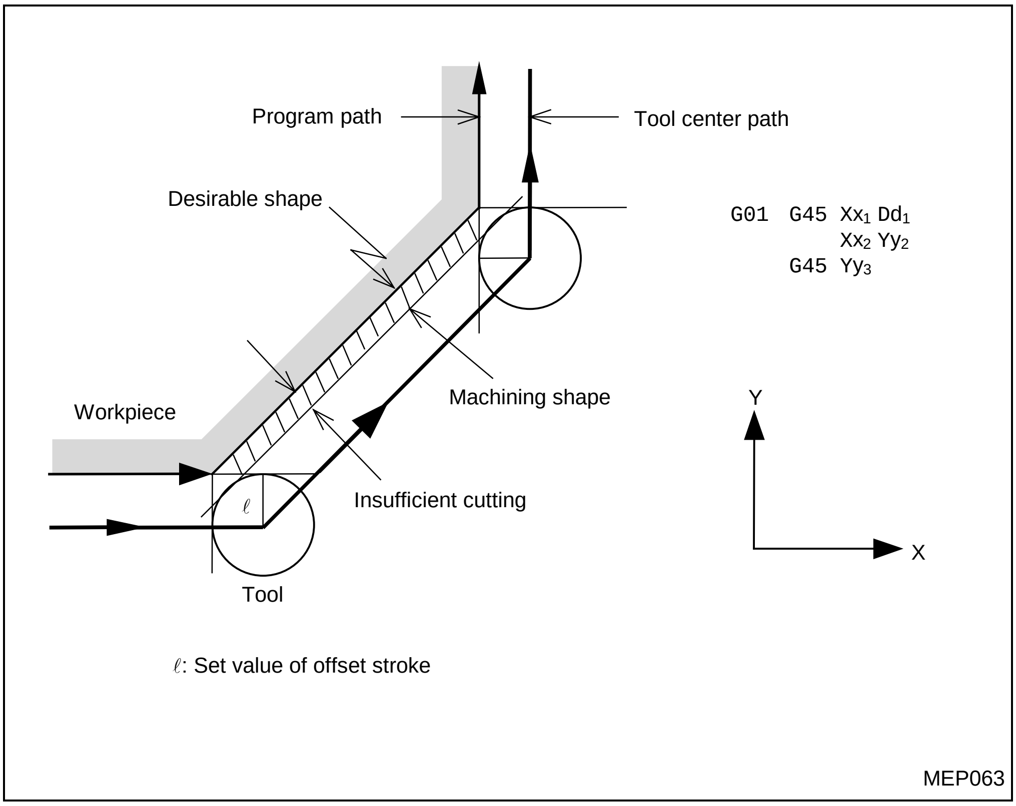
MEP061

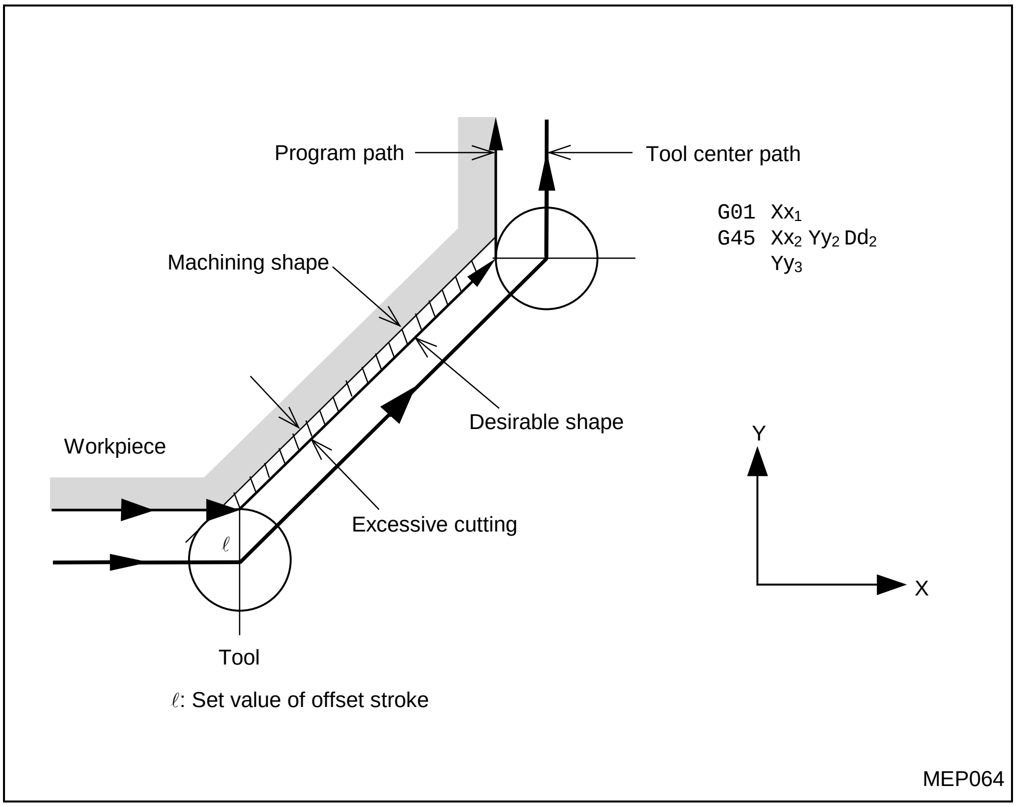
2. If an "n" number of axes are designated at the same time, the same amount of offsetting will be performed on all designated axes. This also applies to additional axes, but within the limits of the simultaneously controllable axis quantity.



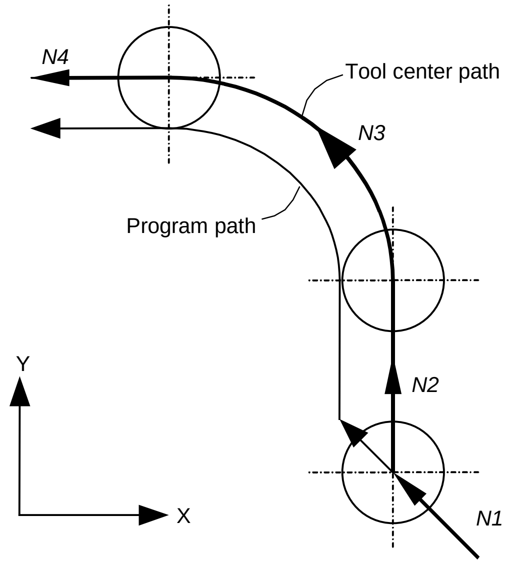
MEP062

Note: Use tool diameter offset commands G40, G41, or G42 if simultaneous offsetting of two axes is likely to result in excessive or insufficient cutting as shown below.





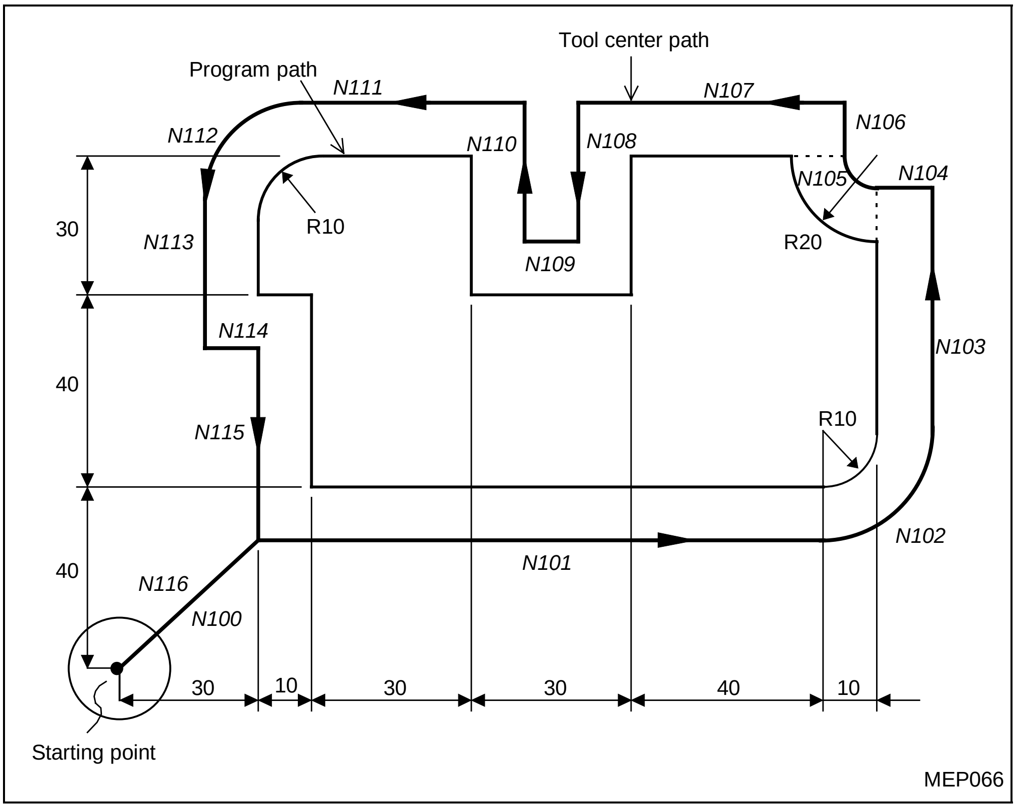
3. Cornering in a 1/4 circle




```
N1  G46   G00    Xx1     Yy1    Dd1
N2  G45   G01    Yy2     Ff2
N3  G45   G03    Xx3     Yy3    Ii3
N4  G01   Xx4
```

MEP065

4. When commands G45 to G48 are set, each of the corresponding amounts of offsetting will become those designated by the offset numbers; unlike the tool length offset command (G43), these commands will not move the axes through the difference from the previous offset amount.



Offset stroke: D01 = 10.000 mm (Tool diameter offset stroke)

```
N100  G91   G46    G00   X40.  Y40.   D01
N101  G45   G01    X100. F200
N102  G45   G03    X10.  Y10.  J10.
N103  G45   G01    Y40.
N104  G46   X0
N105  G46   G02    X–20. Y20.  J20.
N106  G45   G01    Y0
N107  G47   X–30.
N108        Y–30.
N109  G48   X–30.
N110        Y30.
N111  G45   X–30.
N112  G45   G03    X–10. Y–10. J–10.
N113  G45   G01    Y–20.
N114        X10.
N115        Y–40.
N116  G46   X–40.  Y–40.
N117  M02
%
```

13-4 Tool Diameter Offset Function: G40, G41, G42

13-4-1 Overview

1. Function and purpose

Offsetting in any vectorial direction can be done according to the tool radius preselected using G-codes (G38 to G42) and D-codes. This function is referred to as tool diameter offsetting.

For turning tools, nose-R compensation can be performed according to the designated direction (only when TOOL OFFSET type C is selected).

2. Programming format

Command format	Function	Remarks
G40X_Y_	To cancal a tool diameter offset	
G41X_Y_	To offset a tool diameter (Left)	
G42X_Y_	To offset a tool diameter (Right)	
G38 I_J_	To change and hold an offset vector	These commands can be given during the diameter offset mode.
G39	To interpolate a corner arc	

3. Detailed description

The maximum available number of sets of offset numbers is as follows:

Standard: 128 sets: D1 to D128
Optional: 512 sets: D1 to D512
where the maximum available number of sets of offset numbers refers to the total numbers including those concerning the tool length, the tool position, and the tool diameter.

For tool diameter offsetting, all H-code commands are ignored and only D-code commands become valid.
Also, tool diameter offsetting is performed for the plane that is specified by either the plane selection G-code command or two-axis address code command appropriate for tool diameter offsetting. No such offsetting is performed for axes other than those corresponding or parallel to the selected plane. See 6-4 Plane Selection Commands, to select a plane using a G-code command.

13-4-2 Tool diameter offsetting

1. Tool diameter offsetting cancellation

Tool diameter offsetting is automatically cancelled in the following cases:

- After power has been turned on
- After the reset key on the NC operation panel has been pressed
- After M02 or M30 has been executed (these two codes have a reset function)
- After G40 (offsetting cancellation command) has been executed

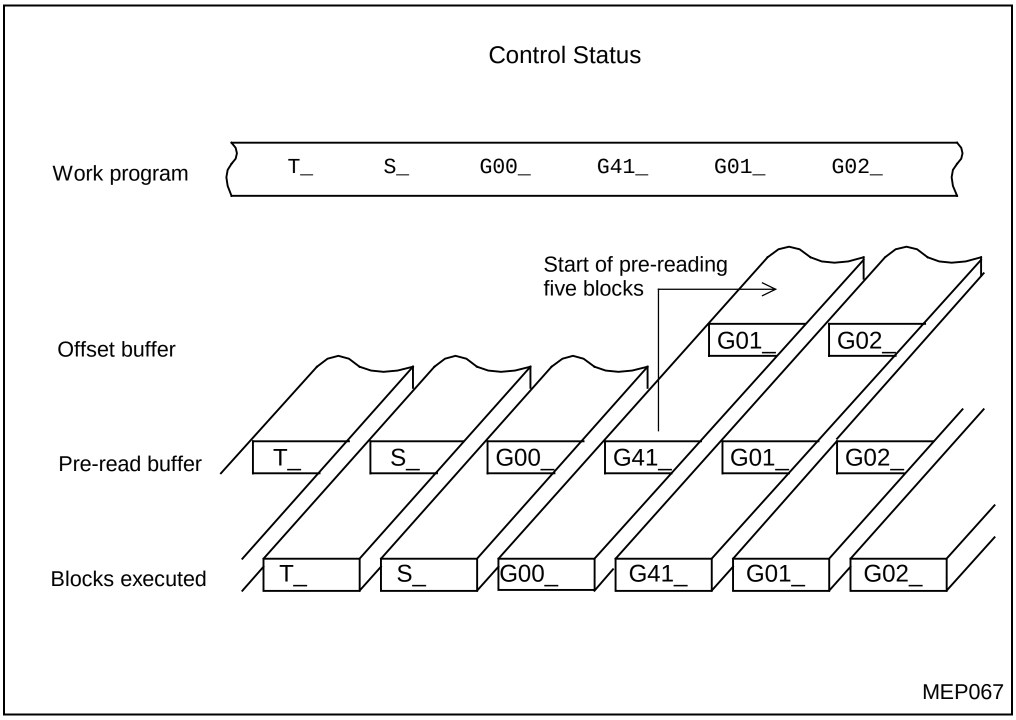
In the offsetting cancellation mode, the offset vector becomes zero and the tool center path agrees with the programmed path.
Programs containing the tool diameter offset function must be terminated during the offsetting cancellation mode. Give the G40 command in a single-command block (without any other G-code). Otherwise it may be ignored.

2. Startup of tool diameter offsetting

Tool diameter offsetting will begin during the offset mode when all the following three conditions are met:

- Command G41 or G42 has been executed.
- The offset number for tool diameter offsetting is larger than zero, but equal to or smaller than the maximum available offset number.
- The command used with the offsetting command is a move command other than those used for arc interpolation.

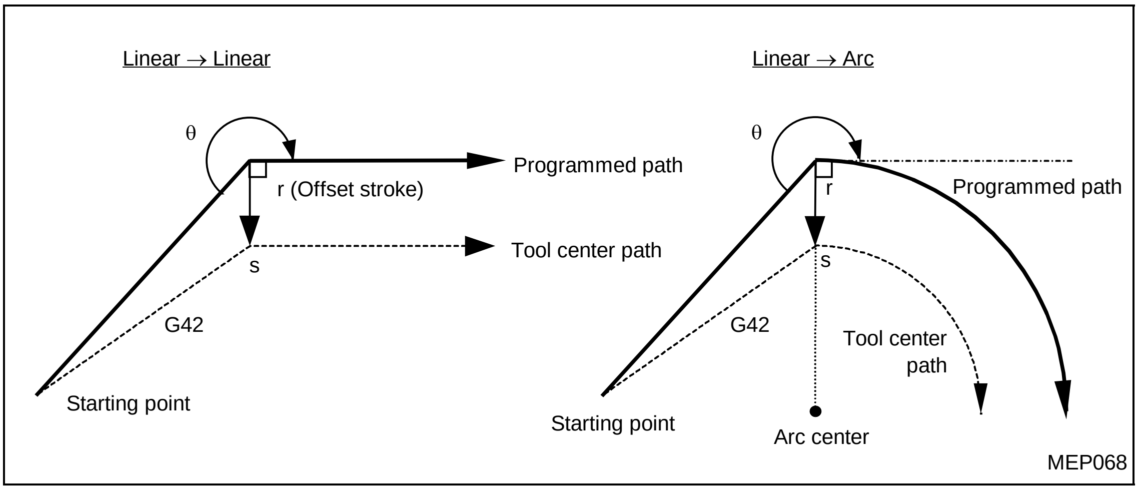
Offsetting will be performed only when reading of five blocks in succession is completed, irrespective of whether the single-block operation mode is used.
During offsetting, five blocks are pre-read and then calculation for offsetting is performed.



There are two types of offsetting startup operation: Type A and Type B.
It depends on the setting of bit 4 of parameter **F92** whether Type A or Type B is automatically selected.
These two types of startup operation are similar to those of offsetting cancellation.
In the descriptive diagrams below, “s” signifies the ending point of single-block operation.

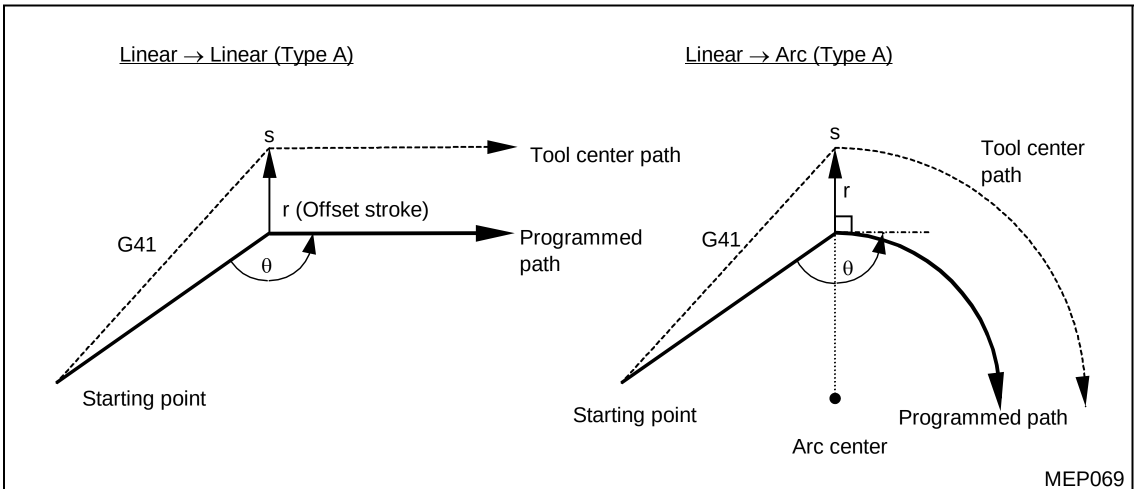
3. Tool diameter offsetting startup operation

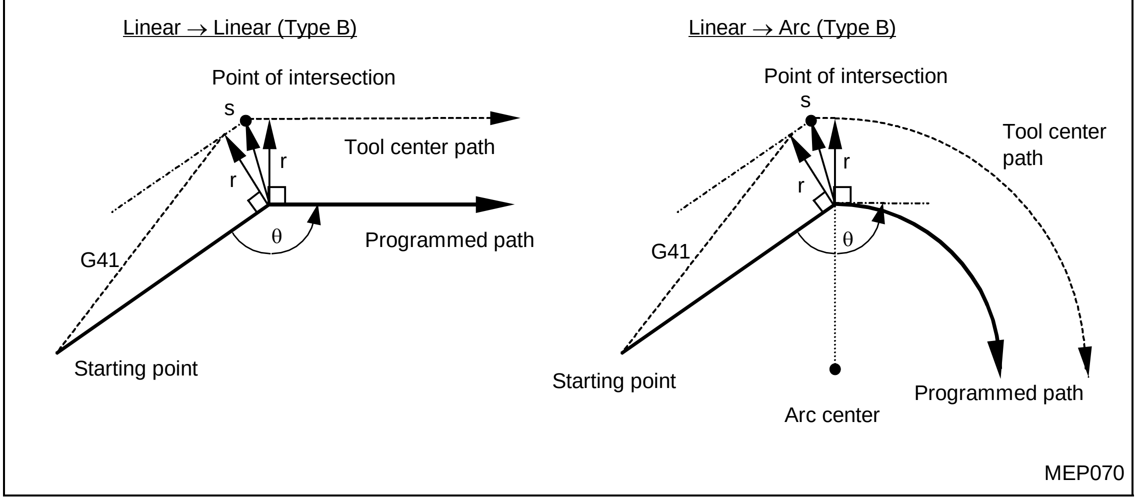
A. For the corner interior



B. For the corner exterior (obtuse angle) [90° ≤ θ < 180°]

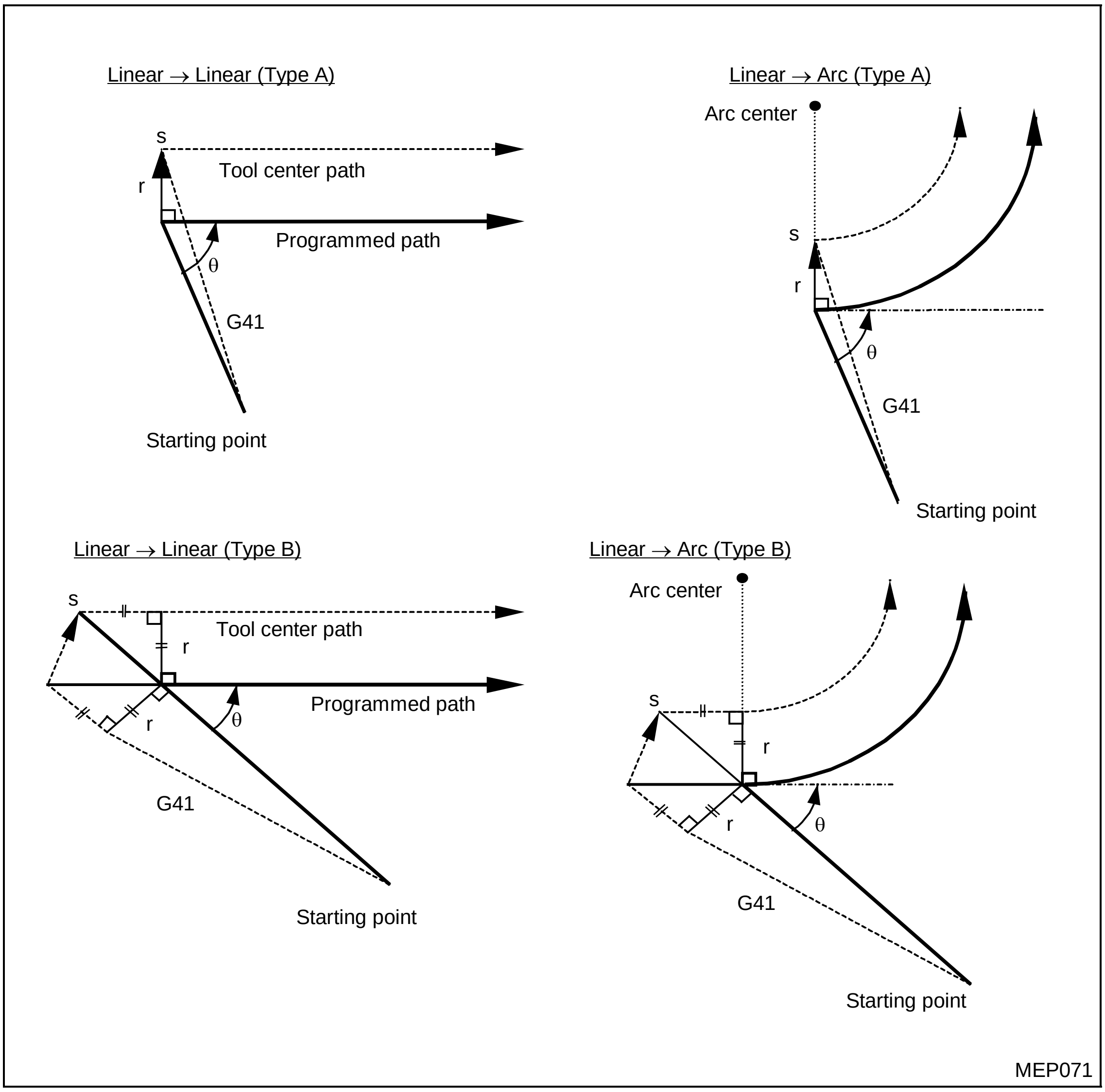
(Type A/B selection is possible with a predetermined parameter.)





C. For the corner exterior (sharp angle) [θ< 90°]

(Type A/B selection is possible with a predetermined parameter.)

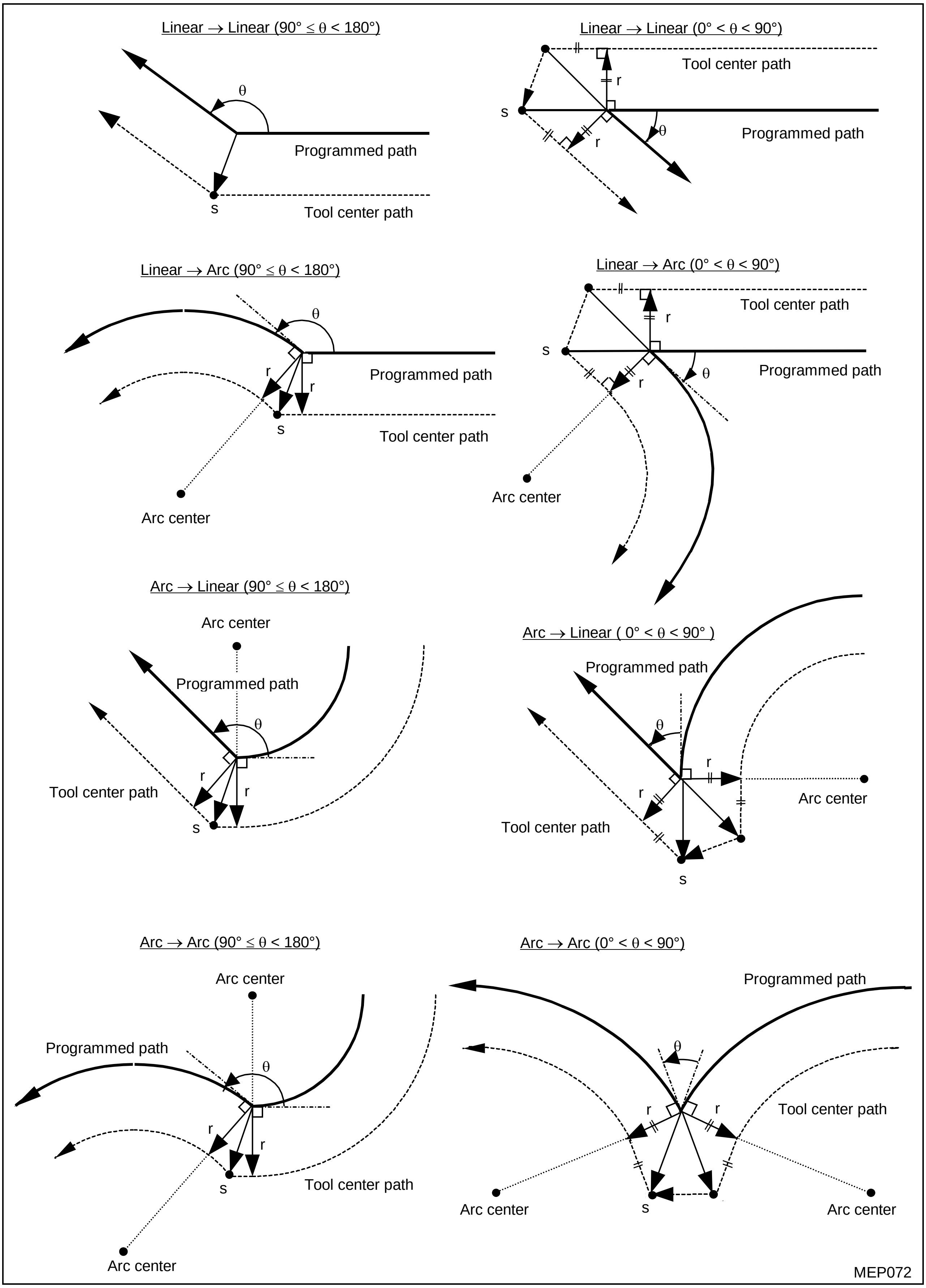


4. Operation during the offset mode

Offsetting is performed for linear or arc interpolation commands and positioning commands. Identical offset commands G41 or G42 will be ignored if they are used during the offset mode.

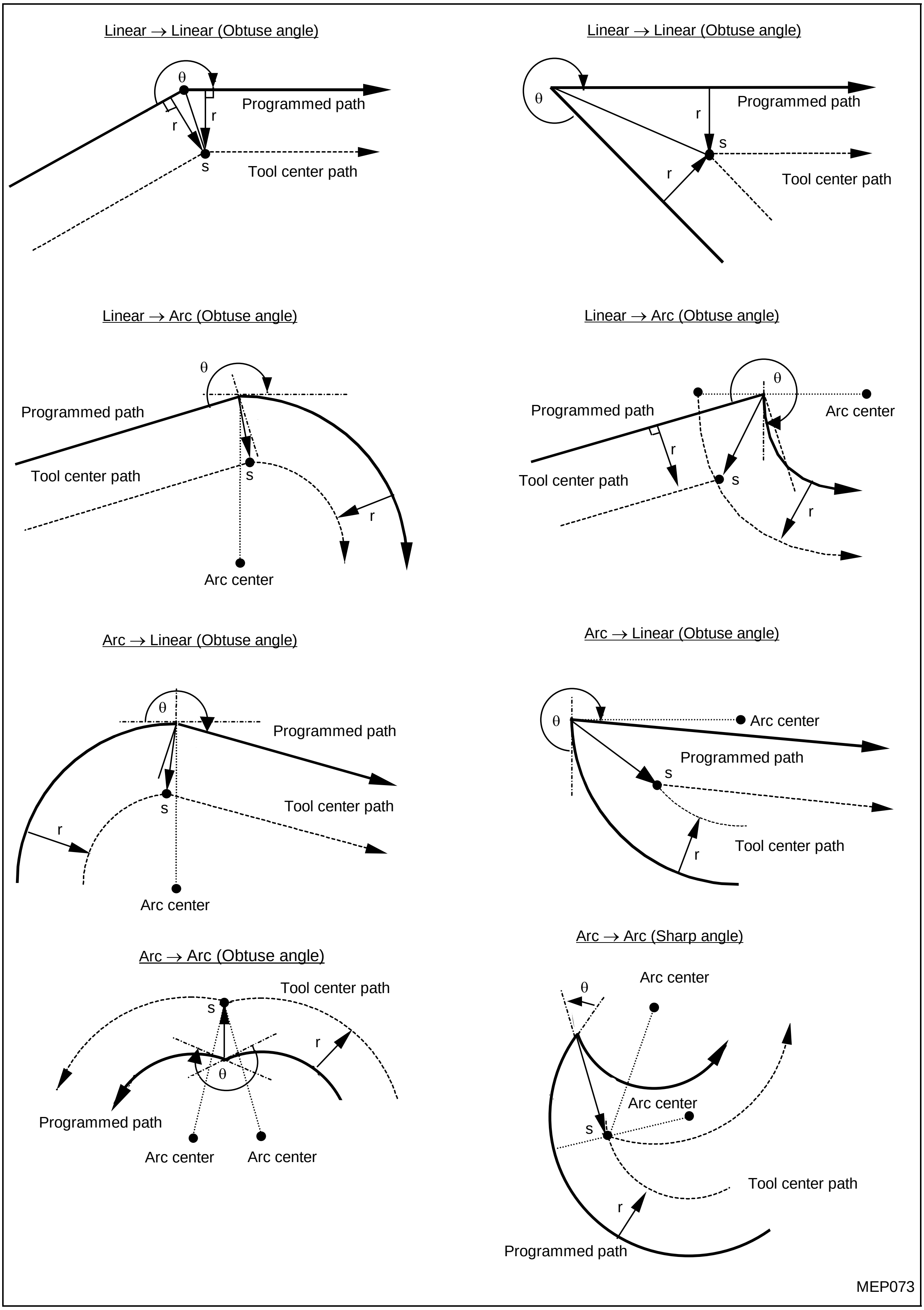
Successive setting of four or more blocks that do not involve movement of axes during the offset mode may result in excessive or insufficient cutting.

A. For the corner exterior



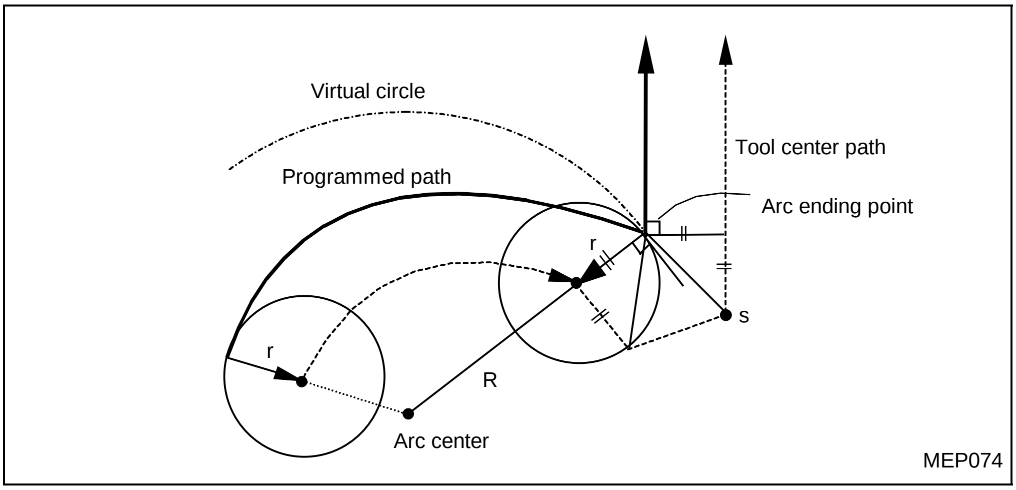
MEP072

B. For the corner interior



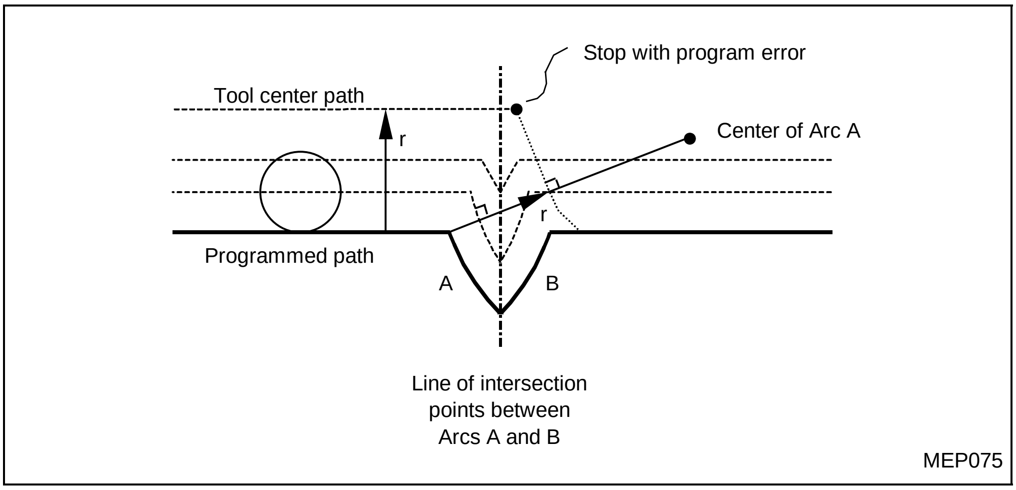
C. For an arc that does not have the ending point on it

The area from the starting point of the arc to the ending point is interpolated as a spiral arc.



D. For arcs that do not have their inner crossing point

In cases such as those shown in the diagram below, there may or may not be a crossing point of arcs A and B, depending on the particular offset data. In the latter case, the program terminates at the ending point of the preceding block after an alarm **836 NO INTERSECTION** has been displayed.



5. Tool diameter offsetting cancellation

During the tool diameter offset mode, tool diameter offsetting will be cancelled in any of the two cases listed below.

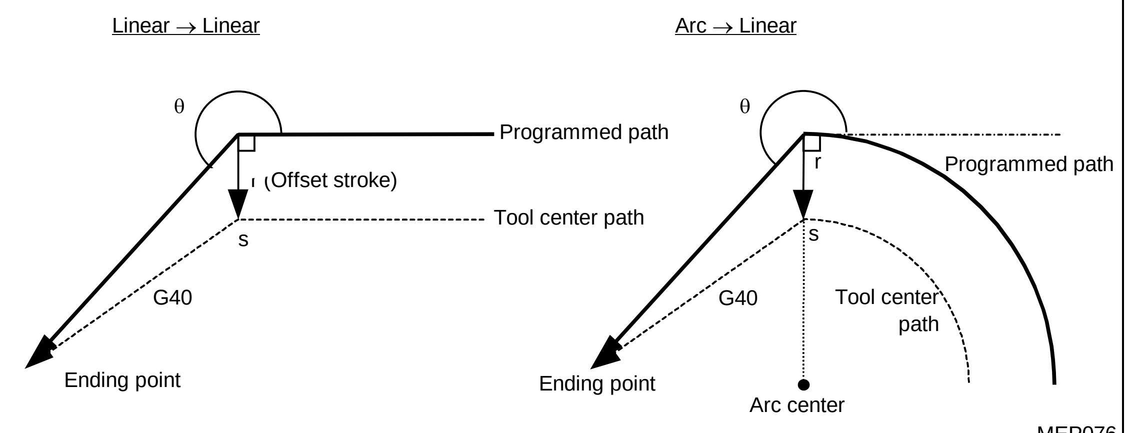
- Command G40 has been executed.
- Offset number code D00 has been executed.

At this time, however, the move command executed must be one other than those used for arc interpolation. An alarm **835 G41, G42 FORMAT ERROR** will occur if an attempt is made to cancel offsetting using an arc command.

After the offsetting cancellation command has been read into the offset buffer, the cancellation mode is set automatically and subsequent blocks of data are read into the pre-read buffer, not the offset buffer.

6. Tool diameter offsetting cancellation operation

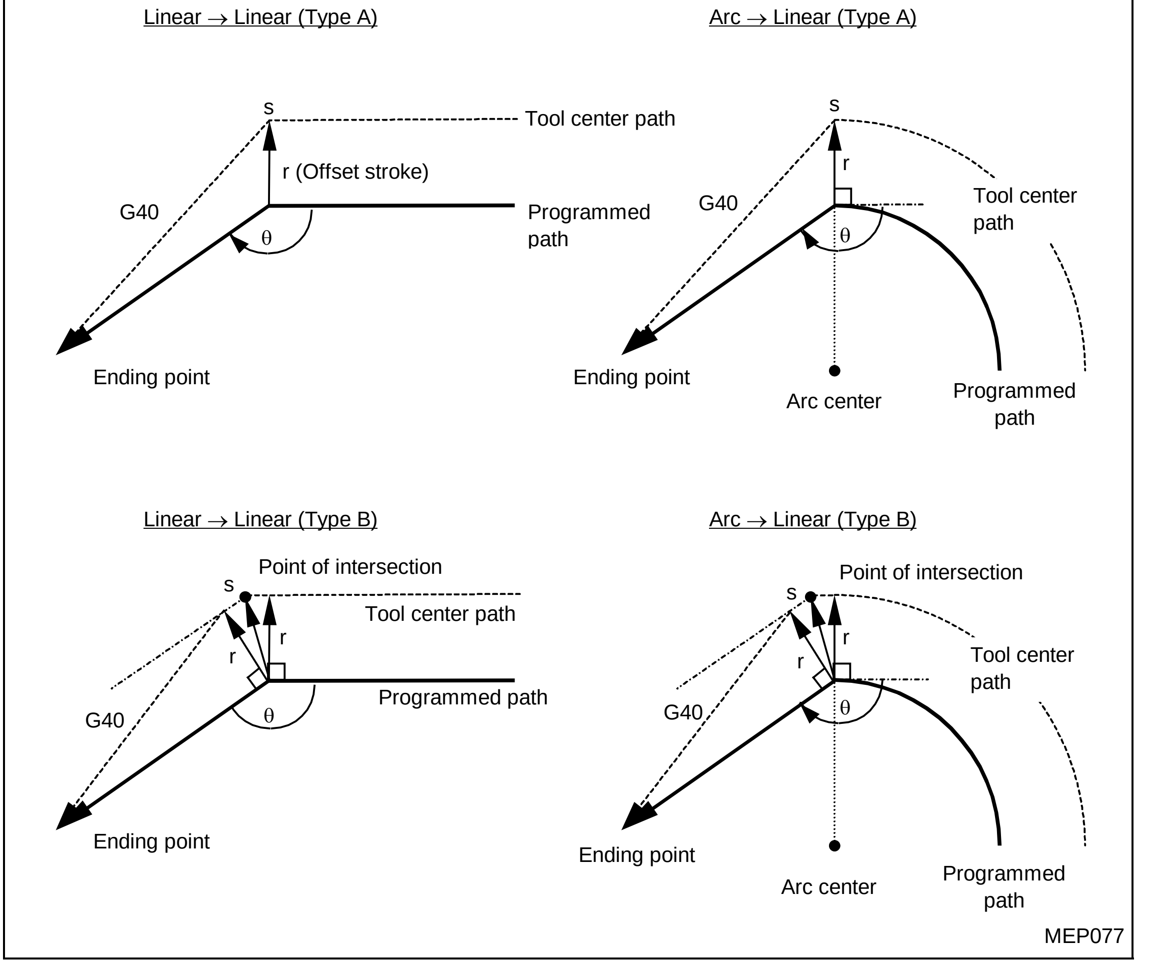
A. For the corner interior



MEP076

B. For the corner exterior (obtuse angle)

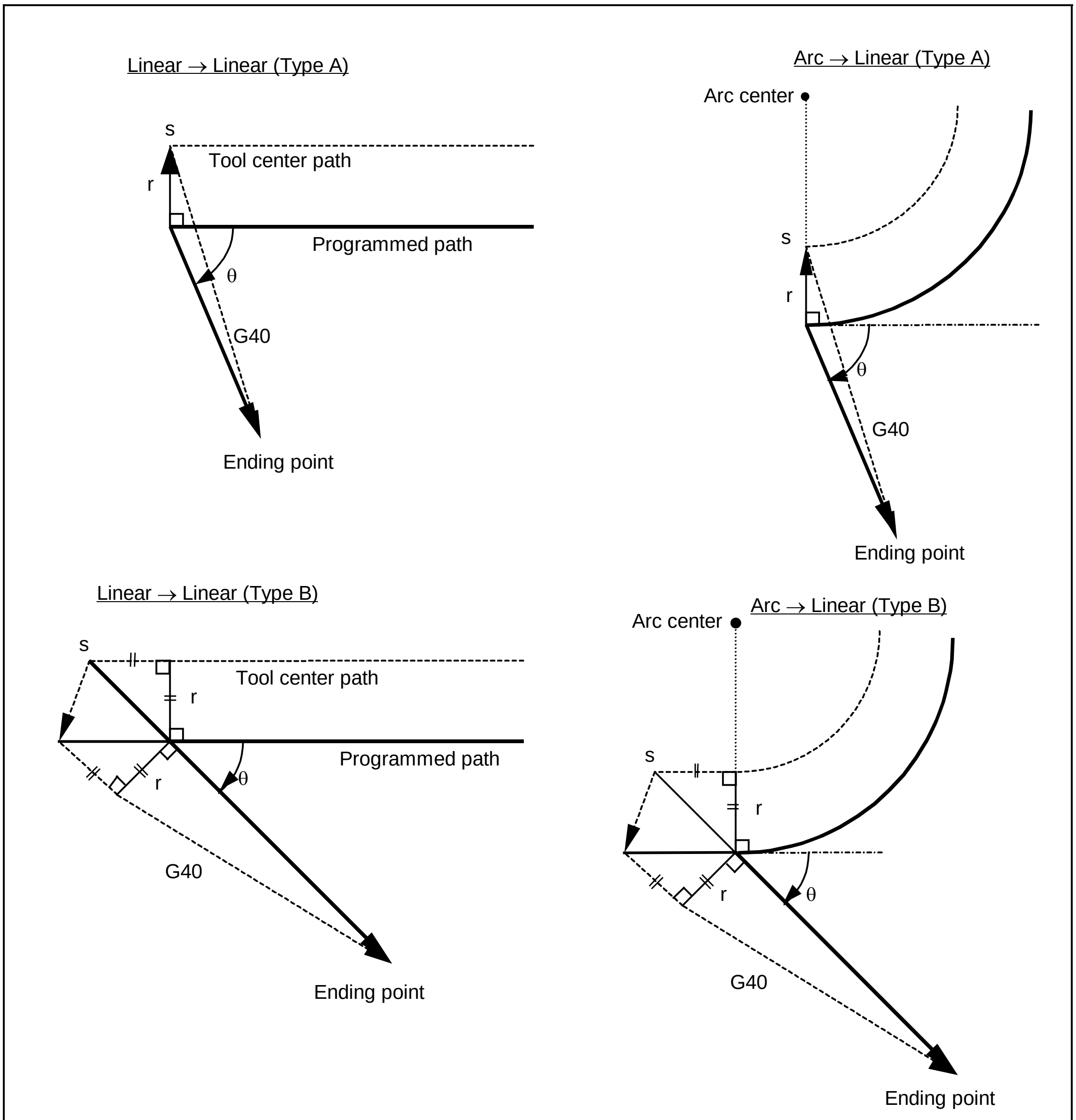
(Type A/B selection is possible with a predetermined parameter)



MEP077

C. For the corner exterior (sharp angle)

(Type A/B selection is possible with a predetermined parameter)

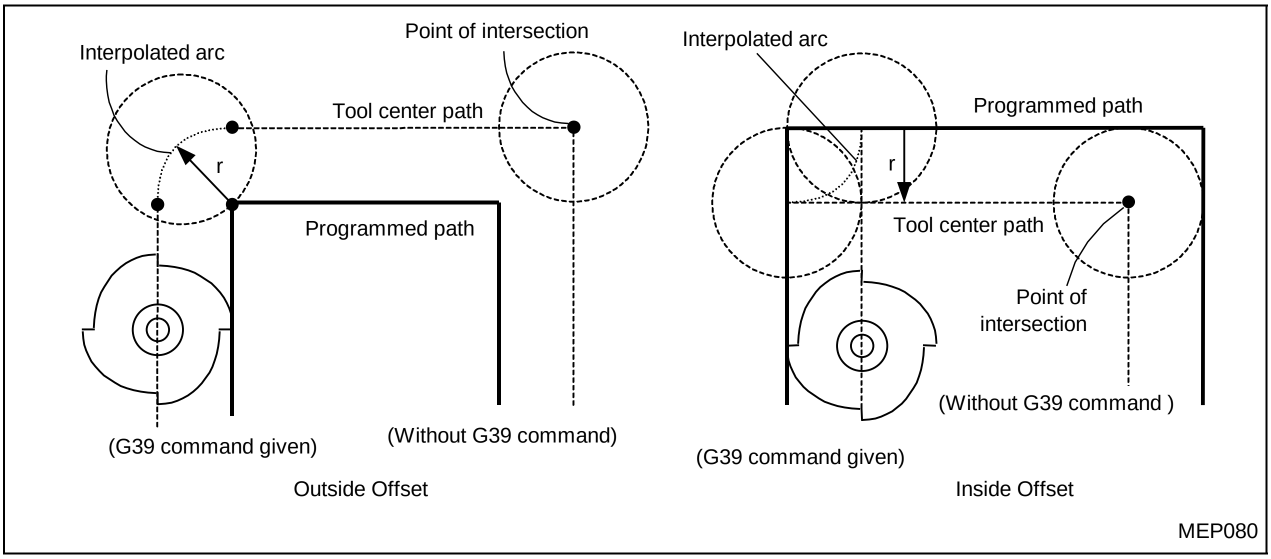


MEP079

13-4-3 Tool diameter offsetting operation using other commands

1. Interpolation of the corner arc

When command G39 (corner-arc interpolation) is used, the coordinates of the crossing points at workpiece corners will not be calculated and an arc with offset data as its radius will be interpolated.



2. Changing/retaining offset vectors

Using command G38, you can change or retain offset vectors during tool diameter offsetting.

- Retaining vectors

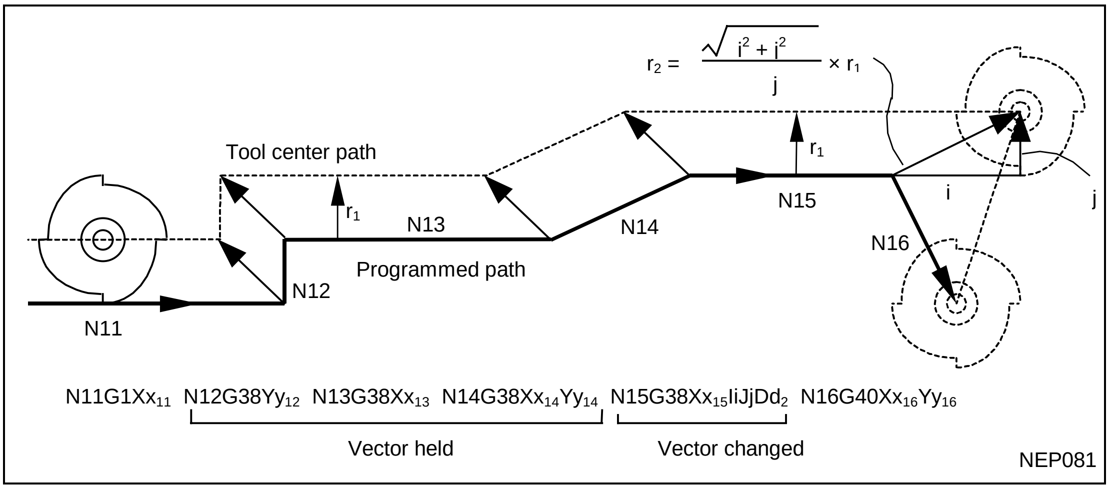
  Setting G38 in block that contains move commands allows crossing-point calculation at the ending point of that block to be cancelled and the vectors in the preceding block to be retained. This can be used for pick and feed operations.

  G38 Xx Yy

- Changing vectors

  The directions of new offset vectors can be designated using I, J, and K (I, J, and K depend on the selected type of plane), and offset data can be designated using D. (These commands can be included in the same block as that which contains move commands.)

  G38 Ii Jj Dd

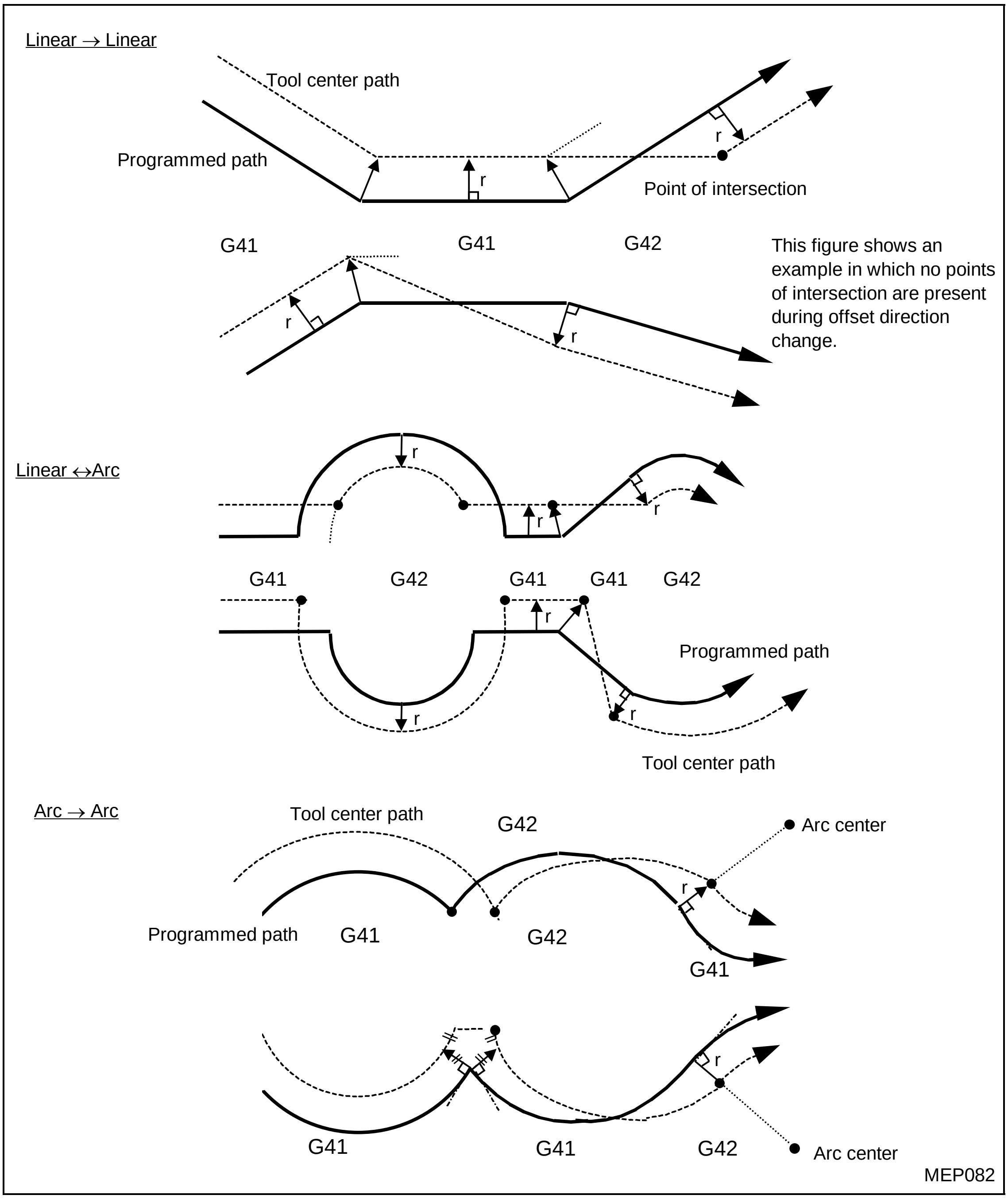


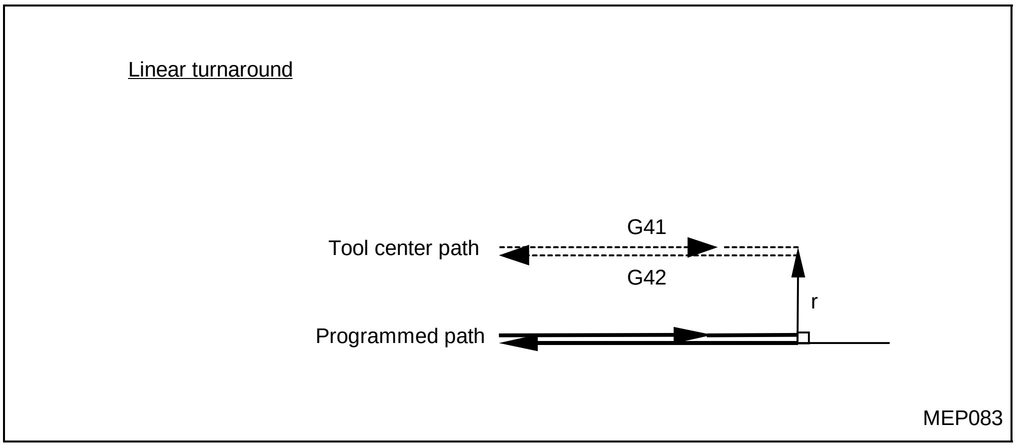
3. Changing the offset direction during tool diameter offsetting

The offset direction is determined by the type of tool diameter offset command (G41 or G42) and the sign (plus or minus) of the offset data.

G-code \ Offset stroke sign	+	–
G41	Left side offset	Right side offset
G42	Right side offset	Left side offset

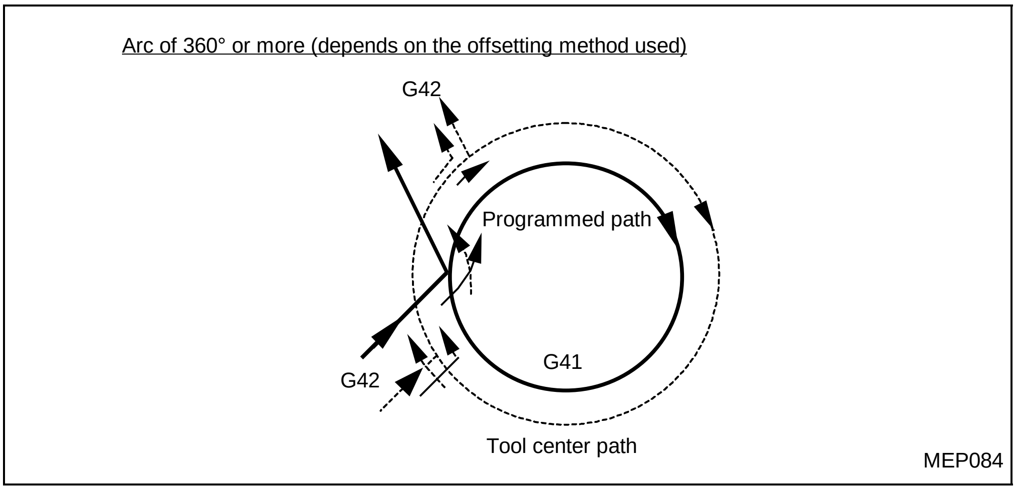
The offset direction can be changed by updating the offset command without selecting the offsetting cancellation function during the offset mode. This can, however, be done only for blocks other than the offset startup block and the next block. See subsection 12-4-7, General precautions on tool diameter offsetting, for NC operation that will occur if the sign is changed.





The arc of more than 360 degrees may result in the following cases:

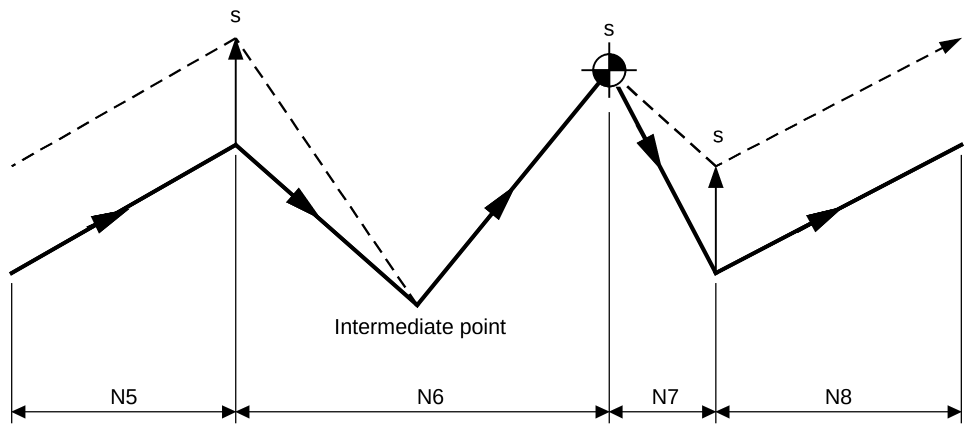
- The offset direction has been changed by G41/G42 selection.
- Commands I, J, and K have been set for G40.



4. Cases where the offset vectors are temporarily lost

If the command listed below is used during the offset mode, the current offset vectors will be lost temporarily and then the NC unit will re-enter the offset mode.
In that case, movements for offsetting cancellation will not occur and program control will be transferred from one crossing-point vector directly to the vector-less point, that is, to the programmed point. Control will also be transferred directly to the next crossing point when the offset mode is re-entered.

A. Reference-point return command




```
(G41)            :
N5  G91  G01  X–60. Y–30.
N6  G28       X–50. Y+40.
N7            X–30. Y+60.
N8            X–70. Y–40.
                 :
```

← Temporary vector 0 as offset at the intermediate point (reference point when the intermediate point is not available)

MEP085

5. Blocks that do not include movement

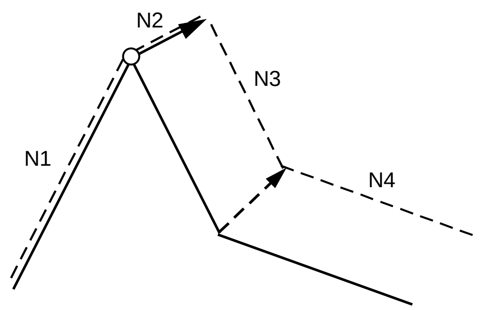
The blocks listed below are referred to as those which do not include movement:

`M03`	M command	Move-free
`S12`	S command	
`T45`	T command	
`G04 X500`	Dwell	
`G22 X200. Y150. Z100.`	To set a machining-prohibited area	
`G10 P01 R50`	To set an offset stroke	
`G92 X600. Y400. Z500.`	To set a coordinate system	
(G17) `Z40.`	To move outside the offsetting plane	
`G90`	G code only	
`G91 X0`	Moving stroke 0	...........Moving stroke is 0.

A. When a block that does not include movement is set during the start of offsetting

Vertical offsetting will be performed on the next move block.

```
N1 X30. Y60.
N2 G41  D10      ← Move-free block
N3 X20. Y–50.
N4 X50. Y–20.
```

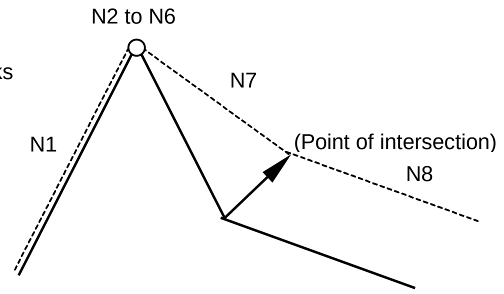


MEP086

Offset vectors, however, will not be generated if four or more blocks that do not include move commands appear in succession.

```
N1   X30. Y60.
N2   G41  D10
N3   G4   X1000
N4   F100
N5   S500
N6   M3
N7   X20. Y-50.
N8   X50. Y-20.
```

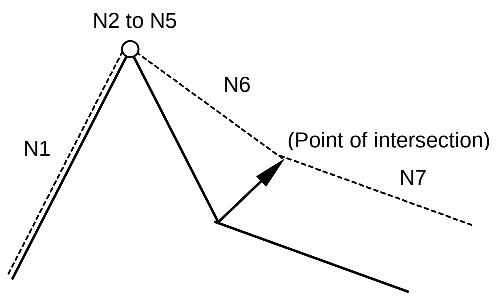
Move-free blocks (N2 to N6)



MEP087

```
N1   G41  X30. Y60. D10
N2   G4   X1000
N3   F100
N4   S500
N5   M3
N6   X20. Y-50.
N7   X50. Y-20.
```

Move-free blocks (N2 to N5)

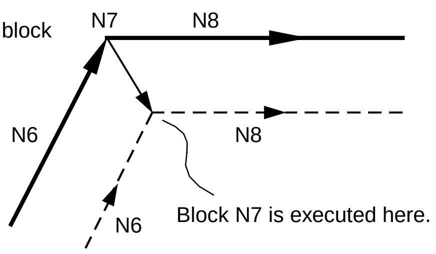


MEP088

B. When a block that does not include movement is set during the offset mode

Usual crossing-point vectors will be generated unless four or more blocks that do not include movement appear in succession.

```
N6   G91   X100.   Y200.
N7   G04   P1000          ← Move-free block
N8   X200.
```



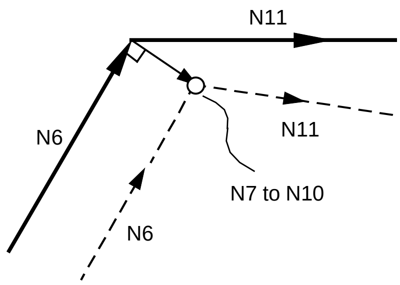
MEP089

Vertical offset vectors will be generated at the end point of preceding block if four or more blocks that do not include movement appear in succession.

Return to Library


```
N6   X100. Y200.
N7   G4    X1000
N8   F100
N9   S500
N10   M4
N11   X100.
```

Move-free blocks (N7 to N11)



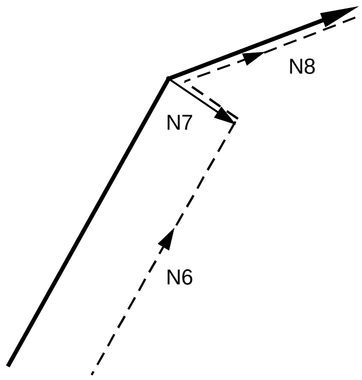
In this case, excessive cutting may occur.

MEP090

C. When a block that does not include movement is set together with offsetting cancellation

Only offset vectors will be cancelled if the block that does not include movement contains G40.

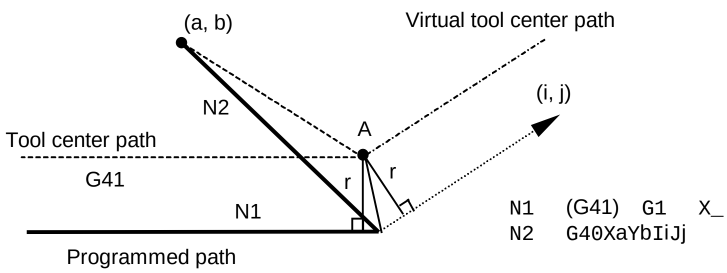
```
N6   X100. Y200.
N7   G40   G04P1000
N8   X100. Y50.
```



MEP091

6. If I, J, and K are set with G40

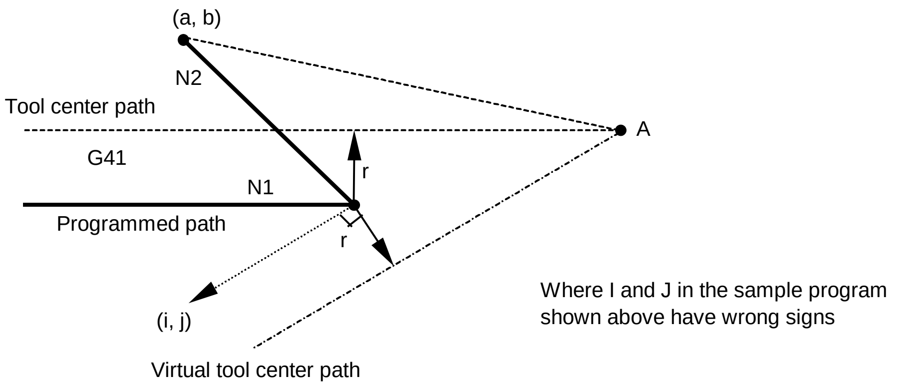
When the last of the four move command blocks which immediately precede the G40 command block contains G41 or G42, movement will be handled as if it had been programmed to occur in the vectorial direction of I, J, and K from the ending point of that last move command. That is, the area up to the crossing point with the virtual tool center path will be interpolated and then offsetting will be cancelled. The offset direction will remain unchanged.



MEP092

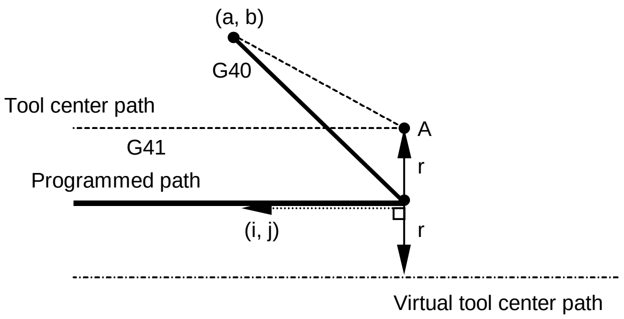
Return to Library

In this case, beware that irrespective of the offset direction, the coordinates of the crossing point will be calculated even if wrong vectors are set as shown in the diagram below.



MEP093

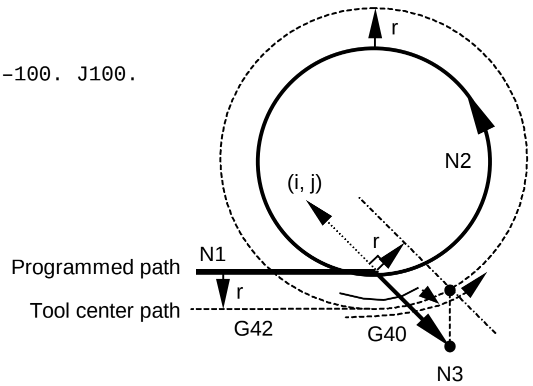
Also, beware that a vertical vector will be generated on the block before that of G40 if crossing-point calculation results in the offset vector becoming too large.



MEP094

Note: Part of the workpiece will be cut twice if the I/J/K command data in G40 preceded by an arc command generates an arc of more than 360 degrees.

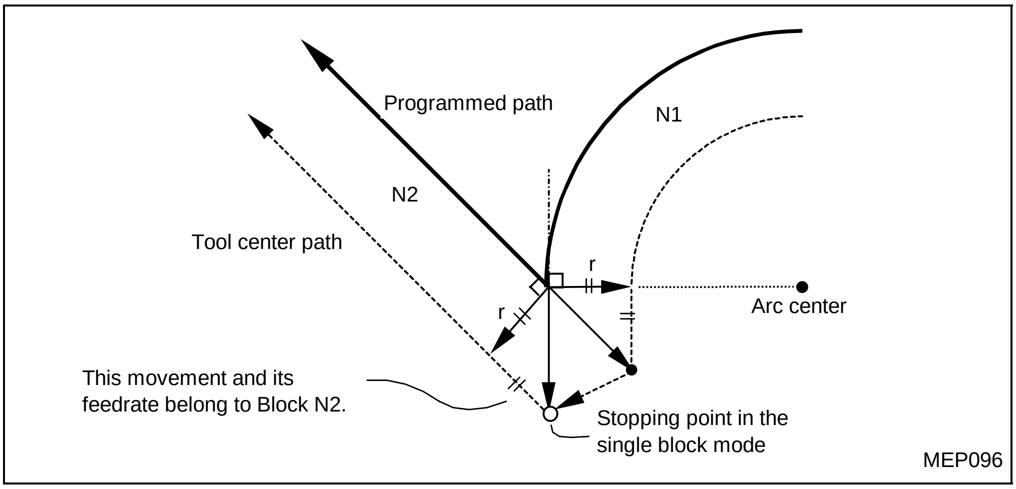
```
N1  (G42, G91)  G01X200.
N2  G02   J150.
N3  G40   G1X150. Y-150.I-100. J100.
```



MEP095

13-4-4 Corner movement

If multiple offset vectors are generated at connections between move command blocks, the tool will move linearly between those vectors. This action is referred to as corner movement.
If the multiple vectors do not agree, the tool will move around the corresponding corners (but this movement belongs to the next block). During single-block operation, the section of (Preceding block + Corner movement) is executed as one block and the remaining section of (Connections movement + Next block) is executed during next movement as another block.



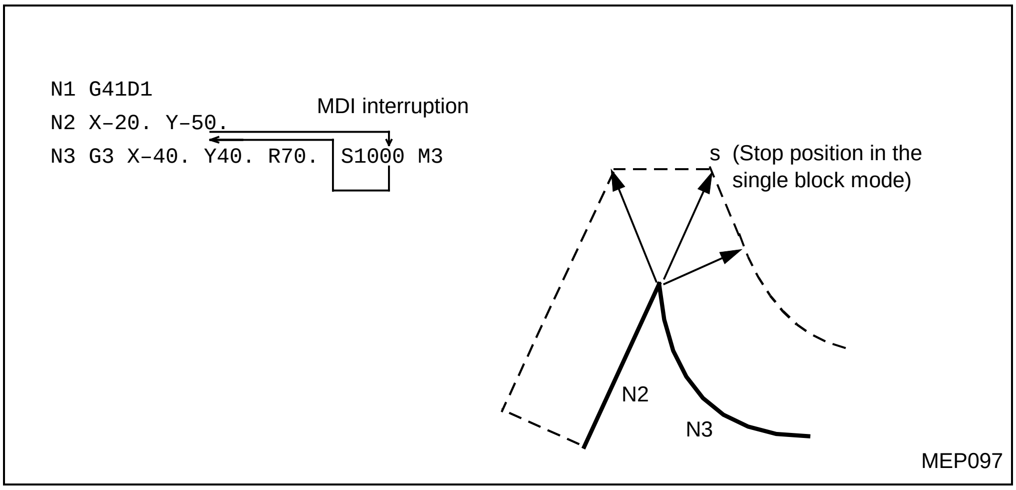
13-4-5 Interruptions during tool diameter offsetting

1. Interruption by MDI

Tool diameter offsetting is valid during automatic operation, whether it is based on the tape, memory, or MDI operation mode.
The following diagrams show what will occur if tape or memory operation is interrupted using the MDI function following termination of the program at a block:

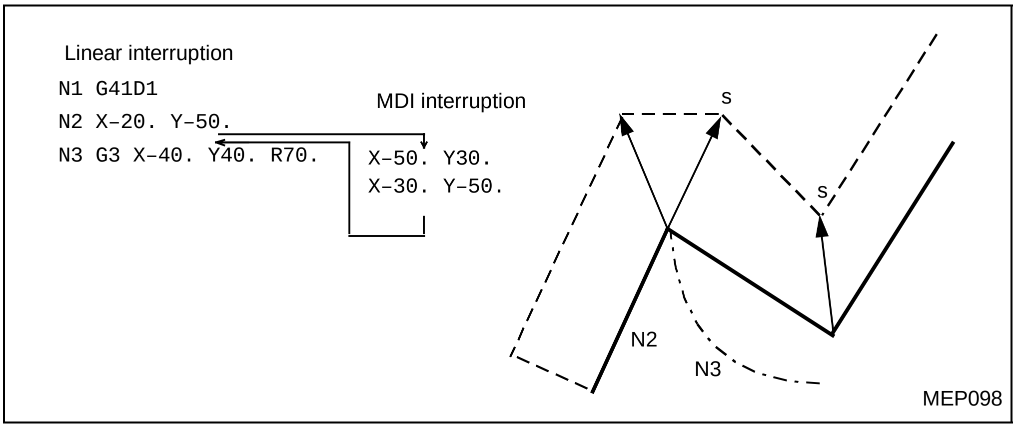
A. Interruption without movement

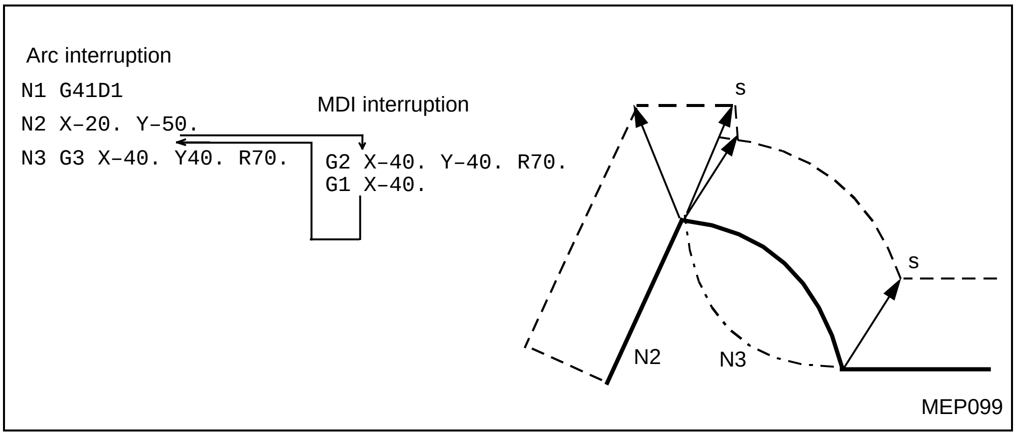
No change in tool path



B. Interruption with movement

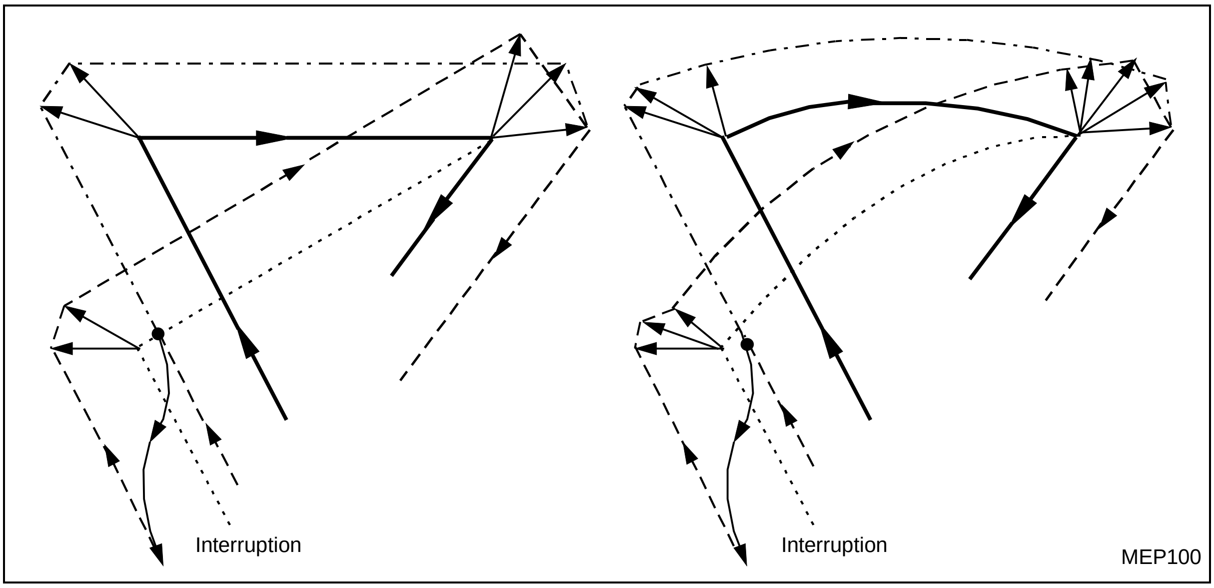
The offset vectors are recalculated automatically at the first effective move block after interruption.





2. Manual interruption

- For the incremental data command mode, the tool path shifts through the interruption amount.
- For the absolute data command mode, the intended tool path is restored at the ending point of the block immediately following that at which interruption has been performed. This state is shown in the diagram below.

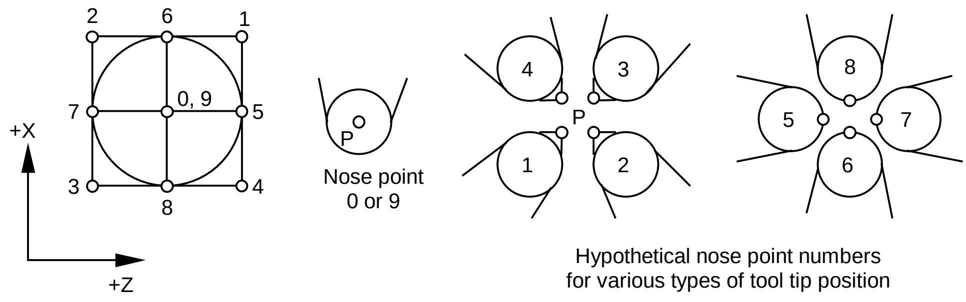


Return to Library

13-4-6 Nose-R compensation

1. Tool nose point (Direction)

To apply the tool diameter offset function to turning tools for nose-R compensation, register the data sets of nose radius and hypothetical nose point (“Nose-R” and “Direction”) for the required tools on the **TOOL OFFSET** display (type C). “Hypothetical nose point” refers here to the reference position for preparing program data of machining with the particular tool (see the figure below).



TEP064

2. Detailed description

1. Register the compensation amount (nose radius) together with the nose point No. (Direction) under a tool offset number.
2. If four or more blocks without move commands exist in five continuous blocks, overcutting or undercutting may result. However, blocks for which optional block skip is valid are ignored.
3. Nose radius compensation function is also valid for fixed cycles (G277 to G279) and roughing cycles (G270, G271, G272 and G273). A roughing cycle, however, is carried out with respect to the finishing contour compensated for nose-R with the compensation being temporarily canceled, and upon completion of the roughing, the compensation mode is retrieved.
4. For threading commands, compensation is temporarily cancelled in one block before.
5. The compensation plane, movement axes and next advance direction vectors depend upon the plane selection with G17, G18 or G19.

   G17 XY plane; X, Y; I, J
   G18 ZX plane; Z, X; K, I
   G19 YZ plane; Y, Z; J, K

13-4-7 General precautions on tool diameter offsetting

1. Selecting the amounts of offset

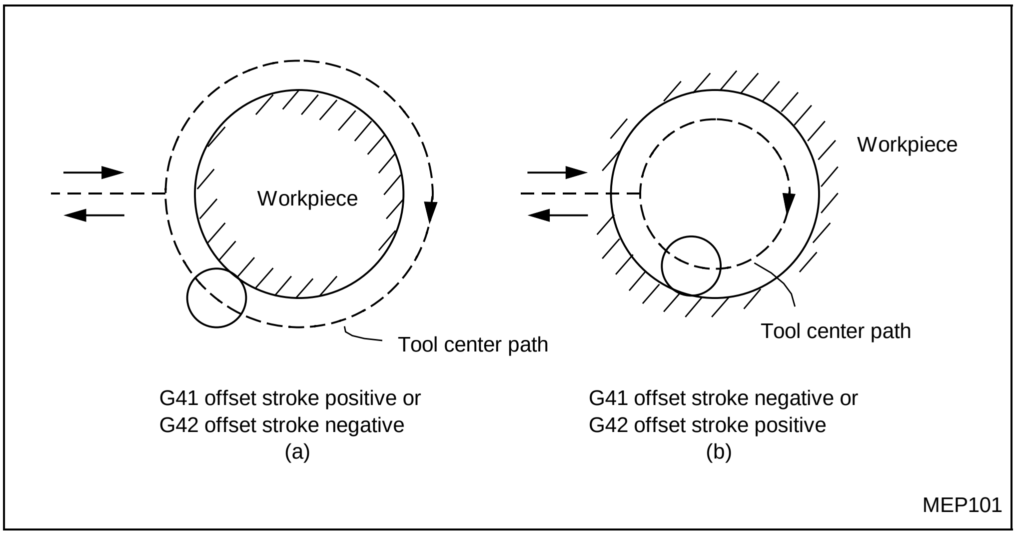
The amounts of offset are selected by specifying an offset number using a D code. Once a D code has been used, it will remain valid until a second D code is used. No H codes can be used to make these selections.
D codes are also used to select tool position offset data.

2. Updating the selected amounts of offset

Updating of the selected amounts of offset is usually to be done after a different tool has been selected during the diameter offsetting cancellation mode. If such updating is done during the offset mode, vectors at the ending point of a block will be calculated using the offset data selected for that block.

3. The sign of offset data and the tool center path

Minus-signed (–) offset data generates the same figure as that obtained when G41 and G42 are exchanged each other. Therefore, the tool center will move around the inside of the workpiece if it has been moving around the outside. Conversely, the tool center will move around the outside of the workpiece if it has been moving around the inside.
Sample programs are shown below. Usually, offset data is to be programmed as plus (+) data. If the tool center has been programmed to move as shown in diagram (a) below, the movement can be changed as shown in diagram (b) below by changing the sign of the offset data to minus (–). Conversely, if the tool center has been programmed to move as shown in diagram (b) below, the movement can be changed as shown in diagram (a) below by changing the sign of the offset data to plus (+). One tape for machining of both inside and outside shapes can be created in this way. Also, a dimensional tolerance between both shapes can be freely set by selecting appropriate offset data (however, Type A is to be used during the start of offsetting or during its cancellation).



4. Offset data item “Direction”

As for data item “Direction” of TOOL OFFSET type C, specify the nose point direction for turning tools. Always set “Direction = 0” for offset numbers to be used for diameter offsetting of milling tools.

Return to Library

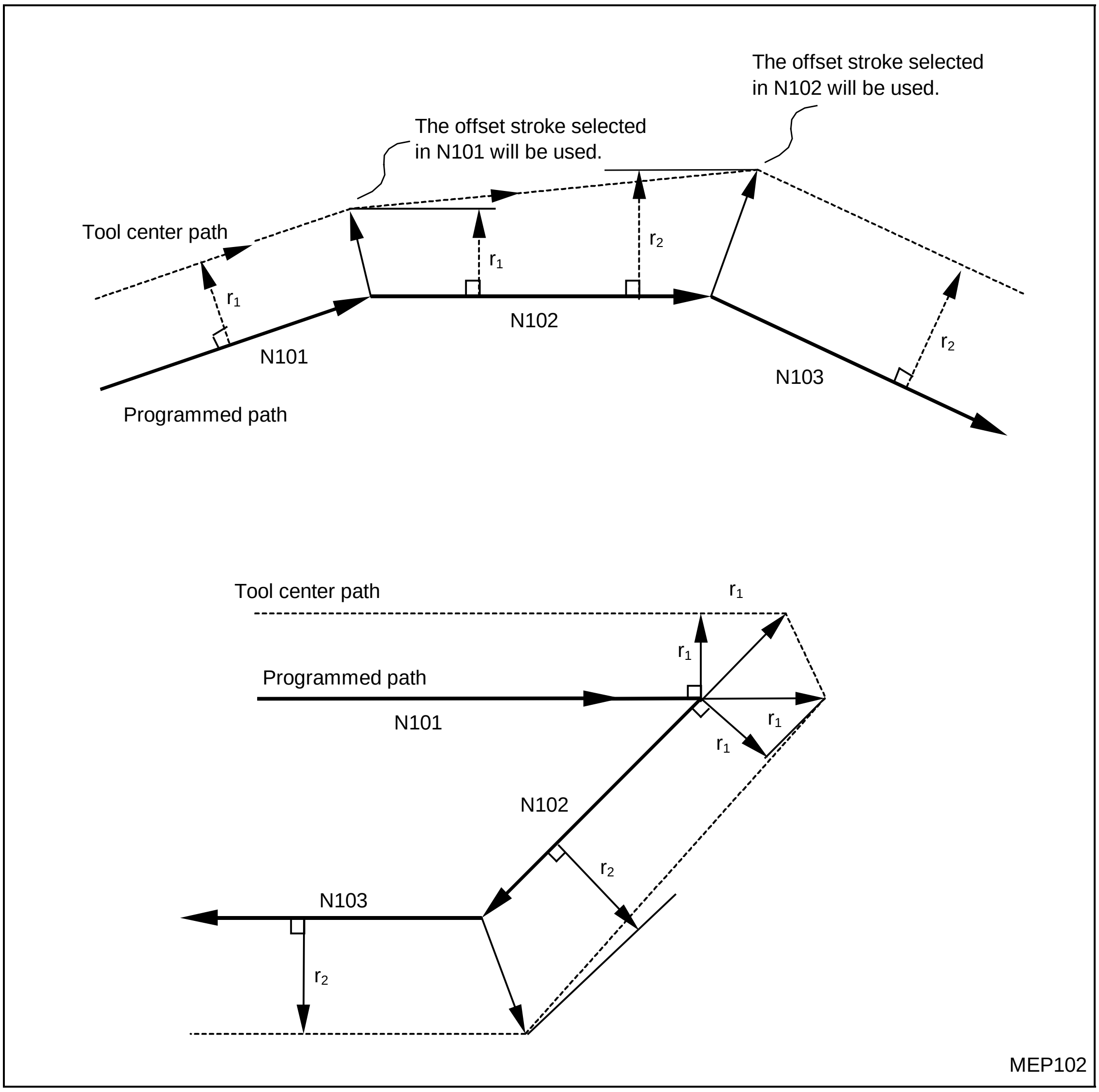
13-4-8 Offset number updating during the offset mode

In principle, offset numbers should not be updated during the offset mode. If updating is done, the tool center will move as shown below.

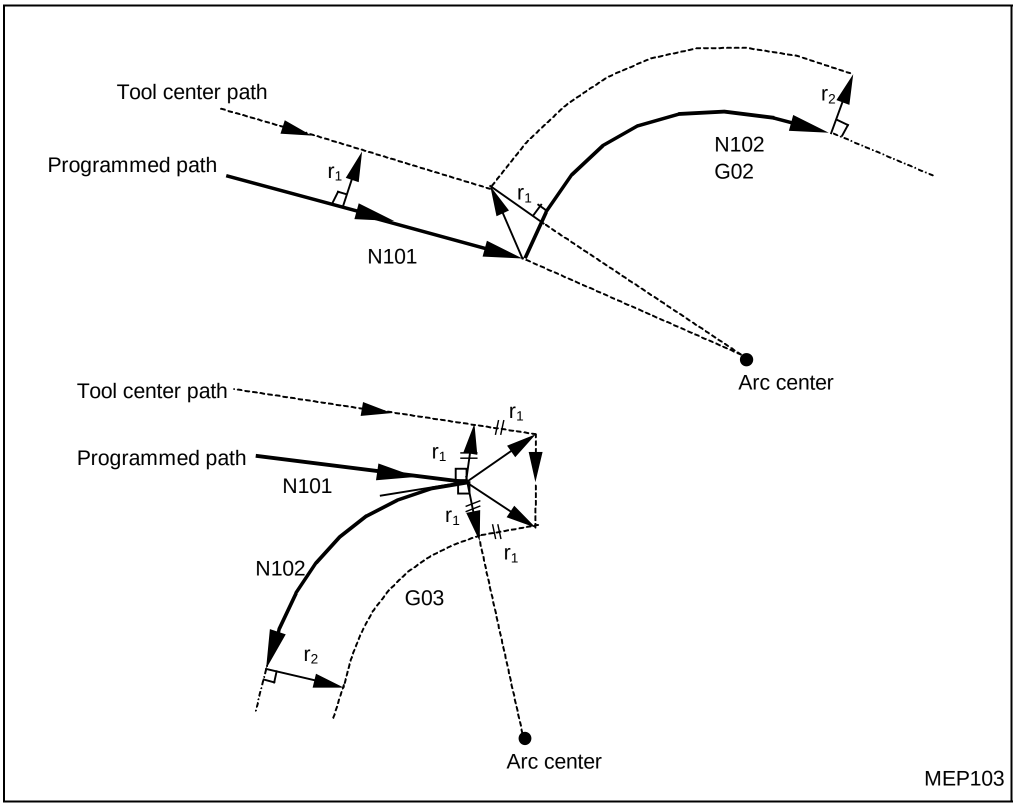
If an offset number (offset data) is updated

G41 G01 Dr_1
⋮
⋮
⋮ α = 0, 1, 2, 3
N101 G0α Xx_1 Yy_1
N102 G0α Xx_2 Yy_2 Dr_2 To change an offset number
N103 Xx_3 Yy_3

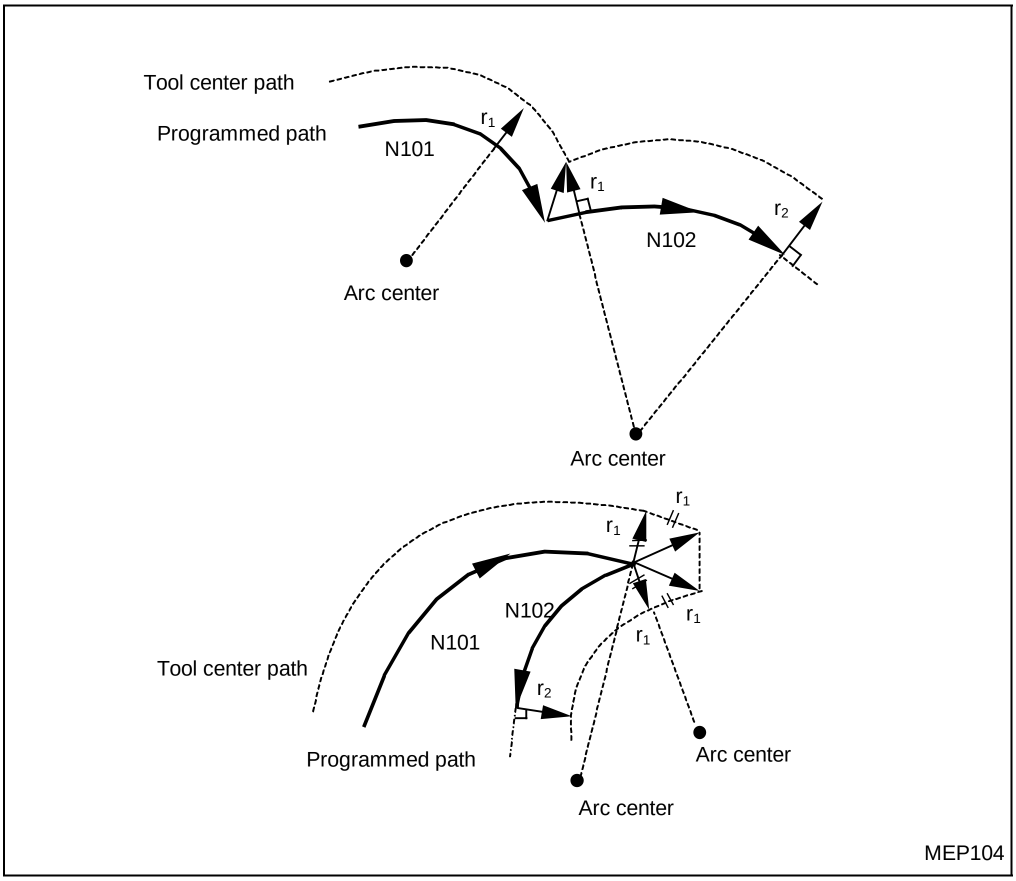
1. Line-to-line movement



2. Line-to-arc movement



3. Arc-to-arc movement

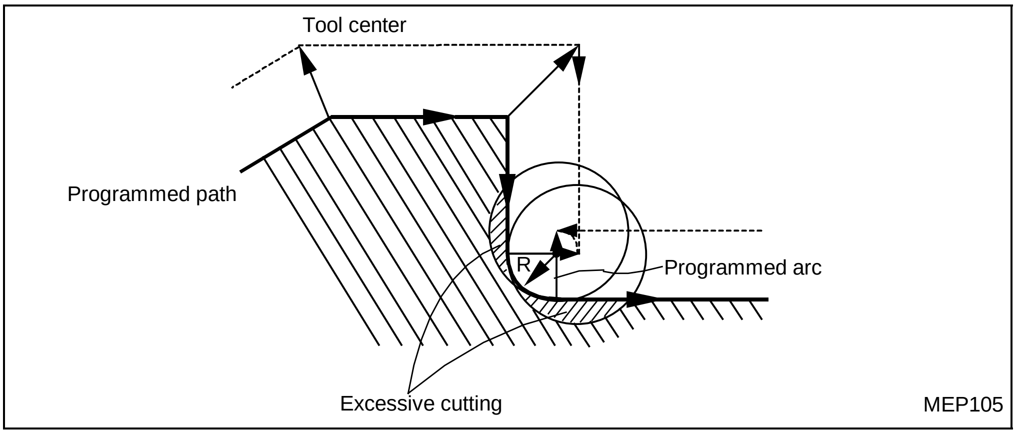


13-4-9 Excessive cutting due to tool diameter offsetting

If an interference check function is not provided, excessive cutting may result in the following three cases:

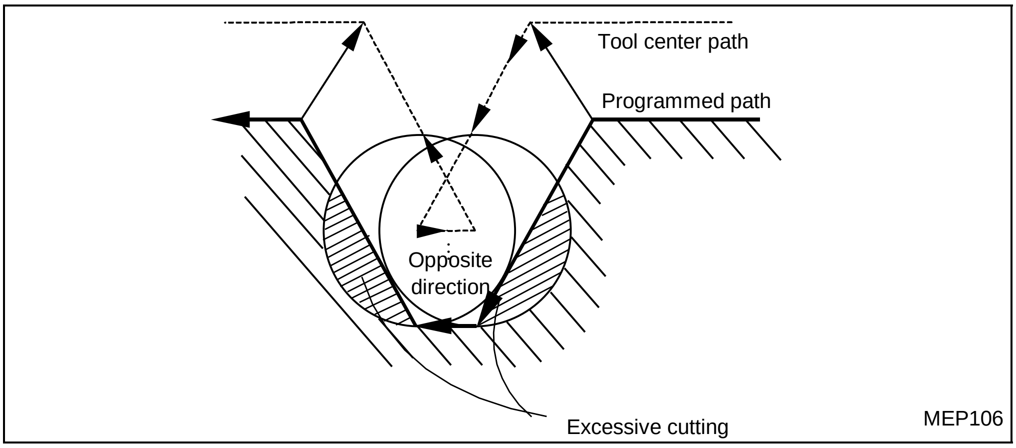
1. Machining of the inside of an arc smaller than the tool radius

If the radius of the programmed arc is smaller than that of the tool, excessive cutting may result from offsetting of the inside of the arc.

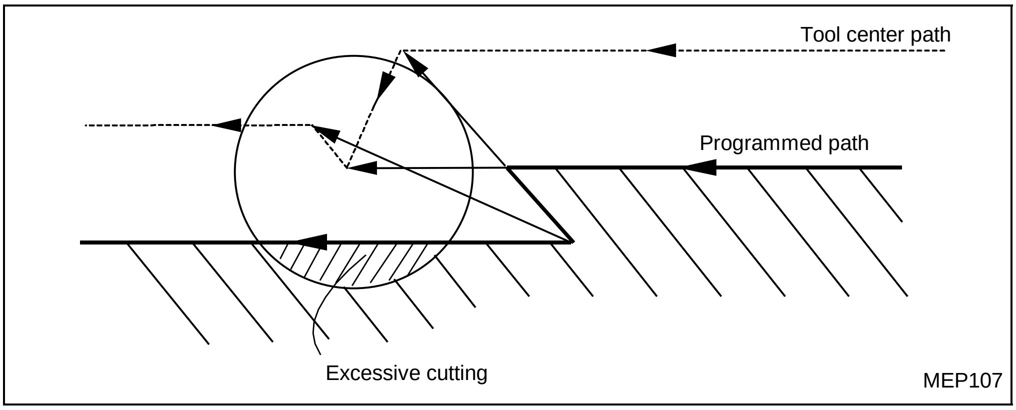


2. Machining of a groove smaller than the tool radius

Excessive cutting may result if tool diameter offsetting makes the moving direction of the tool center opposite to that of the program.



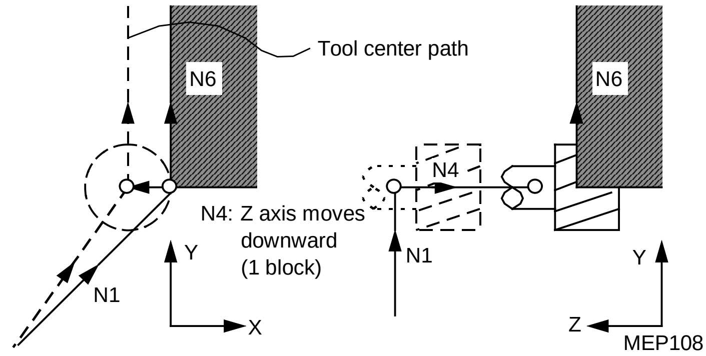
3. Machining of a stepped section smaller than the tool radius



4. Relationship between the start of tool diameter offsetting and the cutting operation in the Z-axis direction

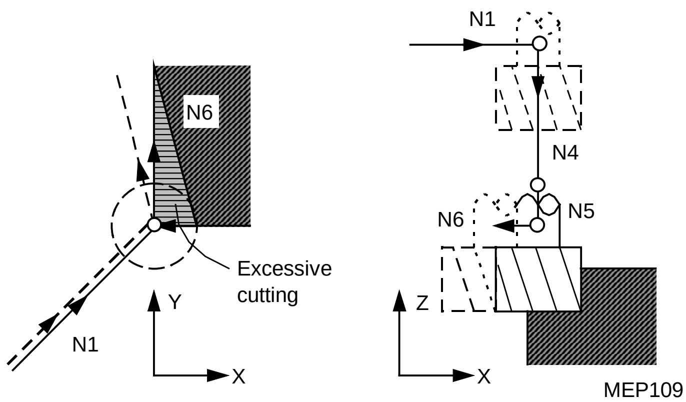
It is generally done that diameter offsetting (usually, on the X-Y plane) is done at a suitable distance from the workpiece during the start of cutting and then the workpiece is cut along the Z-axis. At this time, incorporate the following programming considerations if you want to split the Z-axis action into rapid feed and cutting feed which is to follow only after the Z-axis has moved close to the workpiece:
If you make a program such as that shown below:

```
N1  G91 G00 G41 X500. Y500. D1
N2  S1000
N3  M3
N4  G01   Z-300. F1
N6        Y100.  F2
  .
  .
  .
  .
```



With this program, all blocks up to N6 can be read during the start of offsetting based on N1. Thus, the NC unit will judge the relationship between N1 and N6 and correctly perform the offset operation as shown in the diagram above.
A sample program in which the N4 block in the program shown above has been split into two parts is shown below.

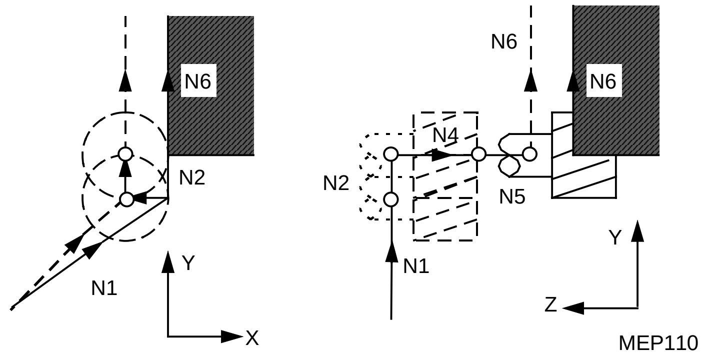
```
N1   G91 G00 G41 X500. Y500. D1
N2   S1000
N3   M3
N4        Z-250.
N5   G01 Z-50. F1
N6        Y100. F2
```



In this case, the N2 through N5 blocks do not have any command corresponding to the X-Y plane and the relevant block N6 cannot be read during the start of offsetting based on N1. As a result, offsetting will be based only on the information contained in the N1 block and thus the NC unit will not be able to create offset vectors during the start of offsetting. This will cause excessive cutting as shown in the diagram above.

Even in such a case, however, excessive cutting can be prevented if a command code that moves the tool in exactly the same direction as that existing after the Z-axis has moved downward is included immediately before the Z-direction cutting block.

```
N1  G91  G00 G41 X500. Y400. D1
N2       Y100. S1000
N3  M3
N4       Z-250.
N5  G01  Z-50.  F1
N6       Y100.  F2
```



For the sample program shown above, correct offsetting is ensured since the moving direction of the tool center at N2 is the same as at N6.

13-4-10 Interference check

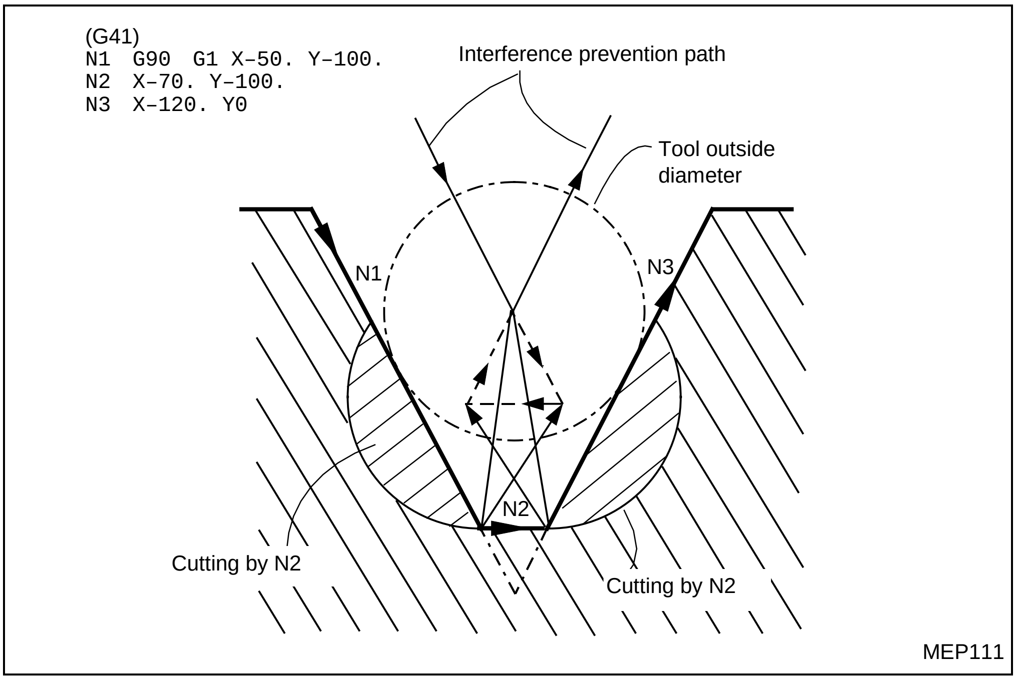
1. Overview

Even a tool whose diameter has been offset by usual tool-diameter offsetting based on two-block pre-reading may move into the workpiece to cut it. This status is referred to as interference, and a function for the prevention of such interference is referred to as interference check.

The following two types of interference check are provided and their selection is to be made using bit 5 of parameter **F92**.

Function	Parameter	Operation
Interference check and alarm	Interference check and prevention off	The system will stop, with a program error resulting before executing the cutting block.
Interference check and prevention	Interference check and prevention on	The path is changed to prevent cutting from taking place.

Example:



MEP111

- For the alarm function

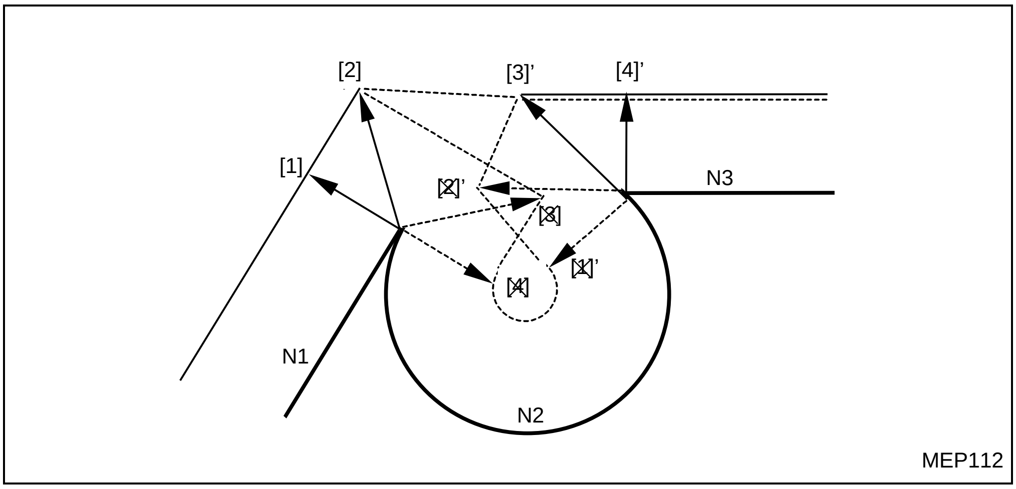
  An alarm occurs before N1 is executed. Machining can therefore be proceeded with by updating the program into, for example,

  `N1 G90 G1 X–20. Y–40.`

  using the buffer correction function.

- For the prevention function

  Interference prevention vectors are generated by N1 and N3 crossing-point calculation.



MEP112

Vector [1] [4]' check → No interference

↓

Vector [2] [3]' check → No interference

↓

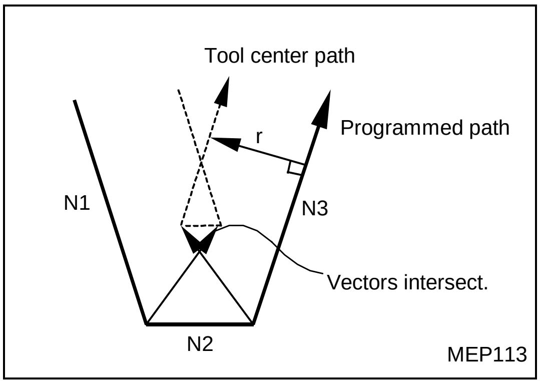
Vector [3] [2]' check → Interference → Vector [3] [2]' deletion → Vector [4] [1]' deletion

The above process is performed to leave vectors [1] [2] [3]' and [4]' as effective ones. Resultantly, the route that connects vectors [1] [2] [3]' and [4] is taken as a bypass for the prevention of interference.

2. Detailed description

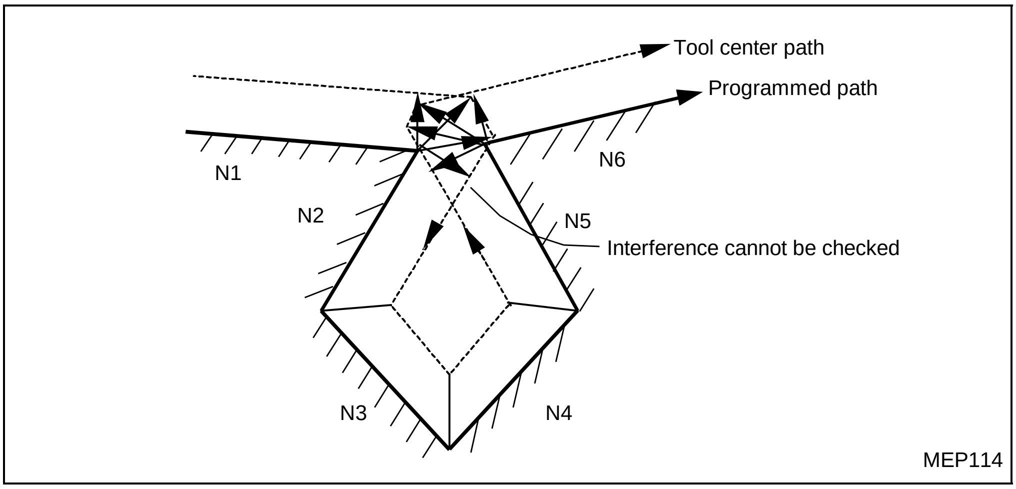
A. The case where interference is regarded as occurring

When move commands are present in three of the five command blocks to be pre-read, interference will be regarded as occurring, if the offset calculation vectors at the block connections of the individual move commands intersect.



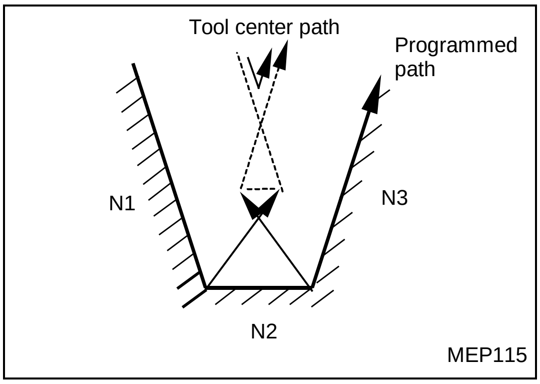
B. Cases where interference check cannot be performed

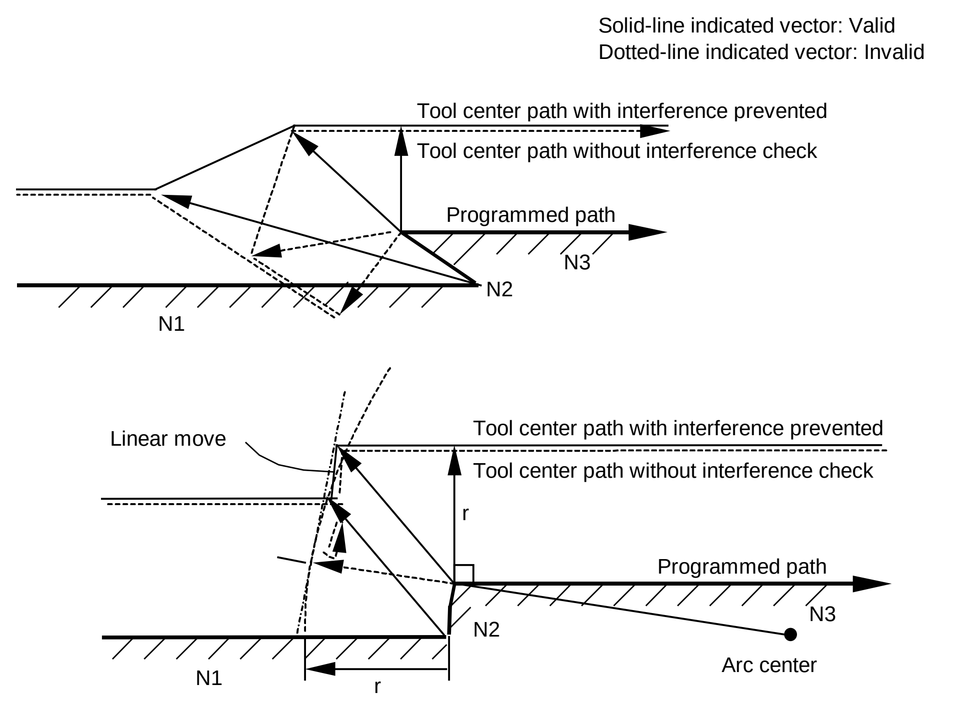
- When pre-reading of three move command blocks of the five to be pre-read is not possible (since the three blocks do not contain move commands).
- When the fourth and subsequent move command blocks themselves interfere.



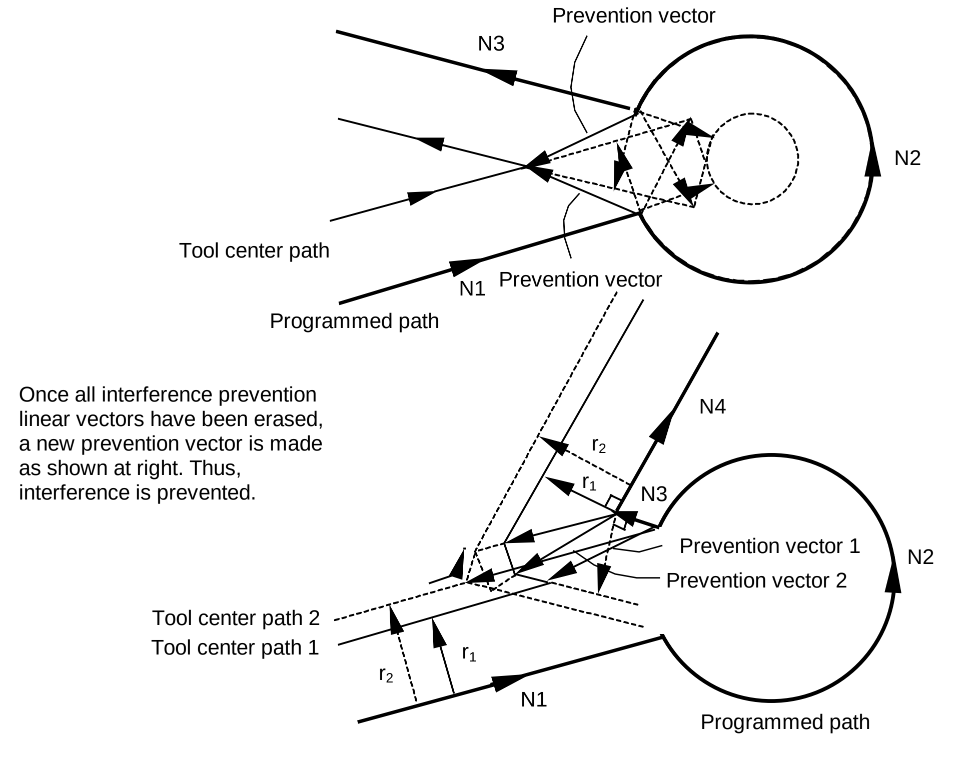
C. Movements during the prevention of interference

The following shows the movements occurring when interference prevention is provided:



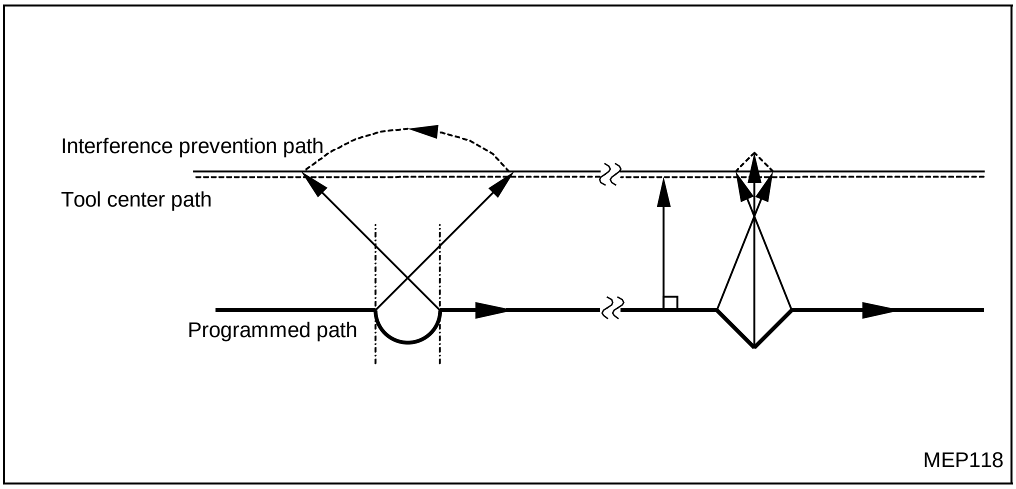


MEP116



MEP117

In the diagram shown below, part of the groove is left uncut:

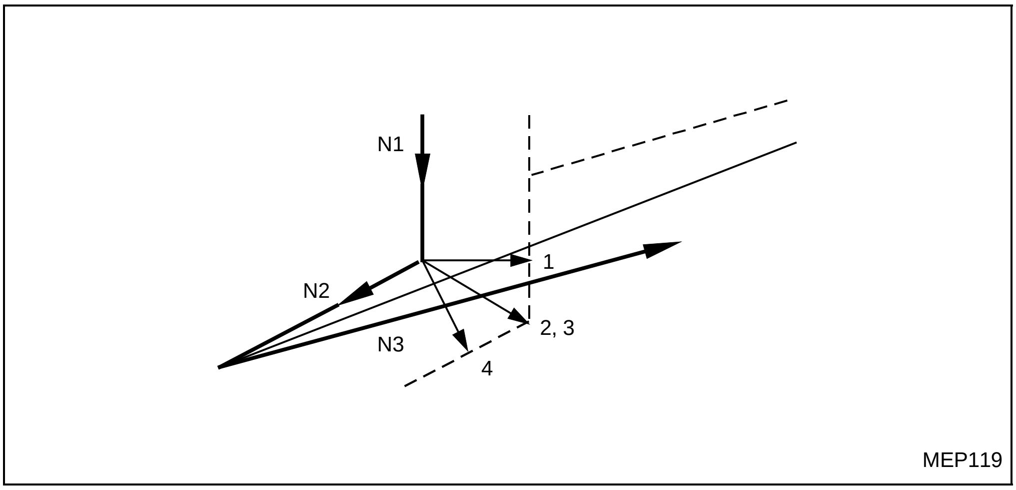


3. Interference alarm

Cases that an interference alarm **837 TOOL OFFSET INTERFERENCE ERROR** occurs are listed below.

<u>When interference check and alarm is selected</u>

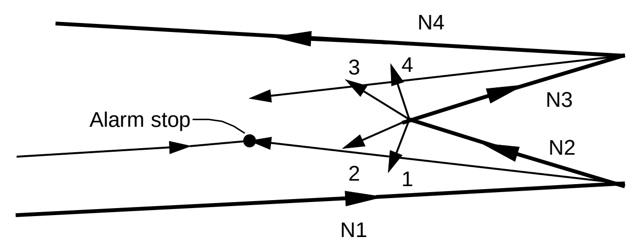
1) If all vectors at the ending point of the current block are erased:
   Prior to execution of N1, a program error will result if vectors 1 through 4 at the ending point of the N1 block are all erased as shown in the diagram below.



When interference check and prevention is selected

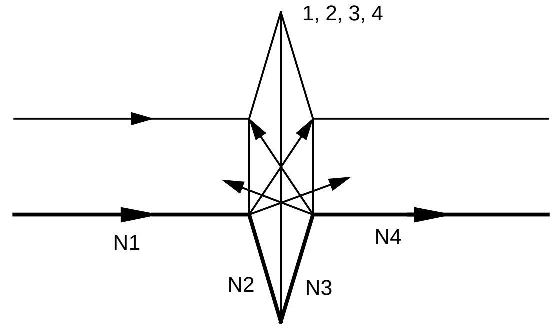
2) If all vectors at the ending point of the current block are erased but an effective vector(s) remains at the ending point of the next block:

- For the diagram shown below, interference checking at N2 will erase all vectors existing at the ending point of N2, but leave the vectors at the ending point of N3 effective.
  At this time, a program error will occur at the ending point of N1.



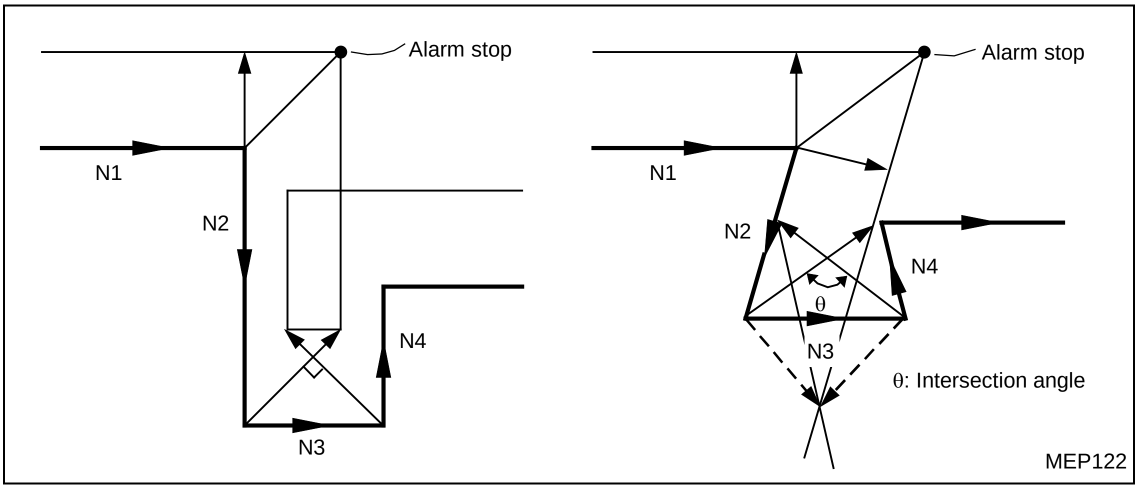
MEP120

- For the diagram shown below, the direction of movement becomes opposite at N2.
  At this time, a program error will occur before execution of N1.

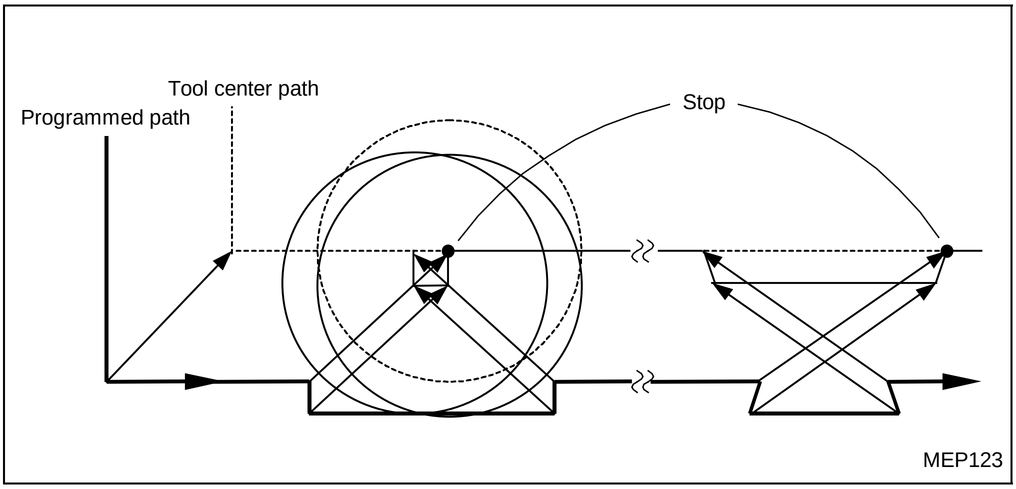


MEP121

3) When prevention vectors cannot be generated:
Prevention vectors may not be generated even when the conditions for generating them are satisfied. Or even after generation, the prevention vectors may interfere with N3.
A program error will therefore occur at the ending point of N1 if those vectors cross at angles of 90 degrees or more.



4) When the after-offsetting moving direction of the tool is opposite to that of the program:
For a program for the machining of parallel or downwardly extending grooves narrower than the tool diameter, interference may be regarded as occurring even if it is not actually occurring.

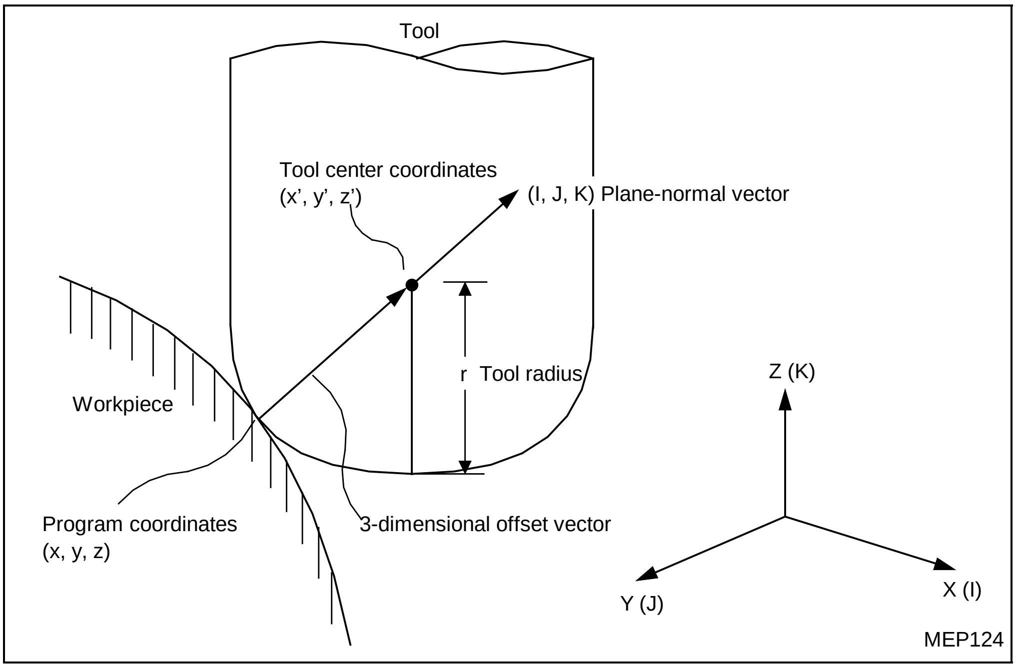


Return to Library

13-5 Three-Dimensional Tool Diameter Offsetting (Option)

Three-dimensional tool diameter offsetting is performed to offset a tool in three-dimensional space according to the previously designated three-dimensional vectors.

13-5-1 Function description



As shown in the diagram above, the tool is moved through the tool radius r in the plane-normal vectorial direction of (I, J, K) from the program coordinates (x, y, z) to the offset tool center coordinates (x', y', z'). Also, unlike two-dimensional tool diameter offsetting, which generates vectors perpendicular to the direction of (I, J, K), three-dimensional tool diameter offsetting generates vectors in the direction of (I, J, K). (The vectors are generated at the ending point of that block.) The axis components of three-dimensional offset vectors become:

$$Hx = \frac{I}{\sqrt{I^2 + J^2 + K^2}} \bullet r$$

$$Hy = \frac{J}{\sqrt{I^2 + J^2 + K^2}} \bullet r$$

$$Hz = \frac{K}{\sqrt{I^2 + J^2 + K^2}} \bullet r$$

Hence, the tool center coordinates (x', y', z') are expressed as

$x' = x + Hx$
$y' = y + Hy$
$z' = z + Hz$

where (x, y, z) denote the program coordinates.

Note 1: The three-dimensional vectors (Hx, Hy, Hz) refer to plane-normal vectors that are identical to the plane-normal vectors (I, J, K) in direction and have a magnitude of r (tool radius).

Note 2: If parameter **F11** is set to a value other than 0, the value of **F11** will be used as $\sqrt{I^2 + J^2 + K^2}$.

13-5-2 Programming methods

1. G-codes and their functions

G-code	Parameter and feature		
	Offset stroke positive	Offset stroke negative	Offset No. D00
G40	To cancel the 3-dimensional tool diameter offset	To cancel	To cancel
G41	To offset in (I, J, K) direction	To offset in the direction opposite to (I, J, K)	To cancel
G42	To offset in the direction opposite to (I, J, K)	To offset in (I, J, K) direction	To cancel

2. Offset data

For the tool radius r that is to be offset, the offset number under which that offset amount has been registered must be selected using D.
The maximum available number of sets of offset numbers is as follows:

Standard: 128 sets: D1 to D128
Optional: 512 sets: D1 to D512 (max.)

3. Space in which offsetting is to be performed

The space in which offsetting is to be performed is determined by the axis address commands (X, Y, Z, U, V, W) that are contained in the starting block of three-dimensional tool diameter offsetting. When the U-, V-, and W-axes are taken as additions to the X-, Y-, and Z-axes, respectively, priority will be given to the X-, Y-, or Z axis if the X axis and the U axis (or Y and V, or Z and W) are selected at the same time. Coordinate axes that have not been addressed will be interpreted as the X axis, the Y axis, and the Z axis, respectively.

Example: G41 Xx_1 Yy_1 Zz_1 Ii_1 Jj_1 Kk_1 XYZ space
G41 Yy_2 Ii_2 Jj_2 Kk_2 XYZ space
G41 Xx_3 Vv_3 Zz_3 Ii_3 Kk_3 XVZ space
G41 Ww_4 Ii_4 Jj_4 Kk_4 XYW space

4. Starting a three-dimensional tool diameter offset operation

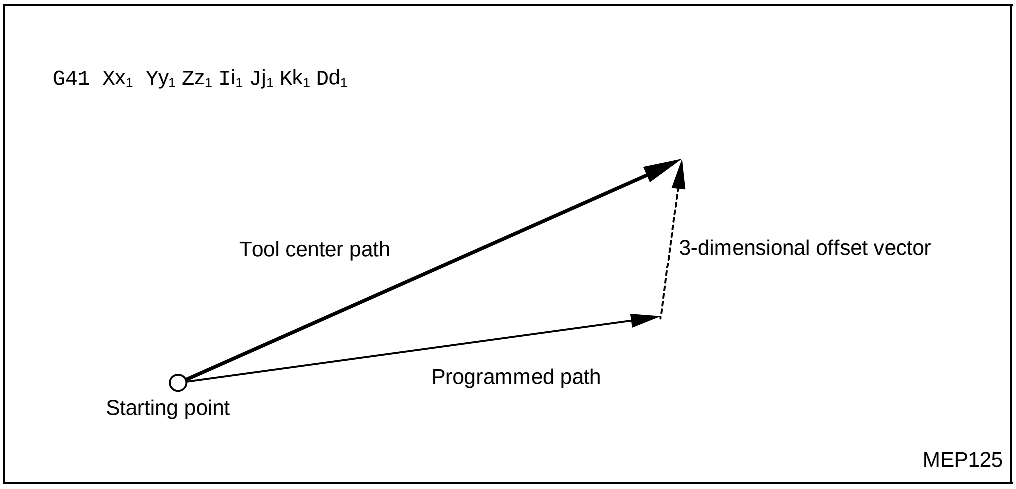
Offset number D and the plane-normal vectors (I, J, K) must be set in the same block as that which contains three-dimensional tool diameter offset command code G41 (or G42). In that case, (I, J, K) must be set for each of the X-, Y-, and Z-axes. If this vector setting is not complete (setting of zero for I, J or K is effective), the usual tool diameter offset mode will be set. If, however, the machine does not have the three-dimensional tool diameter offset function, an alarm **838 3-D OFFSET OPTION NOT FOUND** will result.

G41 (G42) Xx_1 Yy_1 Zz_1 Ii_1 Jj_1 Kk_1 Dd_1

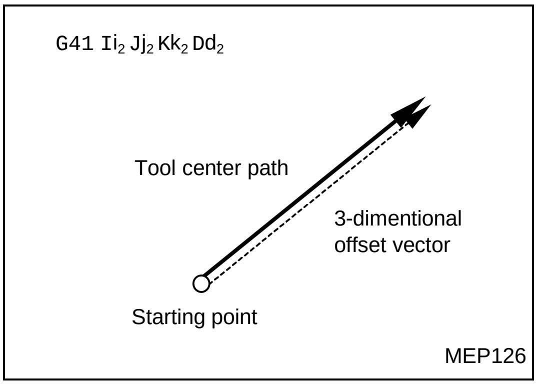
G41 (G42) : 3-dimensional tool diameter offset command
X, Y, Z : Command to move each axis and to determine an offsetting space
I, J, K : To indicate the offsetting direction in plane-normal vectors
D : Offset number

Use the G00 or G01 mode to start the three-dimensional tool diameter offset operation. Use of the G02 or G03 mode results in an alarm **835 G41, G42 FORMAT ERROR.**

Example 1: If move commands are present:



Example 2: If move commands are not present:

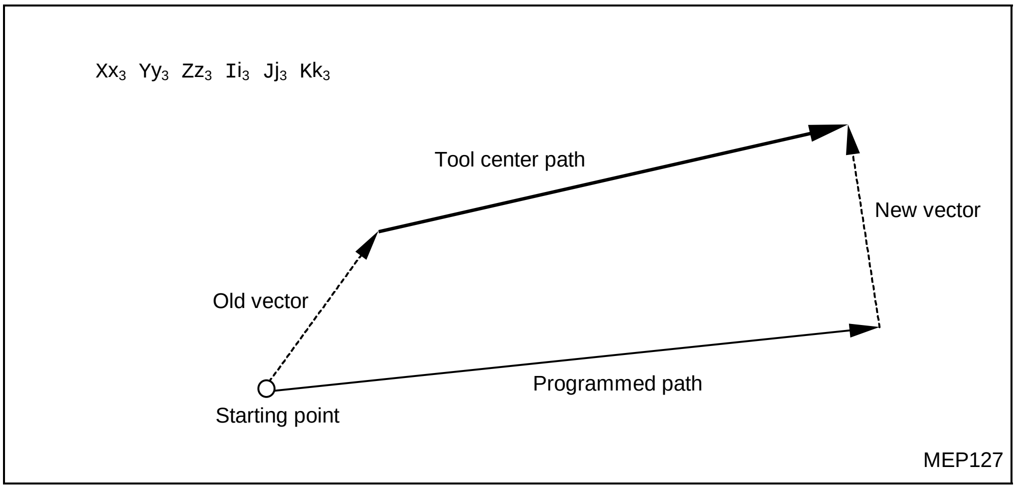


5. During three-dimensional tool diameter offsetting

Set move commands and new plane-normal vector commands as follows:

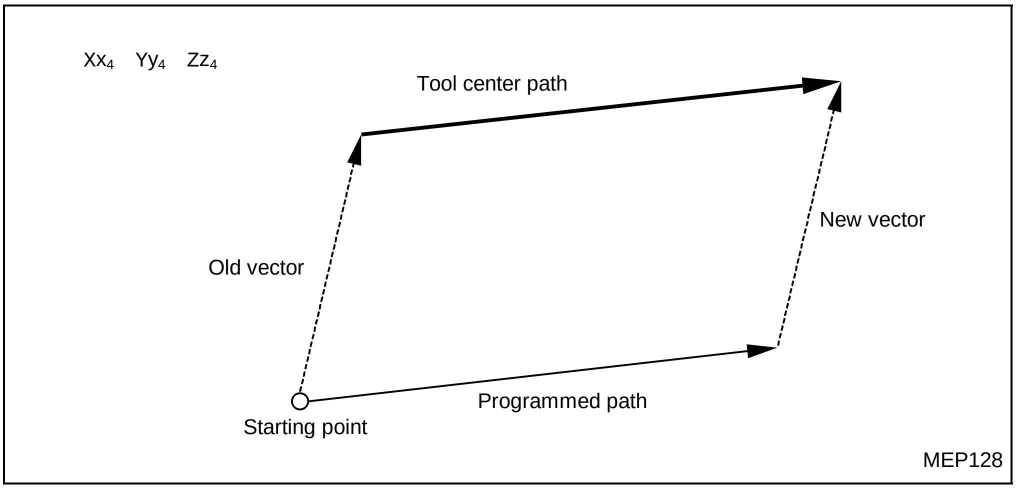
Xx_3 Yy_3 Zz_3 Ii_3 Jj_3 Kk_3

Example 1: If move commands and plane-normal vector commands are present:

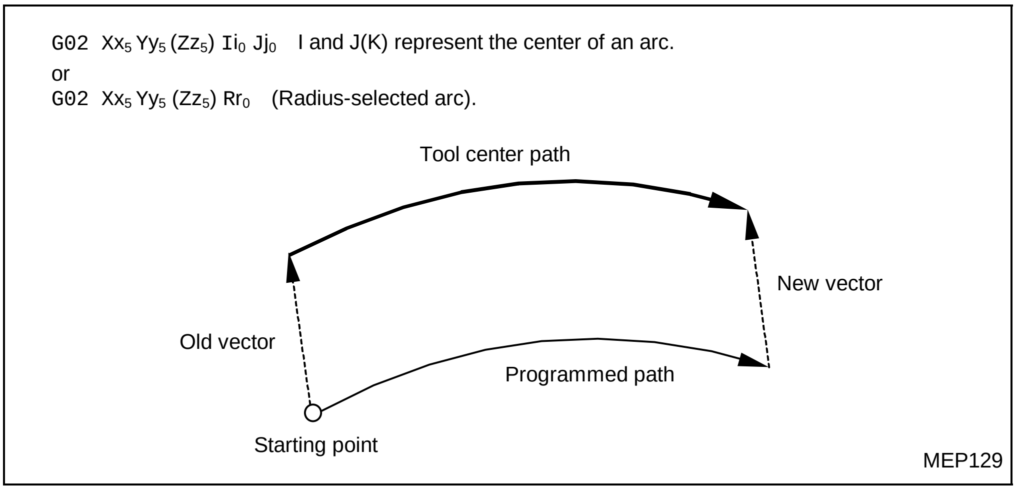


Return to Library

Example 2: If plane-normal vector commands are not present:
The new vector is the same as the old one.

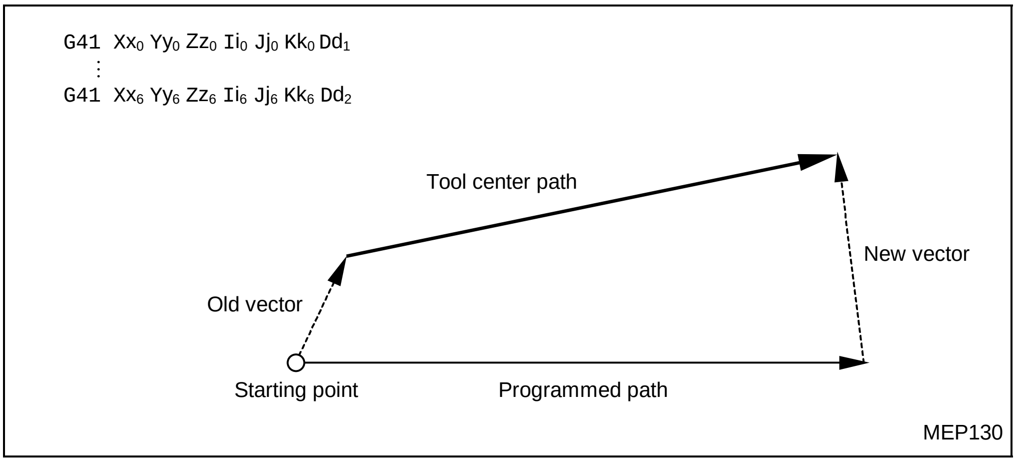


Example 3: For arc or helical cutting:
The new vector is the same as the old one.

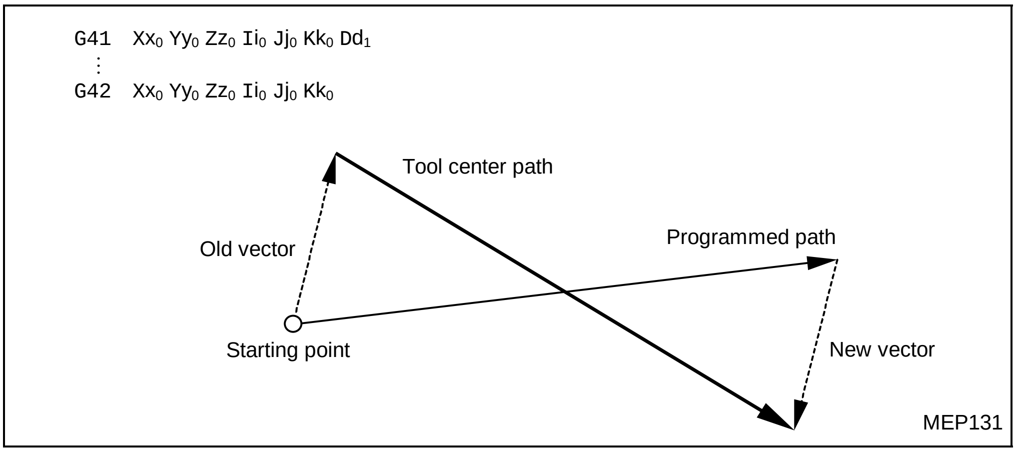


Note: The arc shifts through the amount of vector.

Example 4: For changing the offset data:
Set offset number D in the same block as that of three-dimensional tool diameter offset command G41 or G42. Use the G00 or G01 mode to change the offset data. Use of the arc mode results in **835 G41, G42 FORMAT ERROR.**



Example 5: For changing the offset direction:



Use the G00 or G01 mode to change the offset direction. Use of the arc mode results in an alarm **835 G41, G42 FORMAT ERROR**.

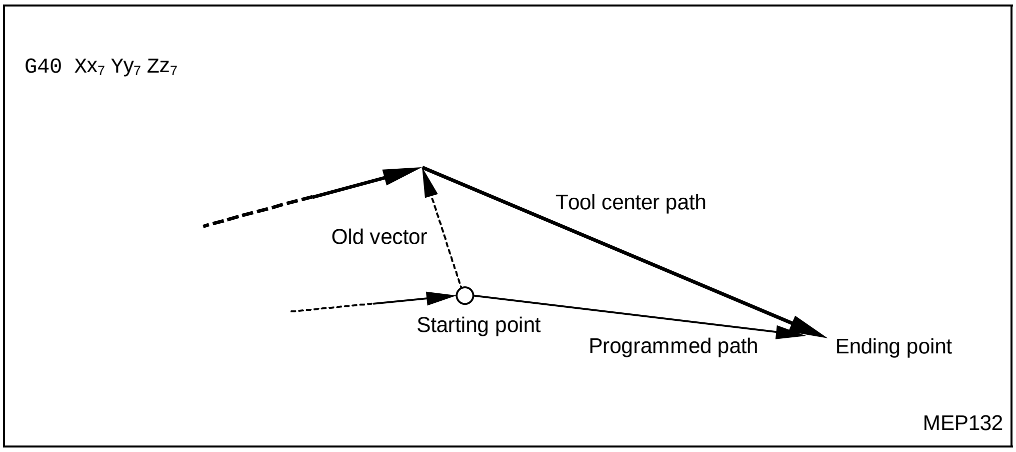
6. Cancelling the three-dimensional tool diameter offset operation

Make the program as follows:

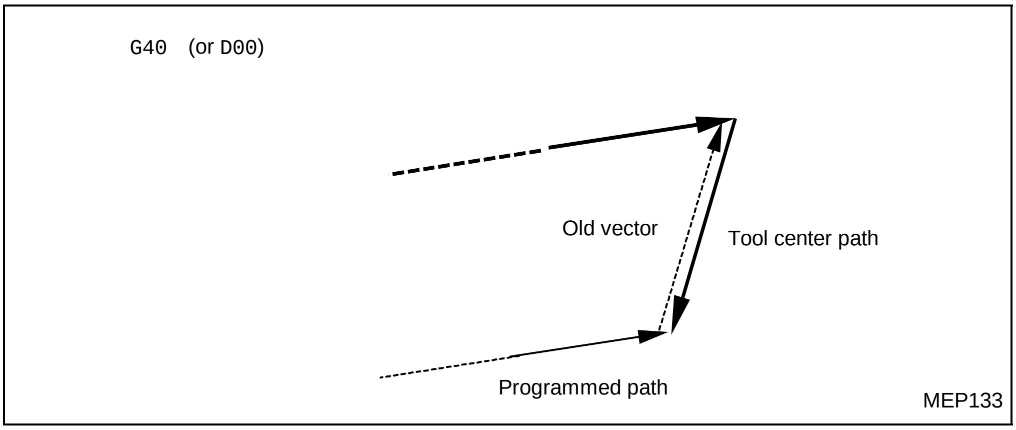
G40 Xx_7 Yy_7 Zz_7

Use the G00 or G01 mode to cancel three-dimensional tool diameter offsetting. Use of the G02 or G03 mode results in an alarm **835 G41, G42 FORMAT ERROR**.

Example 1: If move commands are present:



Example 2: If move commands are not present:



13-5-3 Correlationships to other functions

1. Tool diameter offset
   The usual tool-diameter offset mode will be selected if setting of plane-normal vectors (I, J, K) in the starting block of three-dimensional tool diameter offsetting is not done for each of the X-, Y-, and Z-axes.
2. Tool length offset
   Tool length offsetting is performed according to the coordinates existing after execution of three-dimensional tool diameter offsetting.
3. Tool position offset
   Tool position offsetting is performed according to the coordinates existing after execution of three-dimensional tool diameter offsetting.
4. Selection of fixed-cycle operation results in an alarm **901 INCORRECT FIXED CYCLE COMMAND**.
5. Scaling
   Three-dimensional tool diameter offsetting is performed according to the coordinates existing after execution of scaling.
6. Home position check (G27)
   The current offset data is not cancelled.

13-5-4 Miscellaneous notes on three-dimensional tool diameter offsetting

1. Although they can be used to select offset numbers, D-code commands are valid only after command G41 or G42 has been set. If a D-code command is not present, the previous D-code command becomes valid.
2. Use the G00 or G01 mode to change the offset mode, the offset direction or the offset data. An alarm **835 G41, G42 FORMAT ERROR** will occur if an attempt is made to perform these changes in an arc mode.
3. During the three-dimensional tool diameter offset mode using a space, three-dimensional tool diameter offsetting cannot be done using any other space. The cancel command code (G40 or D00) must be executed to select some other offset space.

   Example:

`G41 X_ Y_ Z_ I_ J_ K_`	To start offsetting in X, Y and Z space
⋮	
`G41 U_ Y_ Z_ I_ J_ K_`	To offset in X, Y and Z space while the U axis moves by the command value

4. Selection of an offset number falling outside the range from 1 to 128 (for standard machine specifications) or from 1 to 512 (for optional machine specifications) results in an alarm **839 ILLEGAL OFFSET No.**
5. Only the G40 or D00 command code can be used to cancel three-dimensional tool diameter offsetting. Cancellation is not possible with the NC reset key or external reset functions.
6. A program error will result if the vectorial magnitude specified by (I, J, K), that is $\sqrt{I^2 + J^2 + K^2}$, overflows.

13-6 Programmed Data Setting: G10

1. Function and purpose

The G10 command allows tool offset data, work offset data and parameter data to be set or modified in the flow of program.

2. Programming formats

A. Programming workpiece offsets

- Programming format for the workpiece origin data

G10 L2 P_ X_ Y_ Z_α_ (α: Additional axis)

P: 0.....Coordinate shift (Added feature)
1.....G54
2.....G55
3.....G56
4.....G57
5.....G58
6.....G59

Data of P-commands other than those listed above are handled as P = 1.
If P-command setting is omitted, the workpiece offsets will be handled as currently effective ones.

- Programming format for the additional workpiece origin data (option)

G10 L20 P_ X_ Y_ Z_α_ (α: Additional axis)

P1: G54.1 P1
P2: G54.1 P2
⋮
P47: G54.1 P47
P48: G54.1 P48

The setting ranges of the data at axial addresses are as follows:

	Micron system		Sub-micron for rotational axes		Sub-micron for all axes	
	Metric	Inch	Metric	Inch	Metric	Inch
Linear	±99999.999 mm	±9999.9999 in.	±99999.999 mm	±9999.9999 in.	±99999.9999 mm	±9999.99999 in.
Rotat.	±99999.999°	±99999.999°	±99999.9999°	±99999.9999°	±99999.9999°	±99999.9999°

B. Programming tool offsets

- Programming format for the tool offset data of Type A

G10 L10 P_R_

P: Offset number
R: Offset amount

- Programming format for the tool offset data of Type B

G10 L10 P_R_ Geometric offset concerning the length
G10 L11 P_R_ Wear compensation concerning the length
G10 L12 P_R_ Geometric offset concerning the diameter
G10 L13 P_R_ Wear compensation concerning the diameter

- Programming format for the tool offset data of Type C

G10 L10 P_R_	Length offset; Geometric Z
G10 L11 P_R_	Length offset; Wear comp. Z
G10 L12 P_R_	Diameter/Nose-R offset (Geometric)
G10 L13 P_R_	Diameter/Nose-R offset (Wear comp.)
G10 L14 P_R_	Length offset; Geometric X
G10 L15 P_R_	Length offset; Wear comp. X
G10 L16 P_R_	Length offset; Geometric Y
G10 L17 P_R_	Length offset; Wear comp. Y
G10 L18 P_R_	Nose-R offset; Direction

The setting ranges for programming tool offset data are as follows:

Offset number (P):

1 to 128 or 512 (according to the number of available data sets)

Offset amount (R):

	Micron system		Sub-micron for rotational axes		Sub-micron for all axes	
	Metric	Inch	Metric	Inch	Metric	Inch
TOOL OFFSET Type A	±9999.999 mm	±845.0000 in.	±1999.999 mm	±84.5000 in.	±1999.9999 mm	±84.50000 in.
TOOL OFFSET Type B Length Geom.	±9999.999 mm	±845.0000 in.	±1999.999 mm	±84.5000 in.	±1999.9999 mm	±84.50000 in.
TOOL OFFSET Type B Length Wear	±99.999 mm	±9.9999 in.	±99.999 mm	±9.9999 in.	±99.9999 mm	±9.99999 in.
TOOL OFFSET Type B Dia. Geom.	±999.999 mm	±99.9999 in.	±999.999 mm	±84.5000 in.	±999.9999 mm	±84.50000 in.
TOOL OFFSET Type B Dia. Wear	±9.999 mm	±0.9999 in.	±9.999 mm	±0.9999 in.	±9.9999 mm	±0.99999 in.
TOOL OFFSET Type C Geom. XYZ	±9999.999 mm	±845.0000 in.	±1999.999 mm	±84.5000 in.	±1999.9999 mm	±84.50000 in.
TOOL OFFSET Type C Geom. Nose-R	±999.999 mm	±99.9999 in.	±999.999 mm	±84.5000 in.	±999.9999 mm	±84.50000 in.
TOOL OFFSET Type C Wear XYZ	±99.999 mm	±9.9999 in.	±99.999 mm	±9.9999 in.	±99.9999 mm	±9.99999 in.
TOOL OFFSET Type C Wear Nose-R	±9.999 mm	±0.9999 in.	±9.999 mm	±0.9999 in.	±9.9999 mm	±0.99999 in.
TOOL OFFSET Type C Direction	0 - 9	0 - 9	0 - 9	0 - 9	0 - 9	0 - 9

C. Programming parameter data

G10 L50................ Parameter input mode ON
N_P_R_
N_R_
G11 .... Parameter input mode OFF

N: Parameter number
P: Axis number (for axis type parameter)
R: Data of parameter

Specify the parameters with address N as indicated below:

Parameter		N: Number	P: Axis No.
A	1 to 108	1001 to 1108	—
B	1 to 108	2001 to 2108	—
C	1 to 108	3001 to 3108	—
D	1 to 108	4001 to 4108	—
E	1 to 108	5001 to 5108	—
F	1 to 154 (47 to 66 excluded)	6001 to 6154	—
I	1 to 18	9001 to 9018	1 to 14
J	1 to 108	10001 to 10108	—
K	1 to 108	11001 to 11108	—
L	1 to 108	12001 to 12108	—
M	1 to 22	13001 to 13022	1 to 14
N	1 to 22	14001 to 14022	1 to 14
P	1 to 5	150001 to 150005	1 to 14
#	0 to 4095	150100 to 154195	1 to 14
S	1 to 22	16001 to 16022	1 to 14
SV	1 to 96	17001 to 17096	1 to 14
SP	1 to 384	18001 to 18384	1 to 4
SA	1 to 88	19001 to 19088	1 to 4
BA	1 to 132	20001 to 20132	—
TC	1 to 154	21001 to 21154	—

Note: As for the setting ranges of parameter data, refer to the Parameter List.

3. Detailed description

A. Workpiece origin data input

1. The G10 command is not associated with movement. However, do not use this command in the same block with a G-code command other than: G21, G22, G54 to G59, G90 and G91.
2. Do not use the G10 command in the same block with a fixed cycle command or a sub-program call command. This will cause a malfunctioning or a program error.
3. Irrespective of workpiece offset type (G54 - G59 and G54.1), the data to the axial addresses have to refer to the origin of the fundamental machine coordinate system.
4. Depending upon the data input mode — absolute (G90) or incremental (G91) — the designated data will overwrite, or will be added to, the existing data.
5. L-code and P-code commands can be omitted, indeed, but take notice of the following when omitting them:
   1) Omit both L-code and P-code commands only when

      The axial data should refer to the coordinate system that was last selected.
   2) The L-code command only may be omitted when the intended axial data refer to a coordinate system of the same type (in terms of L-code: L2 or L20) as the last selected one; give a P-command in such a case as follows:

      - Set an integer from 0 to 6 with address P to specify the coordinate shift data or one of the coordinate systems from G54 to G59.

      - Set an integer from 1 to 48 with address P to specify one of the additional workpiece coordinate systems of G54.1.
   3) If the P-code command only is omitted:

      An alarm will result if the value of L mismatches the coordinate system last selected.
6. Axial data without a decimal point can be entered in the range from –99999999 to +99999999. The data settings at that time depend upon the data input unit.

   Example: `G10 L2 P1 X–100. Y–1000 Z–100 B–1000`

   The above command sets the following data:

Metric system	X –100.	Y –1.	Z –0.1	B –1.
Metric system (up to 4 dec. places)	X –100.	Y –0.1	Z –0.01	B –0.1
Inch system	X –100.	Y –0.1	Z –0.01	B –1.
Inch system (up to 5 dec. places)	X –100.	Y –0.01	Z –0.001	B –0.1
7. The origin data updated by a G10 command are not indicated as they are on the **WORK OFFSET** display until that display has been selected anew.
8. Setting an illegal L-code value causes an alarm.
9. Setting an illegal P-code value causes an alarm.
10. Setting an illegal axial value causes an alarm.
11. The G10 command is invalid (or skipped) during tool path check.

B. Tool offset data input

1. The G10 command is not associated with movement. However, do not use this command in the same block with a G-code command other than: G21, G22, G54 to G59, G90 and G91.
2. Do not use the G10 command in the same block with a fixed cycle command or a sub-program call command. This will cause a malfunctioning or a program error.

3. Depending upon the data input mode — absolute (G90) or incremental (G91) — the designated data will overwrite, or will be added to, the existing data.
4. Offset data (R) without a decimal point can be entered in the range from –999999 to +999999 for geometric offset, or in the range from –99999 to +99999 for wear compensation. The data settings at that time depend upon the data input unit.

   Example: `G10 L10 P1 R1000`

   The above command sets the following data:

Metric system	1.
Metric system (up to 4 dec. places)	0.1
Inch system	0.1
Inch system (up to 5 dec. places)	0.01

5. The offset data updated by a G10 command are not indicated as they are on the **TOOL OFFSET** display until that display has been selected anew.
6. Setting an illegal L-code value causes an alarm.
7. A command of "G10 P_ R_" without an L-code is also available for tool offset data input.
8. Setting an illegal P-code value causes an alarm.
9. Setting an illegal offset value (R) causes an alarm.
10. The G10 command is invalid (or skipped) during tool path check.

C. Parameter data input

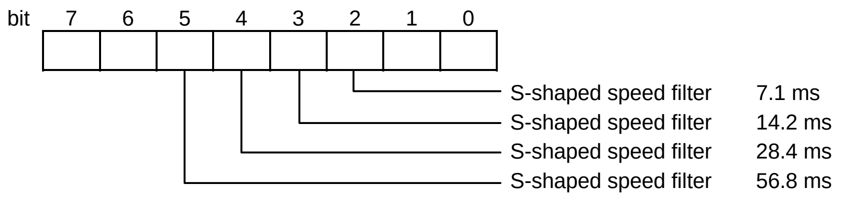
1. The G10 command is not associated with movement. However, do not use this command in the same block with a G-code command other than: G21, G22, G54 to G59, G90 and G91.
2. Do not use the G10 command in the same block with a fixed cycle command or a sub-program call command. This will cause a malfunctioning or a program error.
3. Other NC statements must not be given in the parameter input mode.
4. No sequence number must be designated with address N in the parameter input mode.
5. Irrespective of the data input mode — absolute (G90) or incremental (G91) — the designated data will overwrite the existing parameter. Moreover, describe all the data in decimal numbers (hexadecimal and bit type data, therefore, must be converted).

   Example: For changing a bit type data of 00110110 to 00110111:

   Since $(00110111)_2 = (55)_{10}$ [a binary number of 00110111 corresponds to "55" in decimal notation], set 55 with address R.

6. All decimal places, even if inputted, are ignored.
7. Some specific bit-type parameters require selection of one of multiple bits. For the parameter shown as an example below, set data that turns on only one of bits 2 to 5.

   Example: Parameter **K107**



Setting "1" for bits 2 and 3, for example, could not make valid a speed filter of 21.3 msec (= 7.1 + 14.2).

8. The parameter data updated by a G10 L50 command are not made valid till the execution of a G11 command.
9. The parameter data updated by a G10 L50 command are not indicated as they are on the **PARAMETER** display until that display has been selected anew.
10. Setting an illegal L-code value causes an alarm.
11. Setting an illegal N-code value (parameter No.) causes an alarm.
12. Omission of P-code for an axis type parameter causes an alarm.
13. Setting an illegal parameter value with address R causes an alarm.
14. The G10 command is invalid (or skipped) during tool path check.

4. Sample programs

A. Entering tool offset data from tape

```
··· G10L10P10R–12345 G10L10P05R98765 G10L10P40R2468 ···
```

H10 = –12345   H05 = 98765   H40 = 2468

B. Updating tool offset data

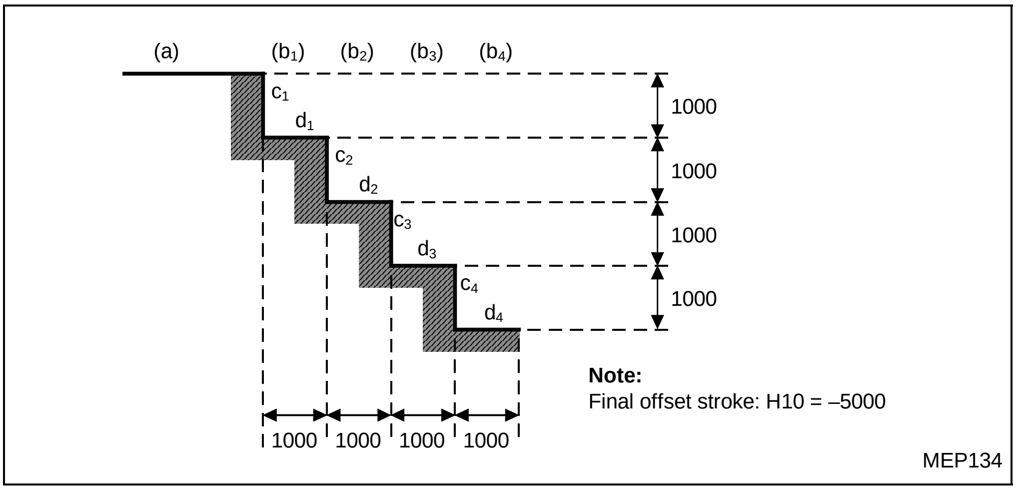
Example 1: Assumes that H10 has already been set equal to –1000.

```
N1  G01  G90  G43  Z–100000  H10     (Z = –101000)
N2  G28  Z0
N3  G91  G10  L10  P10  R–500        (–500 is added in the G91 mode.)
N4  G01  G90  G43  Z–100000  H10     (Z = –101500)
```

Example 2: Assumes that H10 has already been set equal to –1000.

Main program
```
N1  G00  X100000....................................... a
N2  #1=–1000
N3  M98  P1111L4...................................... b1, b2, b3, b4
```

Subprogram O1111
```
N1  G01  G91  G43  Z0  H10  F100........ c1, c2, c3, c4
N2  G01  X1000.......................................... d1, d2, d3, d4
N3  #1=#1–1000
N4  G90  G10  L10  P10  R#1
N5  M99
```



Example 3: The programs in Example 2 above can be rewritten as follows:

Main program
```
N1  G00  X100000
N2  M98  P1111  L4
```

Subprogram O1111
```
N1  G01  G91  G43  Z0  H10  F100
N2  G01  X1000
N3  G10  L10  P10  R-1000
N4  M99
```

Note: Even when the command code is displayed on <Next Command>, the current offset number and variables will remain unupdated until that command is executed.

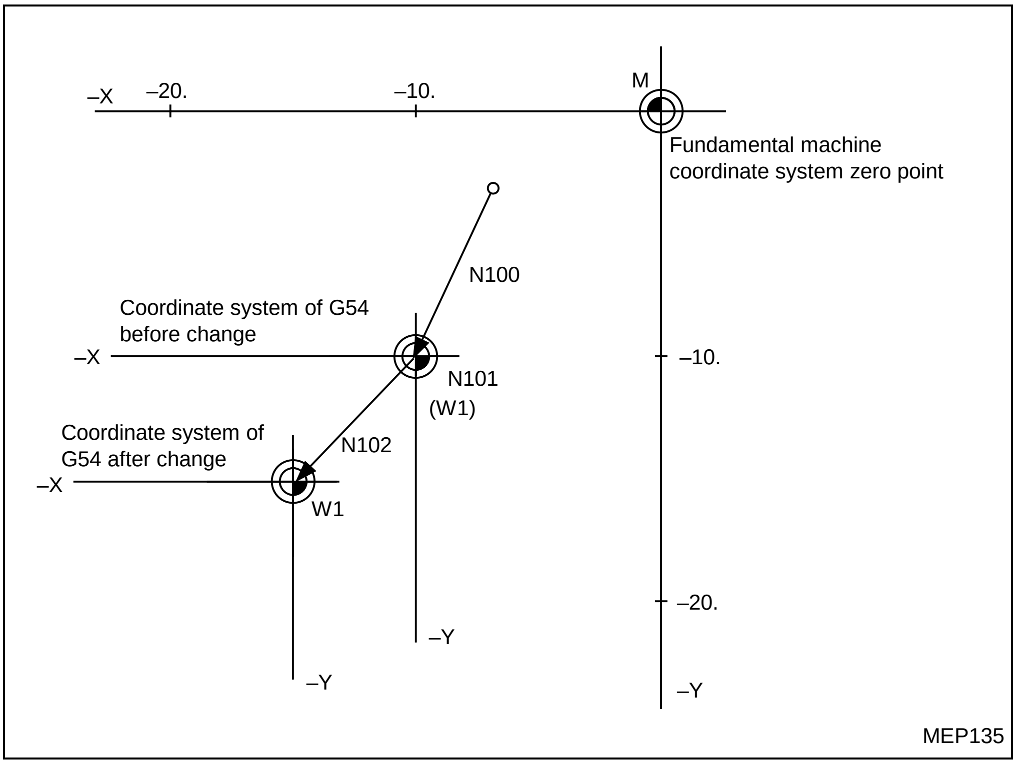
```
N1  G10   L10  P10  R-100
N2  G43   Z-10000   H10
N3  G0    X-10000   Y-10000
N4  G10   L10  P10  R-200
```
Executing block N4 will cause an offset stroke in H10 to be updated.

C. Updating the workpiece coordinate system offset data

Assume that the previous workpiece coordinate system offset data is as follows:

X = –10.000 Y = –10.000

```
  :
N100  G00  G90  G54  X0  Y0
N101  G10  L2  P1  X-15.000  Y-15.000
N102   X0  Y0
  :
M02
```



Note 1: Changes in the display of the workpiece position at N101

At N101, the display of tool position in the G54 coordinate system changes before and after workpiece coordinate system updating with G10.

X = 0 ⇨ X = +5.000

Y = 0   Y = +5.000

Note 2: Prepare the following program to set workpiece coordinate system offset data in G54 to G59:

```
G10L2P1X–10.000  Y–10.000
G10L2P2X–20.000  Y–20.000
G10L2P3X–30.000  Y–30.000
G10L2P4X–40.000  Y–40.000
G10L2P5X–50.000  Y–50.000
G10L2P6X–60.000  Y–60.000
```

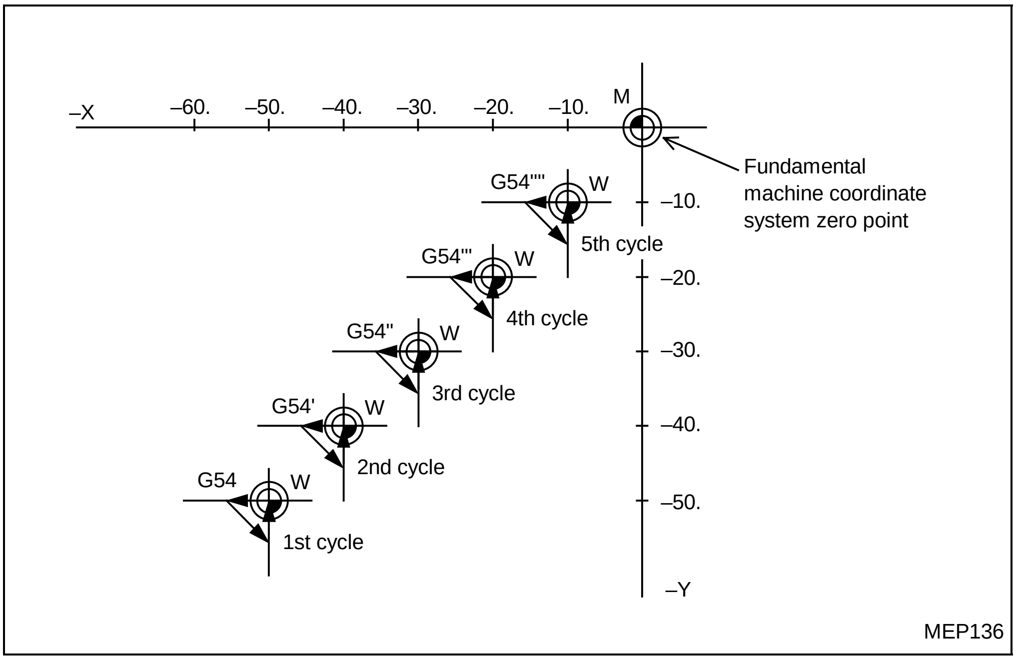
D. Programming for using one workpiece coordinate system as multiple workpiece coordinate systems

Main program

```
  :
#1=–50.   #2=10.
M98  P200  L5
  :
M02
%
```

Subprogram (O200)

```
N1 G90  G54  G10  L2  P1  X#1  Y#1
N2 G00  X0   Y0
N3 X–5. F100
N4 X0   Y–5.
N5 Y0
N6 #1=#1+#2
N7 M99
%
```



E. Programming for parameter data input

```
G10L50            Parameter input mode ON
N4017R10          D17 is set to "10".
N6088R96          F88 is set to "01100000". [ (01100000)2 = (96)10 ]
N12067R–1000      L67 is set to "–1000".
N12072R67         L72 is set to "0x43". [ (43)16 = (67)10 ]
N150004P1R50      P4 data for the 1st axis (X-axis) is set to "50".
G11               Parameter input mode OFF
```

5. Related alarms

Alarm No.	Alarm message	Cause	Remedy
807	ILLEGAL FORMAT	Work offset input: P-command is omitted in a block of G10 L20 (or L2) although the last selected coordinate system is one of the systems from G54 to G59 (or of the G54.1 systems). Parameter input: An illegal parameter number is set.	Review the program data.
809	ILLEGAL NUMBER INPUT	Work offset input: The setting range of the coordinate system number or the offset data is overstepped. Tool offset input: The setting range of the offset data is overstepped. Parameter input: The axis number is not specified for an axis type parameter. The setting range of the axis number or the parameter data is overstepped.	Review the program data.
839	ILLEGAL OFFSET No.	Tool offset input: The specified offset number is greater than the number of available data sets.	Correct the offset number according to the number of available data sets.
903	ILLEGAL G10 L NUMBER	Work offset input: A command of G10 L20 is set although the corresponding function for the G54.1 coordinate systems is not provided.	Give an available L-code command.

13-7 Tool Offsetting Based on MAZATROL Tool Data

Parameter selection allows you to offset both the tool length and the tool diameter using MAZATROL tool data (tool diameter and tool length data).

13-7-1 Selection parameters

Using the following parameters, select whether or not MAZATROL tool data is to be used:
User parameters

F92 bit 7 = 1: Tool diameter offsetting uses the MAZATROL tool data **ACT-ϕ** (tool diameter data).

F93 bit 3 = 1: Tool length offsetting uses the MAZATROL tool data **LENGTH** (tool length data).

F94 bit 2 = 1: Tool length offsetting using the MAZATROL tool data is prevented from being cancelled by a reference-point return command.

F94 bit 7 = 1: Tool offsetting uses the MAZATROL tool data **ACT-ϕ CO.** (or **No.**) and **LENG CO.** (or **No.**).
(Set **F94** bit 7 to 0 to use the data stored on the **TOOL OFFSET** display.)

1. **Tool length offsetting**

Data items used		Parameter F93 bit 3	Parameter F94 bit 7	Programming format	Remarks
TOOL OFFSET	Tool offset No.	0	0	G43/G44 H_ (P_)	
TOOL DATA (MAZATROL)	LENGTH [1]	1	1	T_	
	LENGTH [1] + OFFSET No. or LENGTH + LENG CO. [2]			T_ + H_	- Length offset cancellation not required for tool change. - G43 not required.
	OFFSET No. or LENG CO. [2]	0	1	G43/G44 H_	Length offset cancellation required for tool change. [3]
TOOL OFFSET + TOOL DATA	Tool offset No. + LENGTH [1]	1	0	(G43/G44 H_) + (T_) (P_)	Length offset cancellation required for tool change. [3]

[1] TOOL LENGTH data for milling tools, and LENGTH A and LENGTH B for turning tools.
[2] LENG CO. data are only used for milling tools.
[3] Canceling method - Set G49 before tool change command.
- Set G28/G30 before tool change command (when **F94** bit 2 = 0).

2. **Tool diameter offsetting**

Data items used		Parameter F92 bit 7	Parameter F94 bit 7	Programming format
TOOL OFFSET	Tool offset No.	0	0	G41/G42 D_
TOOL DATA (MAZATROL)	ACT-ϕ + ACT-ϕ CO. or ACT-ϕ + OFFSET No.	1	1	G41/G42 T_
	ACT-ϕ CO. or OFFSET No.	0	1	G41/G42 T_
TOOL OFFSET + TOOL DATA	Tool offset No. + ACT-ϕ	1	0	G41/G42 D_ + T_

3. Nose-R compensation

Data items used		Parameter F92 bit 7	Parameter F94 bit 7	Programming format
TOOL OFFSET	Tool offset No.	0	0	G41/G42 D_
TOOL DATA (MAZATROL)	NOSE-R + OFFSET No.	1	1	G41/G42 T_
	OFFSET No.	0	1	G41/G42 T_
TOOL OFFSET + TOOL DATA	Tool offset No. + NOSE-R	1	0	G41/G42 D_ + T_

13-7-2 Tool length offsetting

1. Function and purpose

Even when offset data is not programmed, tool length offsetting will be performed according to the MAZATROL tool data **LENGTH** that corresponds to the designated tool number.

2. Parameter setting

Set both bit 3 of parameter **F93** and bit 2 of parameter **F94** to 1.

3. Detailed description

1. Tool length offsetting is performed automatically, but its timing and method differ as follows:
   - After a tool change command has been issued, offsetting is performed according to the LENGTH data of the tool mounted in the spindle. (A tool change command code must be set in the program before tool length offsetting can be done.)
   - After command G43 has been set, offsetting is performed according to the **LENGTH** data of the tool mounted in the spindle.
2. Tool length offsetting is cancelled in the following cases:
   - When a command for tool change with some other tool is executed
   - When M02 or M30 is executed
   - When the reset key is pressed
   - When command G49 is issued
   - When a reference-point return command is executed with bit 2 of parameter **F94** set to 0
3. The table below shows how and when the tool length offsetting actually takes place.

F94 bit 7	How and when the tool length offsetting actually takes place
0	For milling tools: Length offsetting in the first movement on the Z-axis. For turning tools: Simultaneous offsetting by LENGTH A and B in the first axis movement, be it on the X-, Y-, Z-, or B-axis.
1	For milling tools: Length offsetting in the first movement on the Z-axis. For turning tools: Offsetting by LENGTH A in the first movement on the Z-axis, and by LENGTH B in the first movement on the X-axis.

4. If this offset function is used with a G43 H-command, offsetting will use as its offset data the sum total of the MAZATROL tool data **LENGTH** and the offset amount specified by the G43 H (or G44 H) command.

Note 1: Set G43 H0 if tool length offsetting is to be done using a G43 H-command and only the offset amount specified by H is to be cancelled.

Return to Library

Note 2: With a G44 command, tool length offsetting based on MAZATROL tool data is not performed.

Note 3: The restart operation must begin from a position before a G43 command code or a tool change command code. Even when the spindle has a mounted tool, G43 or the tool change command must be executed before offsetting based on MAZATROL tool data can take place.

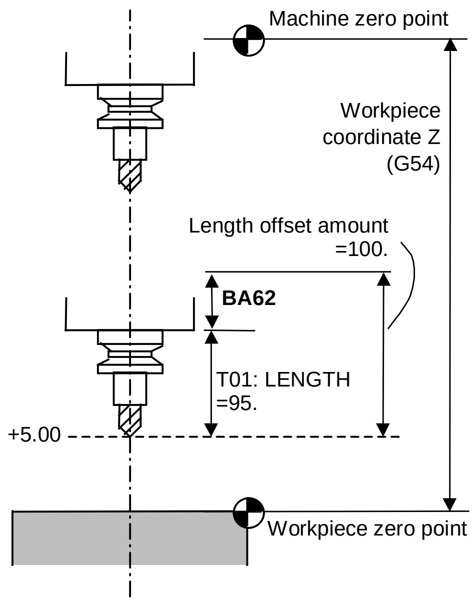
Note 4: Offsetting will fail if registered MAZATROL tool data LENGTH is not present.

Note 5: For an EIA/ISO program, to carry out tool length offset operations using the tool length data included in MAZATROL tool data, it becomes necessary to set data in the validation parameter for the tool length data of the MAZATROL tool data and to insert a tool change T- and M-code command block. It is to be noted that the tool change command block may not be missed particularly in the following cases:

- During automatic operation, if the first tool to be used has already been mounted in the spindle.
- During call of an EIA/ISO program as a subprogram from the MAZATROL main program, if the tool to be used immediately prior to call of the subprogram is the same as that which is to be designated in that subprogram as the first tool to be used.

4. Sample programs

For milling tools



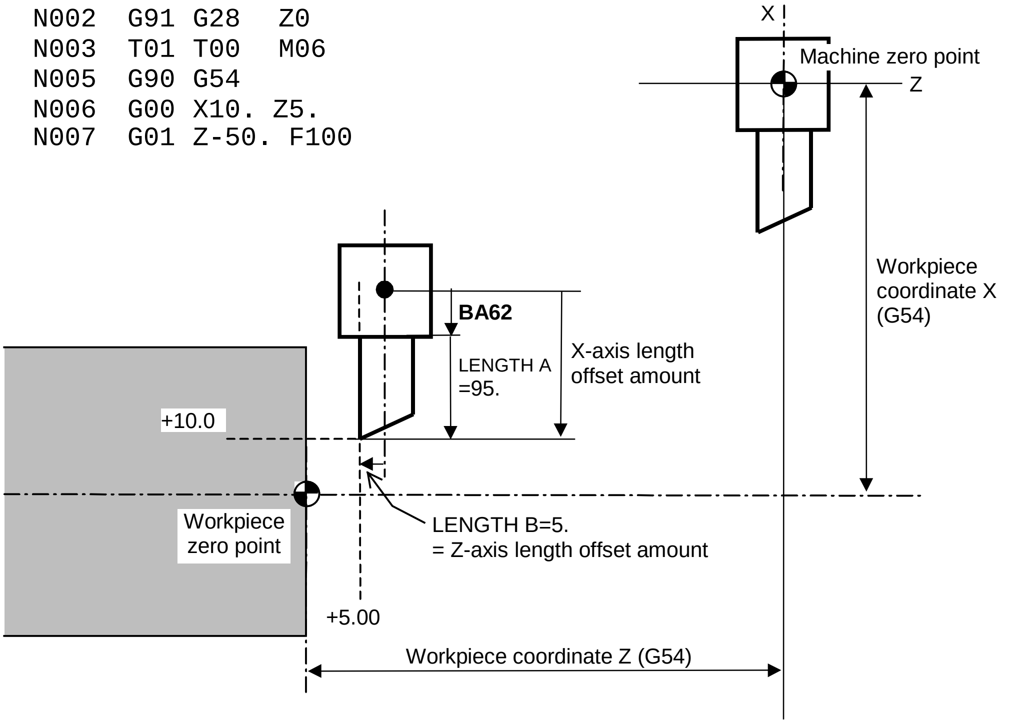
Offsetting values:
(LENGTH = 95.)

```
N001 G90  G94  G00  G40  G80
N002 G91  G28  Z0
N003 T01  T00  M06
N004 G90  G54  X-100. Y0
N005 G0   Z5.
N006 G01  Z-50. F100
```

For turning tools

LENGTH A = 95.
LENGTH B = –5.

```
N001  G90 G94  G00   G40  G80
N002  G91 G28  Z0
N003  T01 T00  M06
N005  G90 G54
N006  G00 X10. Z5.
N007  G01 Z-50. F100
```



13-7-3 Tool diameter offsetting

1. Function and purpose

Tool diameter offsetting by a G41 or G42 command uses MAZATROL tool data **ACT-ϕ** as the offset amounts.

2. Parameter setting

Set bit 7 of parameter **F92** to 1.

3. Detailed description

- Tool diameter offsetting uses as its offset amounts the diameter data of the tool which is mounted in the spindle at the issuance of G41/G42.
- Tool diameter offsetting is cancelled by G40.
- If the tool diameter offset function is used with a D-command, the sum total of the data indicated by the corresponding offset number (D) and the radius of the tool will be used as the offset data.

Note 1: The tool used must be mounted in the spindle before restarting the program.

Note 2: Offsetting based on tool diameter data will not occur if registered MAZATROL tool diameter data is not present or if a tool for which tool diameter data cannot be entered is to be used.

Note 3: To carry out for an EIA/ISO program the tool diameter offset operations using the tool diameter data included in MAZATROL tool data, it is necessary to insert tool change command blocks, as it is the case for tool length offsetting (refer to **Note 5** in Subsection 13-7-2).

13-7-4 Tool data update (during automatic operation)

1. Function and purpose

Tool Data Update allows MAZATROL tool data to be updated during automatic operation based on an EIA/ISO program.

2. Parameter setting

Set parameter **L57** to 1.

3. Detailed description

This function allows the entire tool data, except for spindle tools, to be updated during automatic operation based on an EIA/ISO program.

Parameter	TOOL	NOM-ϕ	ACT-ϕ	LENGTH	COMP.	THR/HP	LIFE	TIME	MAT.	REV.
L57 = 0	No	No	No	No	No	No	Yes	Yes	No	Yes
L57 = 1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Note 1: In the table given above, “Yes” indicates that you can update the data, and “No” indicates that you cannot update the data.
Identification between MAZATROL programs and EIA/ISO programs is automatically made by whether the program currently being executed, is MAZATROL or EIA/ISO, irrespective of whether it is a main program or subprogram.
If, however, the main program is MAZATROL and its subprograms are EIA/ISO, then the currently active set of programs is regarded as a MAZATROL program.

Note 2: An alarm **428 MEMORY PROTECT (AUTO OPERATION)** will occur if the spindle tool data is modified during automatic operation based on an EIA/ISO program.

- NOTE -

14 PROGRAM SUPPORT FUNCTIONS

14-1 Fixed Cycles for Turning

When performing roughing and other such operations during turning, these functions permit to command in a single block the machining program which is normaly commanded in several blocks. In other words, they simplify the machining program. The following types of fixed cycles for turning are available.

G-code	Function
G90	Longitudinal turning cycle
G92	Threading cycle
G94	Transverse turning cycle

1. The programming format is as follows:

   G90 X/U_ Z/W_ R_ F_ ;
   (Same for G92, G94)

   The taper values of fixed cycles G90, G92 and G94 are to be specified by argument R.

2. Fixed cycle commands are modal G-codes and so they are valid until another command in the same modal group or a cancel command is set.
   The following G-code cancels fixed cycle commands.

   G00, G01, G02, G03
   G07,
   G09,
   G10,
   G27, G28, G29, G30, G30.1
   G31,
   G32, G34
   G37,
   G50,
   G52, G53

3. There are two types of fixed cycle call, move command block call and block-by-block call. These are selected by a parameter setting.
   A move command block call calls the fixed cycle macro subprogram only when there is an axial move command in the fixed cycle mode. The block-by-block call calls the fixed cycle macro subprogram in each block in the fixed cycle mode. Both types are executed until the fixed cycle is cancelled.

4. A manual interruption can be applied while a fixed cycle for turning (G90, G92 and G94) is being executed. Upon completion of the interruption, however, the tool must be returned to the position where the manual interruption was applied and then the fixed cycle for turning should be restarted. If it is restarted without the tool having been returned, all subsequent operation movements will deviate by an amount equivalent to the manual interruption value.

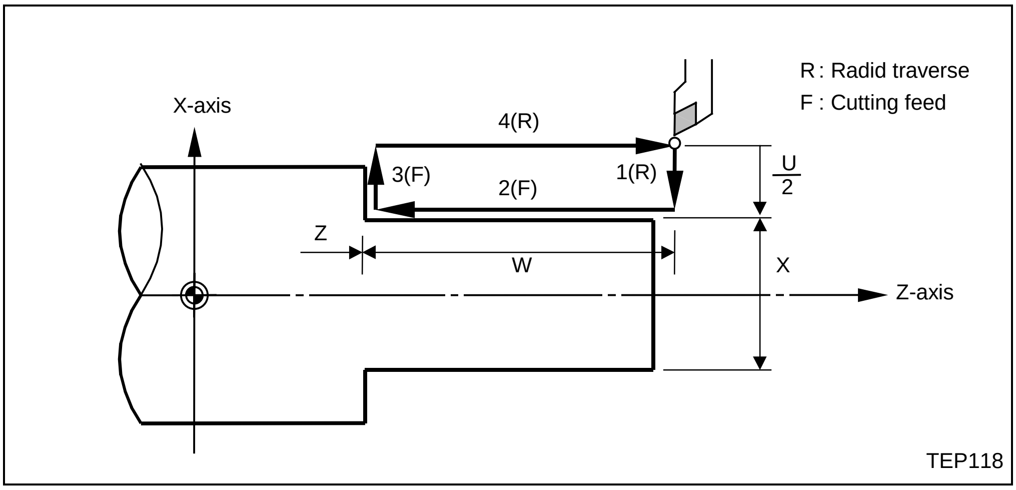
Return to Library

14-1-1 Longitudinal turning cycle: G90 [Series M: G290]

1. Straight turning

This function enables continuous straight turning in the longitudinal direction using the following command.

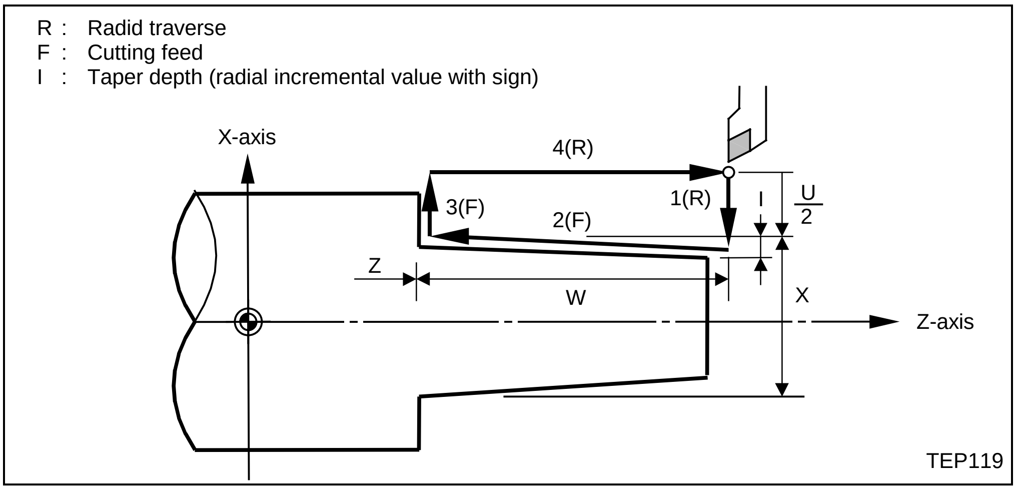
G90 X/U_Z/W_F_;



2. Taper turning

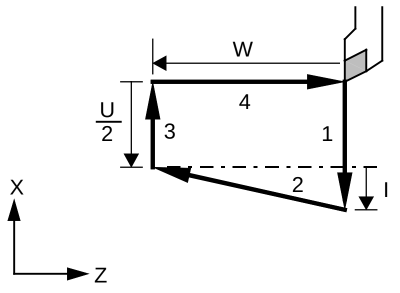
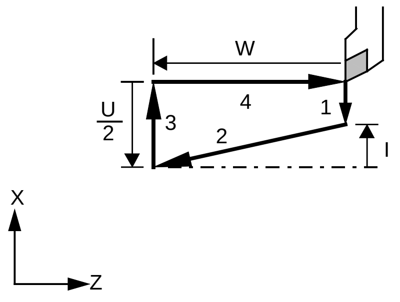
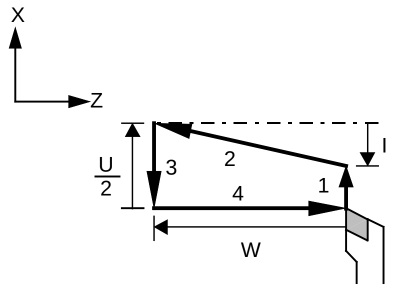
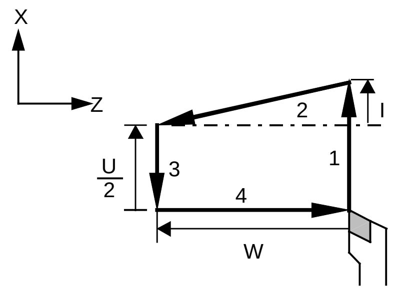
This function enables continuous taper turning in the longitudinal direction using the following command.

G90 X/U_Z/W_I_F_;



3. Remarks

In single-block operation mode, the tool stops either at the ending points of operations 1, 2, 3 and 4, or only on completion of one cycle (depending on bit 7 of parameter **F111**)
Depending on the U, W and I signs, the following shapes are created.

[1] U < 0, W < 0, I < 0	[2] U < 0, W < 0, I > 0
	
[3] U > 0, W < 0, I < 0	**[4] U > 0, W < 0, I > 0**
	 TEP120

Program error **899 ILLEGAL TAPER LENGTH** occurs in shapes [2] and [3] unless the following condition is satisfied.

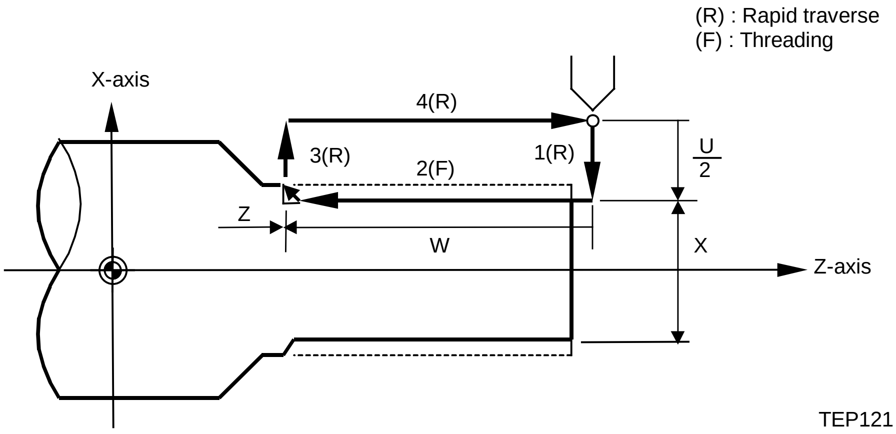
$$\left|\frac{U}{2}\right| \geq |\ I\ |$$

14-1-2 Threading cycle: G92 [Series M: G292]

1. Straight threading

This function enables straight threading using the following command.

G92 X/U_ Z/W_ F/E_ ;

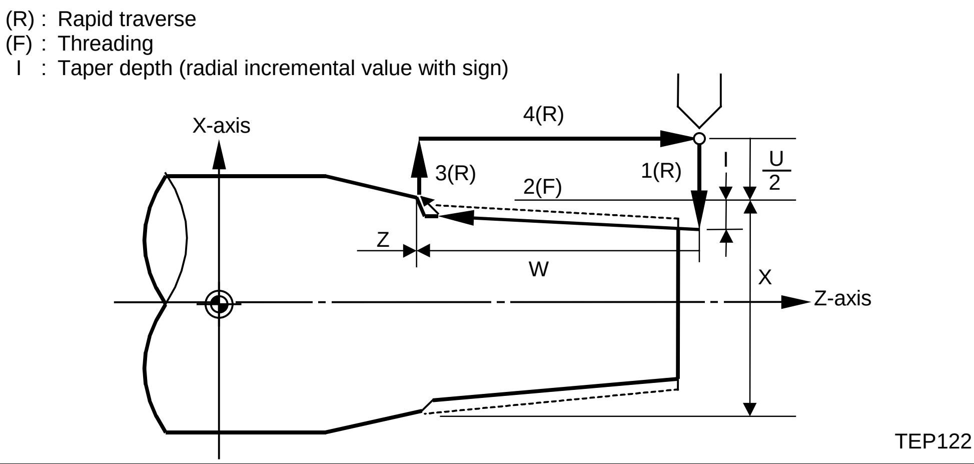


TEP121

2. Taper threading

This function enables taper threading using the following command.

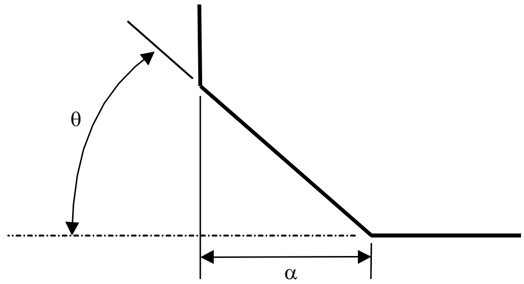
G92 X/U_ Z/W_ I_ F/E_ ;



TEP122

3. Remarks

- Details for thread run-out

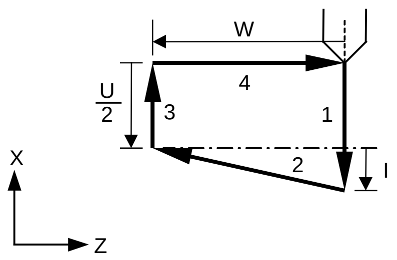
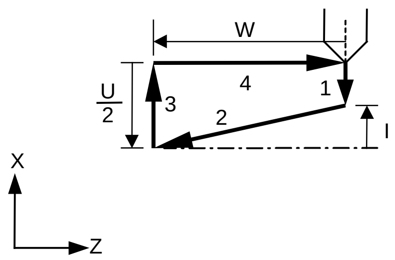
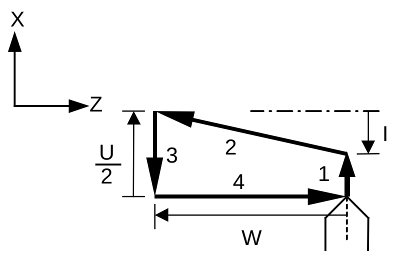
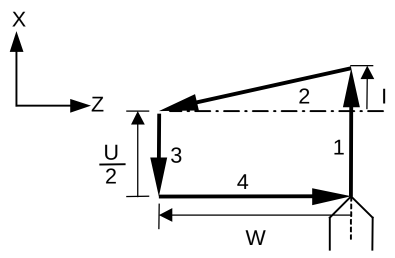


α : Length of thread run-out
If the thread lead is assumed to be L, then the parameter can be set by 0.1 L in the range 0 to 4.0.

θ : Run-out angle of threding
The parameter (**F28**) can be set in 45° or 60°.

TEP123

- In single-block operation mode, the tool stops either at the ending points of operations 1, 3 and 4, or only on completion of one cycle (depending on bit 7 of parameter **F111**).
- When the feed hold function is applied during a threading cycle, automatic operation will stop at that position if not in threading. By setting of parameter **F111** bit 2, threading under way can be stopped either at the next movement completion position (completion of operation 3) of the threading or after chamfering from the position where the feed hold function is applied.
- During threading, use or disuse of dry run will not be changed.
- Depending on the U, W and I signs, the following shapes are created.

[1] U < 0, W < 0, I < 0	[2] U < 0, W < 0, I > 0
	
[3] U > 0, W < 0, I < 0	**[4] U > 0, W < 0, I > 0**
	

TEP124

Program error **899 ILLEGAL TAPER LENGTH** occurs in shapes [2] and [3] unless the following condition is satisfied.

$$\left|\frac{U}{2}\right| \geq |\ I\ |$$

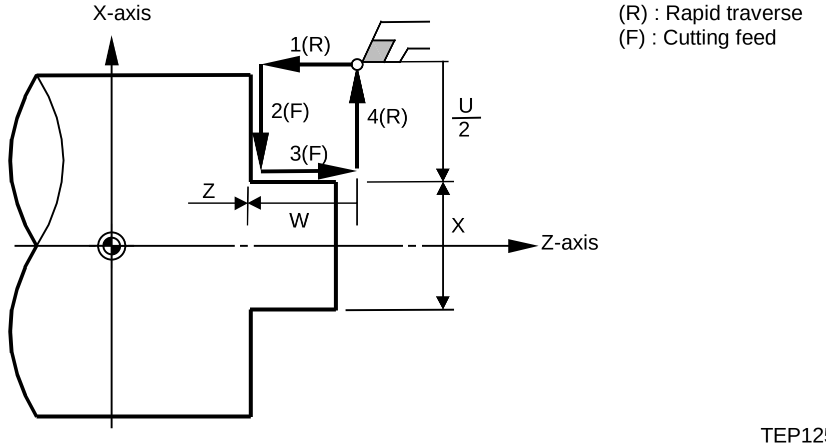
- For machines with the optional function for automatic correction of threading start position, the thread cutting conditions can be changed by “overriding” the spindle speed.
See Subsection 6-13-6 for more information.

14-1-3 Transverse turning cycle: G94 [Series M: G294]

1. Straight turning

This function enables continuous straight turning in the face direction using the following command.

G94 X/U_Z/W_F_;

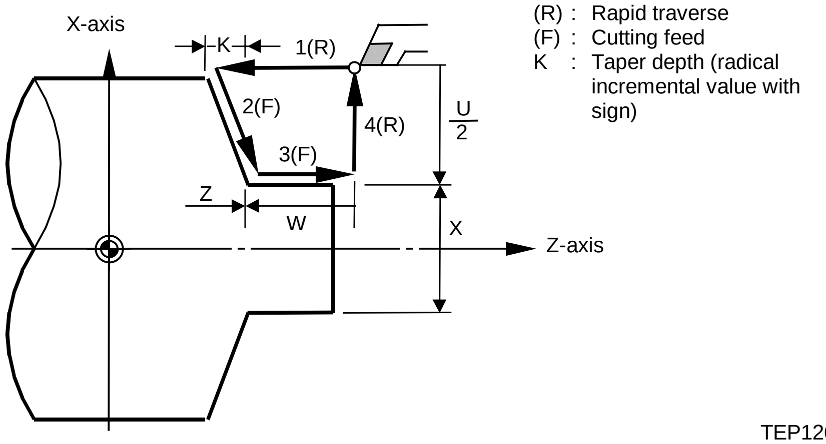


TEP125

2. Taper turning

This function enables continuous taper turning in the face direction using the following command.

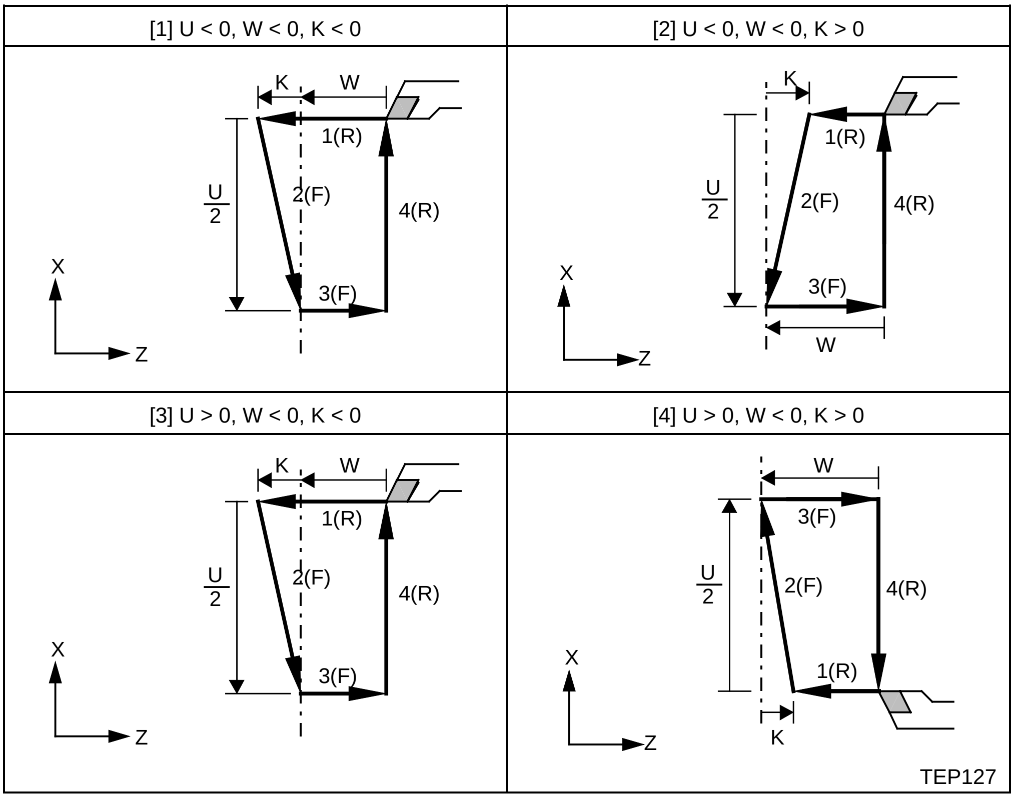
G94 X/U_Z/W_K_F_;



TEP126

3. Remarks

- In single-block operation mode, the tool stops either at the ending points of operations 1, 2, 3 and 4, or only on completion of one cycle (depending on bit 7 of parameter **F111**).
- Depending on the U, W and K signs, the following shapes are created.



Program error **899 ILLEGAL TAPER LENGTH** occurs in shapes [2] and [3] unless the following condition is satisfied.

$| W | \geq | K |$

14-2 Compound Fixed Cycles

These functions permit to execute the fixed cycle by designating a program in a block with corresponding G-code.
The types of compound fixed cycles are listed below.

G-code	Function	
G70	Finishing cycle	Compound fixed cycles I
G71	Longitudinal roughing cycle (roughing along finish shape)	
G72	Transverse roughing cycle (roughing along finish shape)	
G73	Contour-parallel roughing cycle	
G74	Longitudinal cut-off cycle	Compound fixed cycles II
G75	Transverse cut-off cycle	
G76	Compound threading cycle	

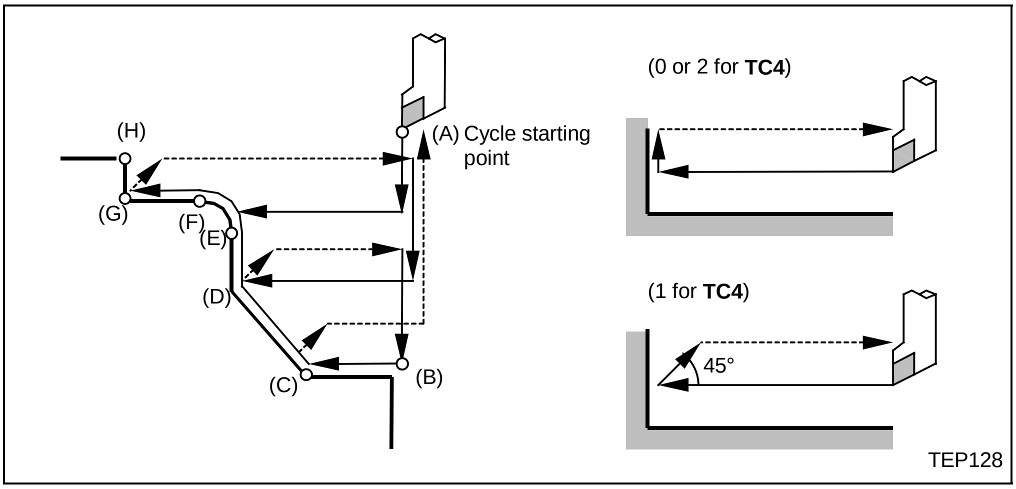
- If the finish shape program has not been entered in the memory, any of the above functions for the compound fixed cycles I (G70 to G73) cannot be used.
- The programming formats are as follows.

G-code	Programming format
G70	G70 A_P_Q_ ;
G71	G71 U_R_ ; G71 A P_Q_U_W_F_S_T_ ;
G72	G72 W_R_ ; G72 A_P_Q_U_W_F_S_T_ ;
G73	G73 U_W_R_; G73 P_Q_U_W_F_S_T_ ;
G74	G74 R_ ; G74 X(U)_Z(W)_P_Q_R_F_S_T_ ;
G75	G75 R_ ; G75 X(U)_Z(W)_P_Q_R_F_S_T_ ;
G76	G76 P_Q_R_ ; G76 X(U)_Z(W)_R_P_Q_F_ ;

14-2-1 Longitudinal roughing cycle : G71 [Series M: G271]

1. Overview

With commands as shown below for finish shape between (A) to (H), roughing by cutting depth D will be executed by leaving finishing allowances U and W.



The parameter **TC4** will determine escape pattern from wall at right angle, whether 45° escape or feedrate accelerated at wall should be made during roughing cycle. By setting 2 for **TC4**, chamfering speed can be changed. (Refer to parameter **TC3**.)

2. Programming format

G71 UΔd R_;
G71 A_ P_ Q_ UΔu W_ F_ S_ T_;

UΔd : Cutting depth
It is commanded without sign (radius value). This command is modal and valid until a new value is commanded.

R : Escape distance
This command is modal and valid until a new value is commanded.
Escape angle is fixed to 45°.

A : Finish shape program No.

P : Head sequence No. for finishing shape

Q : End sequence No. for finishing shape

UΔu : Finishing allowance and direction in X-axis direction (diametral value or radius value)

W : Finishing allowance and direction in Z-axis direction

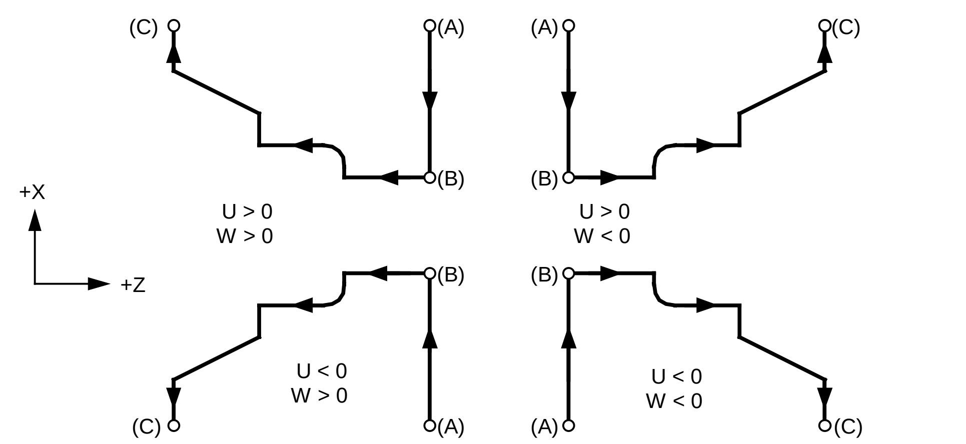
F_ S_ T_: F, S and T command
F, S and T specified in blocks of “P” to “Q” are ignored during cycle, and those specified in or before G71 block become valid.

- Δd and Δu are both specified by address U. The differentiation depends on whether P and Q are specified in the same block.

Note: Even if F and S commands exist in blocks defined by P and Q, they will be ignored during roughing cycle because they are considered for finishing cycle.

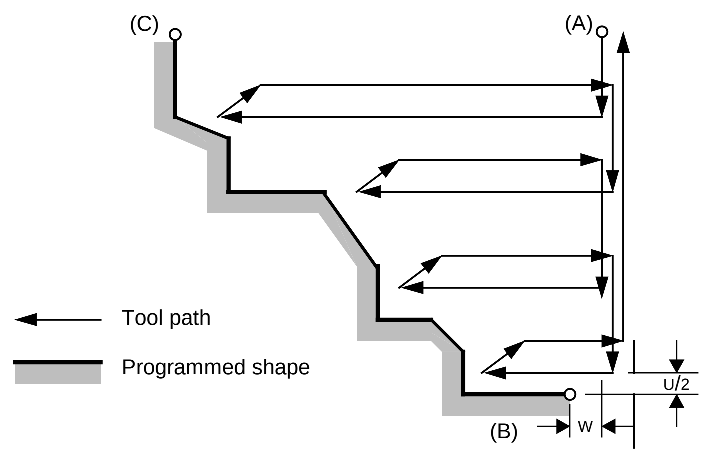
3. Detailed description

Machining shape executed by G71 may be one of the four combinations below.
Basically machining will be executed by Z-axis displacement. Finishing allowances U and W may have different signs.



TEP129

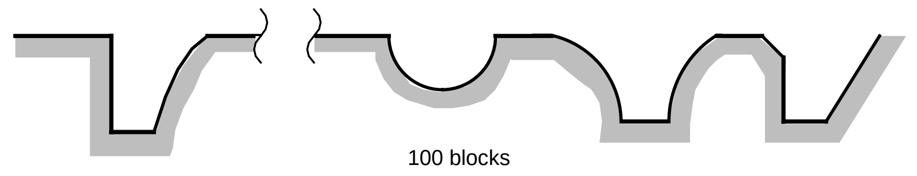
For section between A and B, command should be executed by sequence block of P data, with move command in X axis only.
For section between B and C, a maximum of 32 recesses are allowed.
When G00 is commanded for section between A and B, cutting during the cycle will be made by rapid feed. With G01, cutting feed rate will be used.



TEP130

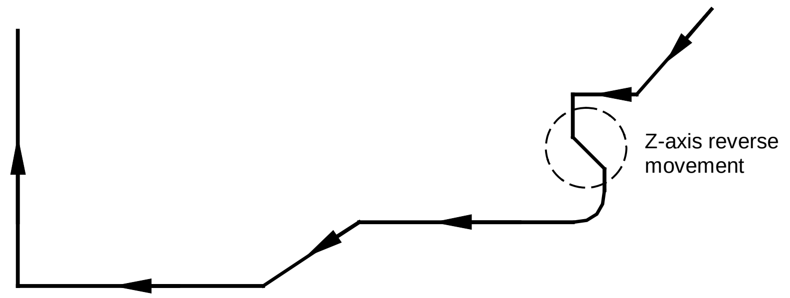
4. Remarks

1. Subprograms can be called from sequence blocks difined by P and Q.
2. Machining may be ended with G02 or G03 in some shapes defined by P and Q. After cycle completion, G-code must be commanded again.
3. The number of blocks for finishing shape is up to 100 blocks including those automatically inserted within NC equipment such as nose radius compensation for example.



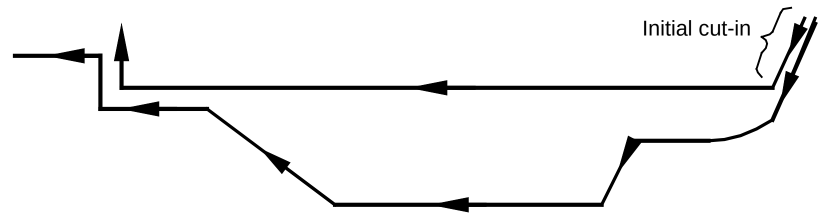
TEP131

Coordinates, however, must be increased monotonously in Z-axis direction. For shapes as shown below, alarm **898 LAP CYCLE ILLEGAL SHAPE DESIGN.** will occur and machining stops.



TEP132

4. The initial cut-in may not be executed in X-axis displacement only. As far as the movement in the coordinates increasing little by little in the Z-axis direction, any shape can be machined.



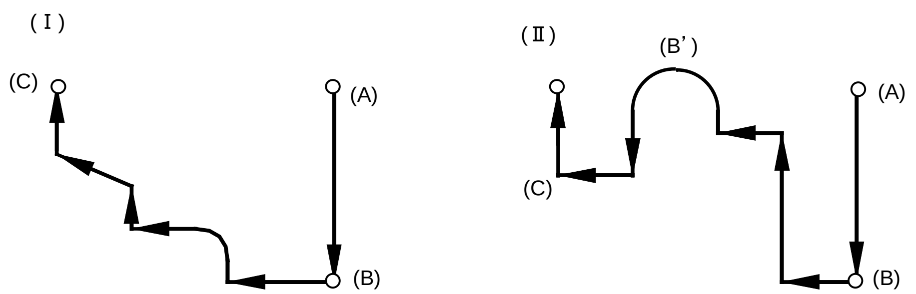
TEP133

5. If this cycle is commanded during nose radius compensation mode, nose radius compensation will be effected on the finishing shape program, and this cycle will be accordingly executed for the compensated shape.

6. For the shape in which X-axis coordinates are not changed little by little, W should be commanded with 0 generally. If this is ignored, excessive cutting will be caused on one wall side.

7. In the first block of repeating section, give a command for each of the pair X(U) and Z(W). If no Z-axis movement is made, W0 must be commanded.

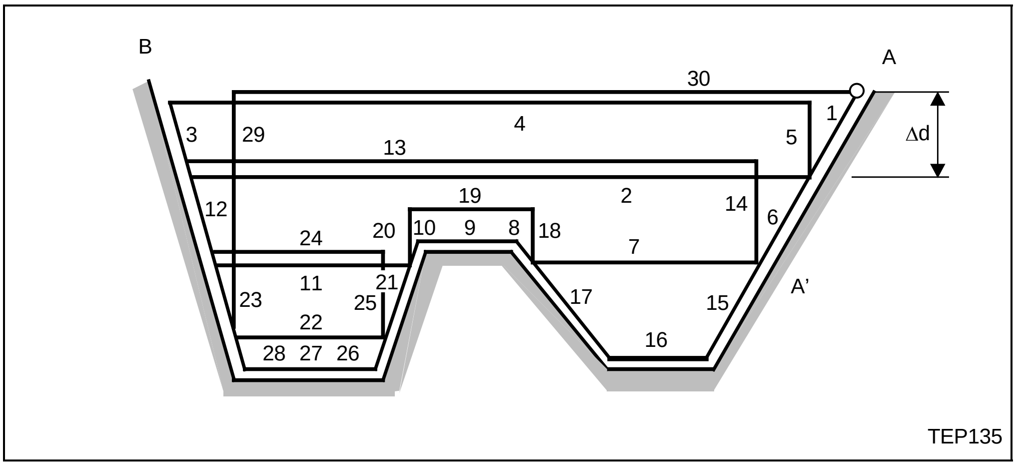
8. In machining shape defined for A-B-C, if the command which results in A ≤ B is made as with B', alarm **898 LAP CYCLE ILLEGAL SHAPE DESIGN.** will be caused. Case (I) is correct while case (II) will cause alarm.



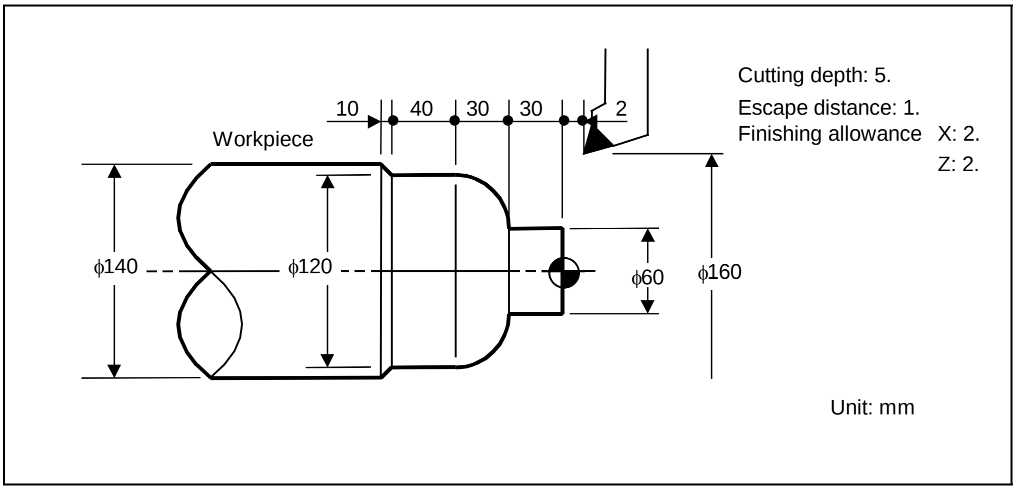
TEP134

9. M-codes in the sequences defined by P and Q will not be effective in G71 cycles and same for G72 and G73 cycles. Such M-codes are effective only in G70 cycles.

10. Cutting path is as shown below.



5. Sample programs




```
N001 G00 G96 G98;
N002 G28 U0 W0;
N003 X160.Z2.;
N010 G71 U5.R1.;
N011 G71 P012 Q016 U4.W2. F150 S150 M03;
N012 G00 X60.S200;
N013 G01 Z-30.F100;
N014 G03 X120.Z-60.R30.;
N015 G01 W-40.;
N016 X140.W-10.;
N017 G70 P012 Q016;
N018 G28 U0 W0 M05;
N019 M30;
```

Return to Library

14-2-2 Transverse roughing cycle: G72 [Series M: G272]

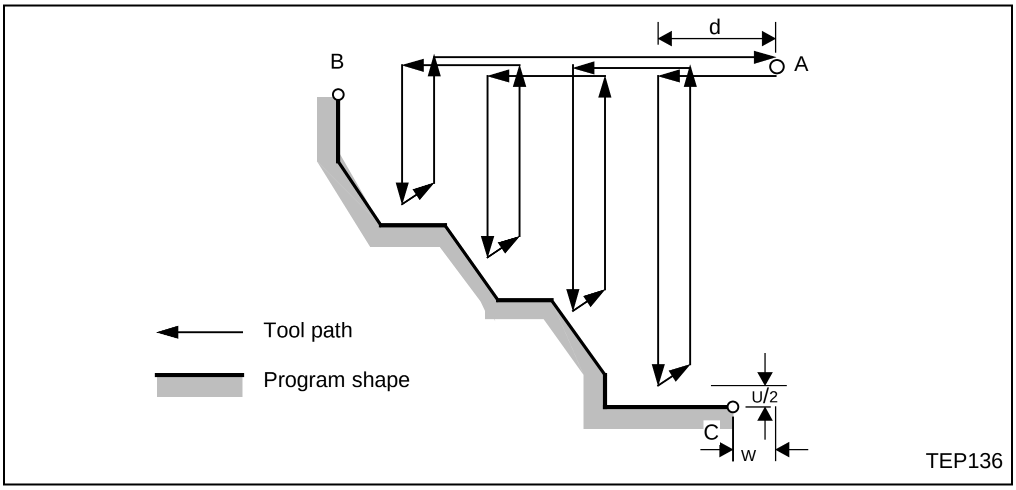
1. Programming format

G72 W$\underline{\Delta d}$ R_;

G72 A_ P_ Q_ U_ W_ F_ S_ T_;

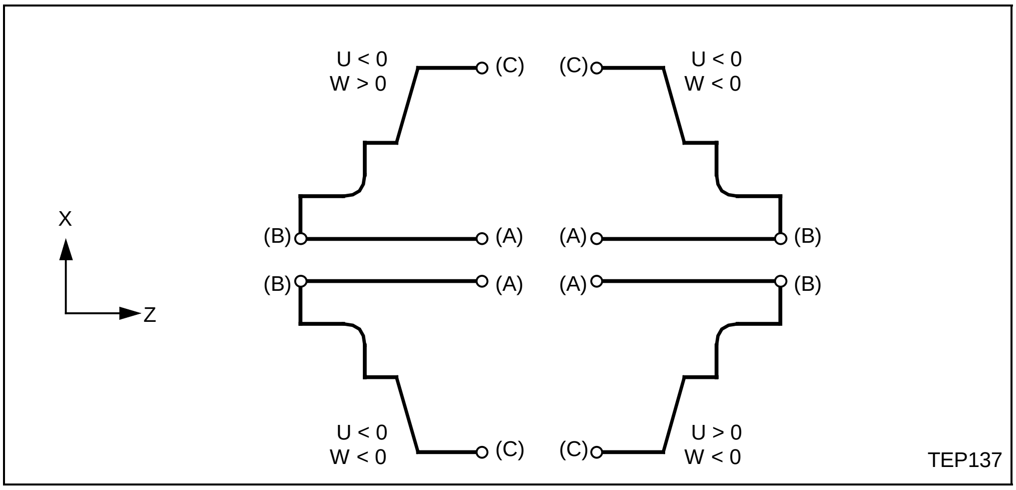
W$\underline{\Delta d}$: Cutting depth

* Other addresses are as with G71.



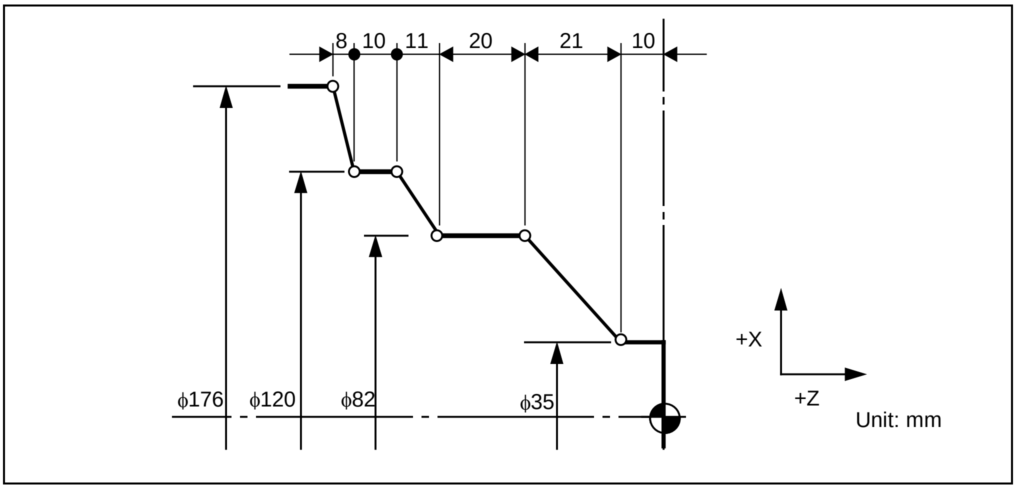
In G72, parameter **TC4** will determine whether escape at 45° from wall should be selected or not in the cycle as in G71. By setting 2 for **TC4**, feed can be accelerated in up-going slope. (Refer to parameter **TC3**.)

Machining shape in G72 may be one of the four combinations below. Basically machining will be executed by X-axis displacement. Finishing allowance U and W may have different signs.



- For section between (A) and (B), command should be made by sequence block of P data, with move command in Z-axis only.
- For section (B) and (C), the number of recesses for finish shape is up to 100 blocks including those automatically inserted within NC unit.
- When G00 is commanded for section between (A) and (B), cutting during the cycle will be made by rapid feed. With G01, cutting feed rate will be used.
- Nose radius compensation amount will be added to the finishing allowances U and W.

2. Sample programs

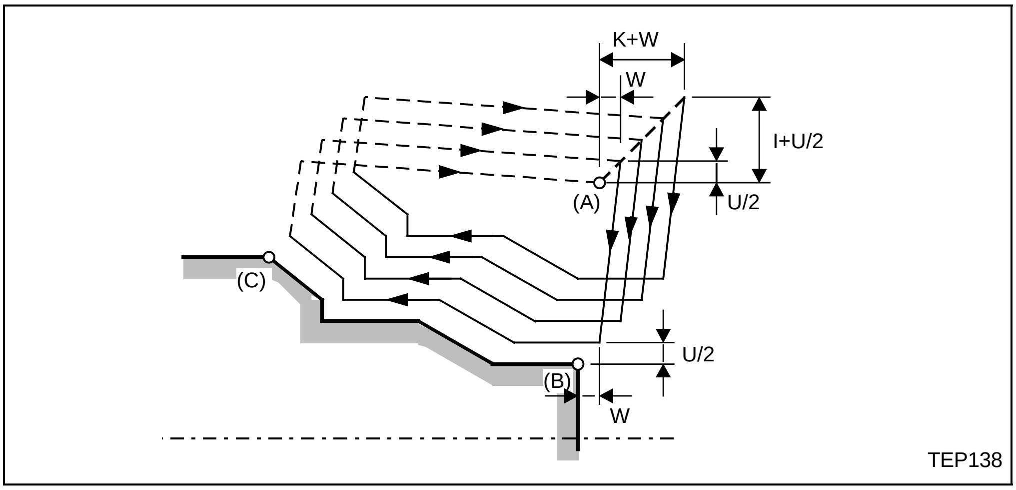



```
N001  G00 G96 G98;
N002  G28 U0 W0;
N003  T001T000M6;
N004  X176.Z2.;
N010  G72 W7.R1.;
N011  G72 P012 Q018 U4.W2.D7. F100 S100 M3;
N012  G00 Z-80.S150;
N013  G01 X120.W8.F100;
N014  W10.;
N015  X82.W11.;
N016  W20.;
N017  X35.W21;
N018  W12.;
N019  G70 P012 Q018;
N020  G28 U0 W0 M5;
N021  M30;
```

14-2-3 Contour-parallel roughing cycle: G73 [Series M: G273]

1. Overview

This function will allow efficient execution in roughing when cast or forged parts are to be cut along finish shape.



2. Programming format

G73 UΔi WΔk Rd;
G73 P_ Q_ UΔu WΔw F_ S_ T_;

Δi : Escape distance and direction in the X-axis direction (radial value)
This command is modal and valid until a new value is commanded.
Δk : Escape distance and direction in the Z-axis direction
This command is modal and valid until a new value is commanded.
d : Times of divisions
It is equal to the times of roughing. This command is modal and valid until a new value is commanded.

* Other addresses are as with G71.

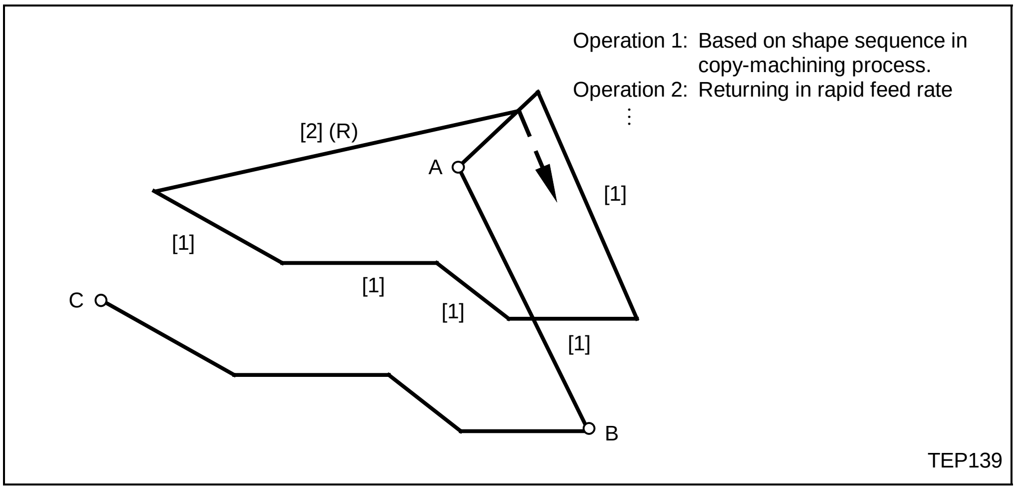
Note: Even if F and S commands exist in blocks defined by P and Q, they will be ignored during roughing cycle because they are considered for finishing cycle.

3. Detailed description

- Finish shape
In the program, (A)→(B)→(C) in the following illustration are commanded.
The section between B and C must be a shape in which coordinates changes little by little in both the X-axis and Z-axis directions.

- One cycle configuration

  A cycle is composed as shown below.



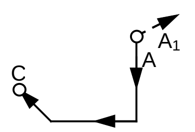
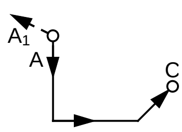
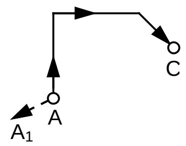
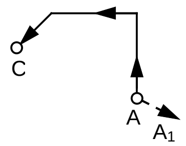
- Tool nose radius compensation

  When this cycle is commanded in the tool nose radius compensation mode, tool nose radius compensation is applied to the finishing shape sequence for this cycle and the cycle is executed for this shape.

  However, when this cycle is commanded in the tool nose radius compensation mode, the compensation is temporarily cancelled immediately before this cycle and started at the head block of the finishing shape sequence.

- Infeed direction

  The shift direction for the infeed is determined by the shape in the finishing program, as shown in the table below.

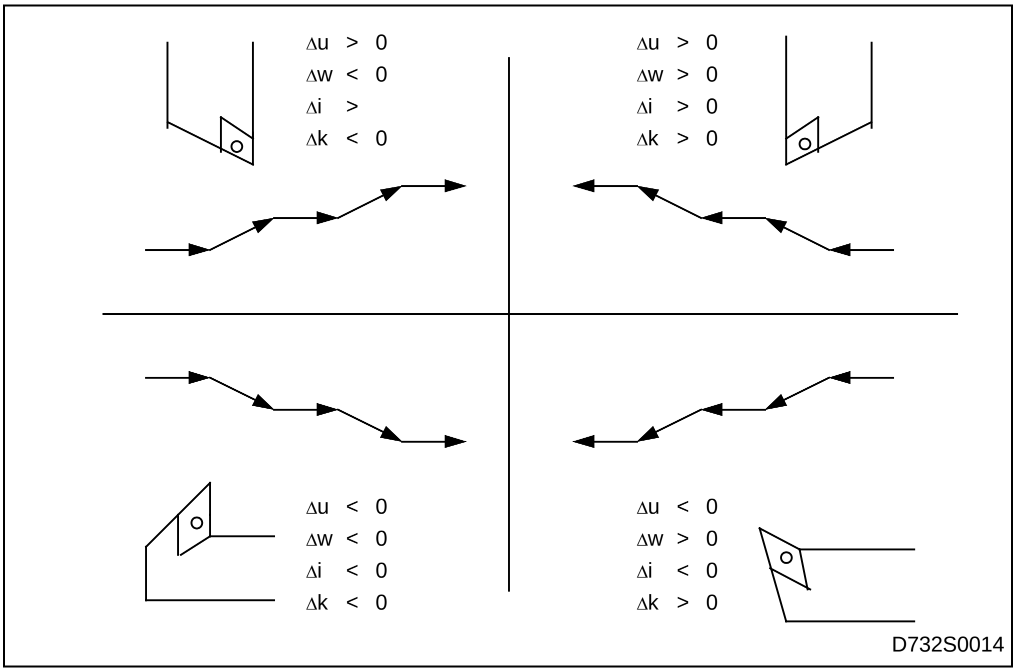
	1	2	3	4
Trace				 TEP140
Initial X-axis	"–" direction	–	+	+
Overall Z-axis	"–" direction	+	+	–
X-axis cutting	"+" direction	+	–	–
Z-axis cutting	"+" direction	–	–	+

4. Parameter

- Escape distance and direction in the X-axis direction can be set by parameter **TC117**. Parameter setting values will be overridden with the program command.
- Escape distance and direction in the Z-axis direction can be set by parameter **TC118**. Parameter setting values vary according to the program command.
- Times of divisions can be set by parameter **TC72**. Parameter setting values vary according to the program command.

5. Remarks

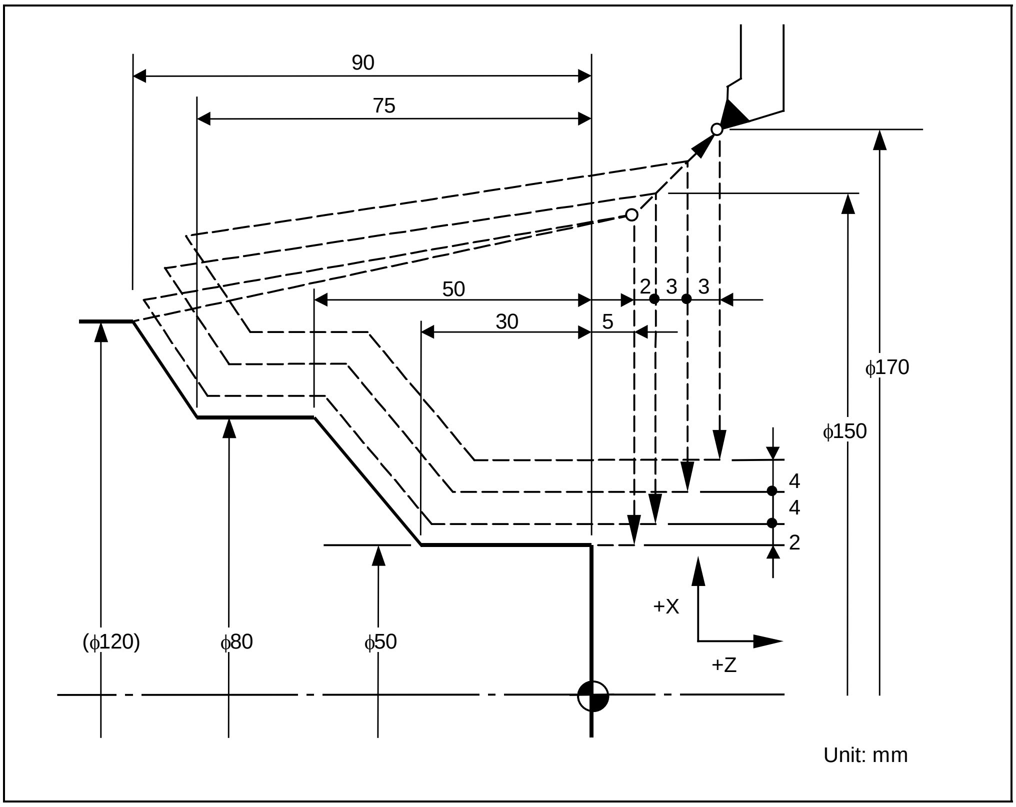
- F, M, S and T command specified in blocks between P and Q are ignored, and those specified in G73 block are valid.
- There are four patterns of machining, therefore, take care to signs of Δu, Δw, Δi and Δk, respectively.



Δi, Δk and Δu, Δw are both specified by addresses U and W. The differentiation is given by whether P and Q are commanded in the same block. That is, addresses U and W when P and Q are not commanded in G73 block represent Δi and Δk respectively, and those when P and Q are commanded represent Δu and Δw respectively.

- When the cycle terminates, the tool is returned to point A.
- In machining where the center of tool nose is aligned with the starting point, if cutting is performed with the tool nose radius compensation applied, the amount of tool nose radius compensation is added to Δu and Δw.
- Others are as with G71.

6. Sample programs




```
N010 G00 G96 G98;
N011 G28 U0 W0;
N012 T001T000M6;
N013 X150.Z5.;
N014 G73 U8.W6.R3.;
N015 G73 P016 Q020 U4.W2. F150 S100 M3;
N016 G00 X50.;
N017 G01 Z-30;
N018 X80.Z-50.;
N019 Z-75.;
N020 X120.Z-90.;
N021 G70 P016 Q020;
N022 G28 U0 W0 M5;
N023 M30.;
```

Return to Library

14-2-4 Finishing cycle: G70 [Series M: G270]

After roughing have been carried out by the G71 to G73 commands, finishing can be performed by the following programming format.

G70 A_ P_ Q_ ;

A : Finish shape program number (program being executed when omitted)
P : Finish shape start sequence number (program head when omitted)
Q : Finish shape end sequence number (end of program when omitted)
Up to M99 command when M99 comes first even if Q command is present

- The F, S and T commands in the finishing shape program are valid during the finishing cycle.
- When the G70 cycle is completed, the tool returns to the starting point by rapid feed and the next block is read.

Example 1: When designating a sequence number

```
 ⋮
N100 G70 P200 Q300;        N200···;
N110                         ⋮
N120                       N300···;
 ⋮
N200⎫
 ⋮  ⎬ Finishing shape program
N300⎭
N310
 ⋮
```

Example 2: When designing a program number

```
 ⋮
N100 G70 A100;
N110···;
N120···;              O100
 ⋮
                 G01 X100 Z50 F0.5;
                          ⋮
                         M99;
```

After execution of the N100 cycle in either Example 1 or Example 2, the N110 block is executed next.

14-2-5 Longitudinal cut-off cycle: G74 [Series M: G274]

1. Overview

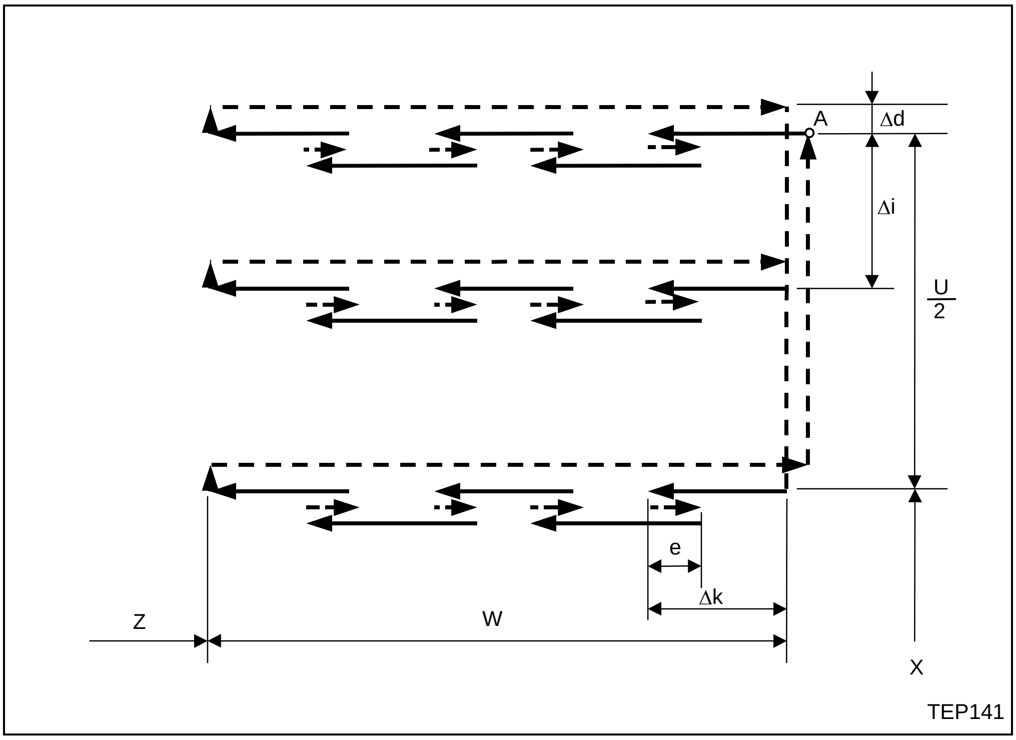
This function is used for smooth disposal of machining chips in longitudinal cut-off machining. For SS materials which produce hard-to-cut machining chips this function can be managed for easy machining chip disposal.

2. Programming format

G74 Re;
G74 Xx/Uu Zz/Ww PΔi QΔk RΔd Ff Ss Tt;

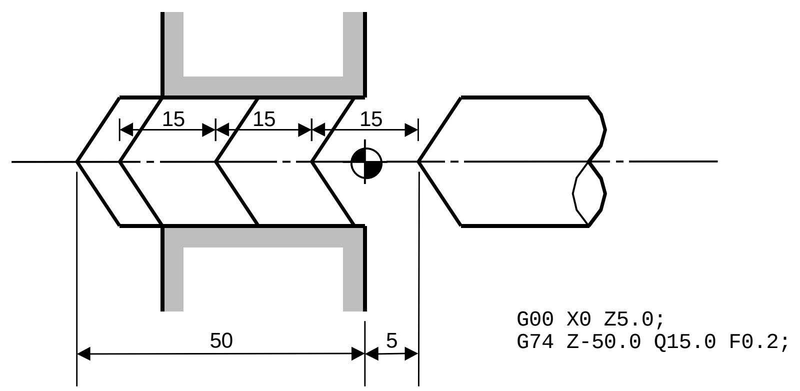
e : Distance of return
This command is modal and valid until a new value is commanded.
x/u : Absolute value/incremental value of X-axis
z/w : Absolute value/incremental value of Z-axis
Δi : X-axis movement distance (commmand without sign)
Δk : Z-axis cut depth (command without sign)
Δd : Tool escape distance at the bottom of cut
It is usually commanded with a plus data.
When address X/U and P are omitted, however, it is commanded with the sign of direction to be escaped.
f : Feed rate
s : S command
t : T command

The distance “e” is set by parameter **TC74** (pecking return distance in grooving process).



3. Detailed description

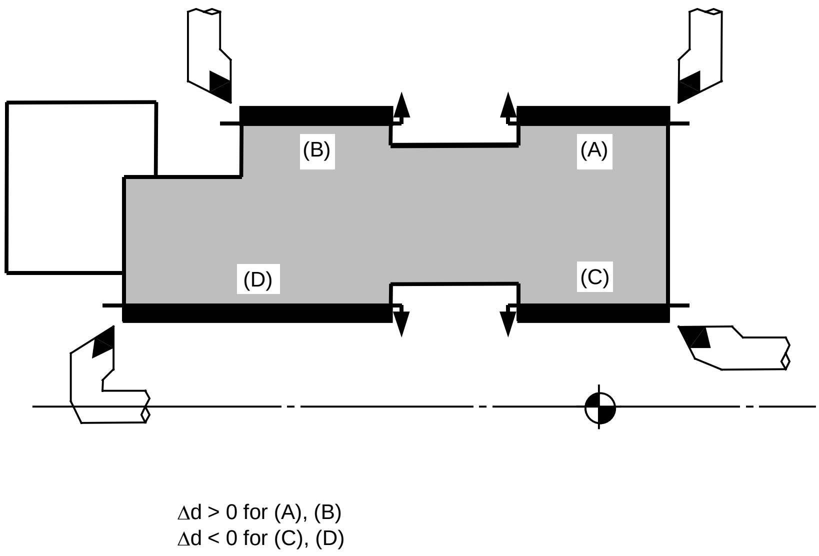
1. For drilling X (U), P and RΔd are not required. Omit these data.



TEP142

2. Without RΔd, escape will be considered as 0. Normally RΔd is specified with plus data. When X (U) and P are omitted in outside or inside diameter machining, however, RΔd requires a sign.

   Four combinations of G74



TEP143

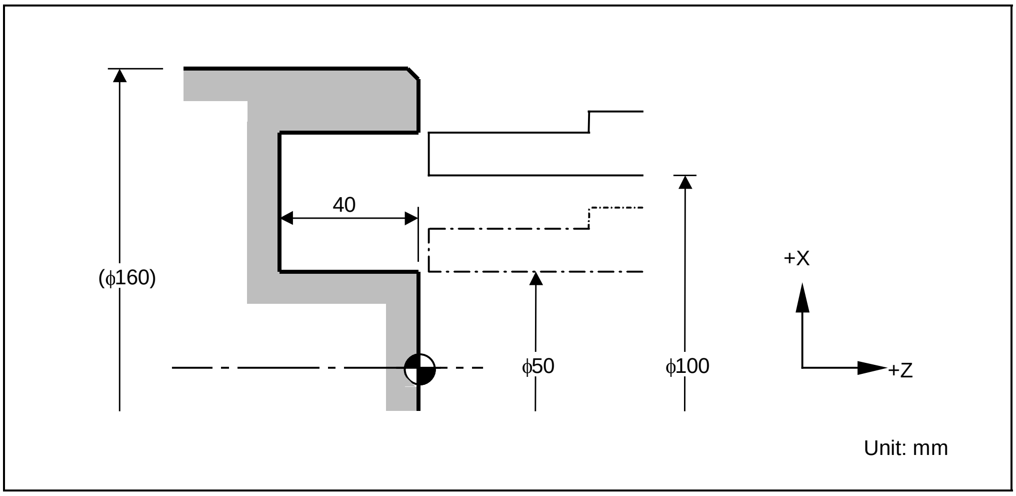
3. During single block operation, all the blocks are executed step by step.

Return to Library

4. Remarks

1. During single block operation, all the blocks are executed step by step.
2. Omission of address X (U), P and RΔd provides the operation of Z-axis alone, resulting in peck drilling cycle.
3. "e" and Δd are both command values of address R. The differentiation is given by whether Z (W) is commanded together. That is, the command R together with Z (W) results in that of Δd.
4. Cycle operation is performed in the block where Z (W) is commanded.

5. Sample programs




```
G00 G96 G98;
G28 U0 W0;
X100.Z2.;
G74 R2.;
G74 U-50.Z-40.P5.Q7.F150 S100 M3;
G28 U0 W0;
M30;
```

14-2-6 Transverse cut-off cycle: G75 [Series M: G275]

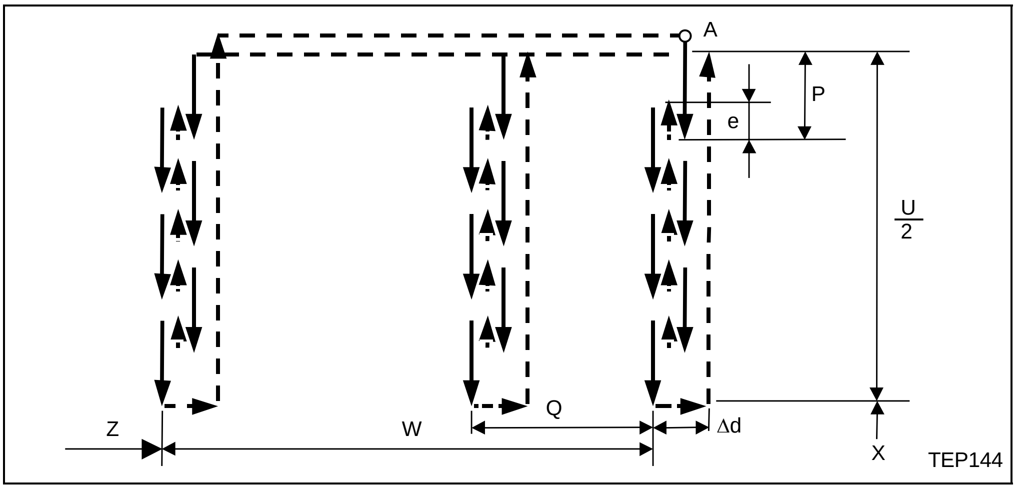
1. Overview

This function is used for smooth disposal of machining chips in transverse cut-off machining. This allows easy disposal of machining chips in face turning as well.

2. Programming format

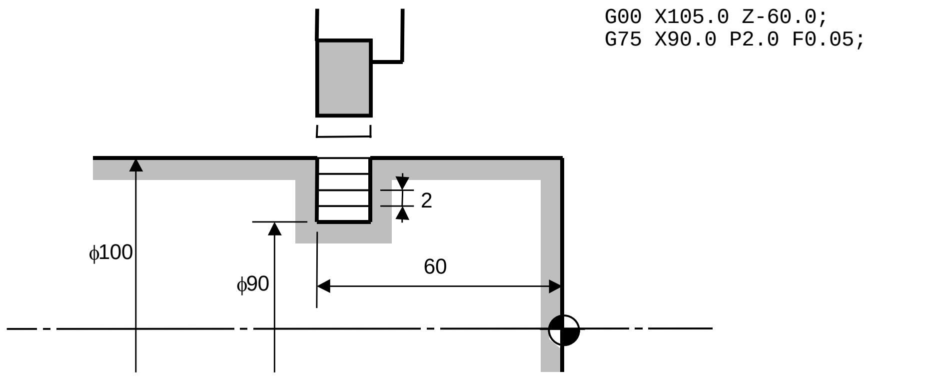
G75 Re ;
G75 X(U)_ Z(W)_ P_ Q_ RΔd F_ S_ T_ ;

G75 executes cycle as shown below.



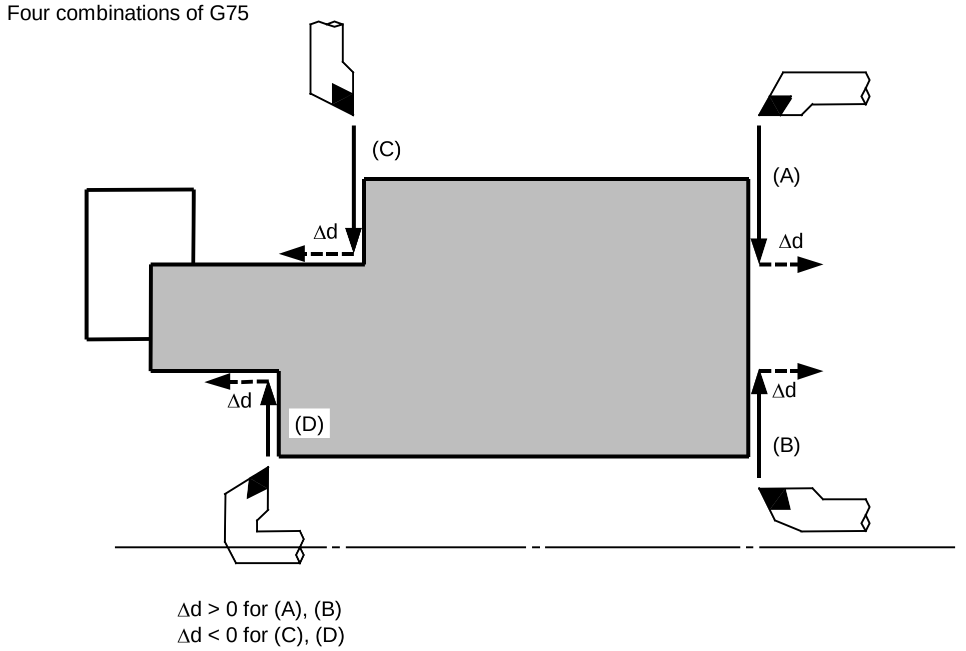
3. Detailed description

1. For outside and inside diameter groove machining, Z (W), Q and RΔd are not required. Omit these data.



TEP145

2. Without RΔd, escape distance in Z-axis direction will be considered as 0. Normally RΔd is specified with plus data. When Z (W) and Q are omitted in edge machining, however, RΔd requires a sign.



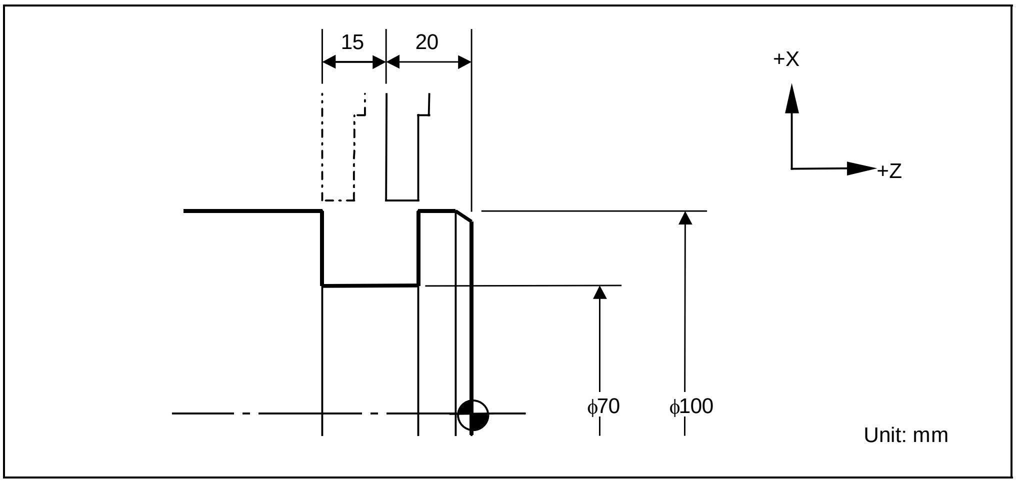
TEP146

3. During single block operation, all the blocks are executed step by step.

4. Remarks

1. Both G74 and G75, which are used for cutting off, grooving or drilling, are a cycle to give the escape of a tool automatically. Four patterns which are symmetrical with each other are available.
2. The return distance “e” can be set by parameter **TC74**. The parameter setting value will be overridden with program command.
3. During single block operation, all the blocks are executed step by step.

5. Sample programs

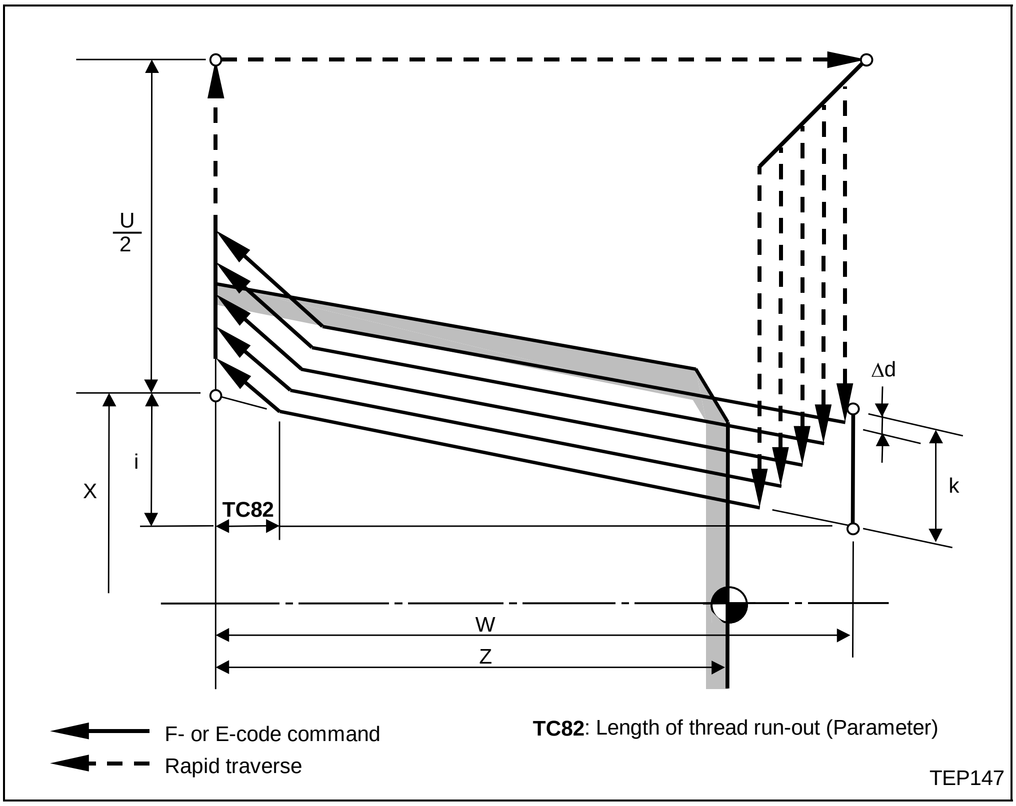


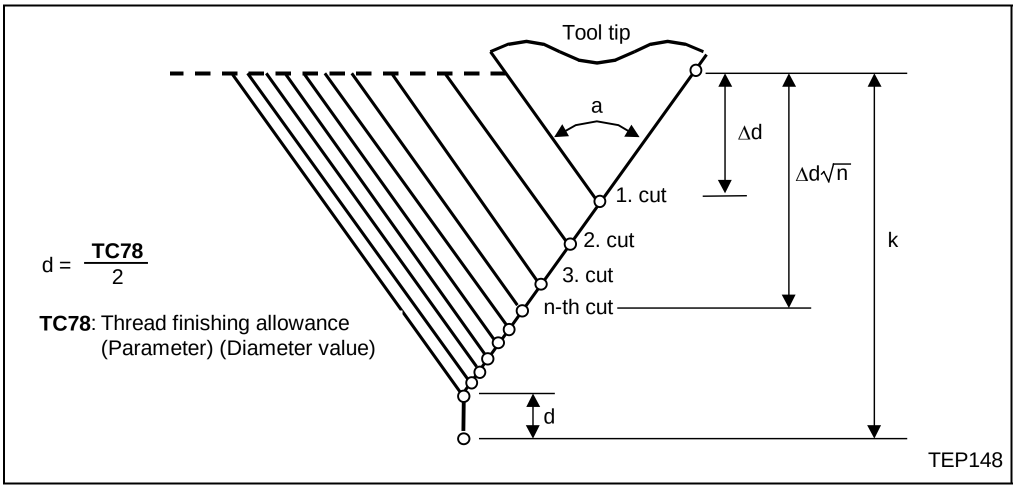

```
G00 G96 G98;
G28 U0 W0;
X102.Z-20.;
G75 R2.;
G75 W-15.X70.P6.Q5.F150 S100 M3;
G28 U0 W0;
M30;
```

Return to Library

14-2-7 Compound threading cycle: G76 [Series M: G276]

1. Cycle configuration





2. Programming format

G76 P<u>mra</u> R<u>d</u>; (omission allowed)
G76 X<u>x</u>/U<u>u</u> Z<u>z</u>/W<u>w</u> R<u>i</u> P<u>k</u> Q<u>Δd</u> F<u>ℓ</u> S_ T_;

m : Repeat times of final finishing (1 to 99)
This command is modal and valid until a new value is commanded.

r : Length of thread run-out
Assuming that the lead is ℓ, the command is given with two numerals of 00 to 99 in 0.1 increments between 0.0 and 9.9. This command is modal and valid until a new value is commanded.

a : Tool tip angle (thread angle)
Six kinds of 80°, 60°, 55°, 30°, 29° and 0° can be selected. The value correspoinding to the angle is commanded with two numerals. This command is modal and valid until a new value is commanded.

d : Finishing allowance
This command is modal and valid until a new value is commanded.

i : Radial difference of threading portion
If i = 0, straight thread cutting is provided.

k : Thread height (Commanded with the distance in the X-axis direction and radius value)

Δd : First cut depth (radial data)

ℓ : Lead of thread (As with G32 thread cutting)

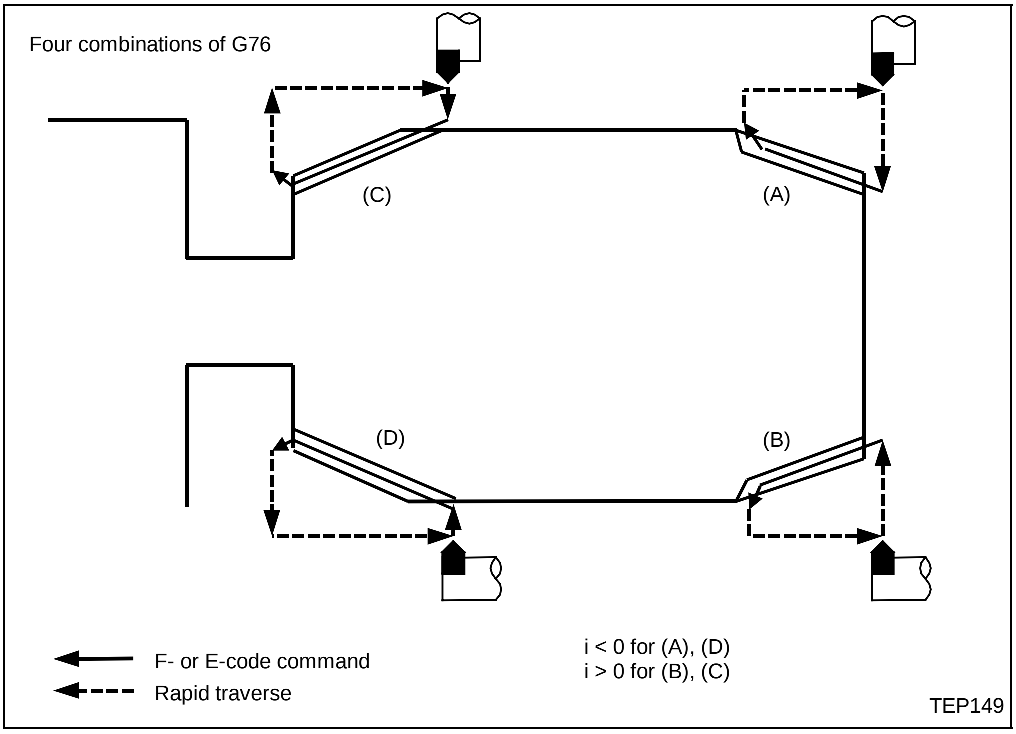
S, T : As with G71

Note: "m", "r" and "a" are commanded together by address P.
When m = 2 times, r = 1.2 ℓ and a = 60° are provided, enter the data as follows.

P <u>02</u> <u>12</u> <u>60</u>
m r a

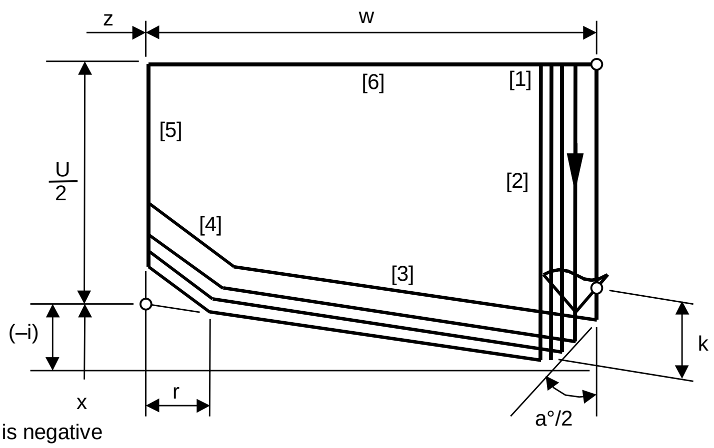
3. Detailed description

- Length of thread run-out can be set by parameter **TC82** by 0.1 × L units in a range 0.1 × L to 4.0 × L (L as lead).
- Cut depth is determined with Δd for initial cut, and $\Delta d\sqrt{n}$ for n-th cut to have a constant depth for each cut.



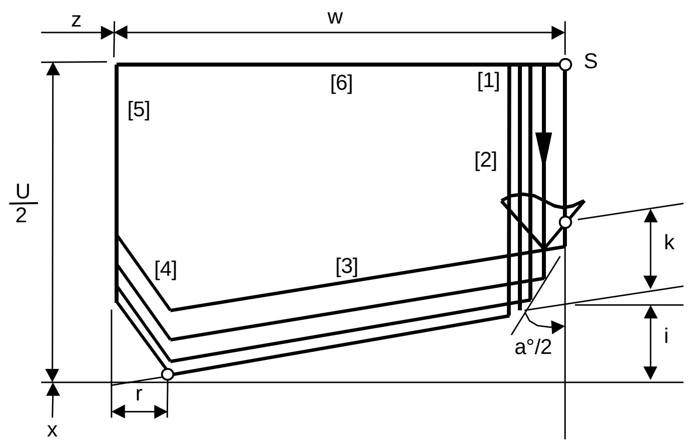
- One cycle configuration

  The tool moves at rapid feed for operations [1], [2], [5] and [6] in the cycle and at the cutting feed based on the value designated to F for operations [3] and [4].



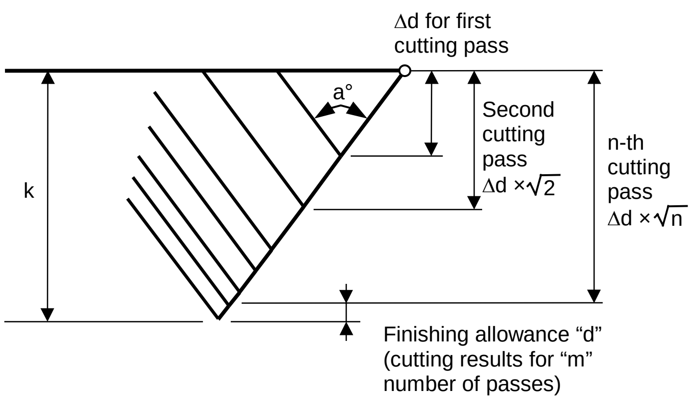
When R is negative

TEP151



When Ri is positive

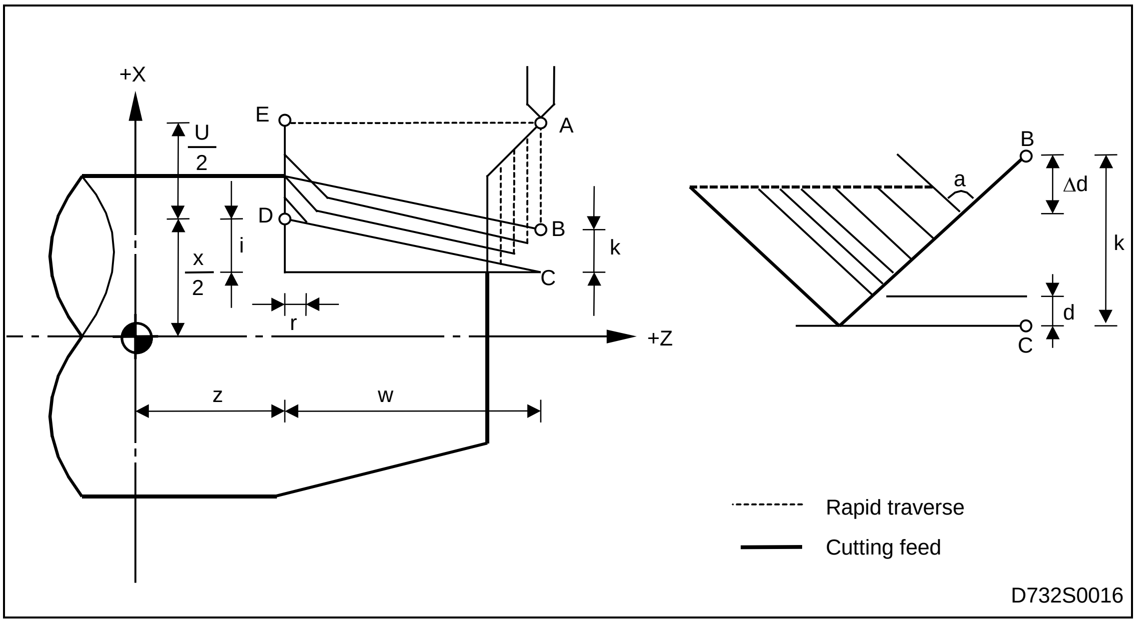
TEP152



TEP153

4. Remarks

1. When the feed hold button is pressed during execution of G76, undergoing threading will be automatically stopped after completion of a block without threading or after completion of chamfering by setting the parameter **F111** bit 2 as in the case of G92. (The feed hold lamp lights immediately in the feed hold mode and it goes off when automatic operation stops.) If threading is not being carried out, the feed hold lamp lights and the feed hold status is established.
2. The machining stops upon completion of operations [1], [4] and [5] when the mode is switched to another automatic mode during the G76 command execution, when automatic operation is changed to manual operations or when single block operation is conducted.
3. During execution of G76, validity or invalidity of dry run will not be changed while threading is under way.
4. During single block operation, all the blocks are executed step by step. For blocks of threading, however, the subsequent block is also executed.
5. For machines with the optional function for automatic correction of threading start position, the thread cutting conditions can be changed by “overriding” the spindle speed.
   See Subsection 6-13-6 for more information.

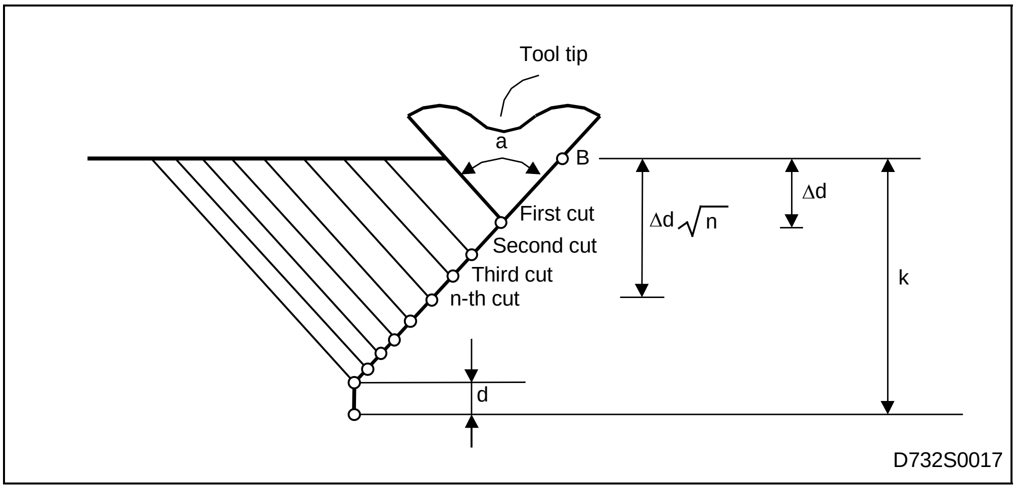


5. Parameter

- Repeat times of final finishing can be set by parameter **TC81**. Parameter setting values will be overridden with program command.
- Length of thread run-out can be set by parameter **TC82**. The parameter setting value will be overridden with program command.
- Tool tip angle can be set by parameter **TC80**. Parameter setting values will be overridden with program command.
- Finishing allowance can be set by a parameter **TC78**. The parameter setting will be overridden with program command.

6. Detailed description

- Setting the tool tip angle provides the machining of a single tip, permitting the decrease in a load applied to the tool tip.
- Cut amount is held constant by setting the first cut depth as Δd and n-th cut depth as $\Delta d\sqrt{n}$.



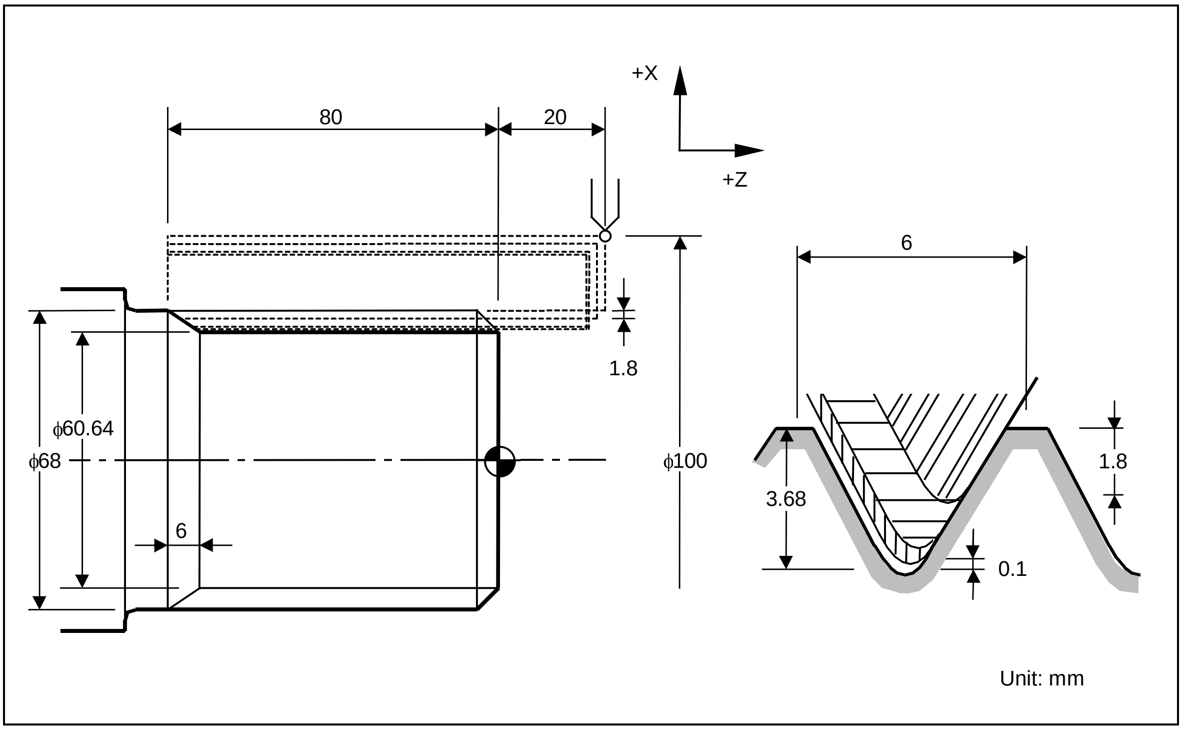
- Allowing for the sign of each address, four patterns are available, and internal threads can also be cut.
- Threading cycle provides a feed commanded by F code or E code only between C and D, and rapid feed for others.
- For the cycle shown above, the signs of increment are as follows:

  u, w ....According to the direction of paths A→C and C→D.
  i.....................According to the direction of path A→C.
  k.....................Plus (always plus)
  Δd ....Plus (always plus)

- Finishing allowance (d; diameter value) can be set by parameter (**TC78**) within the range as follows:

  0 to 65.535 mm (6.5535 inches)

7. Sample programes




```
G00 G97 G99;
G28 U0 W0;
S500 M3;
X100.Z20.;
G76 P011060 R0.2;
G76 X60.64 Z-80.P3.68 Q1.8 F6.0;
G28 U0 W0 M5;
M30;
```

8. Notes

1. For G76 cycle, the notes on threading are as with G32 and G92 threading. If feed hold works during threading, when the parameter of “feed hold during threading” is valid (**F111** bit 2 = 1), the tool stops at the chamferring position at that moment (see item 3 below). Refer to G92 threading cycle for details.

2. Run-out angle can be set in parameter **F28** within the range from 0° to 89°, but it is valid only from 45° to 60°.
   Setting of 90° or more is taken as 45°.
   Setting of 0° to 45° is taken as 45°, and that of 46° to 89° as 60°.

3. During threading, the feed hold during cycle performs one of the follwing two stopping operations according to the parameter (**F111** bit 2).
   - After the block following threading is executed, the tool stops.
   - The tool is stopped at the point where chamferring is accomplished at 60° from the position where the feed hold key is pressed.

   The tool stops immediately except during threading.
   Pressing the cycle start button again causes X and Z together to return to the starting point at rapid feed, and the cycle continues.

4. An alarm occurs in the cases below.
   - Either X or Z is not specified.
   - Either displacement distance of X- or Z-axis is 0°.
   - The thread angle exceeds the range from 0° to 120°.

5. During single block operation, all the blocks are executed step by step. For blocks of threading, however, the subsequent block is also executed.

6. Data commanded by P, Q and R is differentiated by whether addresses X (U) and Z (W) are specified in the same block.

7. The tool performs cycle operation in G76 block where addresses X (U) and Z (W) are commanded.

8. For machines with the optional function for automatic correction of threading start position, the thread cutting conditions can be changed by “overriding” the spindle speed.
   See Subsection 6-13-6 for more information.

14-2-8 Checkpoints for compound fixed cycles: G70 to G76 [Series M: G270 to G276]

1. Except for the parameters which have been preset, set all the required parameters in the blocks for the compound fixed cycle commands.

2. Provided that the finishing shape sequence has been registered in the memory, compound fixed cycle I commands can be executed in the memory, MDI or tape operation mode.

3. When executing commands G70 to G73, ensure that the sequence number of the finishing shape sequence which is specified to addresses P and Q is not duplicated in that program.

4. The finishing shape sequence specified to addresses P and Q in the blocks G71 to G73 should be prepared so that the maximum number of blocks is 100 for all the commands for corner chamfering, corner rounding and other commands including the automatic insertion blocks based on tool nose radius compensation. If this number is exceeded, program error occurs.

5. The finishing shape sequences which are designated by the blocks G71 to G73 should be a program in monotonous changes (increases or reductions only) for both the X- and Z-axes.

6. Blocks without movement in the finishing shape sequence are ignored.
7. N, F, S, M and T commands in the finishing shape sequence are ignored during roughing.
8. When any of the following commands are present in a finishing shape sequence, program error occurs.
   - Commands related to reference point return (G27, G28, G29, G30)
   - Threading (G33)
   - Fixed cycles
   - Skip functions (G31, G37)
9. Subprogram call in the finish shape program can be made.
10. Except for threading cycles, operation stops at the ending (starting) point of each block in the single block mode.
11. Remember that, depending on whether the sequence or program number is designated, the next block upon completion of the G71, G72 or G73 command will differ.
    - When the sequence No. is designated:
      The next block is that which follows the block designated by Q.
      Operation moves to the N600 block upon completion of the cycle.

```
     :
N100 G71 P200 Q500 U_ W_····;
N200 ⎫
N300 ⎪
N400 ⎬ Finishing shape sequence
N500 ⎭
N600
```

    - When the program Nn. is designated:
      The next block is that which follows the cycle command block.
      Operation moves to the N200 block upon completion of the cycle.

```
      :
N100 G71 A100 U_ W_····;
N200
N300
N400
                    O100
                    N10 X100.Z50.;
                    N20
```

12. The next block upon completion of the G70 command is that which follows the command block.
    Operation moves to the N1100 block upon completion of the G70 command.

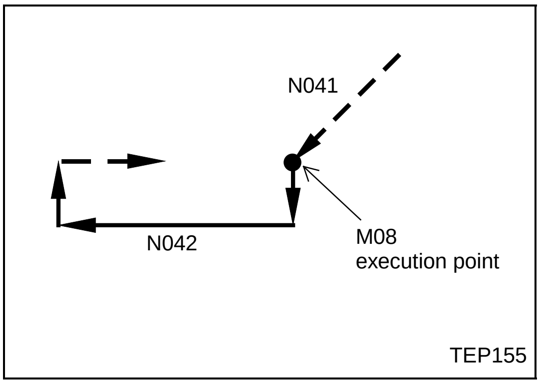
```
 :
N100 ········;
N200 ········;
N300 ········;
N400 ········;
N500 ········;
 :
N1000 G70 P200 Q500;(or G70 A100;)
N1100 ········;
 :
```

13. Manual interruption can be applied during a compound fixed cycle command (G70 to G76). However, upon completion of the interruption, the tool must first be returned to the position where the interruption was applied and then the compound fixed cycle must be restarted.
    If it is restarted without the tool having been returned, all subsequent movements will deviate by an amount equivalent to the manual interruption amount.

14. Compound fixed cycle commands are unmodal commands and so they must be set every time they are required.
15. Programming error No. **898 LAP CYCLE ILLEGAL SHAPE DESIGN.** occurs in the G71 and G72 commands even when, because of tool nose R compensation, there is no further displacement of the Z-axis in the second block or displacement of the Z-axis in the opposite direction is made.
16. Command which must not be entered in blocks for finish shape defined by P and Q in G70 to G73.

    M98/M99
    T code
    G10, G27, G28, G29, G30
    G20, G21, G94, G95, G52, G53, G68, G69
    G32, G77, G78, G79
17. Sequence number specified by P and Q for G70 to G73 must not be entered more than once within a program.
18. In blocks for finishing shape defined by P and Q for G70 to G73, if command for final shape is chamfering (G01 X_ I_) (G01 Z_ K_) or corner rounding (G01 Z_ R_) (G01 X_ R_), alarm **NO DIRECTIVE FOR NEXT MOVE R/C** occurs.
19. Blocks with sequence number specified by P for G71 to G73 must be in G00 or G01 mode.
20. In the case of stopping the machining with the stop button during execution of G70 to G76 and applying the manual interruption, machining must be restarted with the start button after returning to the stopped position (by manual movement of tool tip). If not returned, the tool position at machining restart will be dislocated by pulse movement due to the handle interruption.
    Distance moved by handle interruption can be cancelled by resetting.
21. When setting M, T commands in blocks with G70 to G76, execution point must be considered.

```
N041 G00 X100.Z0;
N042 G71 P101 Q103 U0.5 W0.5
     D4000 F0.5 S150 M08;
       ⋮
N101 G01 X90.F0.5;
N102 Z-20.;
N103 X100.;
```



14-3 Hole Machining Fixed Cycles: G80 to G89 [Series M: G80, G283 to G289]

14-3-1 Outline

1. Function and purpose

When performing predetermined sequences of machining operations such as positioning, hole machining, boring and tapping, these functions permit to command in a single block the machining program which is normaly commanded in several blocks. In other words, they simplify the machining program.
The following types of fixed cycles for hole machining are available.

G-code	Hole machining axis	Hole machining start	Operation at hole bottom	Return movement	Application
G80	—	—	—	—	Cancel
G83	Z	Cutting feed, intermittent feed	Dwell	Rapid feed	Deep hole drilling cycle
G84	Z	Cutting feed	Dwell, spindle reverse rotation	Cutting feed	Tapping cycle
G84.2	Z	Cutting feed	Spindle reverse rotation	Cutting feed	Synchronous tapping cycle
G85	Z	Cutting feed	Dwell	Cutting feed	Boring cycle
G87	X	Cutting feed, intermittent feed	Dwell	Rapid feed	Deep hole drilling cycle
G88	X	Cutting feed	Dwell, spindle reverse rotation	Cutting feed	Tapping cycle
G88.2	X	Cutting feed	Spindle reverse rotation	Cutting feed	Synchronous tapping cycle
G89	X	Cutting feed	Dwell	Cutting feed	Boring cycle

A fixed cycle mode is cancelled when the G80 or any G-code in the 01 group is set. The various data will also be cleared simultaneously to zero.

2. Programming format

A. Face hole machining

G8∆ X/U_ C/H_ Z/W_ R_ Q_ P_ F_ L(K)_ M_;

- M_: M-code
- L(K)_: Number of repetitions
- Z/W_ R_ Q_ P_ F_: Hole machining data
- X/U_ C/H_: Hole positioning data
- G8∆: Hole machining mode (G83, G84, G84.2, G85)

B. Outside hole machining

G8☐ Z/W_ C/H_ X/U_ R_ Q_ P_ F_ L(K)_ M_;

- M_: M-code
- L(K)_: Number of repetition
- X/U_ R_ Q_ P_ F_: Hole machining data
- Z/W_ C/H_: Hole positioning data
- G8☐: Hole machining mode (G87, G88, G88.2, G89)

C. Cancel

G80 ;

D. Data outline and corresponding address

- Hole machining modes:
  These are the fixed cycle modes for drilling (G83, G87), tapping (G84, G84.2, G88, G88.2) and boring (G85, G89).
  These are modal commands and once they have been set, they will remain valid until another hole machining mode command, the cancel command for the hole machining fixed cycle or a G command in the 01 group is set.
- Hole positioning data:
  These are for the positioning of the X(Z)- and C-axes.
  These are unmodal data, and they are commanded block by block when the same hole machining mode is to be executed continuously.
- Hole machining data:
  These are the actual machining data.
  Except for Q, they are modal. Q in the G83 or G87 command is unmodal and is commanded block by block as required.
- Number of repetitions:
  This number is designated for machining holes at equal intervals when the same cycle is to be repeated.
  The setting range is from 0 through 9999 and the decimal point is not valid.
  The number is unmodal and is valid only in the block in which it has been set. When this number is not designated, it is treated as L1. When L0 is deisgnated, the hole machining data are stored in the memory but no holes will be machined.
  Use address K for standard mode.
- M-code:
  Commanding M210 causes M-code for C-axis clamping to be outputted at the start of operation 2 (described later), and M-code for C-axis unclamping to be outputted at the end of operation 5.
  For G84 (G88) and G84.2 (G88.2), M-code for the direction of spindle revolution is specified. If not specified, the preset data of the respective parameter will be used.

Address	Signification
G	Selection of hole machining cycle sequence (G80, G83, G84, G84.2, G85, G87, G88, G88.2, G89)
X/U, (Z/W)*, C/H	Designation of hole position initial point (absolute/incremental value)
Z/W, (X/U)*	Designation of hole bottom position (absolute/incremental value from reference point)
R	Designation of R(apid feed)-point position (incremental value from initial point) (sign ignored.)
Q	Designation of cut amount for each cutting pass in G83 (G87); always incremental value, radial value (sign ignored, decimal point can be commanded in T32 compatible mode, but not in Standard mode)
P	Designation of dwell time at hole bottom point; relationship between time and designated value is same as for G04.
F	Designation of feed rate for cutting feed
L (K)	Designation of number of repetitions, 0 to 9999 (default = 1)
M	Designation of M-code

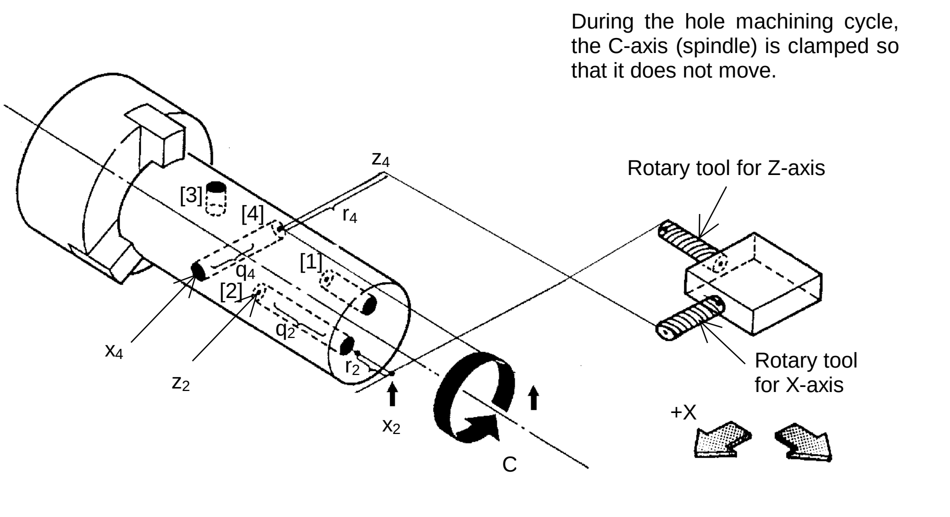
* Addresses in parentheses apply for commands G87, G88 and G89.

E. Use of the hole machining fixed cycles on the 2nd spindle side

The hole machining fixed cycles can also be used for the lower turret on the 2nd spindle side with the aid of the related G-code (G109 L2).

3. Outline drawing

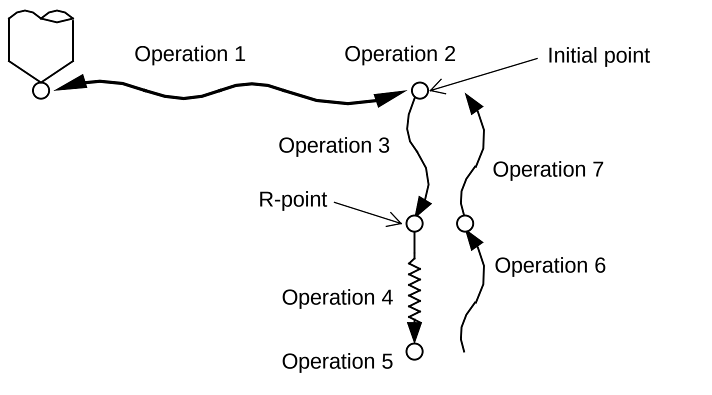
The hole machining axes for the hole machining fixed cycle and the positioning are shown in the outline drawing below.



[1] G83 Xx_1 Cc_1 Zz_1 Rr_1 Qq_1 Pp_1 Ff_1 Ll_1 ;
[2] G83 Xx_2 Cc_2 Zz_2 Rr_2 Qq_2 Pp_2 Ff_2 Ll_2 ;
[3] G87 Zz_3 Cc_3 Xx_3 Rr_3 Qq_3 Pp_3 Ff_3 Ll_3 ;
[4] G87 Zz_4 Cc_4 Xx_4 Rr_4 Qq_4 Pp_4 Ff_4 Ll_4 ;

4. Operations

There are 7 actual operations which are each described in turn below.



TEP156

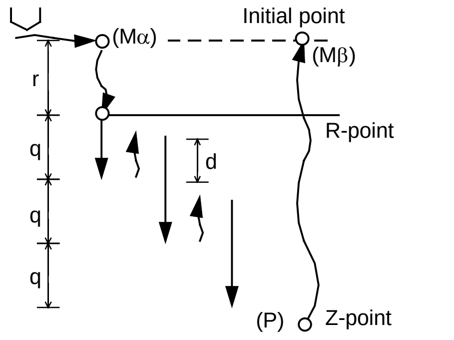
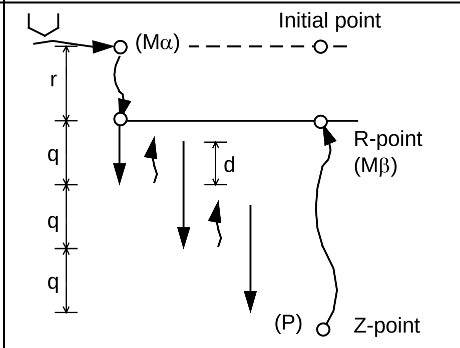
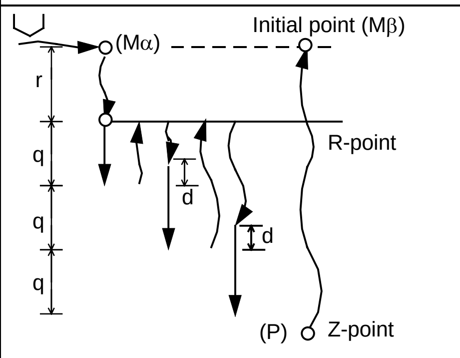
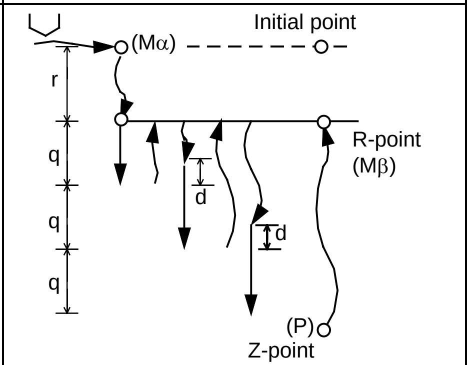
Operation 1 : Positionning by rapid feed to the X(Z) and C-axis initial point
Operation 2 : Output of the M-code for C-axis clamping if it is set
Operation 3 : Positionning to the R-point by rapid feed
Operation 4 : Hole machining by cutting feed
Operation 5 : Operation at the hole bottom position which differs according to the fixed cycle mode. Possible actions include rotary tools reverse rotation (M204), rotary tools forward rotation (M203) and dwell.
Operation 6 : Return to the R-point
Operation 7 : Return to the initial point at rapid feed

(Operation 6 and 7 may be a single operation depending on the fixed cycle mode.)
Whether the fixed cycle is to be completed at operation 6 or 7 can be selected by the user parameter **F162** bit 3.
Parameter **F162** bit 3 = 0: Initial level return
Parameter **F162** bit 3 = 1: R-point level return

14-3-2 Face/Outside deep hole drilling cycle: G83/G87 [Series M: G283/G287]

1. When the Q command is present (deep hole drilling)

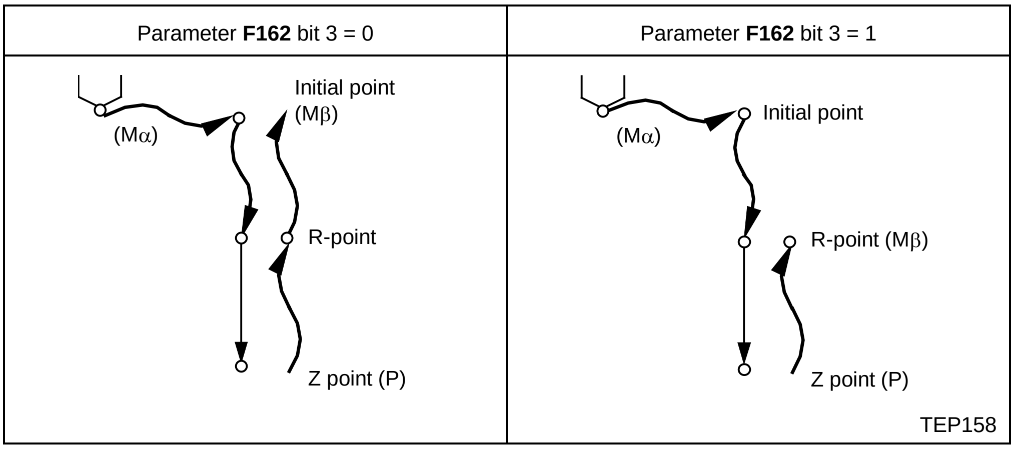
G83(G87)X(Z)_ C_ Z(X)_ Rr Qq Pp Ff Ll Mm ;

Type	Parameter **F162** bit 3 = 0	Parameter **F162** bit 3 = 1
A (high speed)		
B (normal speed)		 TEP157

- Return distance "d" is set by the parameter (**F12**: Pecking return distance in drilling process). The tool returns at rapid feed.
- (Mα): The C-axis clamping M-code (Mm) is outputted here if specified.
- (Mβ): The C-axis unclamping M-code (C-axis clamp M-code + 1 = Mm+ 1) is output when there is a C-axis clamping M-code command (Mm).
- (P): Dwell is performed for the duration equivalent to the time designated by P.

2. When the Q command is not present (drilling)

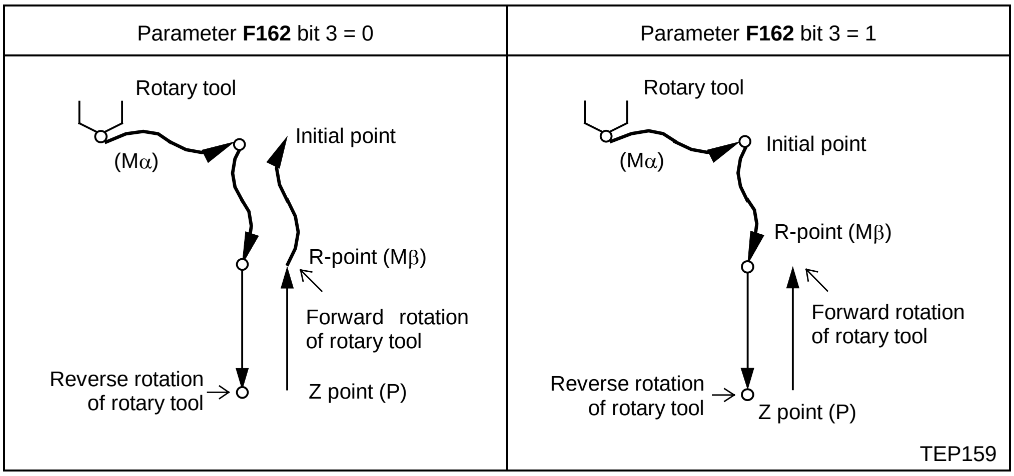
G83 (G87) X(Z)_ C_ Z(X)_ R_ P_ F_ L_ M_ ;



See 1 for details on (Mα), (Mβ) and (P).

14-3-3 Face/Outside tapping cycle: G84/G88 [Series M: G284/G288]

G84 (G88) X(Z)_ C_ Z(X)_ R_ P_ F_ L_ M_ ;



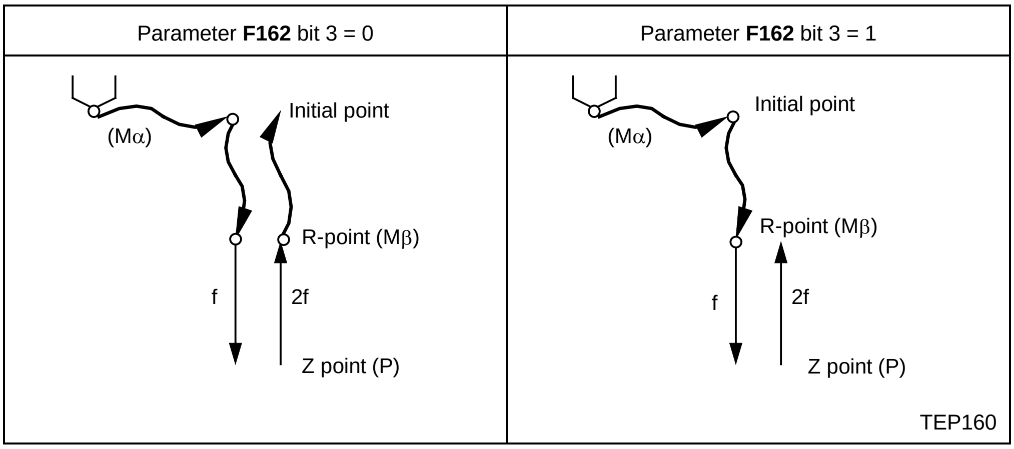
- (Mα), (Mβ) and (P) are as with G83.
- During the execution G84 (G88), the override cancel status is established and 100 % override is automatically applied. Dry run is also ignored.
- When feed hold is applied during the execution of G84 (G88), block stop results after return movement.
- The in-tapping signal is output in a G84 (G88) modal operation.
- The fixed cycle subprograms should be edited if the rotary tool stop (M205) command is required before the rotary tool reverse rotation (M204) or forward rotation (M203) signal is output.

Note: Tapping cycle in the turning mode is not available on the side of secondary spindle.

Return to Library

14-3-4 Face/Outside boring cycle: G85/G89 [Series M: G285/G289]

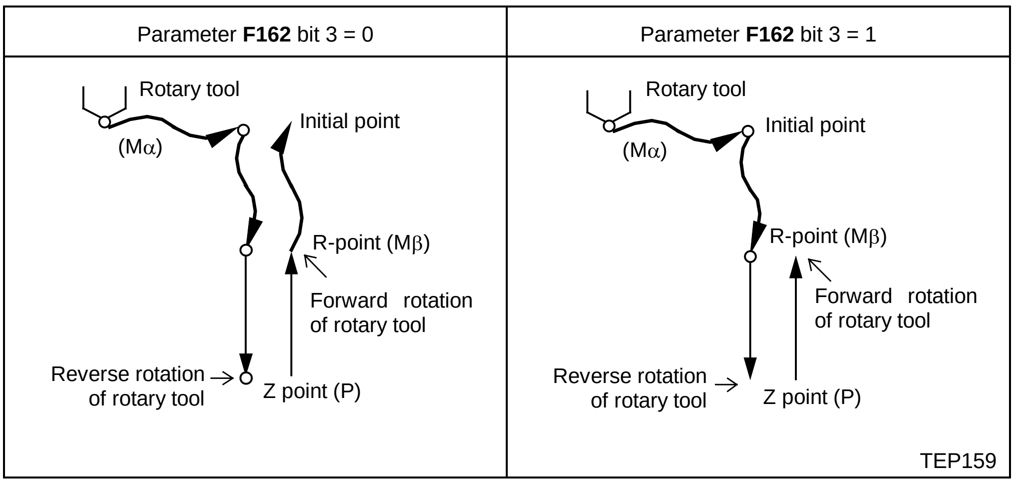
G85 (G89) X(Z)_ C_ Z(X)_ R_ P_ F_ L_ M_ ;



- (Mα), (Mβ) and (P) are as with G83.
- The tool returns to the R-point at a cutting feed rate which is double the designated feed rate command. However, it does not exceed the maximum cutting feed rate.

14-3-5 Face/Outside synchronous tapping cycle: G84.2/G88.2 [Series M: G284.2/G288.2]

G84.2 (G88.2) X(Z)_ C_ Z(X)_ R_ P_ F_ L_ M_ ;



1. Detailed description

- (Mα), (Mβ) and (P) are as with G83.
- The spindle is reversed at the hole bottom to perform tapping cycle. During tapping cycle operation by G84.2 (G88.2), feed rate override is ignored. Even if feed hold is applied, the cycle does not stop until the end of return operation.

- Tapping cycle and reverse tapping cycle can be performed by specifying spindle normal or reverse rotation with M-codes (M03, M04, M203, M204).

  Output to the machine side is as follows:

Programmed command	Z point	R point
M03	M04	M03
M04	M03	M04
M203	M204	M203
M204	M203	M204

- As for synchronous tapping on the face (G84.2), the combination of the direction of the Z-axis movement (in the workpiece coordinate system) and that of the spindle rotation determines the type of tapping: normal or reverse.

Type of tapping	Z-axis movement direction (in the workpiece coordinate system)	Command for the direction of spindle rotation
Normal tapping	Negative	M03/M203
	Positive	M04/M204
Reverse tapping	Negative	M04/M204
	Positive	M03/M203

  Programming example:

  1) `G00 Z0.`
     `G84.2 Z10. F0.1 M4` .... Normal tapping

  2) `G00 Z0.`
     `G84.2 Z–10. F0.1 M4` .... Reverse tapping

- When G84.2 is commanded by feed per revolution (G95), where the unit of cutting feed rate F is set to mm/rev or inch/rev, tap thread pitch can be commanded directly. When X-axis is used as a hole machining axis, G88.2 is commanded in place of G84.2.

- In tapping cycle (G84), the feed rate of Z-axis per spindle rotation must be equal to the thread pitch of a tap. This means that the most desirable tapping always fills the following conditions.

  P = F/S

  P : Tap thread pitch (mm)
  F : Z-axis feed rate (mm/min)
  S : Spindle speed (rpm)

  Spindle rotation and Z-axis feed are independently controlled in tapping cycle (G84). Therefore, the above condition are not always filled. Spindle rotation and Z-axis feed are both decelerated and stopped particularly at the hole bottom, and then the spindle and Z-axis move in the reverse direction, giving acceleration.
  Since each acceleration and deceleration are independently performed, the above conditions are not filled usually. As a result, for improving the accuracy of tapping, it is customary to compensate the feed by mounting a spring in the tap holder.

  On the other hand, for synchronous tapping cycle (G84.2), spinde rotation and Z-axis feed are controlled so that they are always synchronized. In other words, for normal rotation, the spindle is controlled only in relation to speed. However, for synchronous tapping, position control is given also to spindle rotation. And spindle rotation and Z-axis feed are controlled as the linear interpolation of two axes. This fills the condition of P=F/S even in deceleration and acceleration at the hole bottom, permitting tapping of high accuracy.

2. Remarks

1. Synchronous tapping cycle (G84.2) and tapping cycle (G84) differ only in the control method of spindle when Z-axis moves from point R to point Z and when it does from point Z to point R. In synchronous tapping, the movement of spindle is detected by the position coder as shown below, and position control is given. Spindle motor is controlled like a servo motor to give the linear interpolation of two axes of Z-axis and spindle.

   The movement distance of linear interpolation of Z-axis and spindle as well as the feed rate are as given below.

	Movement distance	Feed rate
Z-axis	z = Distance between point R and point Z (mm, inch)	Fz = F command value (mm/min, inch/min)
Spindle	s = z × (S command value / F command value) × 360 (deg)	Fs = S command value (rpm)

   Synchronous tapping cycle is as with G84 except that it differs from tapping cycle in the control method of spindle when Z-axis moves from point R to point Z and when it does from point Z to point R. Refer to the section of fixed cycle G84 for the notes including programming.

2. Z-axis is used as a hole machining axis in the above description. When X-axis is used as a hole machining axis, G88.2 is commanded.

   Example: G88.2 Z/W_ C/H_ X/U_ R_ F_ ; X-axis is used as a hole machining axis.

3. For synchronous tapping cycle (G84.2), feed rate override is invalid, and it is fixed to 100%.

4. Synchronous tapping cycle in the turning mode is not available on the side of secondary spindle.

5. Two types of synchronous tapping are provided: spindle synchronous tapping and mill synchronous tapping.
   However, only either can be used.

14-3-6 Hole machining fixed cycle cancel: G80

This command cancels the hole machining fixed cycles (G83, G84, G84.2, G85, G87, G88, G88.2, G89). The hole machining mode as well as the hole machining data are cancelled.

14-3-7 Checkpoints for using hole machining fixed cycles

1. When the G84 and G88 fixed cycle commands are set, the rotary tool must be rotated in the designated direction beforehand using a miscellaneous function (M3, M4).

2. If the basic axis, additional axis and R data are present in a block, hole machining is performed in a fixed cycle mode; it will not be performed if these data are not present.
   Even if the X-axis data are present, hole machining will not be executed if a dwell (G04) command is present in the block.

3. The hole machining data (Q, P) should be commanded in the block (block including the basic axis, additional axis and R data) in which the holes are machined.
   The modal data will not be updated even if these data are commanded in a non-hole machining block.

4. When resetting is applied during the execution of the G85 (G89) command, the hole machining data will be erased.

5. The hole machining fixed cycles are also cancelled by any G code in the 01 group besides G80. If it is commanded in the same block as the fixed cycle, the fixed cycle will be ignored.
   m = 01 group code, n = hole machining fixed cycle code

   $$\text{-}\ \underbrace{\text{Gm}}_{\text{executed}}\ \underbrace{\text{Gn}}_{\text{ignored}}\ \underbrace{\text{X(Z)\_ C\_ Z(X)\_}}_{\text{executed}}\ \underbrace{\text{R\_ Q\_ P\_ L(K)\_}}_{\text{ignored}}\ \underbrace{\text{F\_;}}_{\text{q}}$$

   $$\text{-}\ \underbrace{\text{Gn}}_{\text{executed}}\ \underbrace{\text{Gm X(Z)\_ C\_ Z(X)\_}}_{\text{executed}}\ \underbrace{\text{R\_ Q\_ P\_ L(K)\_}}_{\text{ignored}}\ \underbrace{\text{F\_;}}_{\text{memorized}}$$

   Example: G01 G83 X100.C30.Z50.R–10.Q10.P1 F100.;
   G83 G01 X100.C30.Z50.R–10.Q10.P1 F100.;

   In both cases, G01 X100.C30.Z50.F100. is executed.

6. When a miscellaneous command is set in the same block as the fixed cycle command, it is outputted after the initial positioning.
   However, the C-axis unclamping M-code (clamp M + 1) is output after the holes have been machined and the tool returns to the return point.
   When the number of repetitions has been designated, the M command execution in above condition is exercised only for the initial operation except for the C-axis clamping M-code.
   In the case of the C-axis clamping/unclamping M commands, as they are modal, the codes are outputted with each repetitions until the operation is cancelled by the fixed cycle cancel command.

7. When a tool position offset command (T function) is set in a hole machining fixed cycle mode, execution will follow the tool position offset function.

8. When a hole machining fixed cycle command is set during tool nose radius compensation, program error occurs.

9. Cutting feed rate by F will be kept after cancelling the drilling cycle.

14-3-8 Sample programs with fixed cycles for hole machining

1. Face deep hole drilling cycle (G83)

```
G00 G97 G98;
G28 UW;
M200;
M203 S800;
X100.Z2.C0;
G83 X50.H30.Z-20.R5.Q5000 P.2 F200 L3 M210;
G80;
G28 UW;
M30;
```

2. Face tapping cycle (G84)

```
G00 G97 G98;
G28 UW;
M200;
M203 S600;
X102.Z-50.C0;
G84 X50.H30.Z-20.R5.P.2 F300 L3 M203 M210;
G80;
G28 UW;
M30;
```

3. Face boring cycle (G85)

```
G00 G97 G98;
G28 UW;
M200;
M203 S600;
X100.Z2.C0;
G85 X50.H30.Z-20.R5.P.2 F150 L3 M210;
G80;
G28 UW;
M30;
```

4. Outside deep hole drilling cycle (G87)

```
G00 G97 G98;
G28 UW;
M200;
M203 S800;
X102. Z-50. C0;
G87 Z-50.H30.X70.R5.Q5000 P.2 F200 L3 M210;
G80;
G28 UW;
M30;
```

5. Outside tapping cycle (G88)

```
G00 G97 G98;
G28 UW;
M200;
M203 S600;
X102.Z-50.C0;
G88 Z-50.H30.X70.R5.P.2 F300 L3 M203 M210;
G80;
G28 UW;
M30;
```

6. Outside boring cycle (G89)

```
G00 G97 G98;
G28 UW;
M200;
M203 S800;
X102. Z-50.C0;
G89 Z-50. H30.X-70. R5.P.2 F200 L3 M210;
G80;
G28 UW;
M30;
```

7. Face synchronous tapping cycle (G84.2)

```
G00 G97 G98;
G28 UW;
M200;
M203 S600;
X100.Z2.C0;
G84.2 X50.H30.Z-20.R5.F2.00 L3 M203 M210;
G80;
G28 UW;
M30;
```

8. Outside synchronous tapping cycle (G88.2)

```
G00 G97 G98;
G28 UW;
M200;
M203 S600
X102. Z-50. C0;
G88.2 Z-50.H30.X70.R5.F2.L3 M203 M210;
G80;
G28 UW;
M30;
```

14-4 Hole Machining Pattern Cycles: G234.1/G235/G236/G237.1 [Series M: G34.1/G35/G36/G37.1]

14-4-1 Overview

1. Function and purpose

Hole machining patterns are used to arrange on a predetermined pattern hole positions at which to execute a hole-machining cycle.

- Give beforehand a command of the desired hole-machining cycle without any axis positioning data (which only causes storage of the hole-machining data to be executed at the arranged hole positions).
- The execution of this command begins with the positioning to the first one of the arranged holes. The type of hole machining depends on the corresponding cycle designated last.
- The current mode of hole-machining cycle will remain active over the execution of this command till it is cancelled explicitly.
- This command will only activate positioning when it is given in any other mode than those of hole-machining cycle.
- These commands only cause positioning at the speed of the current modal condition (of G-code group 01) in default of any preceding hole-machining cycle.

2. List of hole machining pattern cycles

G-code	Description	Argument addresses	Remarks
G234.1	Holes on a circle	X, Y, I, J, K	
G235	Holes on a line	X, Y, I, J, K	
G236	Holes on an arc	X, Y, I, J, P, K	
G237.1	Holes on a grid	X, Y, I, P, J, K	

14-4-2 Holes on a circle: G234.1 [Series M: G34.1]

As shown in the format below, a command of G234.1 determines a circle of radius “r” around the center designated by X and Y. The circumference is then divided, beginning from the point of the central angle “θ”, regularly by “n”, and the hole machining designated beforehand by a fixed cycle (G81 etc.) will be done around all the vertices of the regular n-gon.

The movement in the XY-plane from hole to hole occurs rapidly (under G00). The argument data of the G234.1 command will be cleared upon completion of its execution.

1. Programming format

G234.1 Xx Yy Ir Jθ Kn;

X, Y : Coordinates of the center of the circle.

I : Radius (r) of the circle. Always given in a positive value.

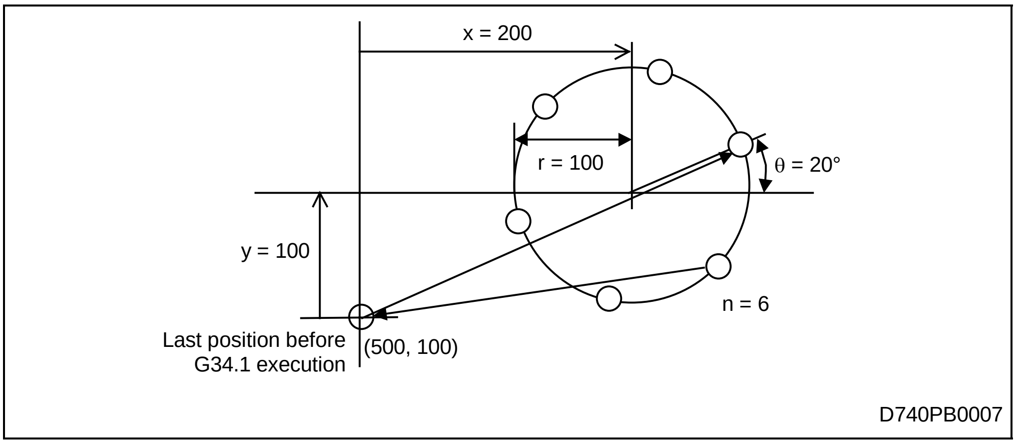
J : Central angle (θ) of the first hole. Positive central angles refer to counterclockwise measurement.

K : Number (n) of holes to be machined (from 1 to 9999). The algebraic sign of argument K refers to the rotational direction of the sequential machining of “n” holes. Set a positive and a negative number respectively for counterclockwise and clock-wise rotation.

2. Sample programes

Given below is an example of G81 hole machining with a figure representing the hole positions.

```
N001 G91;
N002 G81 Z-10. R5. L0. F200;
N003 G90 G34.1 X200. Y100. I100. J20. K6;
N004 G80;
N005 G90 G0 X500. Y100.;
```



3. Notes

- In the use of G-code series T, use the appropriate axis addresses to designate the axis position in an incremental value. As for G-code series M, give a G90 or G91 command as required to designate the position in absolute or incremental values.
- As shown in the above example, the last position of the G234.1 (G34.1) command is on the last one of the arranged holes. Use the method of absolute data input, therefore, to specify the movement to the position for the next operation desired. (An incremental command would require a more or less complicated calculation with respect to that last hole.)

Return to Library

14-4-3 Holes on a line: G235 [Series M: G35]

As shown in the format below, a command of G235 determines a straight line through the starting point designated by X and Y at the angle “θ” with the X-axis. On this line “n” holes will be machined at intervals of “d”, according to the current mode of hole machining.
The movement in the XY-plane from hole to hole occurs rapidly (under G00). The argument data of the G235 command will be cleared upon completion of its execution.

1. Programming format

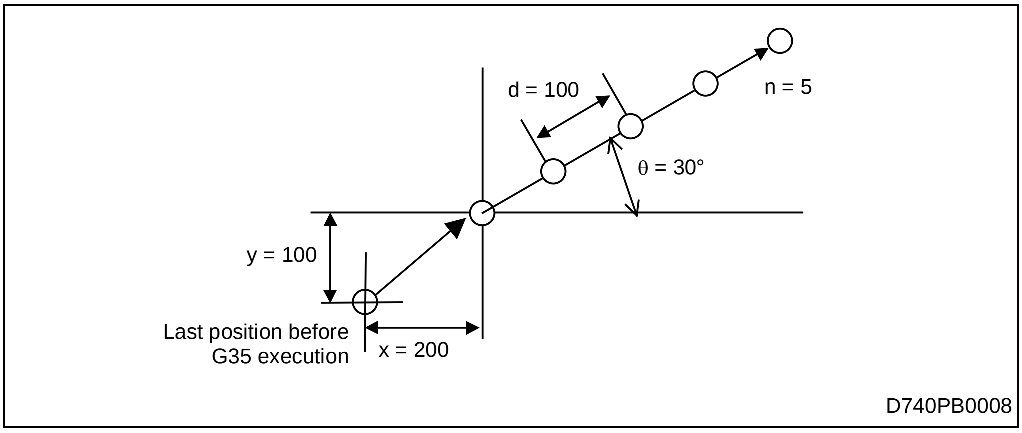
G235 Xx Yy Id Jθ Kn;

- X, Y : Coordinates of the starting point.
- I : Interval (d) between holes. Change of sign for argument I causes a centrically symmetric hole arrangement with the starting point as the center.
- J : Angle (θ) of the line. Positive angles refer to counterclockwise measurement.
- K : Number (n) of holes to be machined (from 1 to 9999), inclusive of the starting point.

2. Sample programes

Given below is an example of G81 hole machining with a figure representing the hole positions.

```
N001 G91;
N002 G81 Z-10. R5. L0. F100;
N003 G35 X200. Y100. I100. J30. K5;
N004 G80;
```



3. Notes

- In the use of G-code series T, use the appropriate axis addresses to designate the axis position in an incremental value. As for G-code series M, give a G90 or G91 command as required to designate the position in absolute or incremental values.
- Omission of argument K or setting “K0” will result in a programming error. A setting of K with five or more digits will lead to the lowest four digits being used.
- In a block with G235 any words with addresses other than G, L, N, X, Y, I, J, K, F, M, S, T and B will simply be ignored.
- Giving a G-code of group 00 in the same block with G235 will cause an exclusive execution of either code which is given later.
- In a block with G235 a G22 or G23 command will simply be ignored without affecting the execution of the G235 command.

14-4-4 Holes on an arc: G236 [Series M: G36]

As shown in the format below, a command of G236 determines a circle of radius “r” around the center designated by X and Y. On the circumference “n” holes will be machined, starting from the point of the central angle “θ”, at angular intervals of “Δθ”, according to the current mode of hole machining.

The movement in the XY-plane from hole to hole occurs rapidly (under G00). The argument data of the G236 command will be cleared upon completion of its execution.

1. **Programming format**

G236 Xx Yy Ir Jθ PΔθ Kn;

X, Y : Coordinates of the center of the arc.

I : Radius (r) of the arc. Always given in a positive value.

J : Central angle (θ) of the first hole. Positive central angles refer to counterclockwise measurement.

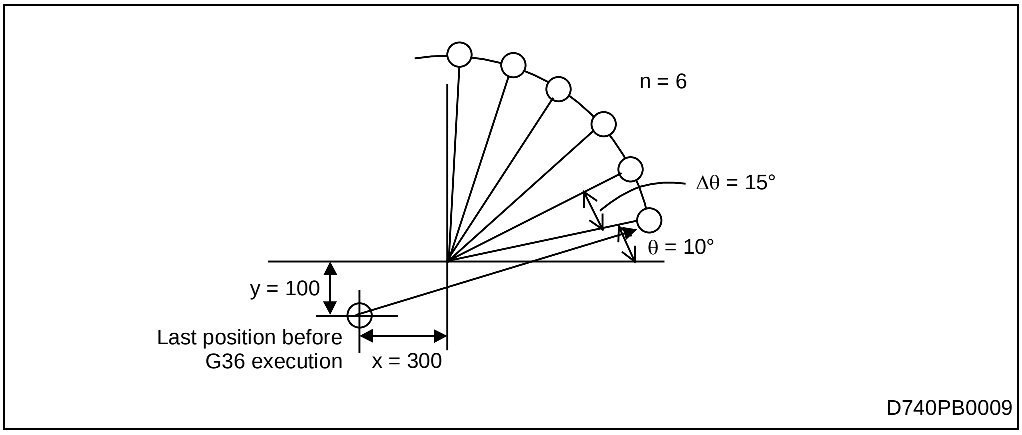
P : Angular interval (Δθ) between holes. The algebraic sign of argument P refers to the rotational direction of the sequential machining of “n” holes. Set a positive and a negative value respectively for counterclockwise and clockwise rotation.

K : Number (n) of holes to be machined (from 1 to 9999).

2. **Sample programes**

Given below is an example of G81 hole machining with a figure representing the hole positions.

```
N001 G91;
N002 G81 Z–10. R5. F100;
N003 G36 X300. Y100. I300. J10. P15. K6;
N004 G80;
```



3. **Notes**

- In the use of G-code series T, use the appropriate axis addresses to designate the axis position in an incremental value. As for G-code series M, give a G90 or G91 command as required to designate the position in absolute or incremental values.

14-4-5 Holes on a grid: G237.1 [Series M: G37.1]

As shown in the format below, a command of G237.1 determines a grid pattern of [Δx]∗[nx] by [Δy]∗[ny] with the point designated by X and Y as starting point. On the grid points the hole machining designated beforehand by a fixed cycle will be done “nx” in number along the X-axis at intervals of “Δx”, and “ny” in number along the Y-axis at intervals of “Δy”. The main progression of machining occurs in the X-axis direction.

The movement in the XY-plane from hole to hole occurs rapidly (under G00). The argument data of the G237.1 command will be cleared upon completion of its execution.

1. Programming format

G237.1 Xx Yy IΔx Pnx JΔy Kny;

X, Y : Coordinates of the starting point.

I : Hole interval (Δx) on the X-axis. Set a positive and a negative value to arrange holes in respective directions from the starting point on the X-axis.

P : Number (nx) of holes to be arranged on the X-axis (from 1 to 9999).

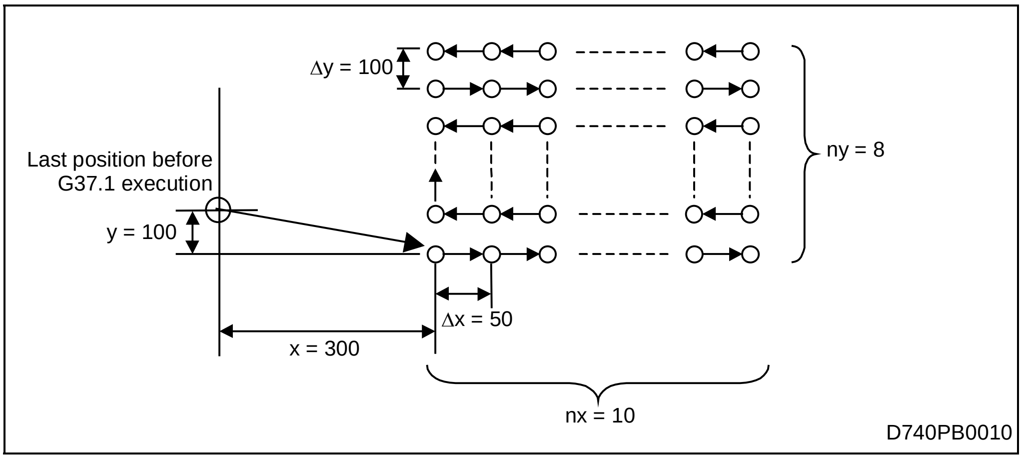
J : Hole interval (Δy) on the Y-axis. Set a positive and a negative value to arrange holes in respective directions from the starting point on the Y-axis.

K : Number (ny) of holes to be arranged on the Y-axis (from 1 to 9999).

2. Sample programs

Given below is an example of G81 hole machining with a figure representing the hole positions.

```
N001 G91;
N002 G81 Z–10. R5. F20;
N003 G37.1 X300. Y-100. I50. P10 J100. K8;
N004 G80;
```



3. Notes

- In the use of G-code series T, use the appropriate axis addresses to designate the axis position in an incremental value. As for G-code series M, give a G90 or G91 command as required to designate the position in absolute or incremental values.
- Omission of argument P or K, or setting “P0” or “K0” will result in a programming error. A setting of K or P with five or more digits will lead to the lowest four digits being used.
- In a block with G237.1 any words with addresses other than G, L, N, X, Y, I, J, K, F, M, S, T and B will simply be ignored.

- Giving a G-code of group 00 in the same block with G237.1 will cause an exclusive execution of either code which is given later.
- In a block with G237.1 a G22 or G23 command will simply be ignored without affecting the execution of the G237.1 command.

14-5 Fixed Cycles (Series M)

14-5-1 Outline

1. Function and purspose

The fixed-cycle functions allow positioning, hole-drilling, boring, tapping, or other machining programs to be executed according to the predetermined job sequence by the commands of a single block. The available job sequences for machining are listed below.
The fixed-cycle function mode is cancelled on reception of G80 or a G-command (G00, G01, G02, G03, G2.1, or G3.1) of group G01. All related types of data are also cleared to zero at the same time.

2. List of fixed cycles

G-Code	Description	Arguments	Notes
G71.1	Chamfering cutter (CW)	[X, Y] Z, Q, R, F [P, D]	
G72.1	Chamfering cutter (CCW)	[X, Y] Z, Q, R, F [P, D]	
G73	High-speed deep-hole drilling	[X, Y] Z, Q, R, F [P, D, K, I, J(B)]	
G74	Reverse tapping	[X, Y] Z, R, F [P, D, J(B), H]	Dwell in seconds
G75	Boring	[X, Y] Z, R, F [Q, P, D, K, I, J(B)]	
G76	Boring	[X, Y] Z, R, F [Q, P, D, J(B)]	
G77	Back spot facing	[X, Y] Z, R, F [Q, P, E, J(B)]	Return to initial point only.
G78	Boring	[X, Y] Z, R, F [Q, P, D, K]	
G79	Boring	[X, Y] Z, R, F [Q, P, D, K, E]	
G81	Spot drilling	[X, Y] Z, R, F	
G82	Drilling	[X, Y] Z, R, F [P, D, I, J(B)]	
G83	Deep-hole drilling	[X, Y] Z, Q, R, F [P, D, K, I, J(B)]	
G84	Tapping	[X, Y] Z, R, F [P, D, J(B), H]	Dwell in seconds
G85	Reaming	[X, Y] Z, R, F [P, D, E]	
G86	Boring	[X, Y] Z, R, F [P]	
G87	Back boring	[X, Y] Z, R, F [Q, P, D, J(B)]	Return to initial point only.
G88	Boring	[X, Y] Z, R, F [P]	
G89	Boring	[X, Y] Z, R, F [P]	

Note 1: The arguments enclosed in brackets ([ ]) can be omitted.

Note 2: Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84**.

Parameter **F84** bit 1 = 1: Argument J-command
= 0: Argument B-command

Return to Library

14-5-2 Fixed-cycle machining data format

1. Setting fixed-cycle machining data

Set fixed-cycle machining data as follows:

G□□X_Y_Z_Q_R_P_D_K_I_J(B)_E_H_F_L_

Hole position data

Repeat times

Hole-machining mode

Hole-machining data

- Hole-machining mode (G-code)
  See the list of the fixed cycles.
- Hole position data (X, Y)
  Set hole positions using incremental or absolute data.
- Hole-machining data

  Z.......... Set the distance from R-point to the hole bottom using incremental data, or set the position of the hole bottom using absolute data.

  Q......... Set this address code using incremental data. (This address code has different uses according to the type of hole-machining mode selected.)

  R .... Set the distance from the initial point of machining to R-point using incremental data, or set the position of R-point using absolute data.

  P .... Set the desired time or the number of spindle revolutions, for dwell at the hole bottom. (Set the overlapping length for the chamfering cutter cycles G71.1 and G72.1.)

  D .... Set this address code using incremental data. (This address code has different uses according to the type of hole-machining mode selected.)

  K .... Set this address code using incremental data. (This address code has different uses according to the type of hole-machining mode selected.)

  I........... Set the feed override distance for the tool to be decelerated during the last cutting operation of drilling with a G73, G82, or G83 command code.

  J(B) ..... For G74 or G84, set the timing of dwell data output; for G75, G76, or G87, set the timing of M3 and M4 output, or; for G73, G82, or G83, set the feed override ratio for deceleration during the last cutting operation.

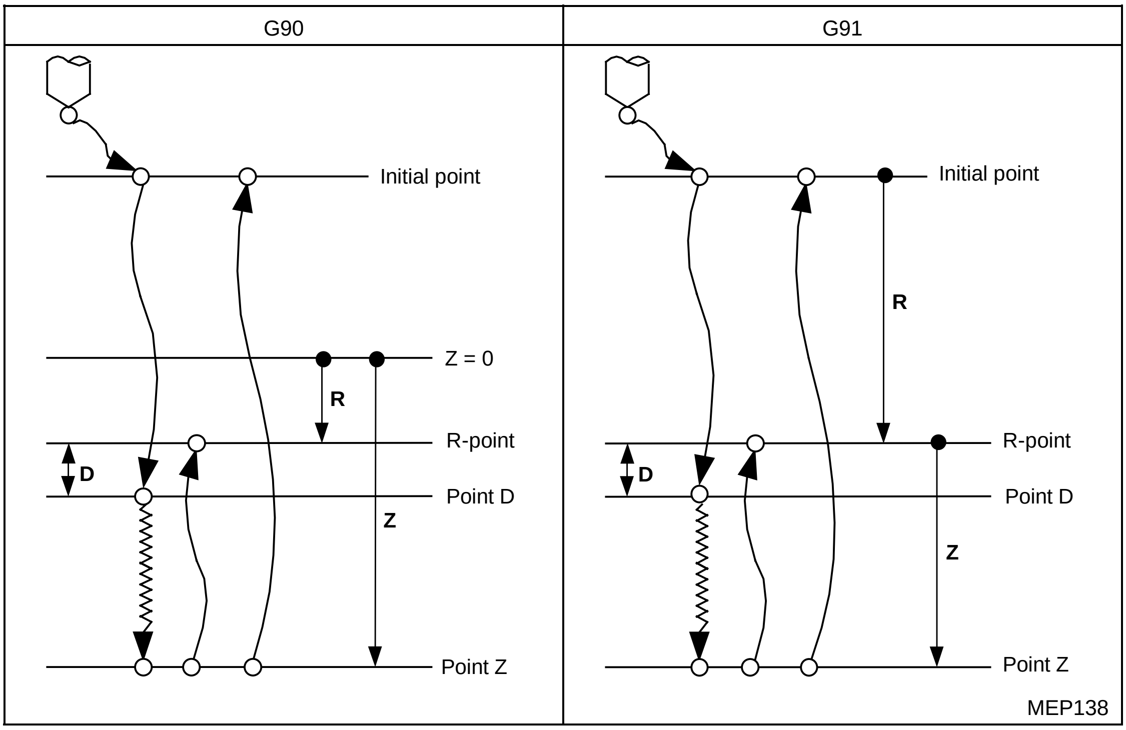
  E .... Set a cutting feed rate (for G77, G79 and G85).

  H .... Select synchronous/asynchronous tapping cycle and set the return speed override during a synchronous tapping cycle.

  F.......... Set a cutting feed rate.

- Repeat times (L)
  If no data is set for L, it will be regarded as equal to 1.
  If L is set equal to 0, hole-machining will not occur; hole-machining data will only be stored into the memory.

- The differences between the G90 mode data setting method and the G91 mode data setting method are shown in the diagram below.



: Signifies signed distance data that begins at ●.
: Signifies unsigned distance data.

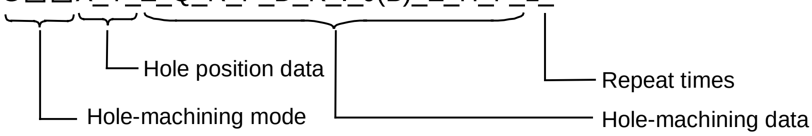
Note 1: The initial point refers to the Z-axis position existing at the moment of the fixed-cycle mode selection.

Note 2: Point D is that at which positioning from R-point can be done further at a rapid feed rate.

2. Programming format

As shown below, the fixed-cycle command consists of a hole-machining mode section, a hole position data section, a hole-machining data section, and a repeat instruction section.

G□□X_Y_Z_Q_R_P_D_K_I_J(B)_E_H_F_L_



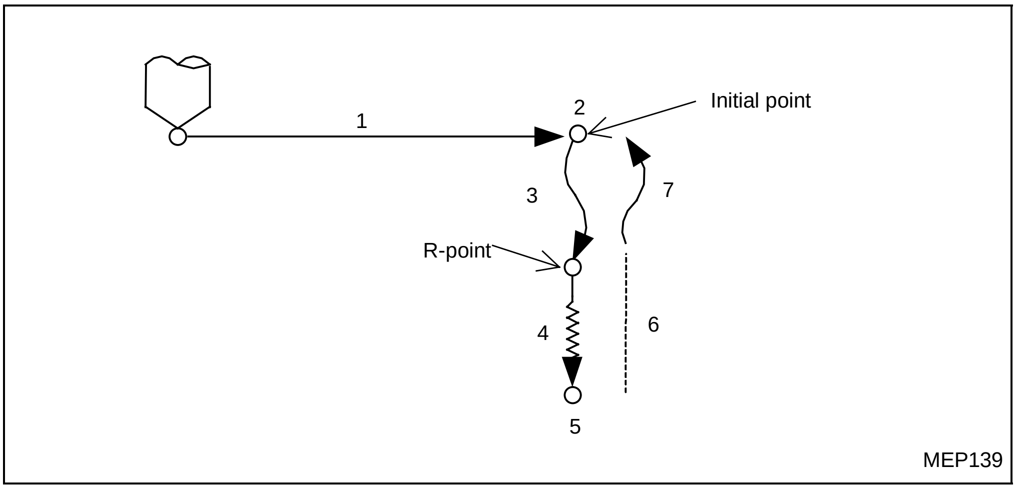
3. Detailed description

1. The hole-machining mode refers to a fixed-cycle mode used for drilling, counterboring, tapping, boring, or other machining operations. Hole position data denotes X- and Y-axis positioning data. Hole-machining data denotes actual machining data. Hole position data and repeat times are non-modal, whereas hole-machining data are modal.

2. If M00 or M01 is set either in the same block as a fixed-cycle command or during the fixed-cycle mode, then the fixed-cycle command will be ignored and then after positioning, M00 or M01 will be outputted. The fixed-cycle command will be executed if either X, Y, Z, or R is set.

3. During fixed-cycle operation, the machine acts in one of the following seven types of manner:

- Action 1 For positioning on the X-, and Y-axes, the machine acts according to the current G-code of group 01 (G02 and G03 will be regarded as G01).
- Action 2 M19 is sent from the NC unit to the machine at the positioning complete point (initial point) in the G87 mode. After execution of this M-command, the next action will begin. In the single-block operation mode, positioning is followed by block stop.



- Action 3 Positioning to R-point by rapid motion.
- Action 4 Hole-machining by cutting feed.
- Action 5 Depending on the selected fixed-cycle type, spindle stop (M05), spindle reverse rotation (M04), spindle normal rotation (M03), dwell, or tool shift is performed at the hole bottom.
- Action 6 Tool relief to R-point is performed by cutting feed or rapid motion (according to the selected fixed-cycle type).
- Action 7 Return to the initial point is performed by rapid motion.

Whether fixed-cycle mode operation is to be terminated at action 6 or action 7 can be selected with the following G-codes:

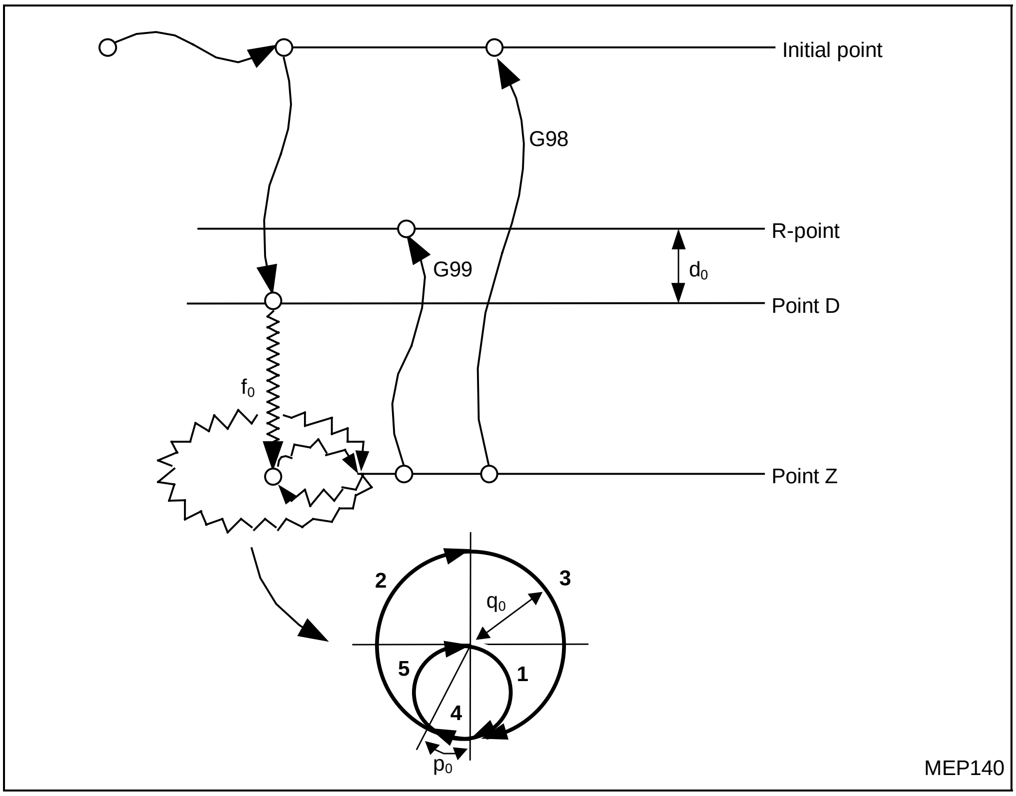
G98: Return to the initial point level

G99: Return to the R-point level

Both commands are modal. Once G98 has been given, for example, the G98 mode remains valid until G99 is given. The G98 mode is the initial state of the NC.
For a block without positioning data, the hole-machining data are only stored into the memory and fixed-cycle operation is not performed.

14-5-3 G71.1 [Chamfering cutter CW] (Series M)

G71.1 [Xx Yy] Rr Zz Qq$_0$ [Pp$_0$ Dd$_0$] Ff$_0$

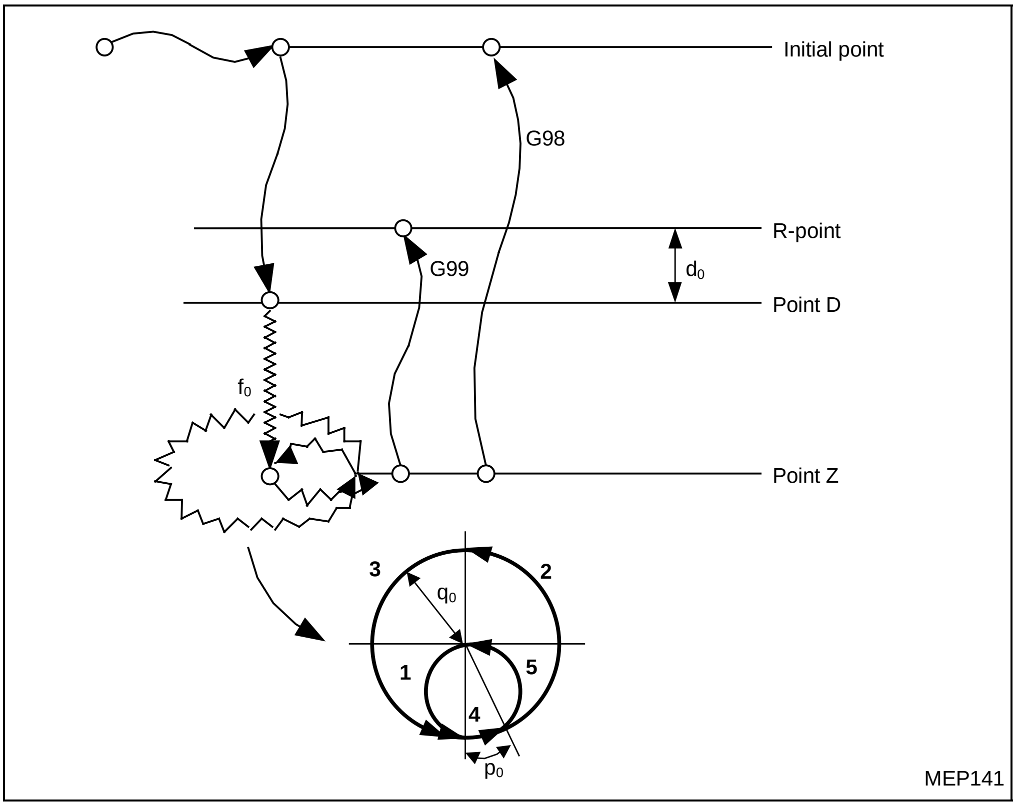


q_0 : Radius
p_0 : Overlapping length (in arc)
d_0 : Distance from R-point
f_0 : Feed rate

- X, Y, P, and/or D can be omitted.
- Omission of Q or setting “Q0” results in a program error.

14-5-4 G72.1 [Chamfering cutter CCW] (Series M)

G72.1 [Xx Yy] Rr Zz Qq_0 [Pp_0 Dd_0] Ff_0

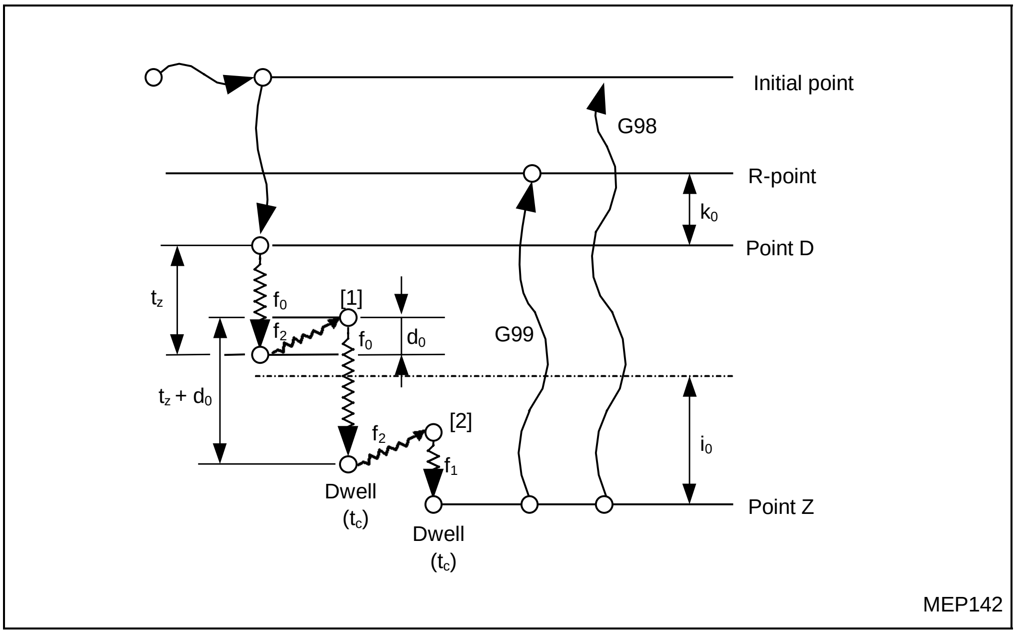


q_0 : Radius	d_0 : Distance from R-point
p_0 : Overlapping length (in arc)	f_0 : Feed rate

- X, Y, P, and/or D can be omitted.
- Omission of Q or setting “Q0” results in a program error.

14-5-5 G73 [High-speed deep-hole drilling] (Series M)

G73 [Xx Yy] Rr Zz Qt$_z$ [Pt$_c$] Ff$_0$ [Dd$_0$ Kk$_0$ Ii$_0$ Jj$_0$(Bb$_0$)]



t_z	: Depth of cut per pass	j_0	: Feed override ratio (%)
t_c	: Dwell (in time or No. of revolutions)	(b_0)	
d_0	: Return distance	f_0	: Feed rate
k_0	: Distance from R-point to the starting point of cutting feed	f_1	: Feed overridden $f_1 = f_0 \times j_0(b_0)/100$
i_0	: Feed override distance	f_2	: Return speed (fixed) Max. speed: 9999 mm/min (for mm-spec.) 999.9 in./min (for in.-spec.)

- The feed rate will remain unchanged if either **I** or J(B) is omitted.
- X, Y, P, D, K, **I**, and/or J(B) can be omitted.
  If D is omitted or set to 0, the machine operates according to the value of parameter **F12**.
- The alarm **809 ILLEGAL NUMBER INPUT** will occur if Q is set to 0.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84**.

  Parameter **F84** bit 1 = 1: Argument J-command
  = 0: Argument B-command

  Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

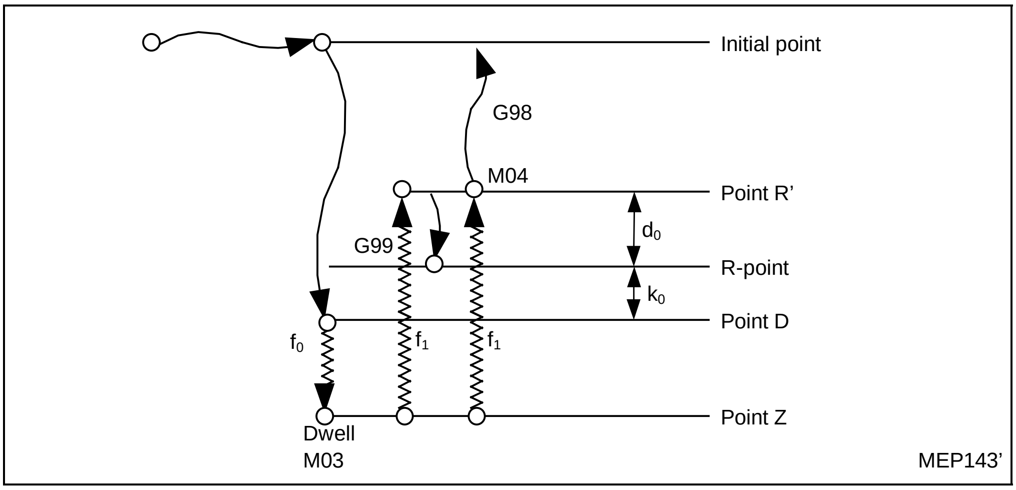
- The feed rate is f_1 only if the starting point of a cutting pass is within the range of i_0.

  Example: In the diagram shown above, during the second cutting operation, since pecking return point [1] falls outside the range of feed override distance i_0, feeding does not decelerate and cutting is performed at feed rate f_0; during the third cutting operation, since pecking return point [2] falls within the range of i_0, feeding decelerates and cutting is performed at feed rate f_1.

Return to Library

14-5-6 G74 [Reverse tapping] (Series M)

G74 [Xx Yy] Rr Zz [Pt$_c$] Ff$_0$ [Jj$_0$(Bb$_0$) Dd$_0$ Hh$_0$ Kk$_0$]



t_c : Dwell (always in time)
f_0 : Feed rate
j_0 (b_0) : 1···M03 after dwell at hole bottom
2···M03 before dwell at hole bottom
4···M04 after dwell at R-point
d_0 : Distance from R-point (Tap lifting distance)

h_0 : Flag for synchronous/asynchronous tapping and the return speed override (%) for synchronous tapping
$h_0 = 0$ Asynchronous tapping
$h_0 > 0$ Synchronous tapping
k_0 : Distance from R-point

- X, Y, P, J(B), D, H, and/or K can be omitted.
  If, however, J(B) is omitted or set to 0, the setting of J(B) will be regarded as 2.
  If H is omitted, the selection between synchronous/asynchronous tapping is performed by the bit 6 of parameter **F94**.
- For synchronous tapping, see Subsection 14-5-21.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84.**

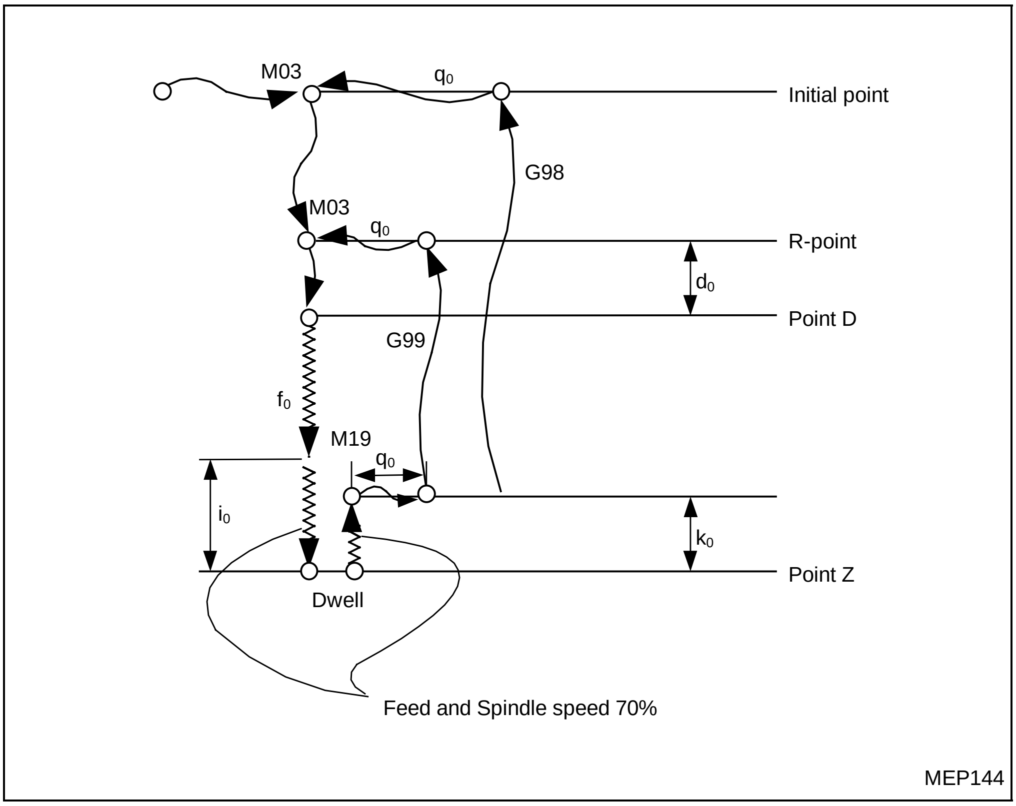
  Parameter **F84** bit 1 = 1: Argument J-command
  = 0: Argument B-command

Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

Return to Library

14-5-7 G75 [Boring] (Series M)

G75 [Xx Yy] Rr Zz [Pt_c Qq_0] Ff_0 [Dd_0 Jj_0(Bb_0) Kk_0 Ii_0]



t_c : Dwell (in time or No. of revolutions)
q_0 : Amount of relief on the XY-plane (Direction determined by bits 3 & 4 of **I14**)
f_0 : Feed rate
d_0 : Distance from R-point
j_0 : 0 or omitted······· M03 after machining
(b_0) Value except 0···· M04 after machining
k_0 : Distance from point Z
i_0 : Distance from point Z

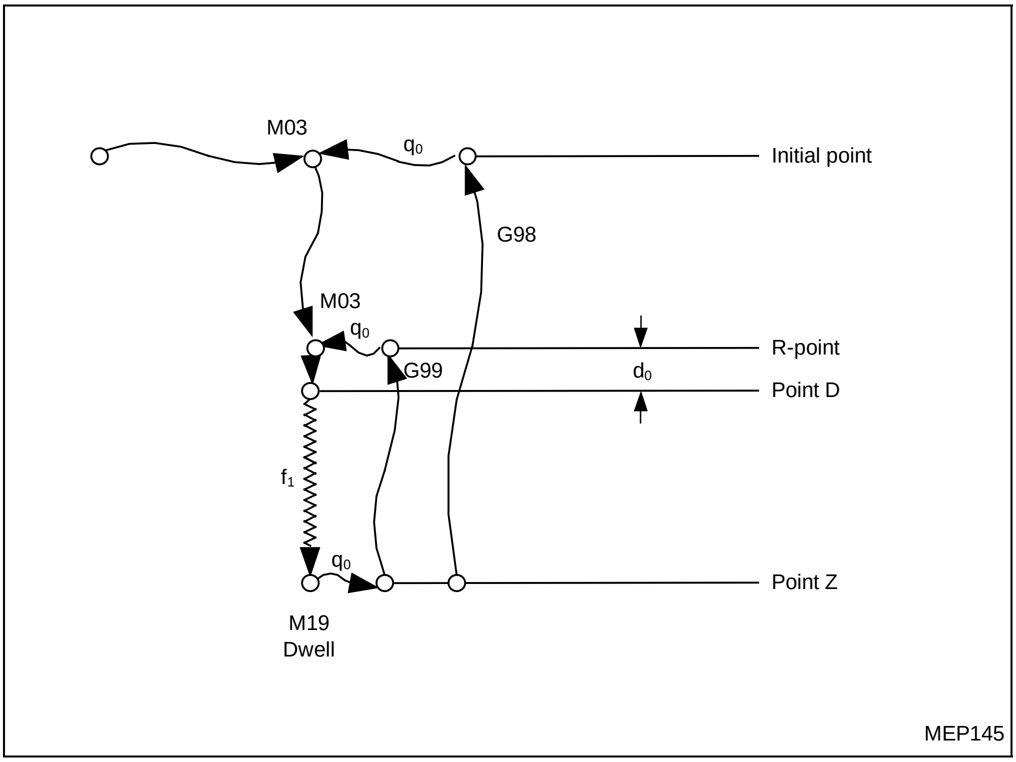
- X, Y, P, Q, D, J(B), K, and/or I can be omitted.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84.**

Parameter **F84** bit 1 = 1: Argument J-command
= 0: Argument B-command

Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

14-5-8 G76 [Boring] (Series M)

G76 [Xx Yy] Rr Zz [Pt$_c$ Qq$_0$] Ff$_1$ [Dd$_0$ Jj$_0$(Bb$_0$)]



t_c : Dwell (in time or No. of revolutions)
q_0 : Amount of relief on the XY-plane
(Direction determined by bits 3 & 4 of **I14**)

f_1 : Feed rate
j_0 : 0 or omitted ······· M03 after machining
(b_0) Value except 0 ···· M04 after machining

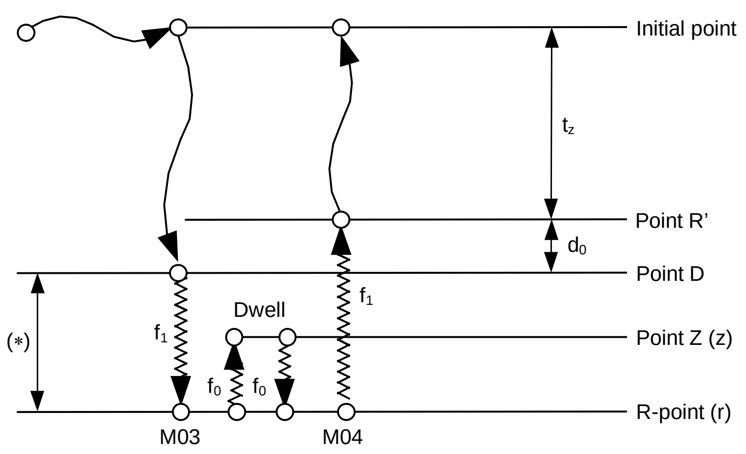
- X, Y, P, Q, D, and/or J(B) can be omitted.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84.**

Parameter **F84** bit 1 = 1: Argument J-command
= 0: Argument B-command

Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

14-5-9 G77 [Back spot facing] (Series M)

G77 [Xx Yy] Rr Zz [Pt_c Qt_z] Ff_0 [Ef_1 Jj_0(Bb_0) Dd_0]



MEP146'

t_c	: Dwell (in time or No. of revolutions)	$j_0(b_0)$	: Output order of M03 and M04 at hole bottom.
t_z	: Distance from the initial point		0: M03, then M04 (for normal spindle rotation)
f_0	: Feed rate 0		1: M04, then M03 (for reverse spindle rotation)
f_1	: Feed rate 1	d_0	: Distance from point R'

- Normally, asynchronous feed (G94) is used for the pass marked with (∗). If $f_1 = 0$, or if f_1 is omitted, however, synchronous feed (G95) is used (feed rate = 0.5 mm/rev).
- X, Y, P, Q, E, J (B), and/or D can be omitted.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84.**

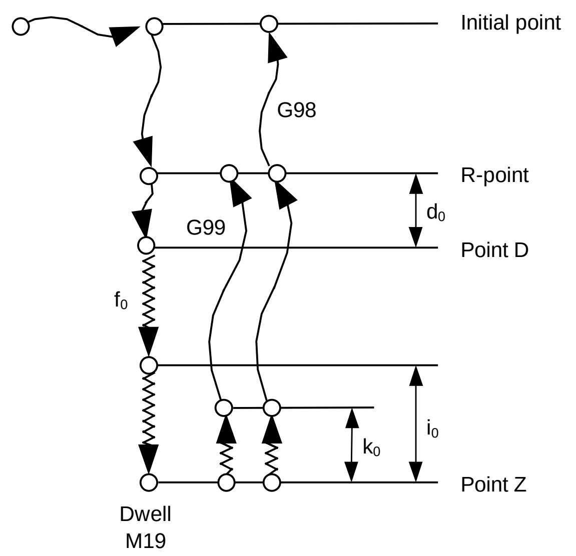
  Parameter **F84** bit 1 = 1: Argument J-command
  = 0: Argument B-command

Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

- In G91 (incremental data input) mode, the direction of hole machining is automatically determined according to the sign of Z data (the sign of data at address R will be ignored).

14-5-10 G78 [Boring] (Series M)

G78 [Xx Yy] Rr Zz [Pt_c] Ff_0 [Dd_0 Kk_0 Qi_0]



MEP147

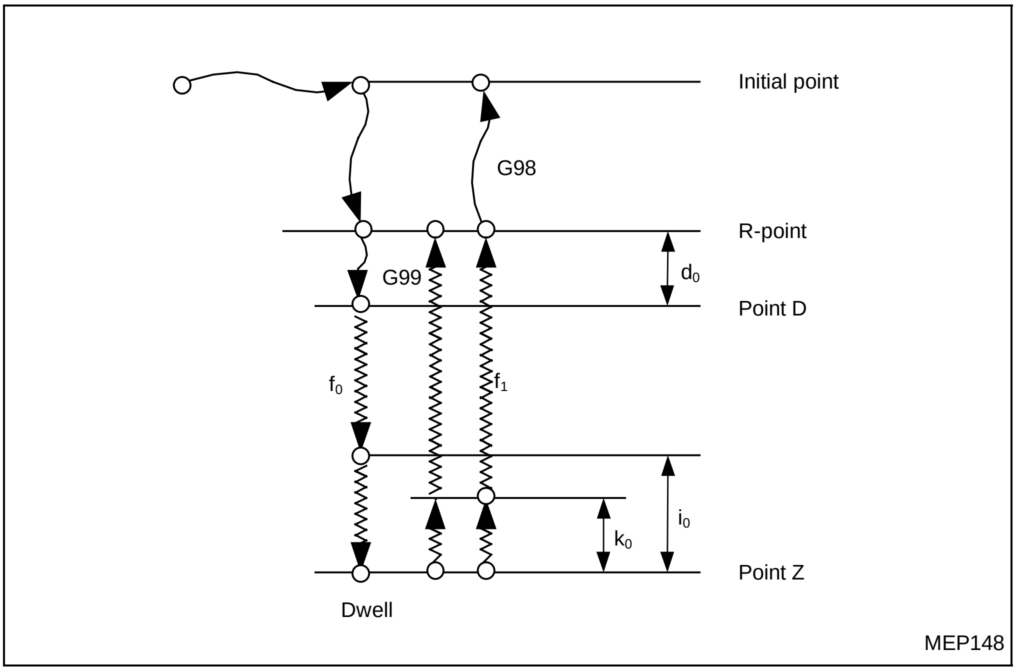
t_c :	Dwell (in time or No. of revolutions)	k_0 :	Distance from point Z
d_0 :	Distance from R-point	i_0 :	Distance from point Z

- X, Y, P, D, K, and/or Q can be omitted.

Return to Library

14-5-11 G79 [Boring] (Series M)

G79 [Xx Yy] Rr Zz [Pt_c] Ff_0 [Dd_0 Kk_0 Qi_0 Ef_1]

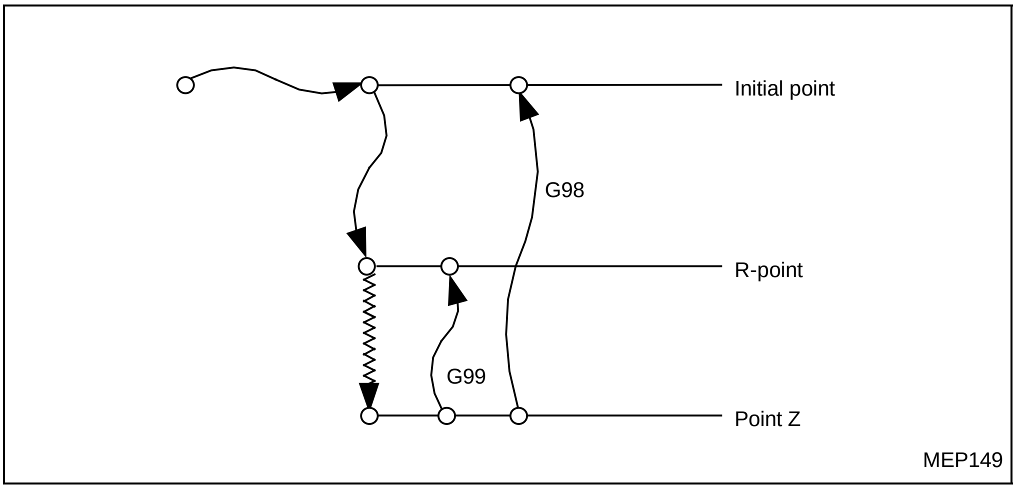


t_c :	Dwell (in time or No. of revolutions)	k_0 :	Distance from point Z
f_0 :	Feed rate 0	i_0 :	Distance from point Z
d_0 :	Distance from R-point	f_1 :	Feed rate 1

- Asynchronous feed is used for f_1.
  If, however, f_1 is set equal to 0 or is not set, then the tool is fed at the setting of f_0.
- X, Y, P, D, K, Q, and/or E can be omitted.

14-5-12 G81 [Spot drilling] (Series M)

G81 [Xx Yy] Rr Zz

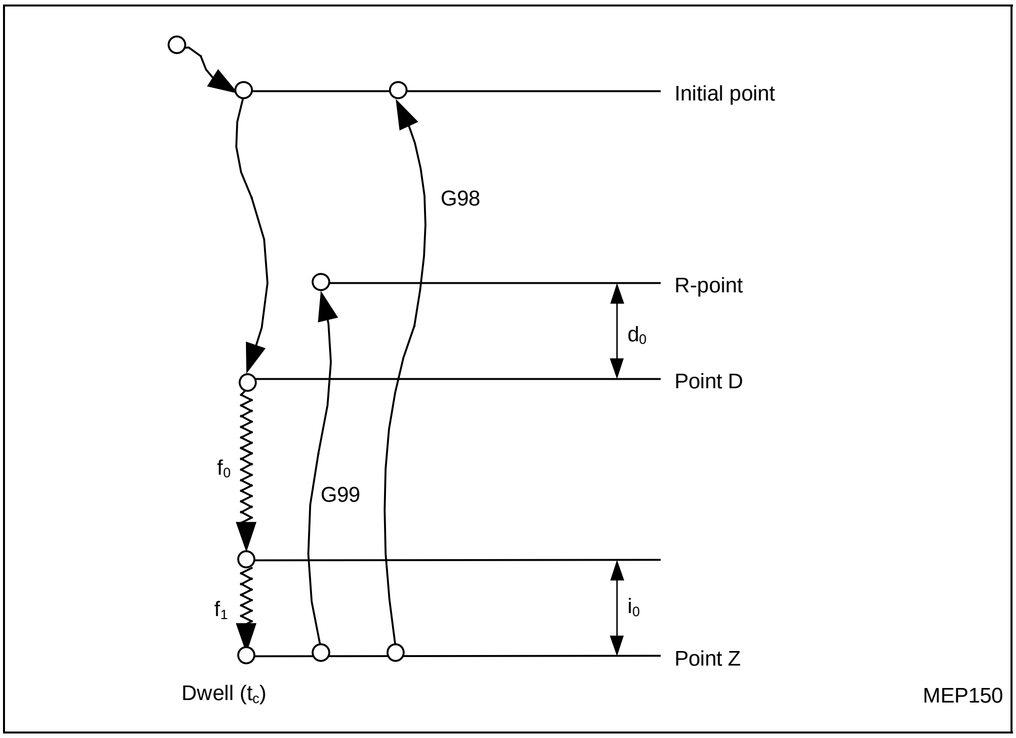


- X and/or Y can be omitted.

Return to Library

14-5-13 G82 [Drilling] (Series M)

G82 [Xx Yy] Rr Zz [Pt$_c$] Ff$_0$ [Dd$_0$ Ii$_0$ Jj$_0$(Bb$_0$)]



t_c : Dwell (in time or No. of revolutions)
d_0 : Distance from R-point to the starting point of cutting feed
i_0 : Feed override distance
j_0 : Feed override ratio (%)
(b_0)
f_0 : Feed rate
f_1 : Feed overridden $f_1 = f_0 \times j_0(b_0)/100$

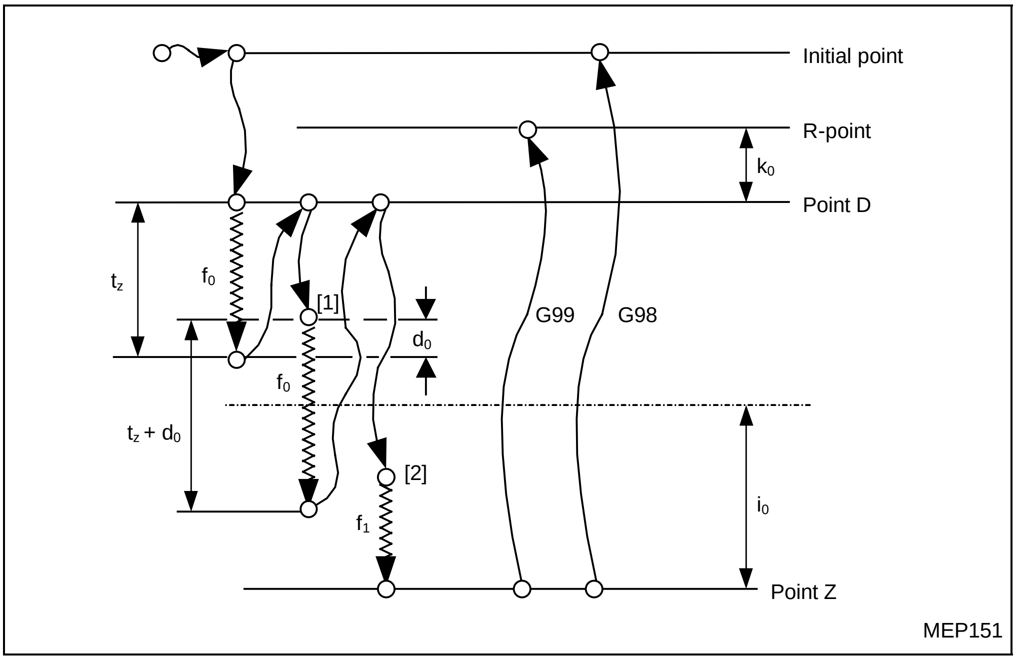
- The feed rate will remain unchanged if either **I** or J(B) is omitted.
- X, Y, P, D, **I**, and/or J(B) can be omitted.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84.**

Parameter **F84** bit 1 = 1 : Argument J-command
= 0 : Argument B-command

Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

14-5-14 G83 [Deep-hole drilling] (Series M)

G83 [Xx Yy] Rr Zz Qt$_z$ Ff$_0$ [Dd$_0$ Kk$_0$ Ii$_0$ Jj$_0$(Bb$_0$)]



t_z : Depth of cut per pass
d_0 : Rapid motion stopping allowance
k_0 : Distance from R-point to the starting point of cutting feed
i_0 : Feed override distance
j_0 : Feed override ratio (%)
(b_0)
f_0 : Feed rate
f_1 : Feed overridden $f_1 = f_0 \times j_0(b_0)/100$

- The feed rate will remain unchanged if either I or J(B) is omitted.
- X, Y, P, D, K, I, and/or J(B) can be omitted.
  If D is omitted or set to 0, the machine will operate according to the value of parameter **F13**.
- The alarm **809 ILLEGAL NUMBER INPUT** will occur if Q is set to 0.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84.**

  Parameter **F84** bit 1 = 1: Argument J-command
  = 0: Argument B-command

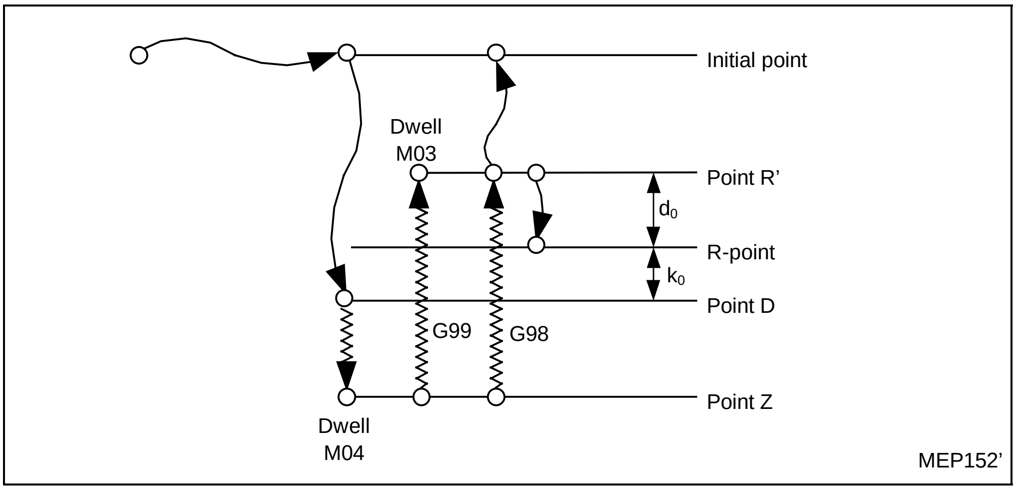
  Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

- The feed rate is f_1 only if the starting point of a cutting pass is within the range of i_0.

  Example: In the diagram shown above, during the second cutting operation, since rapid feed positioning point [1] falls outside the range of feed override distance i_0, feeding does not decelerate and cutting is performed at feed rate f_0; during the third cutting operation, since rapid feed positioning point [2] falls within the range of i_0, feeding decelerates and cutting is performed at feed rate f_1.

14-5-15 G84 [Tapping] (Series M)

G84 [Xx Yy] Rr Zz [Pt$_c$] Ff$_0$ [Jj$_0$(Bb$_0$) Dd$_0$ Hh$_0$ Kk$_0$]



t_c : Dwell (always in time)
f_0 : Feed rate
j_0 (b_0) : 1···M04 after dwell at hole bottom
2···M04 before dwell at hole bottom
4···M03 after dwell at R-point
d_0 : Distance from R-point (Tap lifting distance)

h_0 : Flag for synchronous/asynchronous tapping and the return speed override (%) for synchronous tapping
$h_0 = 0$ Asynchronous tapping
$h_0 > 0$ Synchronous tapping
k_0 : Distance from R-point

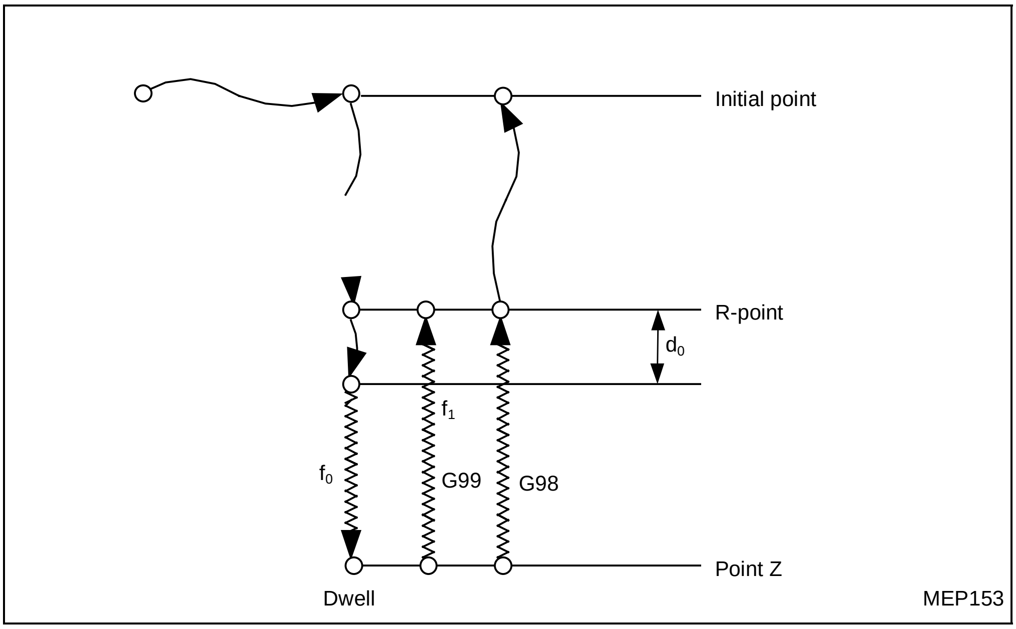
- X, Y, P, J(B), D, H, and/or K can be omitted.
  If, however, J(B) is omitted or set to 0, the setting of J(B) will be regarded as 2.
  If H is omitted, the selection between synchronous/asynchronous tapping is performed by the bit 6 of parameter **F94**.
- For synchronous tapping, see Subsection 14-5-21.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84.**

Parameter **F84** bit 1 = 1 : Argument J-command
= 0 : Argument B-command

Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

14-5-16 G85 [Reaming] (Series M)

G85 [Xx Yy] Rr Zz [Pt_z] Ff_0 [Ef_1 Dd_0]

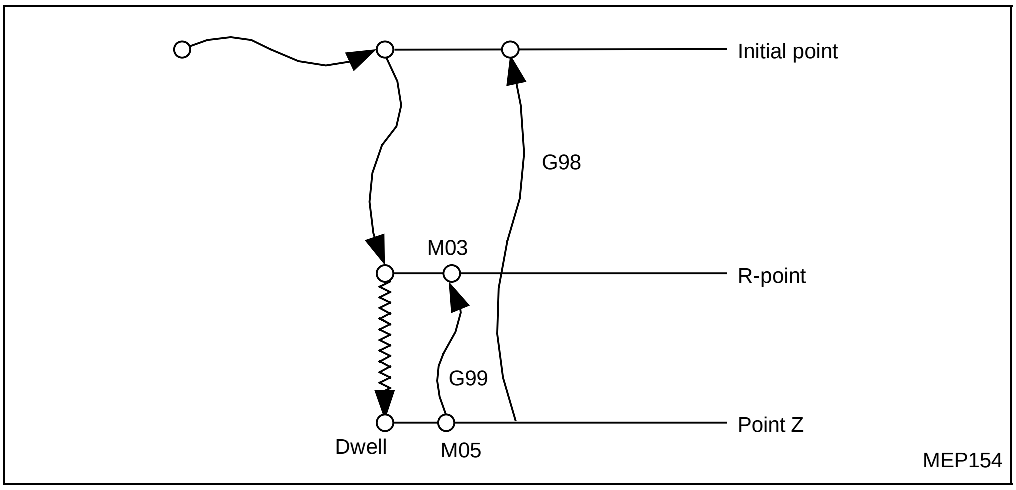


t_z : Dwell (in time or No. of revolutions)
f_1 : Feed rate 1
f_0 : Feed rate 0
d_0 : Distance from R-point

- Asynchronous feed is used for f_1.
  If, however, f_1 is set equal to 0 or is not set, then the tool is fed at the setting of f_0.
- X, Y, P, E, and/or D can be omitted.

14-5-17 G86 [Boring] (Series M)

G86 [Xx Yy] Rr Zz [Pt_c]



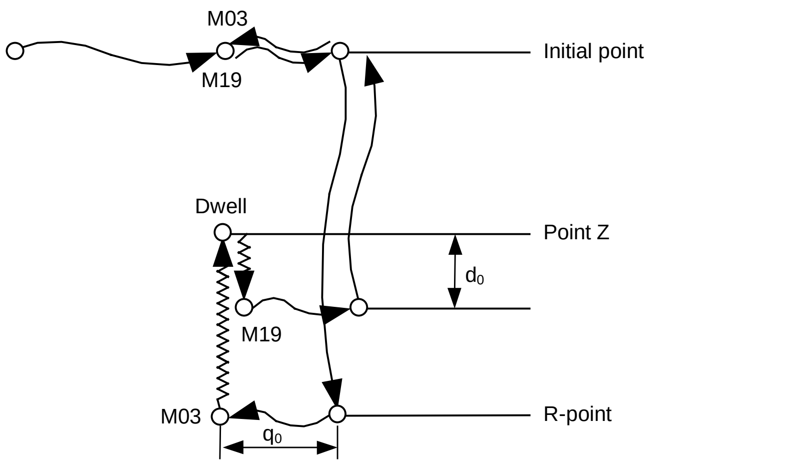
t_c : Dwell (in time or No. of revolutions)

- X, Y, and/or P can be omitted.

Return to Library

14-5-18 G87 [Back boring] (Series M)

G87 [Xx Yy] Rr Zz [Pt_c Qq_0] Ff_0 [Dd_0 Jj_0(Bb_0)]



t_c : Dwell (in time or No. of revolutions)
q_0 : Amount of relief on the XY-plane
(Direction determined by bits 3 & 4 of **I14**)
f_0 : Feed rate
d_0 : Distance from point Z
j_0 : 0 or omitted······· M03 at R-point
(b_0) Value except 0···· M04 at R-point

- X, Y, P, Q, D, and/or J(B) can be omitted.
- Initial-point return is always used for G87 (even if the current modal is of G99).
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84.**

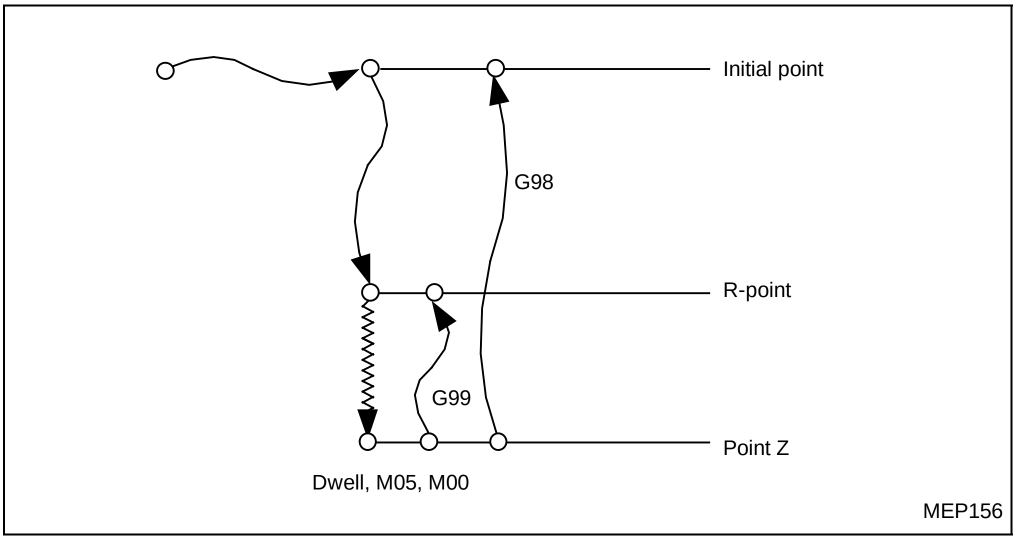
Parameter **F84** bit 1 = 1: Argument J-command
= 0: Argument B-command

Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

- In G91 (incremental data input) mode, the direction of hole machining is automatically determined according to the sign of Z data (the sign of data at address R will be ignored).

14-5-19 G88 [Boring] (Series M)

G88 [Xx Yy] Rr Zz [Pt$_c$]

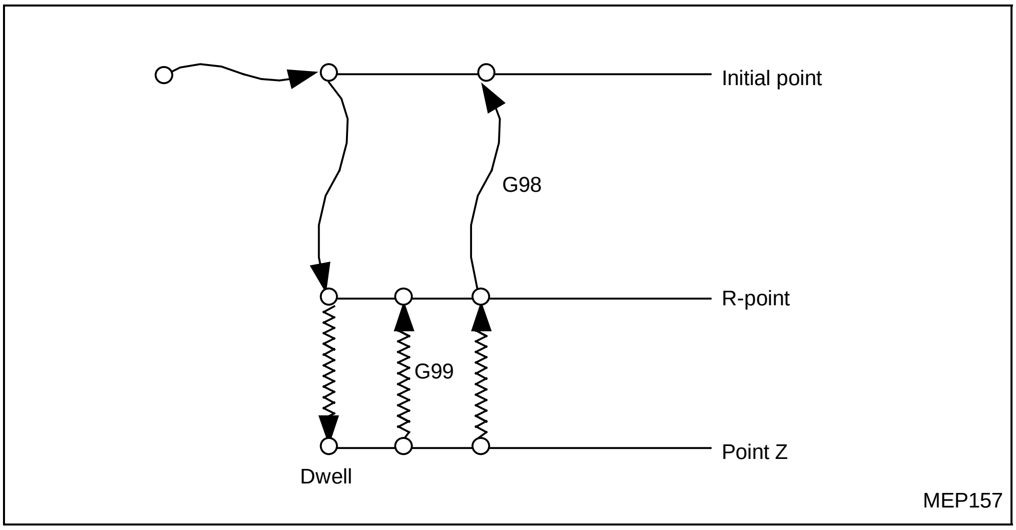


t$_c$: Dwell (in time or No. of revolutions)

- X, Y, and/or P can be omitted.
- At the hole bottom, M05 and M00 are outputted.

14-5-20 G89 [Boring] (Series M)

G89 [Xx Yy] Rr Zz [Pt$_c$]



t$_c$: Dwell (in time or No. of revolutions)

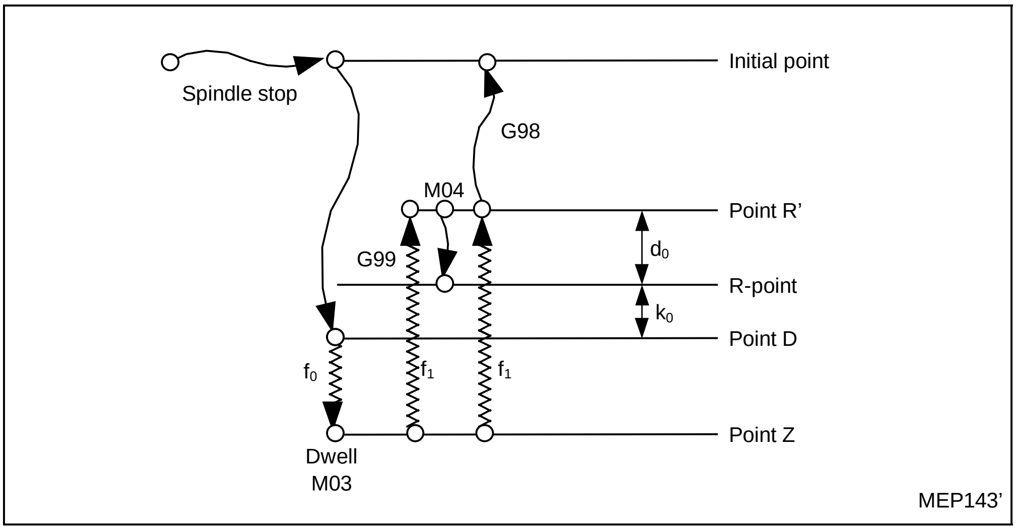
- X, Y, and/or P can be omitted.

14-5-21 Synchronous tapping [Option] (Series M)

In an EIA/ISO program, synchronous tapping can be selected by additionally setting data at the address H in the tapping cycle block of G74 or G84. Address H is used to select a synchronous/asynchronous tapping and to designate the override of return speed. Special preparatory functions G84.2 and G84.3 are also provided for both types of synchronous tapping.

1. G74 [Reverse tapping]

G74 [Xx Yy] Rr Zz [Pt$_c$] Ff$_0$ [Jj$_0$(Bb$_0$) Dd$_0$ Hh$_0$ Kk$_0$]



t$_c$: Dwell (always in time)
f$_0$: Feed rate
(Set the pitch for synchronous tapping)
j$_0$: 1···M03 after dwell at hole bottom
(b$_0$) 2···M03 before dwell at hole bottom
4···M04 after dwell at R-point

d$_0$: Distance from R-point (Tap lifting distance)
h$_0$: Return speed override (%)
h$_0$ = 0 ..... Asynchronous tapping
h$_0$ ≥ 1 .... Synchronous tapping
k$_0$: Distance from R-point

- X, Y, P, J(B), D, H, and/or K can be omitted.
  If, however, J(B) is omitted or set to 0, the setting of J(B) will be regarded as 2.
  If H is omitted, the selection between synchronous/asynchronous tapping is performed by the bit 6 of parameter **F94**.
- H is used to select whether synchronous tapping cycle operation or asynchronous tapping cycle operation is to be performed using a machine capable of synchronous tapping. This code is also used to override the return speed for synchronous tapping cycle operation. H becomes invalid for a machine not capable of synchronous tapping, or if your machine has synchronous tapping function but bit 6 of parameter **F94** is not set to 1.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84.**

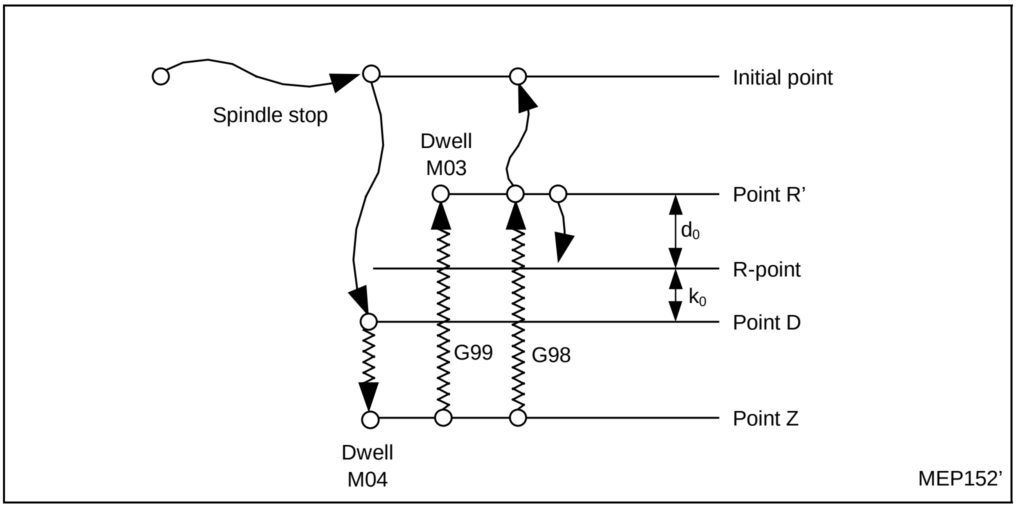
  Parameter **F84** bit 1 = 1: Argument J-command
  = 0: Argument B-command

  Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.
- During gear selection for tapping, due consideration must be given to ensure the minimum spindle acceleration/deceleration time. Refer to the machine-operating manual.

Return to Library

2. G84 [Normal tapping]

G84 [Xx Yy] Rr Zz [Pt$_c$] Ff$_0$ [Jj$_0$(Bb$_0$) Dd$_0$ Hh$_0$ Kk$_0$]



t_c : Dwell (always in time)
f_0 : Feed rate
(Set the pitch for synchronous tapping)
j_0 : 1···M04 after dwell at hole bottom
(b_0) 2···M04 before dwell at hole bottom
4···M03 after dwell at R-point

d_0 : Distance from R-point (Tap lifting distance)
h_0 : Return speed override (%)
$h_0 = 0$ ..... Asynchronous tapping
$h_0 \geq 1$ .... Synchronous tapping
k_0 : Distance from R-point

- X, Y, P, J(B), D, H, and/or K can be omitted.
  If, however, J(B) is omitted or set to 0, the setting of J(B) will be regarded as 2.
  If H is omitted, the selection between synchronous/asynchronous tapping is performed by the bit 6 of parameter **F94**.
- H is used to select whether synchronous tapping cycle operation or asynchronous tapping cycle operation is to be performed using a machine capable of synchronous tapping. This code is also used to override the return speed for synchronous tapping cycle operation. H becomes invalid for a machine not capable of synchronous tapping, or if your machine has synchronous tapping function but bit 6 of parameter **F94** is not set to 1.
- Whether argument J or B is to be used depends on the value that has been set in bit 1 of parameter **F84.**

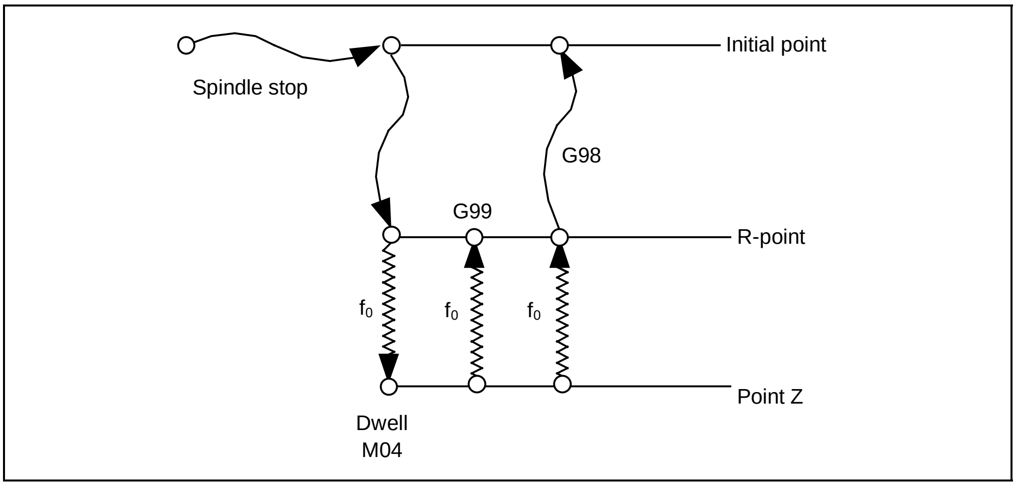
  Parameter **F84** bit 1 = 1: Argument J-command
  = 0: Argument B-command

  Note: For a horizontal machining center, if the value of bit 1 of parameter **F84** is 1 (argument J-command), setting a B-command will cause the table to rotate. Be careful in that case to ensure no interference between the workpiece and the tool.

- During gear selection for tapping, due consideration must be given to ensure the minimum spindle acceleration/deceleration time. Refer to the machine-operating manual.

3. G84.2 [Normal tapping]

G84.2 [Xx Yy] Rr Zz [Pt_c] Ff_0

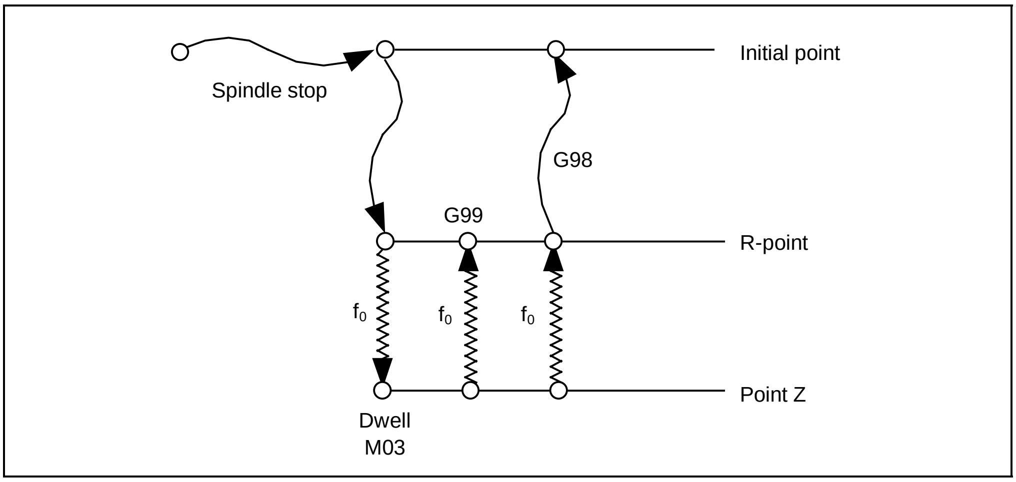


t_c : Dwell (in time) at point Z and upon return to R-point
f_0 : Feed rate (in pitch)

- X, Y, and/or P can be omitted.
- G84.2 and G84.3 always performs a synchronous tapping, irrespective of the setting in bit 6 of parameter **F94**.
- Designation of G84.2 or G84.3 without the corresponding option would cause the alarm No. **952 NO SYNCHRONIZED TAP OPTION**.
- During gear selection for tapping, due consideration must be given to ensure the minimum spindle acceleration/deceleration time. Refer to the machine-operating manual.
- The value of parameter **K90** is always referred to as the return speed override (%).

4. G84.3 [Reverse tapping]

G84.3 [Xx Yy] Rr Zz [Pt$_c$] Ff$_0$



t_c : Dwell (in time) at point Z and upon return to R-point
f_0 : Feed rate (in pitch)

- X, Y, and/or P can be omitted.
- G84.2 and G84.3 always performs a synchronous tapping, irrespective of the setting in bit 6 of parameter **F94**.
- Designation of G84.2 or G84.3 without the corresponding option would cause the alarm No. **952 NO SYNCHRONIZED TAP OPTION**.
- During gear selection for tapping, due consideration must be given to ensure the minimum spindle acceleration/deceleration time. Refer to the machine-operating manual.
- The value of parameter **K90** is always referred to as the return speed override (%).

14-6 Initial Point and R-Point Level Return: G98 and G99 (Series M)

1. Function and purpose

Commands G98 or G99 can be used to select whether the return level of the final sequence during fixed-cycle operation is to be set at R-point or at the initial point of machining.

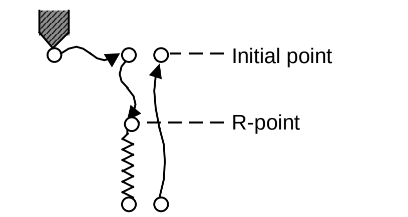
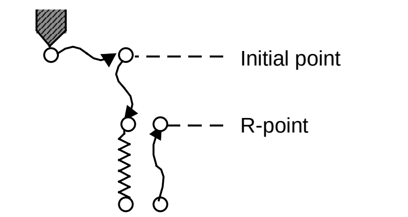
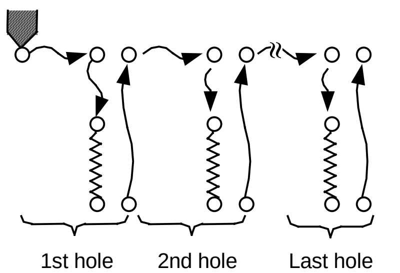
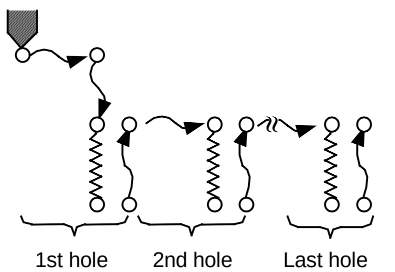
2. Programming format

G98: Initial point level return

G99: R-point level return

3. Detailed description

The following represents the relationship between the G98/G99 mode and repeat times:

Number of holes	Sample program	G98 (At power-on or after cancellation using M02, M30, or RESET key)	G99
Only one	`G81 X100. Y100. Z–50. R25. F1000`	 Return to initial point level.	 Return to R-point level.
Two or more	`G81 X100. Y100. Z–50. R25. L5 F1000`	 Always return to initial point.	 MEP158

Return to Library

14-7 Scaling ON/OFF: G51/G50 (Series M)

1. Function and purpose

The shape specified in a machining program can be enlarged or reduced in size using scaling command G51. The range of scaling (enlargement/reduction) factors is from 0.000001 to 99.999999.
Use command G51 to specify a scaling axis, the center of scaling, and a scaling factor.
Use command G50 to specify scaling cancellation.

2. Programming format

G51 Xx Yy Zz Pp	Scaling on (specify a scaling axis, the center of scaling (incremental/absolute), and a scaling factor)
G50	Scaling cancel

3. Detailed description

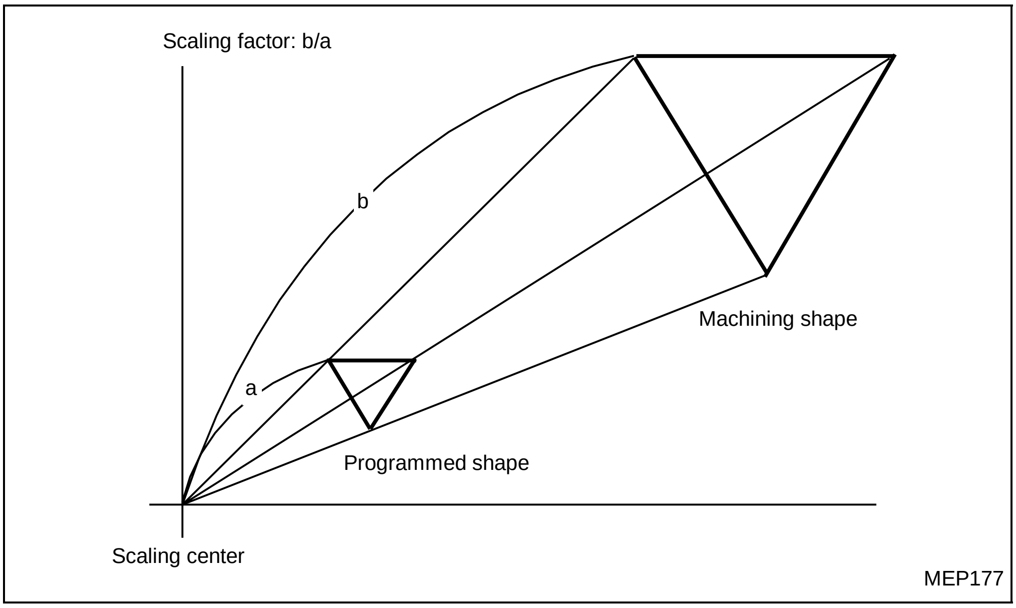
A. Specifying a scaling axis

The scaling mode is set automatically by setting G51. Command G51 does not move any axis; it only specifies a scaling axis, the center of scaling, and a scaling factor.
Scaling becomes valid only for the axis to which the center of scaling has been specified.

<u>Center of scaling</u>
The center of scaling must be specified with the axis address according to the absolute or incremental data command mode (G90 or G91). This also applies even when specifying the current position as the center.

<u>Scaling factor</u>
Use address P to specify a scaling factor.

Minimum unit of specification:	0.000001
Specifiable range of factors:	1 to 99999999 or 0.000001 to 99.999999 (times) (Although both are valid, the latter with a decimal point must be preceded by G51.)



The scaling factor set in parameter **F20** will be used if you do not specify any scaling factor in the same block as that of G51. The current setting of this parameter will be used if it is updated during the scaling mode. That is, the parameter setting existing when G51 is set is valid.
Data will be calculated at a scaling factor of 1 if neither the program nor the parameter has a specified scaling factor.

Program errors occur in the following cases:

- If scaling is specified for a machine not capable of scaling (Alarm **872 G51 OPTION NOT FOUND**)
- If a scaling factor exceeding its maximum available value is specified in the same block as that of G51 (Alarm **809 ILLEGAL NUMBER INPUT**) (All scaling factors less than 0.000001 are processed as 1.)

B. Cancellation of scaling

The scaling cancel mode is set automatically by setting G50. Setting this command code offsets any deviation between the program coordinates and the coordinates of the actual machine position. Even for axes that have not been designated in the same block as that of G50, the machine moves through the offset amount specified by scaling.

4. Precautions

1. Scaling does not become valid for tool diameter offsetting, tool length offsetting, or tool position offsetting. Offsets and other corrections are calculated only for the shape existing after scaling.
2. Scaling is valid only for move commands associated with automatic operation (tape, memory, or MDI); it is not valid for manual movement.
3. After-scaling coordinates are displayed as position data.
4. Scaling is performed on the axis for which the center of scaling is specified by G51. In that case, scaling becomes valid for all move commands associated with automatic operation, as well as for the parameter-set return strokes of G73 and G83 and for the shift strokes of G76 and G87.
5. If only one axis of the plane concerned is selected for scaling, circular interpolation is performed with the single scaling on that axis.
6. Scaling will be cancelled if either M02, M30, or M00 (only when M0 contains reset) is issued during the scaling mode. Scaling is also cancelled by an external reset command or any other reset functions during the reset/initial status.
7. Data P, which specifies a scaling factor, can use a decimal point. The decimal point, however, becomes valid only if scaling command code G51 precedes data P.

`G51P0.5`	0.5 time
`P0.5G51`	1 time (regarded as P = 0)
`P500000G51`	0.5 time
`G51P500000`	0.5 time

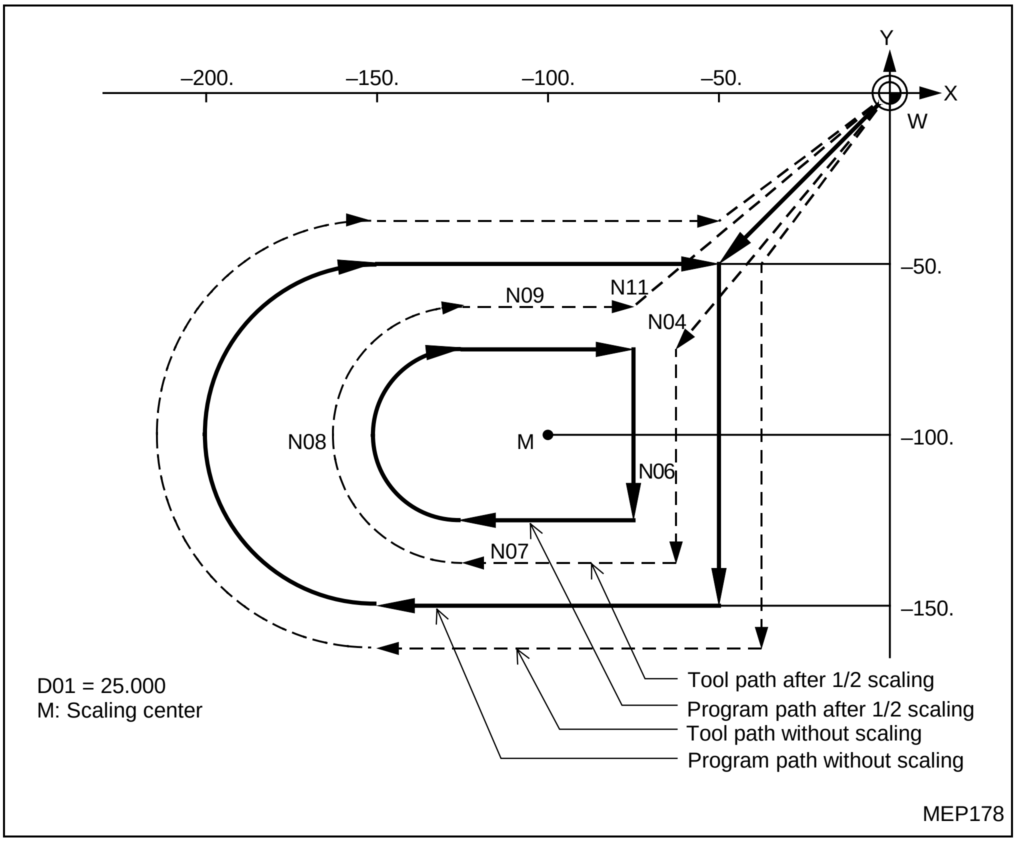
8. The center of scaling is shifted accordingly if the coordinate system is shifted using commands G92 or G52 during scaling.

Return to Library

5. Sample programs

1. Basic operation I

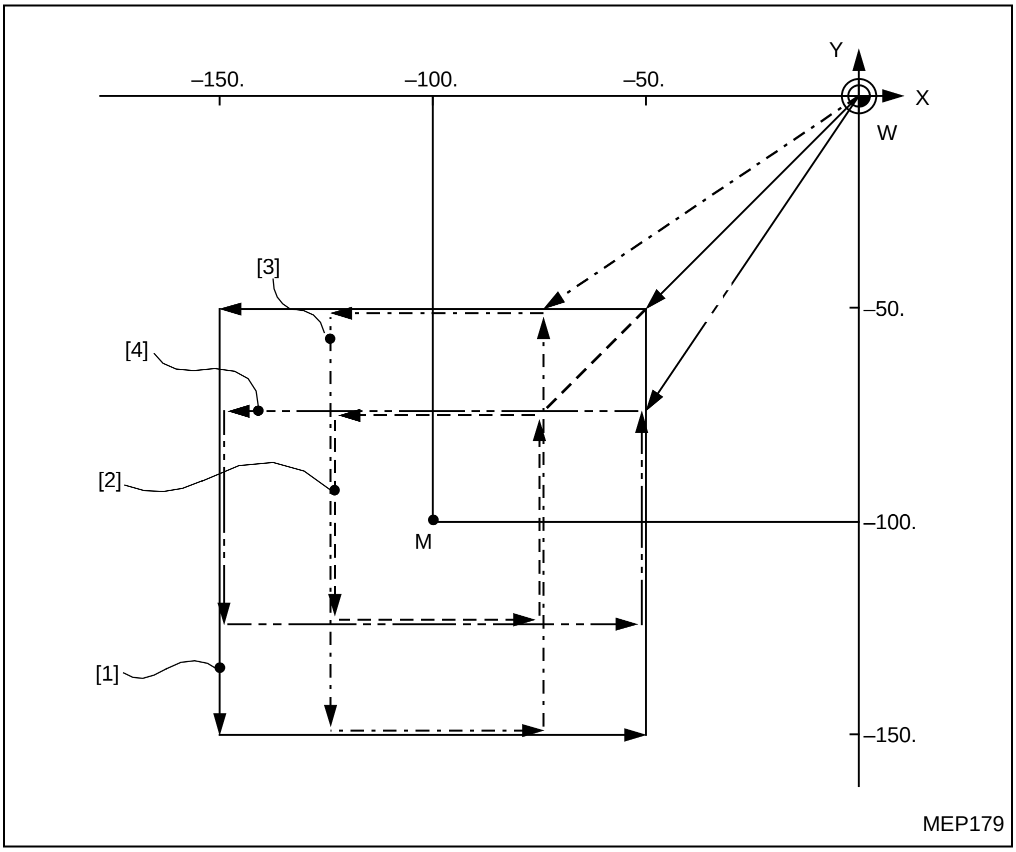
```
N01 G92X0Y0Z0
N02 G90G51X–100.Y–100.P0.5
N03 G00G43Z–200.H02
N04 G41X–50.Y-50.D01
N05 G01Z–250.F1000
N06 Y–150.F200
N07 X–150.
N08 G02Y–50.J50.
N09 G01X–50.
N10 G00Z0
N11 G40G50X0Y0
N12 M02
```



2. Basic operation II

```
N01 G92X0Y0
N02 G90G51P0.5 .................. See [1] to [4] below.
N03 G00X–50.Y–50.
N04 G01X–150.F1000
N05 Y–150.
N06 X–50.
N07 Y–50.
N08 G00G50
N09 M02
```

[1] Without scaling	`N02 G90G51P0.5`
[2] If scaling is to be done for X, Y	`N02 G90G51X–100.Y–100.P0.5`
[3] If scaling is to be done for X only	`N02 G90G51X–100.P0.5`
[4] If scaling is to be done for Y only	`N02 G90G51Y–100.P0.5`



3. Basic operation III

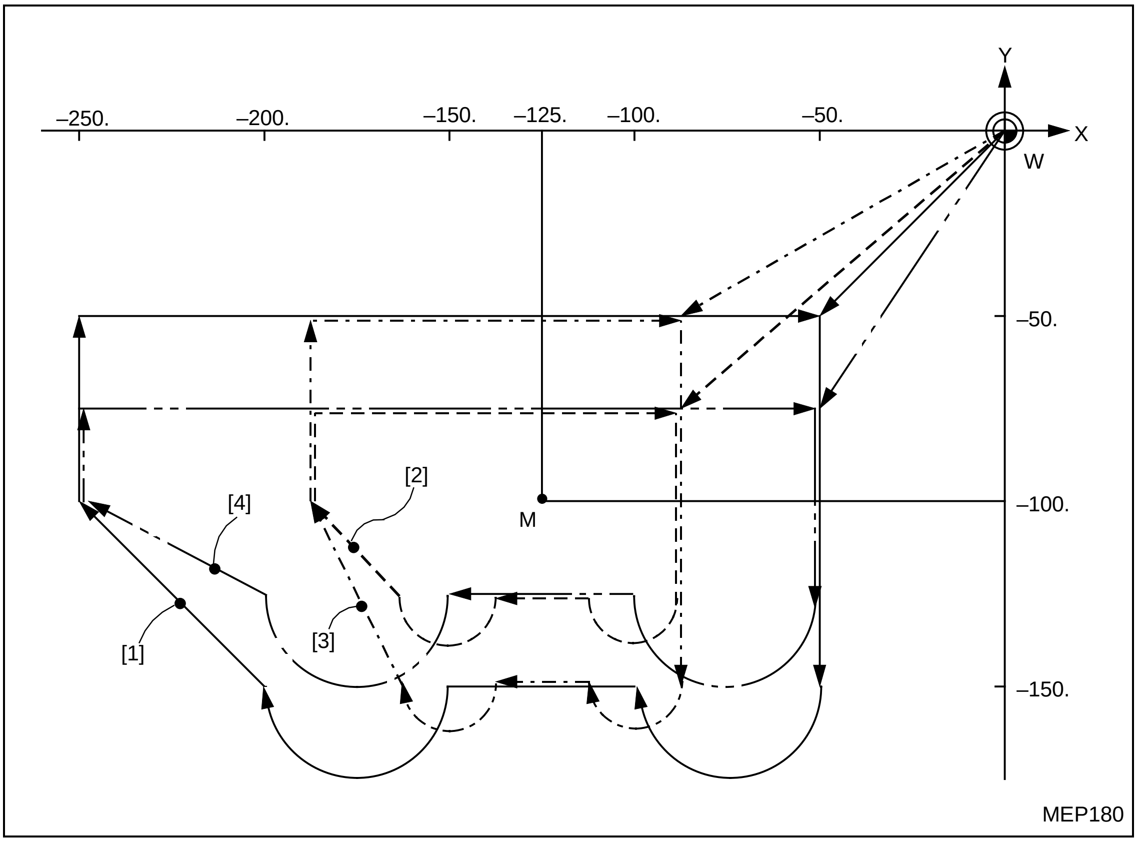
```
N01 G92X0Y0
N02 G90G51P0.5 .................. See [1] to [4] below.
N03 G00X-50.Y-50.
N04 G01Y-150.F1000
N05 G02X-100.I-25.
N06 G01X-150.
N07 G02X-200.I-25.
N08 G01X-250.Y-100.
N09 Y-50.
N10 X-50.
N11 G00G50
N12 M02
```

[1] Without scaling N02 G90G51P0.5

[2] If scaling is to be done for X, Y N02 G90G51X–125.Y–100.P0.5

[3] If scaling is to be done for X only N02 G90G51X–125.P0.5

[4] If scaling is to be done for Y only N02 G90G51Y–100.P0.5

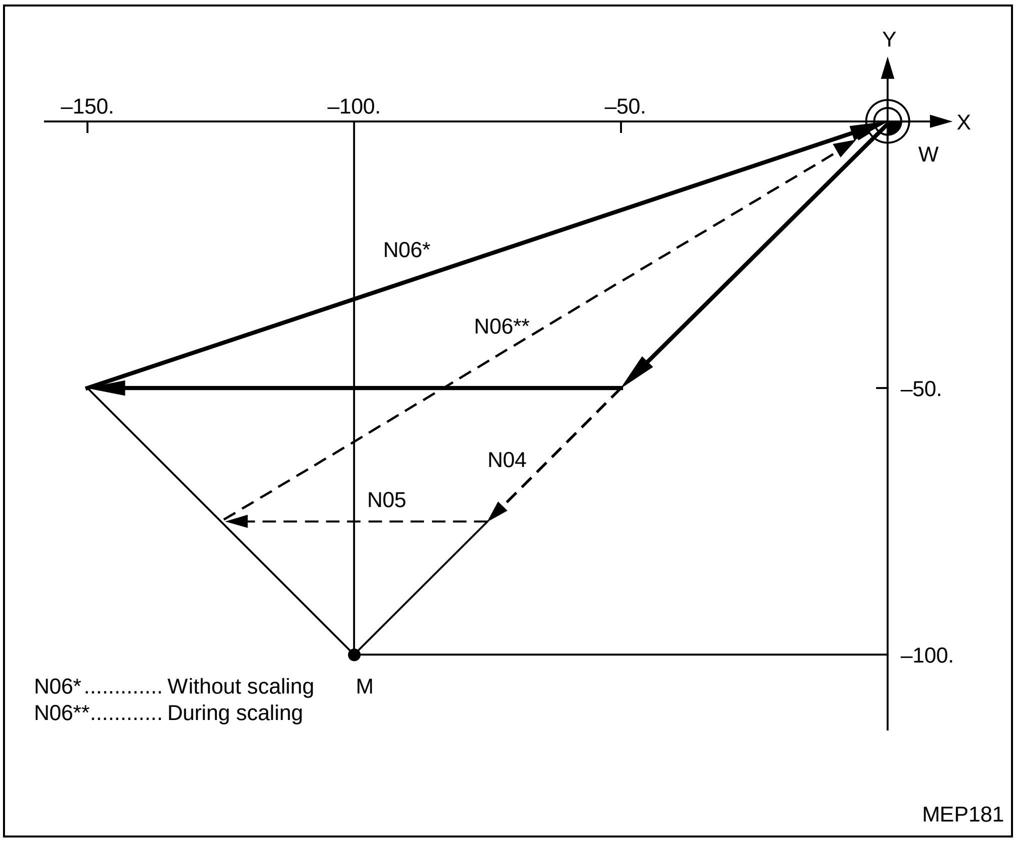


4. Reference-point (zero point) check (G27) during scaling

Setting G27 during scaling cancels the scaling mode after G27 has been executed.

```
N01 G28X0Y0
N02 G92X0Y0
N03 G90G51X–100.Y–100.P0.5
N04 G00X–50.Y–50.
N05 G01X–150.F1000
N06 G27X0Y0
        ⋮
```

If a program is constructed in the manner that the reference point is reached under normal mode, it will also be reached even under scaling mode.

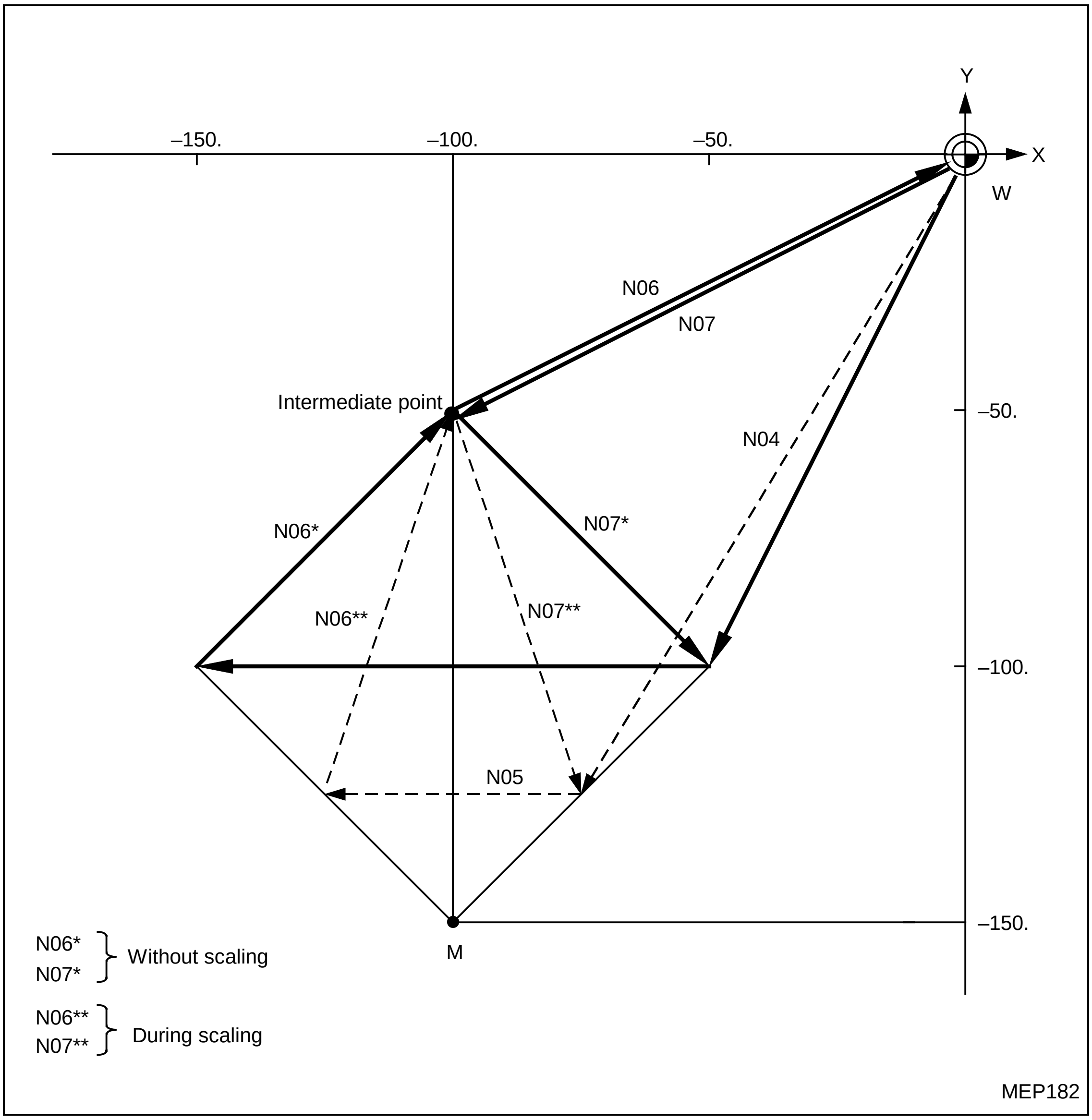


Return to Library

5. Reference-point (zero point) return (G28, G29, or G30) during scaling

Setting G28 or G30 during scaling cancels the scaling mode at the middle point and then executes the reference-point (zero point) return command. If the middle point has not been set, the reference-point (zero point) return command is executed with the point where scaling has been cancelled as middle point.
If G29 is set during the scaling mode, scaling will be performed for the entire movement after the middle point.

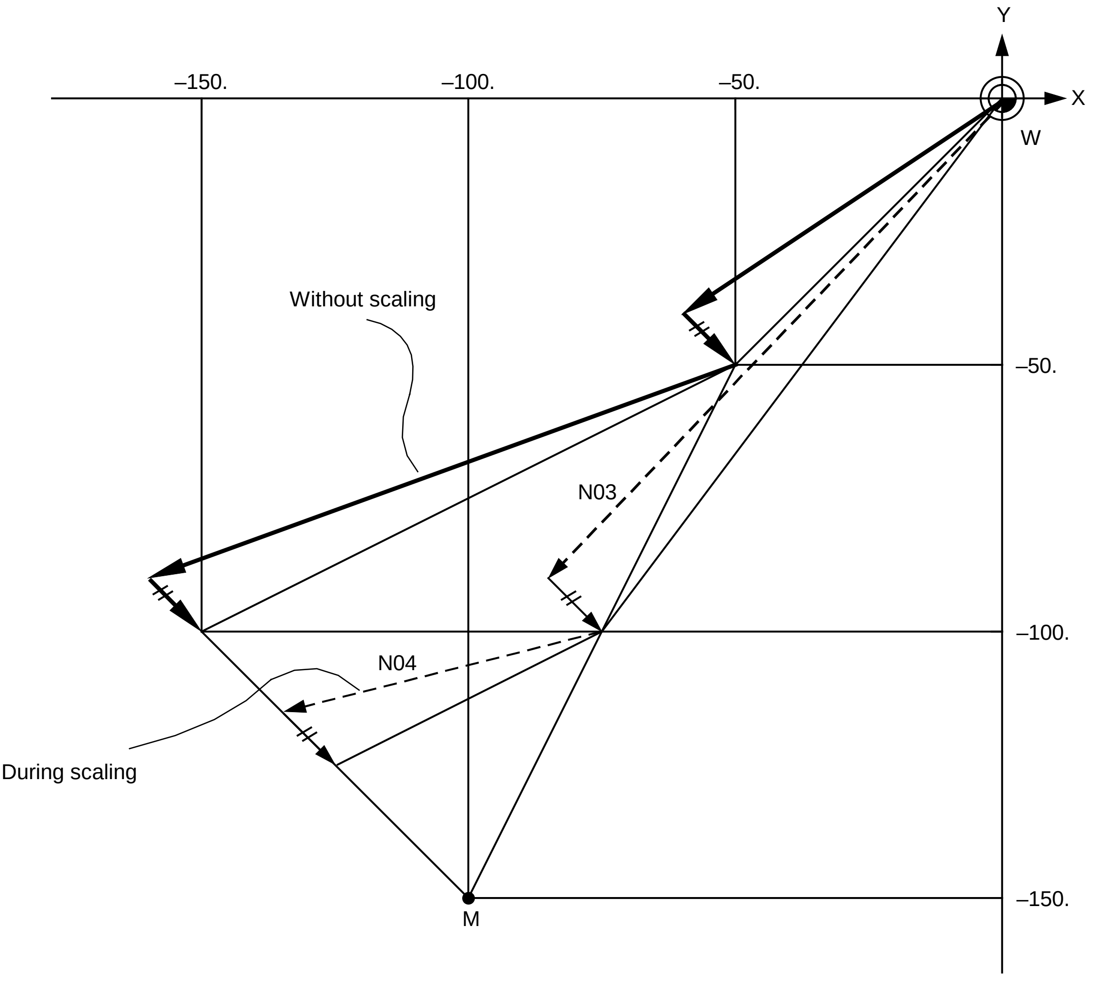
```
N01 G28X0Y0
N02 G92X0Y0
N03 G90G51X–100.Y–150.P500000
                          ||
N04 G00X–50.Y–100.       0.5
N05 G01X–150.F1000
N06 G28X–100.Y–50.
N07 G29X–50.Y–100.
```



6. One-way positioning (G60) during scaling

Setting G60 during the scaling mode executes scaling at the final point of positioning, and thus no scaling is performed for the parameter **l1** of creeping. That is, the amount of creeping remains constant, irrespective of whether scaling is valid.

```
N01 G92X0Y0
N02 G91G51X–100.Y–150.P0.5
N03 G60X–50.Y–50.
N04 G60X–150.Y–100.
```

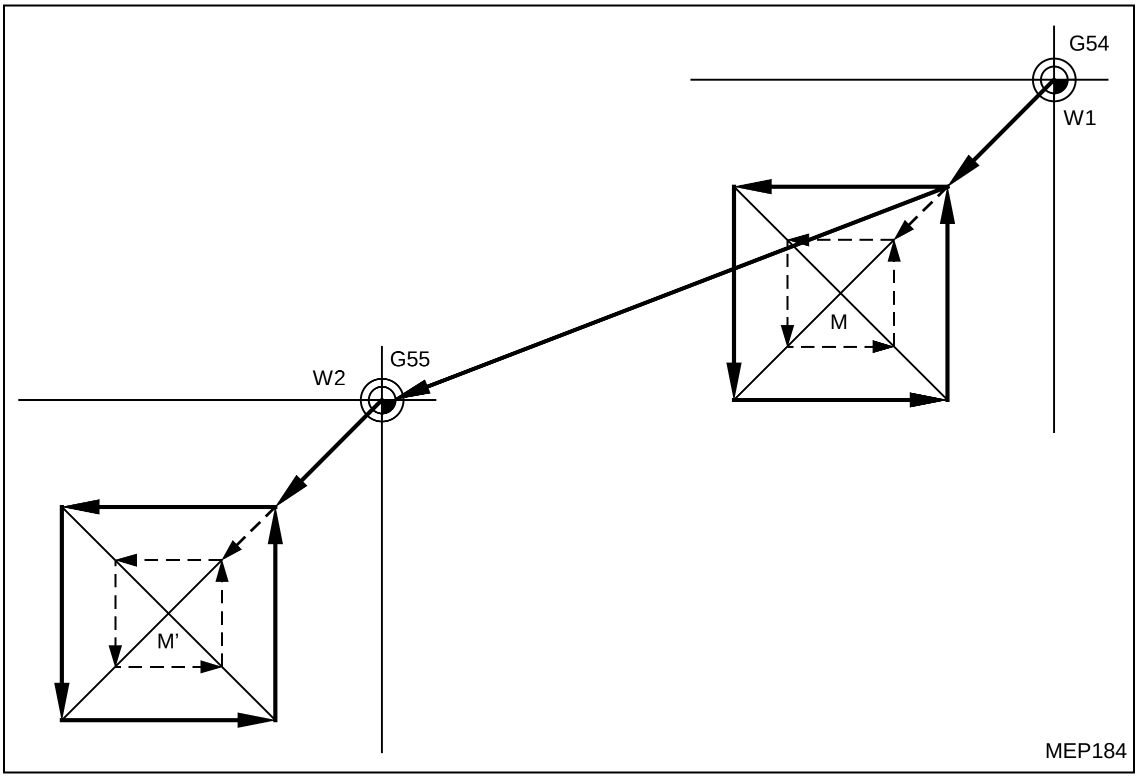


MEP183

7. Workpiece coordinate system updating during scaling

Updating of the workpiece coordinate system during scaling causes the center of scaling to be shifted according to the difference in offset amount between the new workpiece coordinate system and the old one.

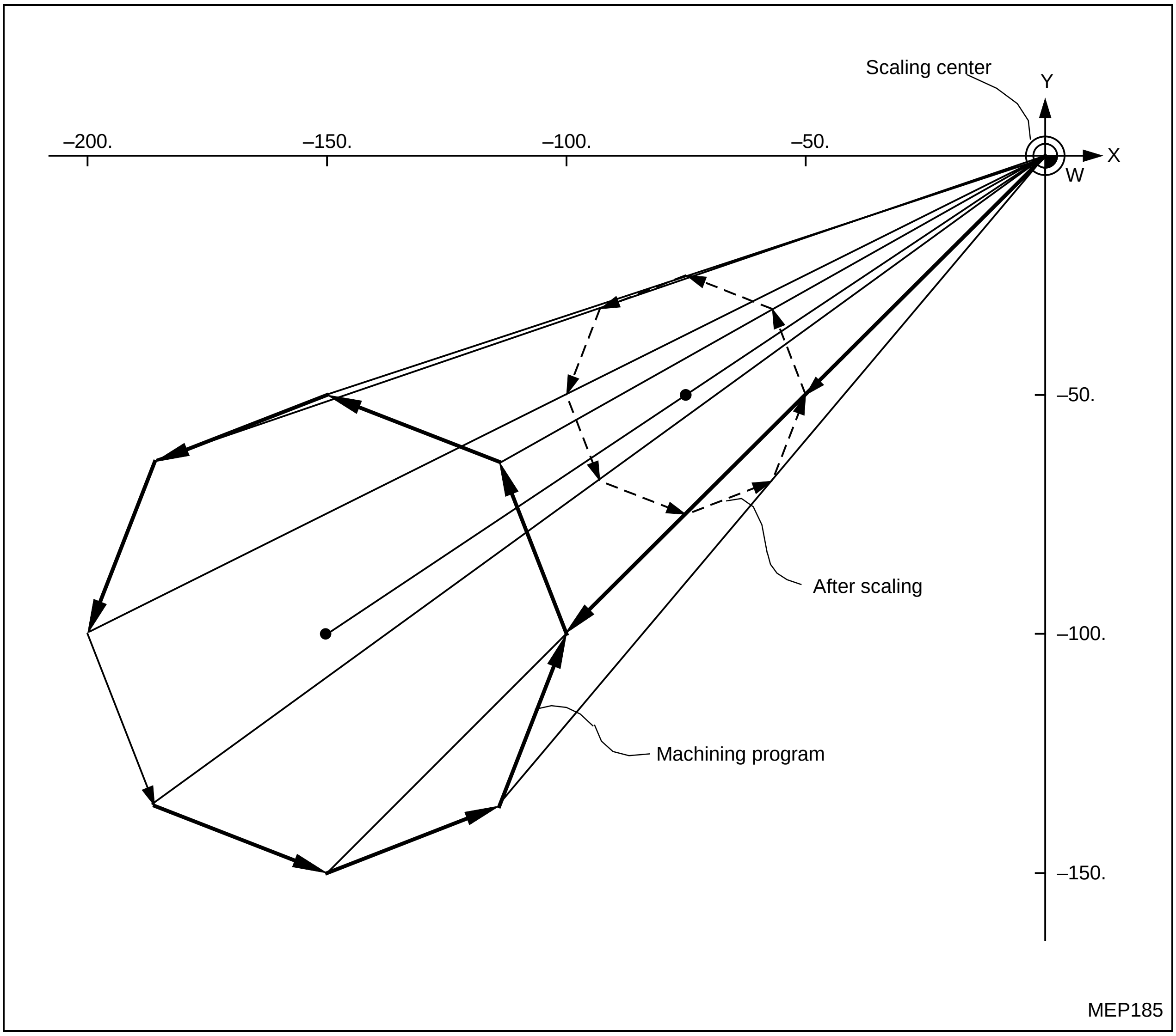
	Subprogram
`N01 G90G54G00X0Y0`	`O100`
`N02 G51X–100.Y–100.P0.5`	`G00X–50.Y–50.`
`N03 G65P100`	`G01X–150.F1000`
`N04 G90G55G00X0Y0`	`Y–150.`
`N05 G65P100`	`X–50.`
	`Y–50.`
	`M99`
	`%`



8. Figure rotation during scaling

Setting a figure rotate command during scaling executes scaling for both the center and radius of rotation of the figure.

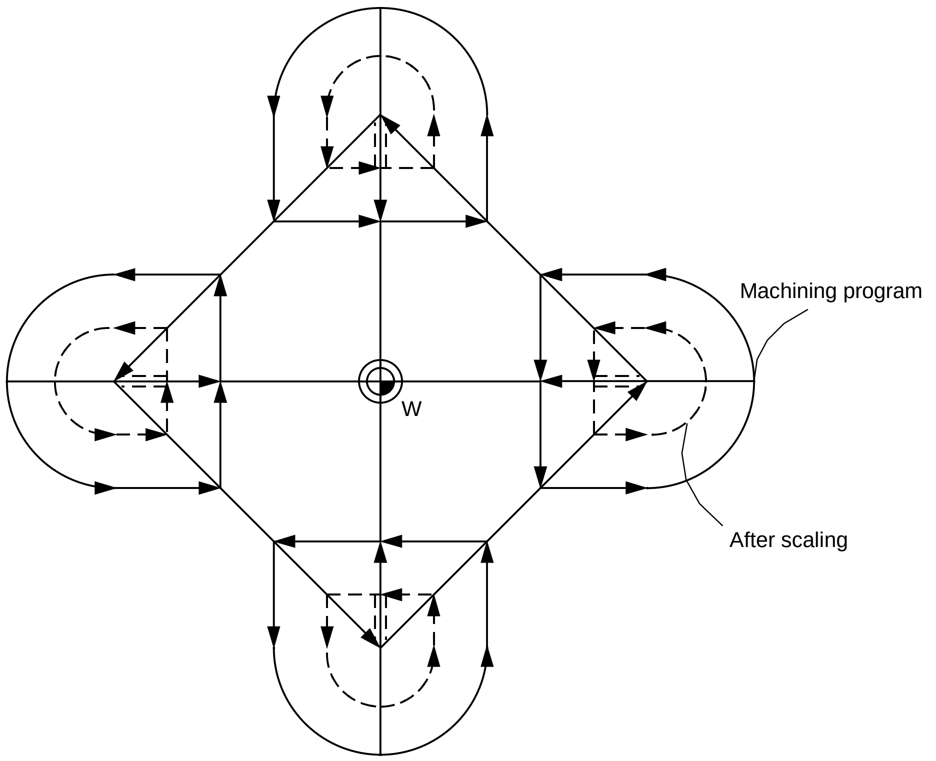
	Subprogram
N01 G92X0Y0	O200
N02 G90G51X0Y0P0.5	G91G01X–14.645Y35.355F1000
N03 G00X–100.Y–100.	M99
N04 M98P200I–50.L8	%



9. Scaling using a figure rotation subprogram

Setting a scaling command in a figure rotation subprogram executes scaling only for the shape predefined in the subprogram. Scaling is not executed for the radius of rotation of the figure.

	Subprogram
G92X0Y0	O300
G90G00X100.	G91G51X0Y0P0.5
M98P300I–100.L4	G00X–40.
G90G00X0Y0	G01Y–40.F1000
M02	X40.
	G03Y80.J40.
	G01X–40.
	Y–40.
	G00G50X40.
	X–100.Y100.
	M99
	%

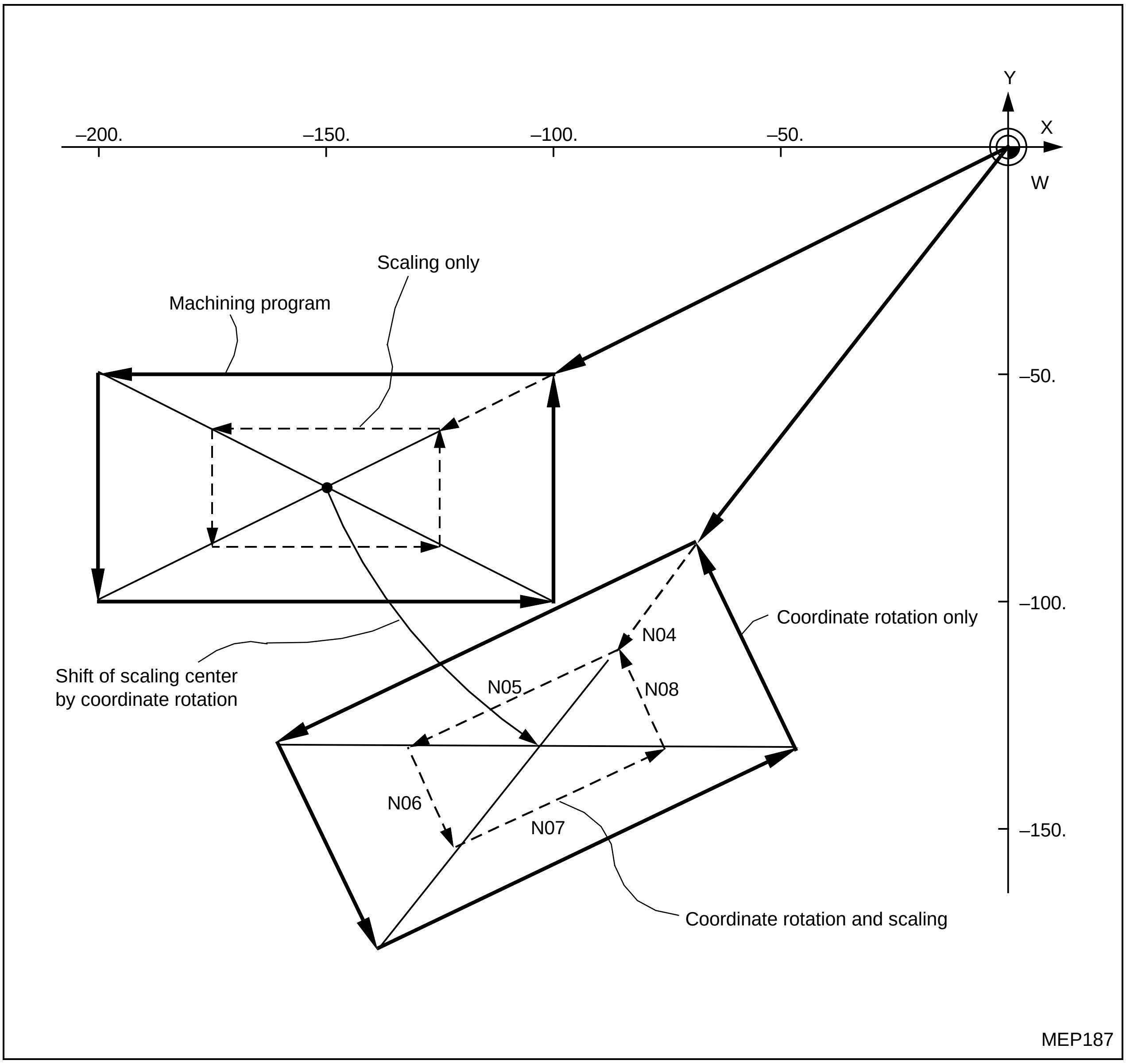


MEP186

10. Scaling during coordinate rotation

If scaling during coordinate rotation is programmed the center of scaling will rotate and scaling will be performed at that rotated center of scaling.

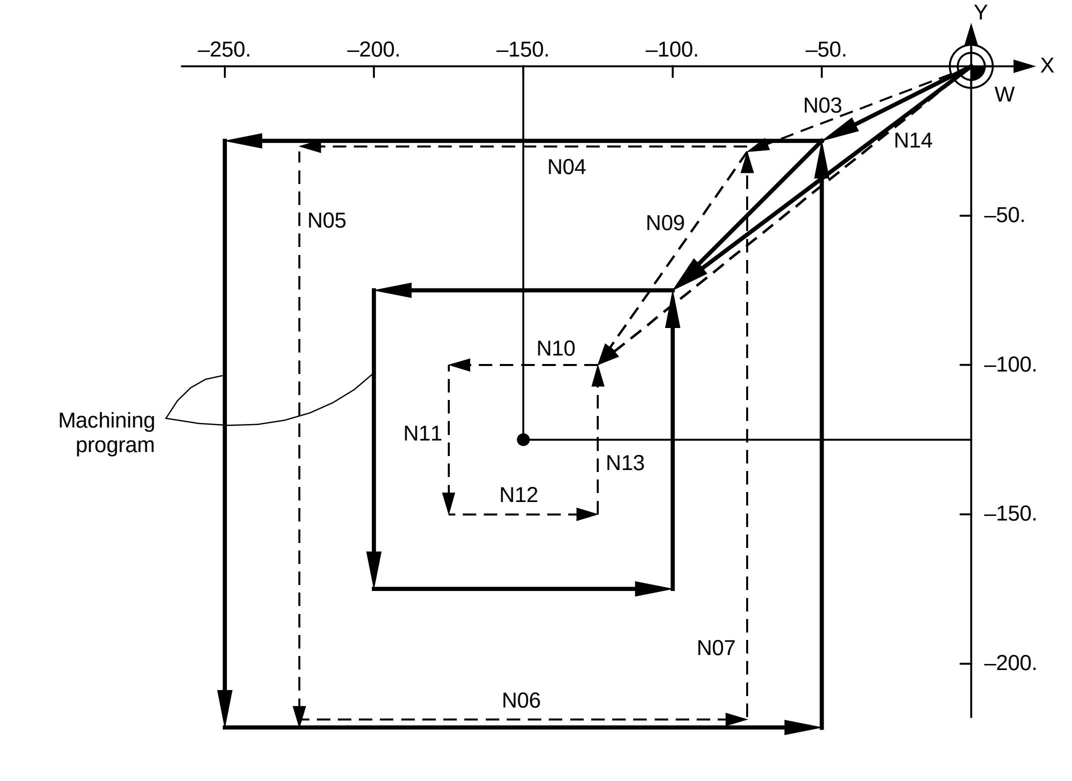
```
N01 G92X0Y0
N02 M00                            (Coordinate rotation data setting)
N03 G90G51X–150.Y–75.P0.5
N04 G00X–100.Y–50,
N05 G01X–200.F1000
N06 Y–100.
N07 X–100.
N08 Y–50.
N09 G00G50X0Y0
```



11. Setting G51 during scaling

If command G51 is set during the scaling mode, the axis for which the center of scaling is newly specified will also undergo scaling. The scaling factor specified by the latest G51 command becomes valid in that case.

```
N01 G92X0Y0
N02 G90G51X–150.P0.75      Scaling axis X; P = 0.75
N03 G00X–50.Y–25.
N04 G01X–250.F1000
N05 Y–225.
N06 X–50.
N07 Y–25.
N08 G51Y–125.P0.5          Scaling axes X and Y; P = 0.5
N09 G00X–100.Y–75.
N10 G01X–200.
N11 Y–175.
N12 X–100.
N13 Y–75.
N14 G00G50X0Y0             Cancel
```



MEP188

14-8 Mirror Image ON/OFF: G51.1/G50.1 (Series M)

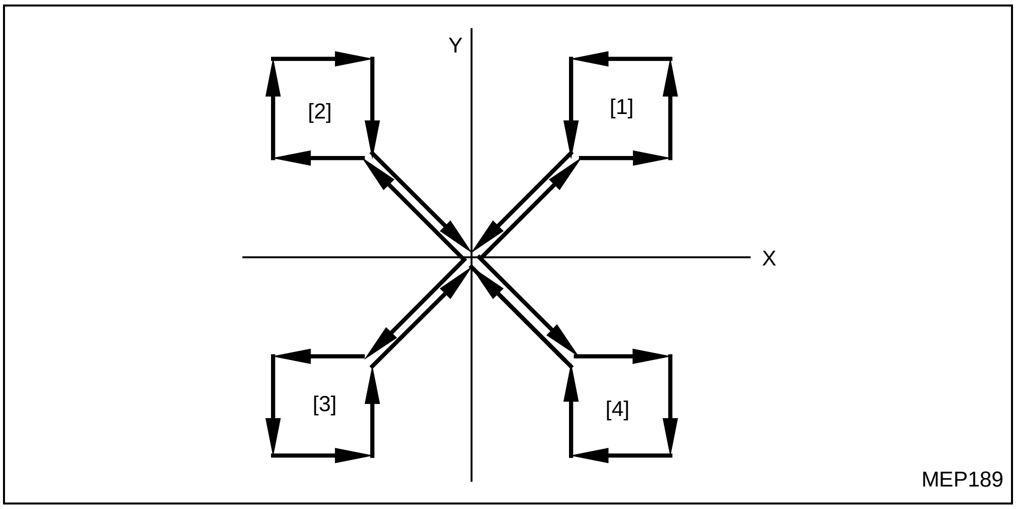
You can use G-code commands to turn the mirror image mode on or off for each axis. Higher priority is given to setting of the mirror image mode using G-code commands over setting using any other methods.

Programming format:

G51.1 Xx_1 Yy_1 Zz_1 (Mirror image On)
G50.1 Xx_2 Yy_2 Zz_2 (Mirror image Off)

Detailed description

- When using command G51.1, the name of the axis for which mirror image processing is to be performed must be designated using the appropriate coordinate word, and the mirror image center coordinates must be designated using absolute or incremental data as the coordinate data.
- If the coordinate word is designated in G50.1, then this denotes the axis for which the mirror image is to be cancelled. Coordinate data, even if predefined, is ignored in that case.
- After mirror image processing has been performed for only one of the axes forming a plane, the rotational direction and the offset direction become reverse during arc interpolation, tool diameter offsetting, or coordinate rotation.
- Since the mirror image processing function is valid only for local coordinate systems, the center of mirror image processing moves according to the particular counter preset data or workpiece coordinate offsetting data.



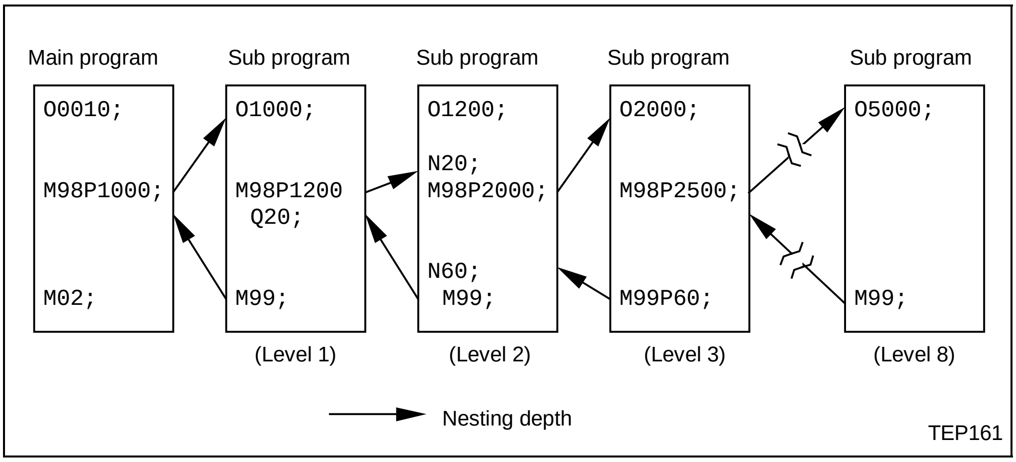
Specific examples

(Main program)		X	Y	(Subprogram O100)
G00G90G40G49G80				G91G28X0Y0
M98P100	[1]	OFF	OFF	G90G00X20.Y20.
G51.1X0 / M98P100	[2]	ON	OFF	G42G01X40.D01F120
				Y40.
G51.1Y0 / M98P100	[3]	ON	ON	X20.
				Y20.
G50.1X0 / M98P100	[4]	OFF	OFF	G40X0Y0
G50.1Y0		OFF	OFF	M99
M30				

14-9 Subprogram Control: M98, M99

1. Function and purpose

Fixed sequences or repeatedly used programs can be stored in the memory as subprograms which can then be called from the main program when required. M98 serves to call subprograms and M99 serves to return from the subprogram. Furthermore, it is possible to call other subprograms from particular subprograms and the nesting depth can include as many as 8 levels.



The table below shows the functions which can be executed by adding and combining the tape storing and editing functions, subprogram control functions and fixed cycle functions.

	Case 1	Case 2	Case 3	Case 4
1. Tape storing and editing 2. Subprogram control 3. Fixed cycles	Yes No No	Yes Yes No	Yes Yes Yes	Yes No Yes
Function				
1. Memory operation	○	○	○	○
2. Tape editing (main memory)	○	○	○	○
3. Subprogram call	×	○	○	×
4. Subprogram nesting level call (**Note 2**)	×	○	○	×
5. Fixed cycles	×	×	○	○
6. Fixed cycle subprogram editing	×	×	○	○

Notes:

1. "○" denotes a function which can be used and "×" a function which cannot be used.

2. The nesting depth can include as many as 8 levels.

2. Programming format

<u>Subprogram call</u>

M98 <_> Q_ L_;

- Number of subprogram repetitions (L1 if omitted)
- Sequence number in subprogram to be called (head block if omitted)
  (Use the address H for G-code series M.)
- Program name of subprogram to be called (own program if omitted).
  Can only be omitted during memory operation.

Alternatively,

M98 P_ Q_ L_;

- Number of subprogram repetitions (L1 if omitted)
- Sequence number in subprogram to be called (head block if omitted)
  (Use the address H for G-code series M.)
- Program number [composed of numerals only] of subprogram to be called (own program if omitted). P can only be omitted during memory operation.

<u>Return to main program from subprogram</u>

M99 P_ L_;

- Number of times after repetition number has been changed
- Sequence number of return destination (returned to block following block of call if omitted)

3. Creating and entering subprograms

Subprograms have the same format as machining programs for normal memory operation except that the subprogram completion instruction M99 (P_ L_) is entered as an independent block at the last block.

OΔΔΔΔ ;	Program number as subprogram
...............;	
...............;	
⋮	Main body of subprogram
...............;	
M99;	Subprogram return command
%(EOR)	End of record code (% with ISO code and EOR with EIA code)

The above program is registered by editing operations. For further details, refer to the section on program editing.

Only those subprograms numbers ranging from 1 through 9999 designated by the optional specifications can be used. When there are no program numbers on the tape, the setting number for "program input" is used.

Up to 8 nesting levels can be used for calling programs from subprograms, and program error occurs if this number is exceeded.

Main programs and subprograms are registered in order in which they were read because no distinction is made between them. This means that main programs and subprograms should not be given the same numbers. (If the same numbers are given, error occurs during entry.)

Return to Library

Example:

```
       ;
OOOOO  ;
.................;      Subprogram A
       :
M99;
      %

       ;
OΔΔΔΔ  ;
.................;      Subprogram B
       :
M99;
      %

       ;
O□□□□  ;
.................;      Subprogram C
       :
M99;
      %
```

Note 1: Main programs can be used during memory and tape operation but subprograms must have been entered in the memory.

Note 2: The following commands are not the object of subprogram nesting and can be called even beyond the 8th nesting level.

- Fixed cycles
- Pattern cycles

4. Subprogram execution

M98: Subprogram call command
M99: Subprogram return command

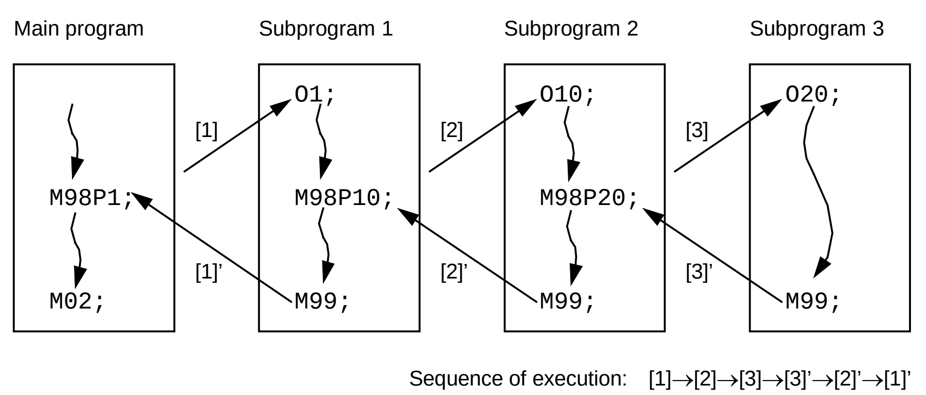
<u>Programming format</u>

M98 <_> Q_ L_; or M98 P_ Q_ L_;

Where < > : Name of the subprogram to be called (up to 32 characters)
P : Number of the subprogram to be called (up to 8 digits)
Q : Any sequence number within the subprogram to be called (up to 5 digits) (Use the address H for G-code series M.)
L : Number of repetitions from 1 to 9999 with numerical value of four figures; if L is omitted, the subprogram is executed once ; with L0, there is no execution.

For example,
M98 P1 L3; is equivalent to the following :
M98 P1;
M98 P1;
M98 P1;

Example 1: When there are 3 subprogram calls (known as 3 nesting levels)

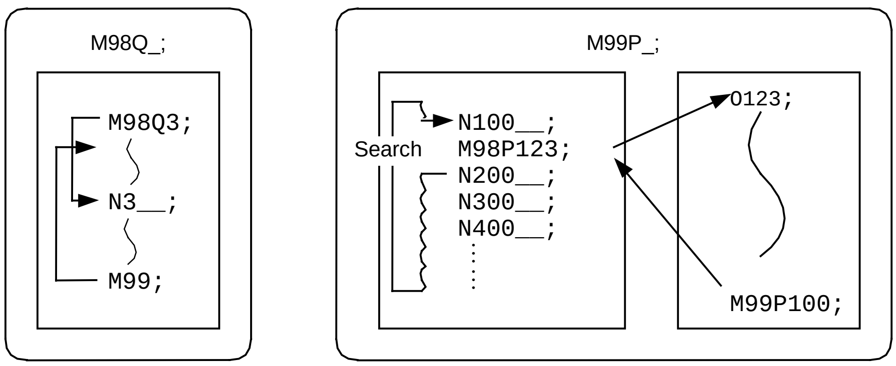


TEP162

For nesting, the M98 and M99 commands should always be paired off on a 1 : 1 basis [1]' for [1], [2]' for [2], etc.

Modal information is rewritten according to the execution sequence without distinction between main programs and subprograms. This means that after calling a subprogram, attention must be paid to the modal data status when programming.

Example 2: The M98 Q_ ; and M99 P_ ; commands designate the sequence numbers in a program with a call instruction.



TEP163

Example 3: Main program M98 P2 ;

```
O1;   ⎫
 :    ⎪ Subprogram 1
M99;  ⎪
%     ⎭
O2;   ⎫
 :    ⎪
N200  ⎪ Subprogram 2
 :    ⎪
M99;  ⎪
%     ⎭
O3;   ⎫
 :    ⎪
N200  ⎪ Subprogram 3
 :    ⎪
M99;  ⎪
%     ⎭
```

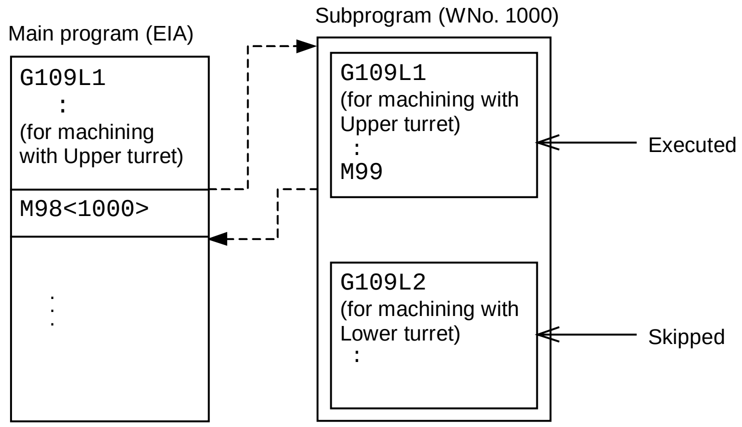
- When the O2 N200 block is searched with the memory search function, the modal data are updated according to the related data of O2 to N200.
- The same sequence number can be used in different subprograms.
- When the subprogram (No. p_1) is to be repeatedly used, it will be repeatedly executed for I_1 times provided that M98 Pp_1 Ll_1 ; is programmed.

5. Other precautions

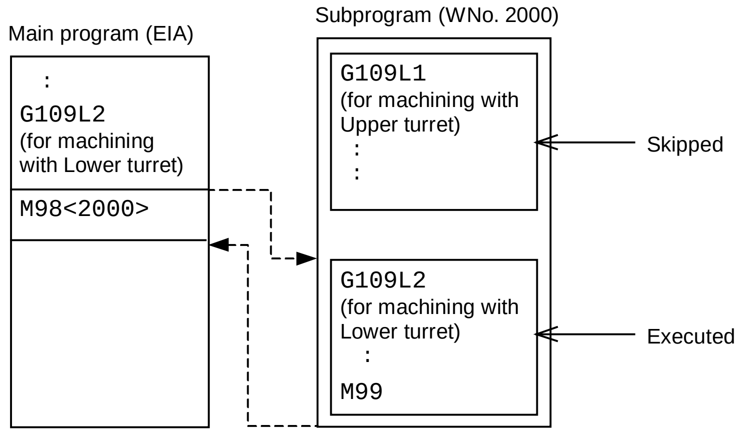
- Programming error occurs when the designated program number (P) is not found.
- Single block stop does not occur in the M98P _ ; and M99 ; block. If any address except O, N, P, Q or L is used, single block stop can be executed. (With X100. M98 P100 ; operation branches to O100 after X100. is executed.)
- When M99 is commanded in the main program, operation returns to the head.
- Operation can branch from tape or PTR operation to a subprogram by M98P_ but the sequence number of the return destination cannot be designated with M99P_ ;. (P_ is ignored.)
- Care should be taken that the search operation will take time when the sequence number is designated by M99P_ ;

- In the execution of a subprogram composed of sections each proper to either of the upper and lower turrets with the aid of G109L_ blocks, only the program sections for that system (headstock or turret) which is currently active in the main program at the call of the subprogram will selectively be executed with the other sections being appropriately skipped, as shown below:

Pattern 1:



Pattern 2:



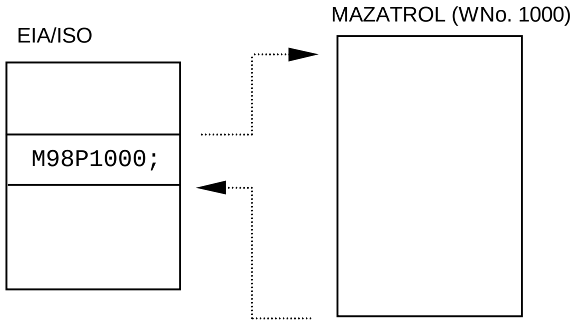
- The Z- and C-offset values stored on the **SET UP MANAG.** display for the main program will remain intact for the execution of a subprogram prepared in the EIA format.

6. MAZATROL program call from EIA/ISO program

A. Overview

MAZATROL machining program can be called as a subprogram from the machining program described with EIA/ISO codes.

EIA/ISO → MAZATROL (Program call)

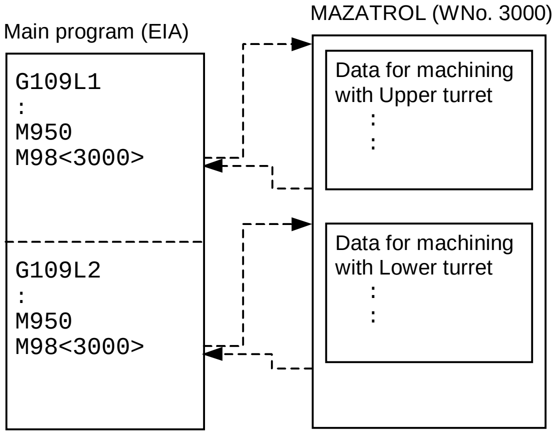


MAZATROL machining program is called from EIA/ISO program, and entire machining program can be used.

Note 1: When the execution of MAZATROL machining program is completed, the execution is returned again to EIA/ISO program.
It should be noted that the used tool, current position and others are changed though EIA/ISO modal information is not changed.

Note 2: MAZATROL programs (with commands for both the upper and lower turret) can successfully be called up as a subprogram from two positions of an EIA program of similar structure on condition that one and the same program is called up on completion of blocks of the same waiting command.

Example:



Note 3: The Z- and C-offset values used for the execution of a MAZATROL program as a subprogram called up from an EIA program depend upon the setting of parameter **F161** bit 6 as follows:

F161 bit 6 = 0: Values of the subprogram (MAZATROL)
= 1: Values of the main program (EIA)

B. Programming format

M98 <_> L_; or M98 P_ L_;

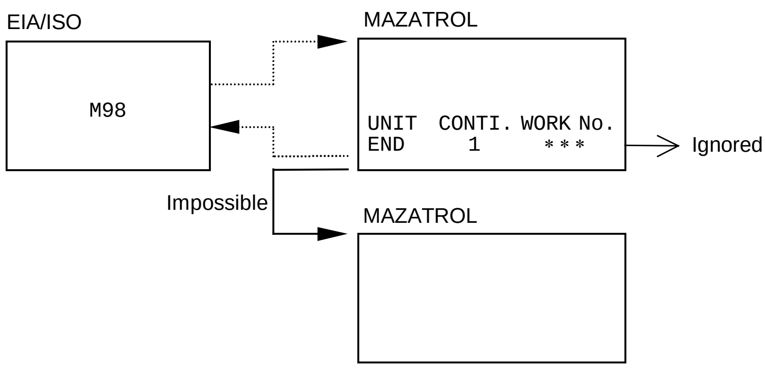
< > or P: Name, or number, of the MAZATROL machining program to be called.
When not specified, the alarm **744 NO DESIGNATED PROGRAM** will be displayed. Also, when the specified program is not stored, the alarm **744 NO DESIGNATED PROGRAM** will be displayed.

L: Number of repetitions of program execution (1 to 9999).
When omitted or L=0, the called program will be executed one time (as if L=1).

C. Detailed description

1. END unit of the MAZATROL program

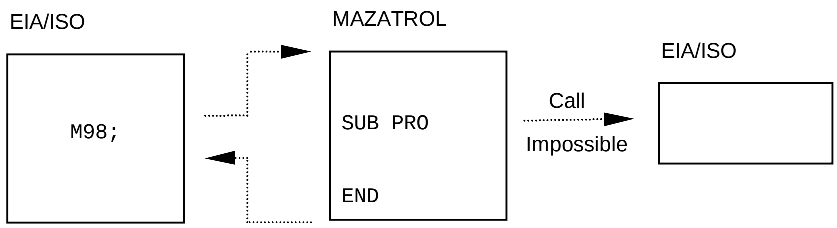
End unit does not have to be specified at the end of MAZATROL machining program.
When end unit is specified: Even if **WORK No.** and **CONTI.** are specified, they are ignored.
This means that program chain cannot be made with MAZATROL program called from EIA/ISO program.



Also, **REPEAT** and **SHIFT** are ignored even if they are specified.

2. MAZATROL program execution
When MAZATROL program is called from EIA/ISO program, the MAZATROL program is executed like automatic operation of MAZATROL.
MAZATROL program is executed independently of EIA/ISO program which has made the call. In other words, it performs the same machining as MAZATROL program alone is executed. When calling MAZATROL program, always place a tool outside the safety profile beforehand. Failure to do this may cause interference of a workpiece with the tool.

3. Nesting
Within a MAZATROL program called from EIA/ISO program, the subprogram unit (SUB PRO) cannot be used.



Refer to the MAZATROL Programming Manual for SUB PRO unit.

Note: As is the case with a SUB PRO unit, alarm **742 SUB PROGRAM NESTING OVER** will occur if a point-machining unit is present in the MAZATROL program that has been called up as a subprogram from the EIA program.

D. Remarks

1. MDI interruption and macro interruption signal during MAZATROL program execution are ignored.
2. MAZATROL program cannot be restarted halfway.
3. MAZATROL program call in the mode of a fixed cycle results in an alarm.
4. MAZATROL program call in the mode of nose radius compensation results in an alarm.
5. MAZATROL program call is not available in the MDI operation mode (results in an alarm).
6. A MAZATROL program called by M98 cannot be executed but in its entirety (from the head to the end).
7. Commands to addresses other than O, N, P, Q, L and H in a block of M98 for MAZATROL program call will not be processed till completion of the called program.

Return to Library

14-10 End Processing: M02, M30, M998, M999

If the program contains M02, M30, M998, M999 or EOR (%), the block containing one of these codes will be executed as the end of the program in the NC unit. The program end processing will not be commanded by M98 or M99. In end processing, tool life processing, parts count, and work No. search will be executed.

1. M02, M30
   Tool life processing only will be executed.
2. M998, M999
   Tool life processing, parts count, and work No. search will be executed.

M998(999) <111> Q1;

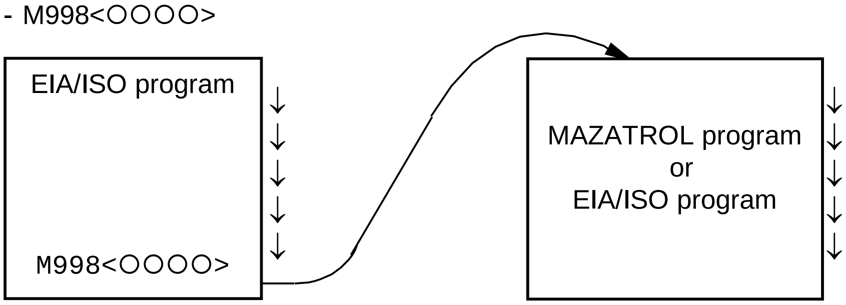
- Q1: Specification of execution or non-execution of parts count (counting updated on **POSITION** display)
  0: Parts count non-execution
  1: Parts count execution
- <111>: Name of the program to be executed next
- M998(999): M-code for program chain
  M998: Continuous execution after parts count and work No. search
  M999: Ending after parts count and work No. search

Alternatively,

M998(999) P111 Q1;

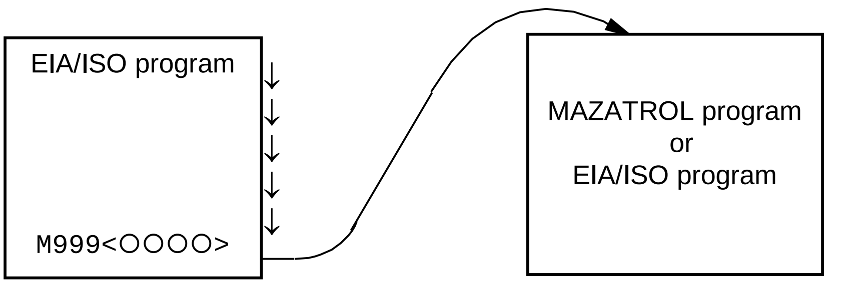
- Q1: Specification of execution or non-execution of parts count (counting updated on **POSITION** display)
  0: Parts count non-execution
  1: Parts count execution
- P111: Number of the program to be executed next
- M998(999): M-code for program chain
  M998: Continuous execution after parts count and work No. search
  M999: Ending after parts count and work No. search

- M998<○○○○>



MAZATROL or EIA/ISO program is called from EIA/ISO program and executed as the next program.

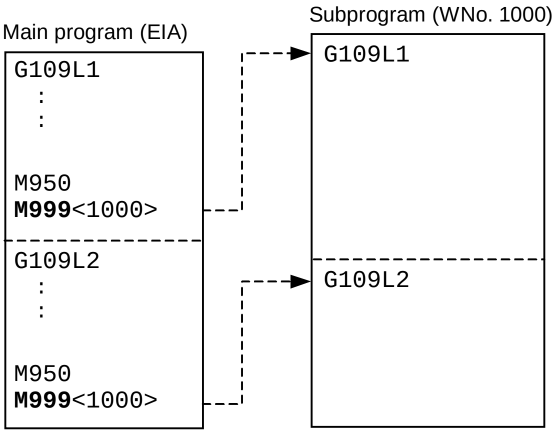
- M999<○○○○>



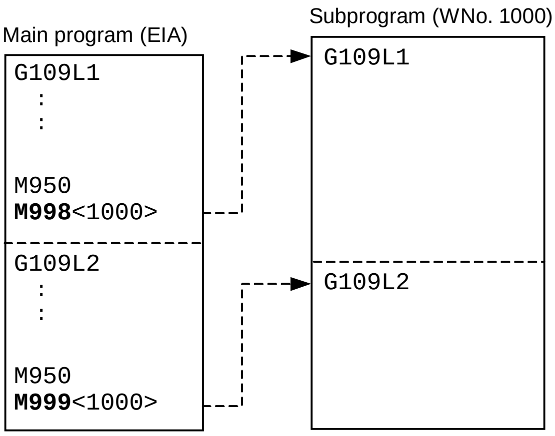
MAZATROL or EIA/ISO program is only called from EIA/ISO program and the operation is terminated.

Note: The programs to be called up at the end of both the upper and lower turrets' program sections must be of the same work number; otherwise an alarm will be caused. Moreover, use either M998 or M999 for both turrets' sections in their respective ending blocks; otherwise an alarm will likewise be caused.

Example 1: Correct use



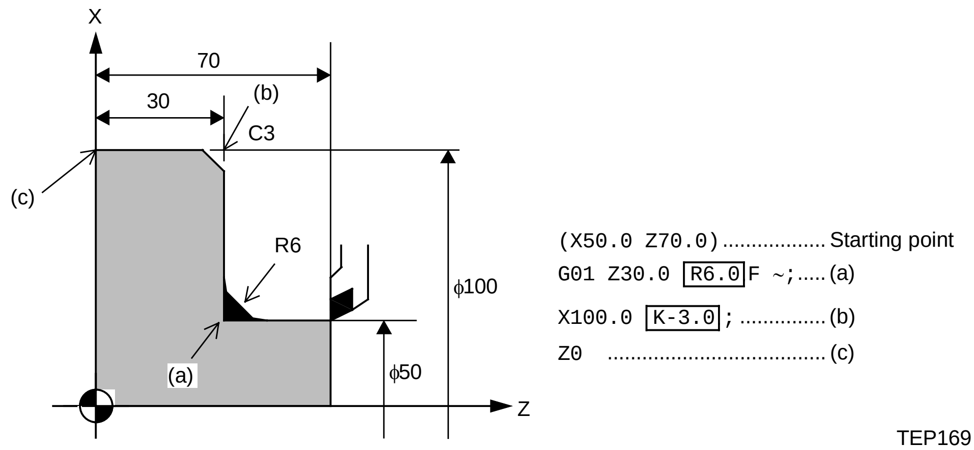
Example 2: Wrong use



14-11 Chamfering and Corner Rounding at Right Angle Corner

1. Overview

Chamfering or corner rounding can be commanded between two blocks specified by linear interpolation (G01). For I, J and K, radial data must be always set.



TEP169

2. Detailed description

1. For chamfering or corner rounding, movement commanded by G01 must be displacement in the X- or Z-axis only. In the second block, a command perpendicular to the first axis must be given in the Z- or X-axis.
2. The starting point of the second block is the ending point of the first block.

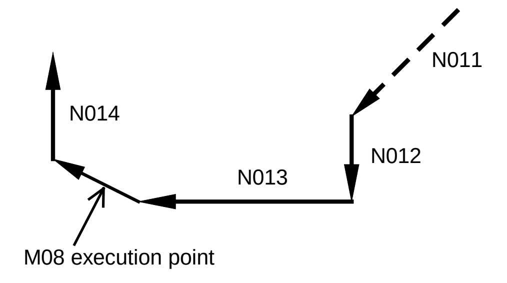
   Example: `G01 Z270.0 R6;`
   `X860.0 K–3;` The starting point of this block has Z270.0 as Z coordinate.

3. The commands below will cause an alarm.
   - I, J, K or R is commanded while two axes of X and Z are commanded in G01.
   - Two of I, J, K or R are commanded in G01.
   - X and I, Y and J or Z and K are commanded at the same time in G01.
   - In a block commanding chamfering or corner rounding, movement distance in X- or Z-axis is smaller than chamfering data or corner radius data.
   - In a block next to the block commanding chamfering or corner rounding, command G01 is not perpendicular to the command in the preceding block.
4. In threading block, chamfering or corner rounding command will be ignored.
5. Execution by single step mode will require two steps to complete the operation.

Return to Library

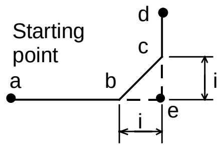
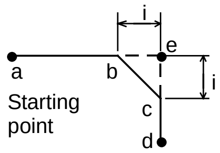
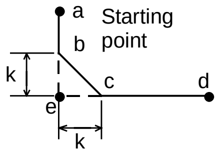
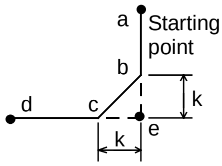
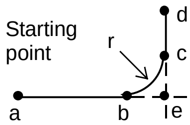
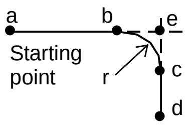
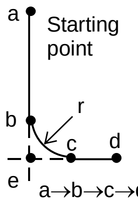
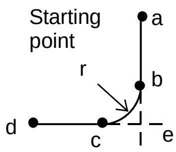
6. When M, T commands are included in the same block, execution point must be considered.

```
N011  G00 X100.0 Z0;
N012  G01 X90.0 F0.5;
N013  Z-20. R0.5 M08;
N014  X100.;
```



TEP170

7. Chamfering and corner rounding programming format

Operation	Command	Tool movement	Remarks
Chamfering Z → X	`G01 Z(W)e Ii;` `X(U)d;` .... Next block	 a→b→c→d	Specify the point “e”. Specify the data only for X-axis in the next block. d ≥ e + 2i
	`G01 Z(W)e I–i;` `X(U)d;` .... Next block	 a→b→c→d	Specify the point “e”. Specify the data only for X-axis in the next block. d ≤ e – 2i
Chamfering X → Z	`G01 X(U)e Kk;` `Z(W)d;` .... Next block	 a→b→c→d	Specify the point “e”. Specify the data only for Z-axis in the next block. d ≥ e + k
	`G01 X(U)e K–k;` `Z(W)d;` .... Next block	 a→b→c→d	Specify the point “e”. Specify the data only for Z-axis in the next block. d ≤ e – k
Corner rounding Z → X	`G01 Z(W)e Rr;` `X(U)d;` ····· Next block	 a→b→c→d	Specify the point “e”. Specify the data only for X-axis in the next block. d ≥ e + 2r
	`G01 Z(W)e R–r;` `X(U)d;` .... Next block	 a→b→c→d	Specify the point “e”. Specify the data only for X-axis in the next block. d ≤ e – 2r
Corner rounding X → Z	`G01 X(U)e Rr;` `Z(W)d;` .... Next block	 a→b→c→d	Specify the point “e”. Specify the data only for Z-axis in the next block. d ≥ e + r
	`G01 X(U)e R–r;` `Z(W)d;` .... Next block	 a→b→c→d TEP171	Specify the point “e”. Specify the data only for Z-axis in the next block. d ≤ e – r

14-12 Chamfering and Corner Rounding at Arbitrary Angle Corner Function

Chamfering or corner rounding at any angle corner is performed automatically by adding ",C_" or ",R_" to the end of the block to be commanded first among those command blocks which form the corner with lines only.

14-12-1 Chamfering at arbitrary angle corner: , C_

1. Function

The arbitrary corner is chamfered between two points on the two lines which form this corner and displaced by the lengths commanded by ", C_" from their intersection point.

2. Programming format

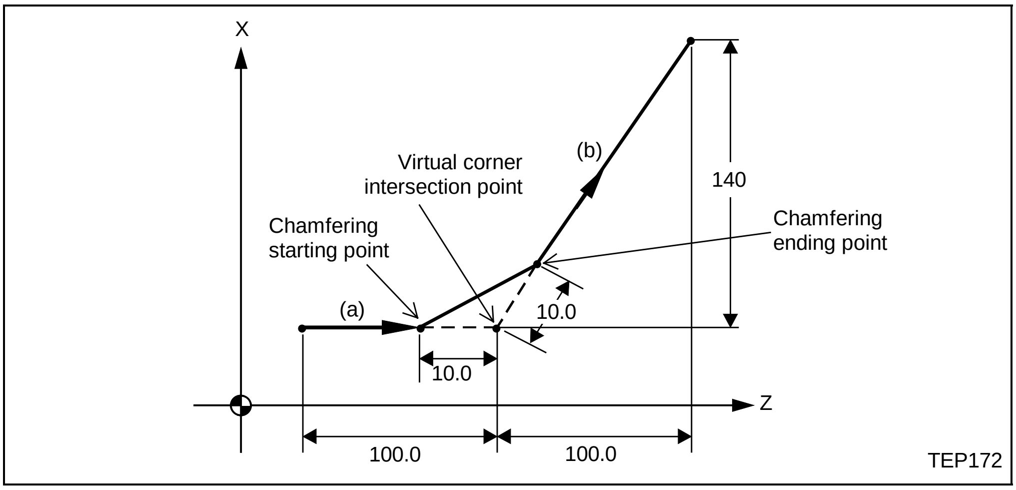
N100 G01 X_ Z_ ,C_;
N200 G01 X_ Z_;

Chamfering is performed at the point where N100 and N200 intersect.

Length up to chamfering starting point or ending point from virtual corner intersection point

3. Example of program

(a) `G01 W100.,C10.F100;`
(b) `U280.W100.;`



4. Detailed description

1. The starting point of the block following the corner chamfering is the virtual corner intersection point.
2. When the comma in " , C " is not present, it is considered as a C command.
3. When both, C_ and , R_ are commanded in the same block, the latter command is valid.
4. Tool offset is calculated for the shape which has already been subjected to corner chamfering.
5. Program error occurs when the block following the block with corner chamfering does not contain a linear interpolation command.
6. Program error occurs when the movement amount in the block commanding corner chamfering is less than the chamfering amount.

7. Program error occurs when the movement amount in the block following the block commanding corner chamfering is less than the chamfering amount.

14-12-2 Rounding at arbitrary angle corner: , R_

1. Funciton

The arbitrary corner is rounded with the arc whose radius is commanded by ",R_" and whose center is on the bisecter of this corner angle.

2. Programming format

N100 G01 X_ Z_ ,R_;
N200 G01 X_ Z_;

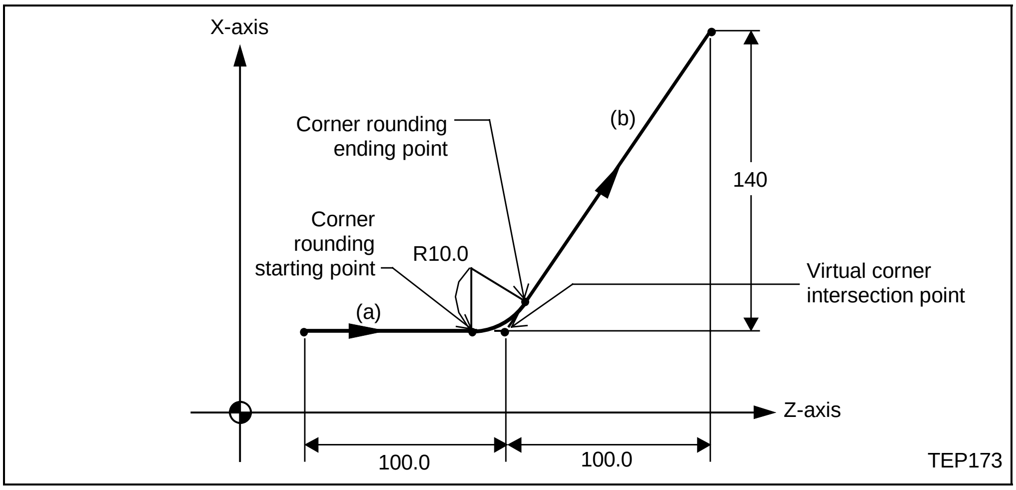
Rounding is performed at the point where N100 and N200 intersect.

Arc radius of corner rounding

3. Example of program

(a) `G01 W100.,R10.F100;`

(b) `U280.W100.;`



4. Detailed description

1. The starting point of the block following the corner rounding is the virtual corner intersection point.
2. When the comma in " , R" is not present, it is considered as an R command.
3. When both , C_ and , R_ are commanded in the same block the latter command is valid.
4. Tool offset is calculated for the shape which has already been subjected to corner rounding.
5. Program error occurs when the block following the block with corner rounding does not contain a linear command.
6. Program error occurs when the movement amount in the block commanding corner rounding is less than the R value.
7. Program error occurs when the movement amount in the block following the block commanding corner rounding is less than the R value.

Return to Library

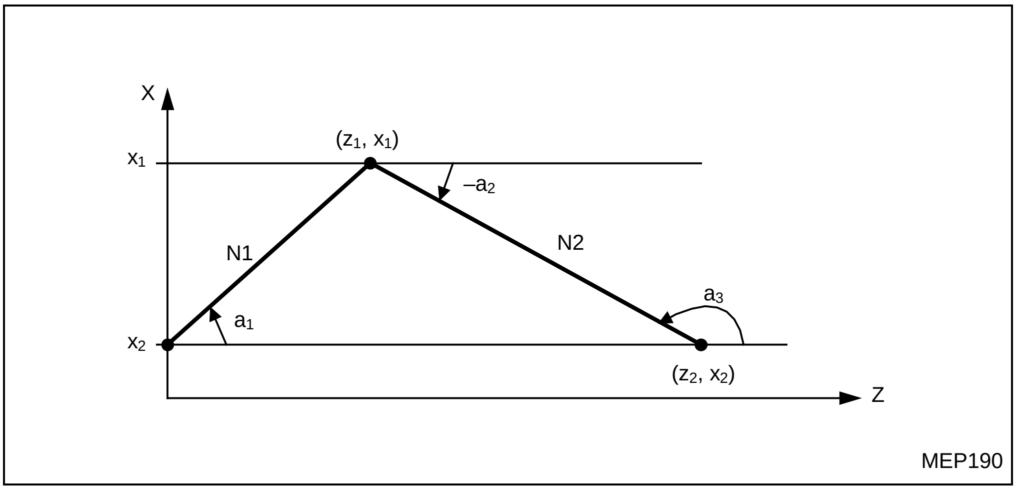
14-13 Linear Angle Commands

1. Function and purpose

Programming the linear angle and one of the coordinates of the ending point makes the NC unit automatically calculate the coordinates of that ending point.

2. Programming format

N1 G01 Aa_1 Zz_1 (Xx_1)   Designate the angle and the coordinates of the X-axis or the Z-axis.
N2 G01 A–a_2 Zz_2 Xx_2 (Setting Aa_3 means the same as setting A–a_2.)



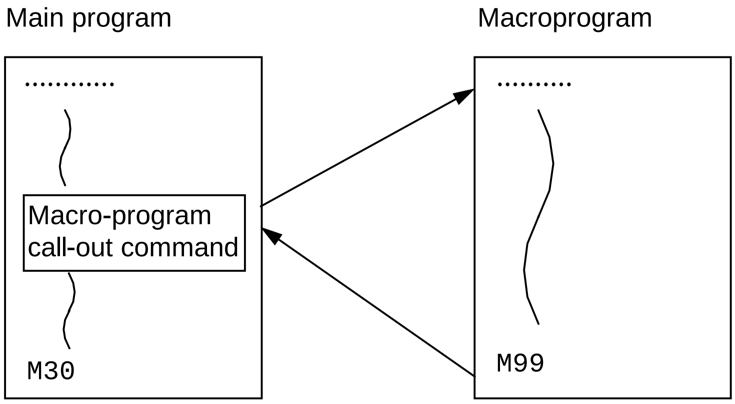
3. Detailed description

1. The angle denotes that relative to the plus (+) direction of the first axis (horizontal axis) on the selected plane.
   Assign the sign + for a counterclockwise direction (CCW) or the sign – for a clockwise direction (CW).
2. Set the ending point on one of the two axes of the selected plane.
3. Angle data will be ignored if the coordinates of both axes are set together with angles.
4. If angles alone are set, the command will be handled as a geometric command.
5. For the second block, the angle at either the starting point or the ending point can be specified.
6. The linear angle command function does not work if address A is to be used for an axis name or for the No. 2 auxiliary function.
7. This function is valid only for the G01 command; it is not valid for other interpolation or positioning commands.

14-14 Macro Call Function: G65, G66, G66.1, G67

14-14-1 User macros

Macroprogram call, data calculation, data input to/output from a personal computer, data control, judgment, branching, and various other instructions can be used with variables commands to perform measurements and other operations.



A macroprogram is a subprogram which is created using variables, calculation instructions, control instructions, etc. to have special control features.
These special control features (macroprograms) can be used by calling them from the main program as required. These calls use macro call instructions.

Detailed description

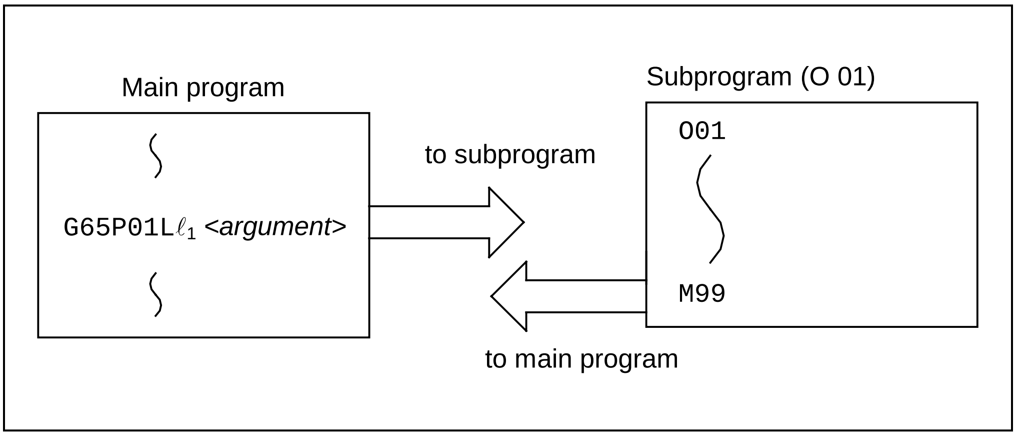
- When command G66 is entered, the designated user macro subprogram willbe called every time after execution of the move commands within a block until G67 (cancellation) is entered.
- Command codes G66 and G67 must reside in the same programm in pairs.

Return to Library

14-14-2 Macro call instructions

Two types of macro call instructions are provided: single-call instructions used to call only at the designated block, and modal call instructions used to call at each block within a macro call mode. Modal call instructions are further divided into type A and type B.

1. Single call



The designated user macro subprogram ends with M99.
Instruction G65 calls the designated user macro subprogram only once.

Format:
G65 <__> L__ *<argument>*

L__ — Repeat times
<__> — Program Name (When omitted, own program will be repeated.)

Alternatively,

G65 P__ L__ *<argument>*

L__ — Repeat times
P__ — Program No. (When P is omitted, own program will be repeated.)

<Argument>
When argument is to be delivered to the user macro subprogram as a local variable, designate the required data with the respective addresses. (Argument designation is not available for a user macro subprogram written in MAZATROL language.)
In such a case, the argument can have a sign and a decimal point, irrespective of the address. Arguments can be specified using method I or II, as shown below.

Return to Library

A. Argument specification I

Format: A_B_C_ ······· X_Y_Z_

Detailed description

- An argument can be specified using all addresses, except G, L, N, O, and P.
- Except for I, J, and K, addresses does not need be specified in an alphabetical order.
  I_J_K_ ... Correct
  J_I_K_ ... Wrong
- Addresses whose specification is not required can be omitted.
- The relationship between addresses that can be specified using argument specification I, and variables numbers in a user macro unit, is shown in the following table:

Relationship between address and variables number		Call commands and usable addresses	
Address specified using method I	Variable in macro-program	G65, G66	G66.1
A	#1	○	○
B	#2	○	○
C	#3	○	○
D	#7	○	○
E	#8	○	○
F	#9	○	○
G	#10	×	×*
H	#11	○	○
I	#4	○	○
J	#5	○	○
K	#6	○	○
L	#12	×	×*
M	#13	○	○
N	#14	×	×*
O	#15	×	×
P	#16	×	×*
Q	#17	○	○
R	#18	○	○
S	#19	○	○
T	#20	○	○
U	#21	○	○
V	#22	○	○
W	#23	○	○
X	#24	○	○
Y	#25	○	○
Z	#26	○	○

○: Usable ×: Unusable *: Usable in G66.1 modal

B. Argument specification II

Format: A_B_C_I_J_K_I_J_K_·····

<u>Detailed description</u>

- Up to a maximum of 10 sets of arguments that each consist of addresses I, J, and K, as well as A, B, and C, can be specified.
- If identical addresses overlap, specify them in the required order.
- Addresses whose specification is not required can be omitted.
- The relationship between addresses that can be specified using argument specification II, and variables numbers in a user macro unit, is shown in the following table:

Argument specification II addresses	Variables in macro-programs	Argument specification II addresses	Variables in macro-programs
A	#1	K5	#18
B	#2	I6	#19
C	#3	J6	#20
I1	#4	K6	#21
J1	#5	I7	#22
K1	#6	J7	#23
I2	#7	K7	#24
J2	#8	I8	#25
K2	#9	J8	#26
I3	#10	K8	#27
J3	#11	I9	#28
K3	#12	J9	#29
I4	#13	K9	#30
J4	#14	I10	#31
K4	#15	J10	#32
I5	#16	K10	#33
J5	#17		

Note: In the table above, the numerals 1 through 10 have been added to addresses I, J, and J just to denote the order of arrangement of the designated sets of arguments: these numerals are not included in actual instructions.

C. Combined use of argument specification I and II

When both method I and method II are used to specify arguments, only the latter of two arguments which have an address corresponding to the same variable will become valid.

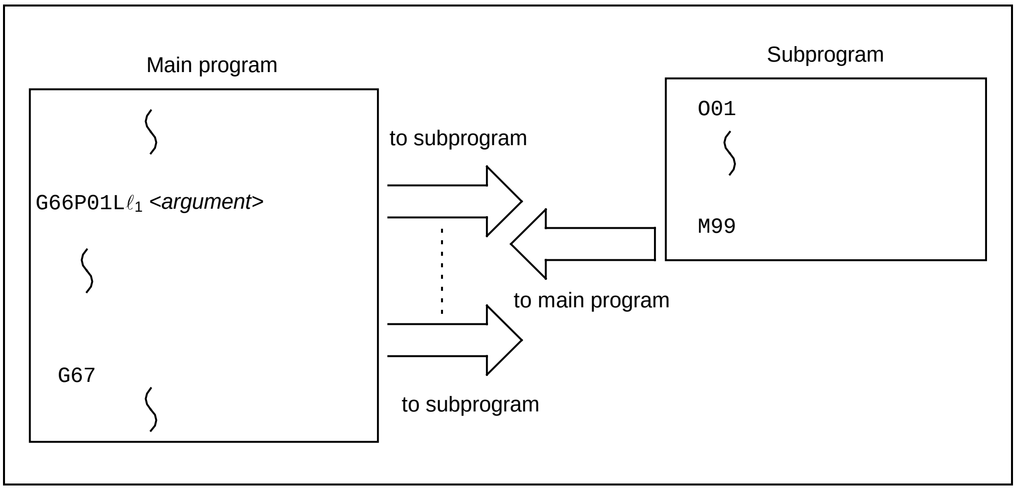
Example: Call command `G65 A1.1 B–2.2 D3.3 I4.4 I7.7`

Variables
#1: 1.1
#2: –2.2
#3:
#4: 4.4
#5:
#6:
#7: 7.7

If two arguments (D3.3 and I7.7) are designated for the variable of #7, only the latter argument (I7.7) will be used.

Return to Library

2. Modal call, type A (Move command call)



For a block that has a move command code between G66 and G67, the designated user macro subprogram is executed after that move command has been executed. The subprogram is executed an ℓ_1 number of times for the first call, or once for subsequent calls.
For modal call of type A, the methods of specifying <argument> are the same as used for single call.

Format:
G66 <__> L__ *<argument>*

- L__ : Repeat times
- <__> : Program name

Alternatively,

G66 P__ L__ *<argument>*

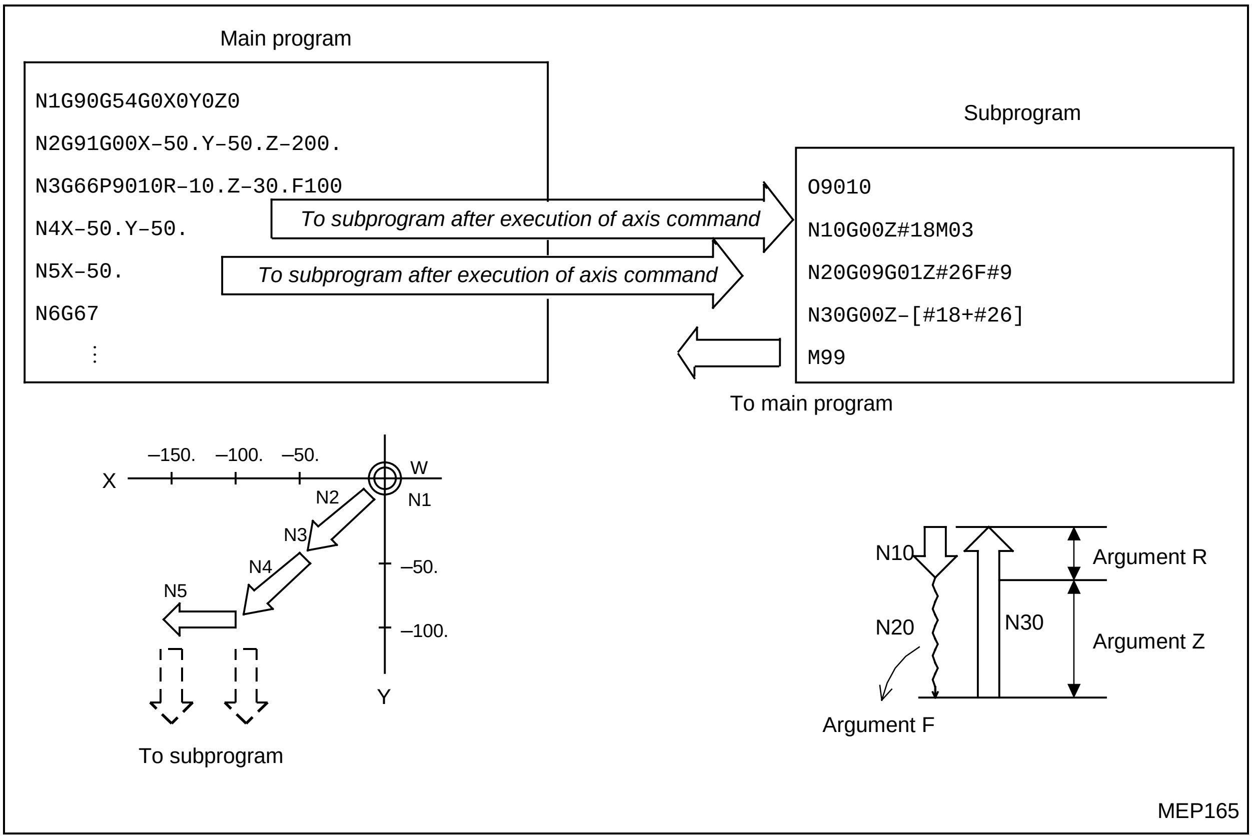
- L__ : Repeat times
- P__ : Program No.

<u>Detailed description</u>

- When command G66 is entered, the designated user macro subprogram will be called every time after execution of the move commands within a block until command G67 (cancellation) is entered.
- Command codes G66 and G67 must reside in the same program in pairs.
  Entry of a G67 command without a G66 command results in an alarm **857 INCORRECT USER MACRO G67 PROG**.

Return to Library

Drilling cycle



Note 1: The designated subprogram is executed after the axis commands in the main program have been executed.

Note 2: No subprograms are executed for the G67 block and its successors.

3. Modal call, type B (Block-to-block call)

The designated user macro subprogram is called unconditionally for each of the command blocks present between G66.1 and G67. Execution of the macro program is repeated as specified with L for the first call, and only once for each of subsequent calls.

Format:
G66.1 <__> L__ *<argument>*

- L__ — Repeat times
- <__> — Program name

Alternatively,

G66.1 P__ L__ *<argument>*

- L__ — Repeat times
- P__ — Program No.

Return to Library

Detailed description

- During the G66.1 mode, only the codes O, N, and G in each of the read command blocks are executed. No other codes in those blocks are executed; codes other than O, N, and G are handled as arguments. However, only the last G-code and the N-codes following a code other than O or N become arguments.
- All significant blocks in the G66.1 mode are regarded as preceded by the command G65P_.

  For example, the block of

  `N100G01G90X100. Y200. F400R1000`

  in the G66.1P1000 mode is handled as equivalent to

  `N100G65P1000G01G90X100. Y200. F400R1000.`

  Note: Call is executed even for the G66.1 command block of the G66.1 mode, with the relationship between the addresses of the arguments and the variables numbers being the same as for G65 (single call).
- The data range of the G, L, P, and N commands that you can set as new variables using the G66.1 mode is the same as the data range of usual NC commands.
- Sequence number N, modal G-codes, and O are all updated as modal information.

4. G-code macro call

The user macro subprograms of the required program number can be called just by setting G-codes.

Format:
G×× *<argument>*
G-code which calls macro-subprogram

Detailed description

- The instruction shown above performs the same function as those of the instructions listed below. Which of these listed instructions will apply is determined by the parameter data to be set for each G-code.

  M98PΔΔΔΔ

  G65PΔΔΔΔ *<argument>*

  G66PΔΔΔΔ *<argument>*

  G66.1PΔΔΔΔ *<argument>*
- Use parameters to set the relationship between G×× (macro call G-code) and PΔΔΔΔ (program number of the macro to be called).
- Of G00 through G255, up to a maximum of 10 command codes can be used with this instruction unless the uses of these codes are clearly predefined by EIA Standards, such as G00, G01, G02, etc.
- The command code cannot be included in user macro subprograms that have been called using G-codes.

Return to Library

5. Auxiliary command macro call (M-, S-, T-, or B-code macro call)

The user macro subprograms of the required program number can be called just by setting M-, S-, T-, or B-codes.

```
Format:
Mm (or Ss, Tt and Bb)
 └────────── M (or S, T and B) code which calls macro-subprogram
```

Detailed description (The following description also applies to S-, T-, and B-codes.)

- The instruction shown above performs the same function as those of the instructions listed below. Which of these listed instructions will apply is determined by the parameter data to be set for each M-code.

```
M98PΔΔΔΔ
G65PΔΔΔΔMm
G66PΔΔΔΔMm
G66.1PΔΔΔΔMm
```

- Use parameter to set the relationship between Mm (macro call M-code) and PΔΔΔΔ (program number of the macro to be called).
  Up to a maximum of 10 M-codes, ranging from M00 to M95, can be registered. Do not register the M-codes that are fundamentally required for your machine, nor M0, M1, M2, M30, and M96 through M99.
- If registered auxiliary command codes are set in the user macro subprograms that have been called using M-codes, macro calls will not occur since those special auxiliary command codes will be handled as usual ones (M-, S-, T-, or B-codes).

6. Differences in usage between commands M98, G65, etc.

- Arguments can be designated for G65, but cannot be designated for M98.
- Sequence numbers can be designated for M98, but cannot be designated for G65, G66, or G66.1.
- Command M98 executes a subprogram after M98 block commands other than M, P, H, and L have been executed, whereas G65 just branches the program into a subprogram without doing anything.
- Single-block stop will occur if the block of command M98 has addresses other than O, N, P, H, and L. For G65, however, single-block stop will not occur.
- The level of local variables is fixed for M98, but for G65 does change according to the depth of nesting. (For example, #1s, if present before and after M98, always mean the same, but if present before and after G65, they have different meanings.)
- Command M98 can have up to a maximum of eight levels of call multiplexity when combined with G65, G66, or G66.1, whereas the maximum available number of levels for command G65 is four when it is combined with G66 or G66.1.

7. Multiplexity of macro call commands

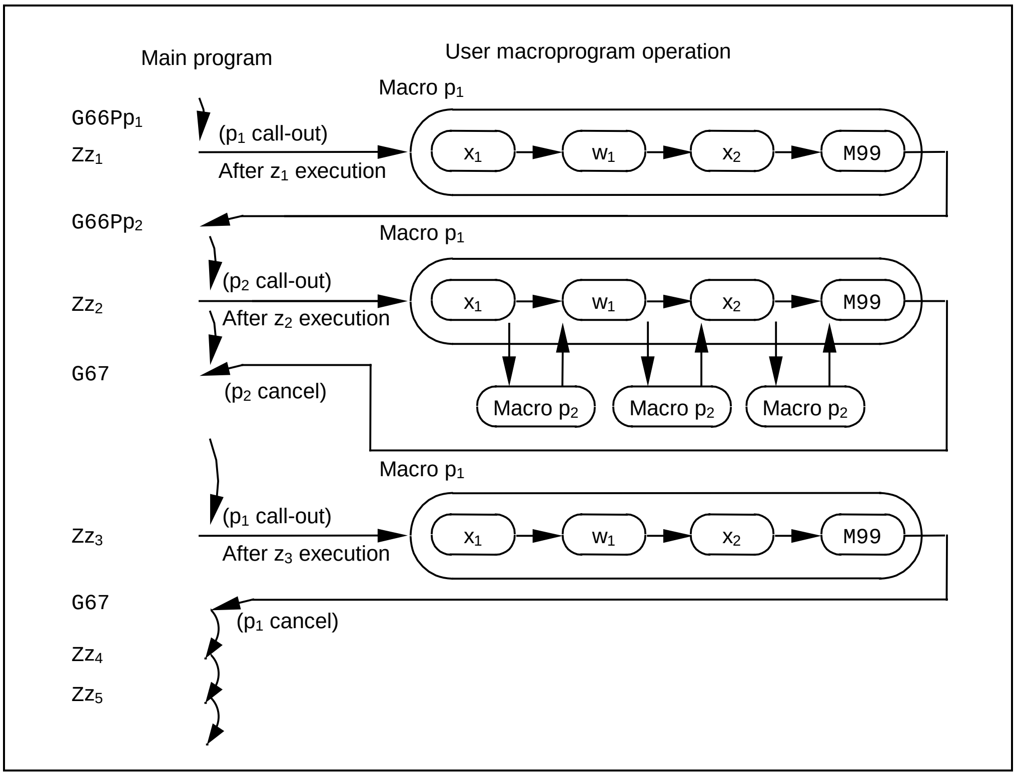
The maximum available number of levels of macro subprogram call is four, whether it is single or modal. Arguments in macro call instructions become valid only within the level of the called macro. Since the multiplexity of macro call is of up to a maximum of four levels, arguments can be included in a program as local variables each time a macro call is made.

Note 1: When a G65, G66, or G66.1 macro call or an auxiliary command macro call is made, nesting will be regarded as single-level and thus the level of local variables will also increase by 1.

Note 2: For modal call of type A, the designated user macro subprogram is called each time a move command is executed. If, however, multiple G66s are present, the next user macro subprogram will be called even for the move commands in the macro each time axis movement is done.

Note 3: User macro subprograms are cancelled in a reverse order to that in which they have been arranged.

Example:



Return to Library

8. User macro call based on interruption

Outline

Prior creation of special user macros for interrupt processing allows the user macros to be executed during automatic operation when a user macro interrupt signal is input. After the user macro has been executed, the program can also be returned to the interrupted program block and then started from this block.

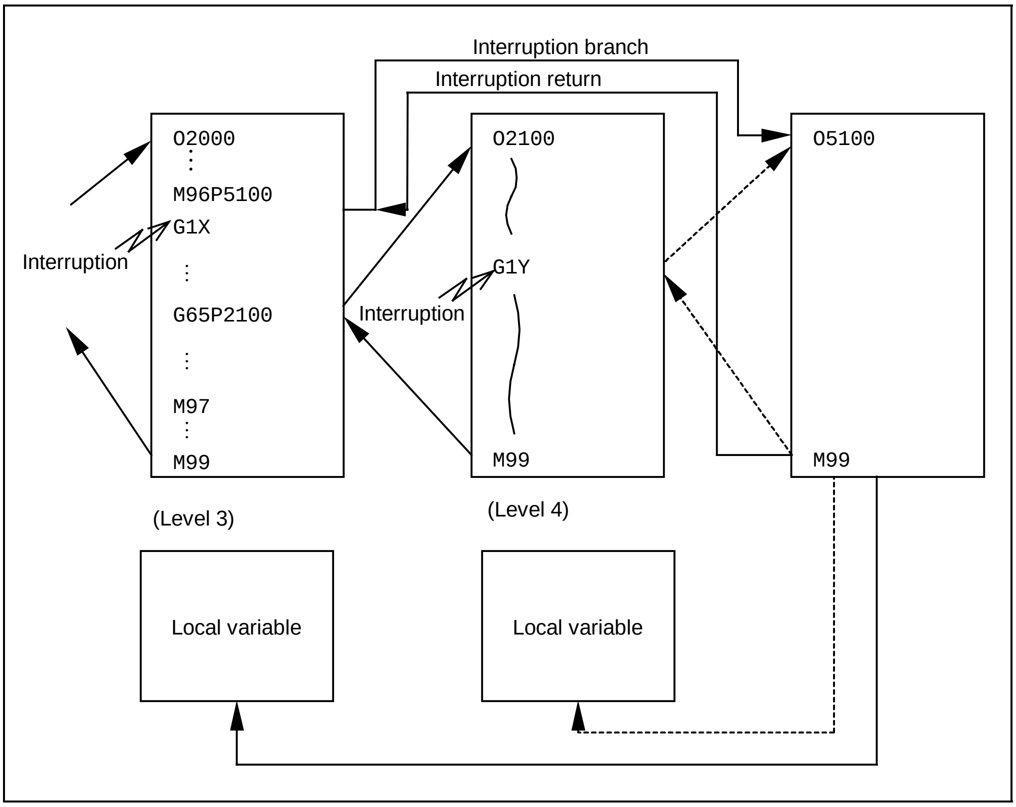
Detailed description

- Format for selecting the user macro branching destination

⋮
M96<_>L_ (or M96P_L_)
⋮
⋮
M97 (Branching mode off)
⋮

(Branching mode on)
When user macroprogram interruption signal is input during this space, the branch into the specified user macroprogram will be applied.

- User macro interrupts can be processed even when the number of levels of macro call multiplexity during the occurrence of an interrupt is four. The local variables' level of the user macros used for interruption is the same as the level of the user macros existing during the occurrence of an interrupt.



14-14-3 Variables

Of all types of variables available for the NC unit, only local variables, common variables, and part of system variables are retained even after power-off.

1. Multiplexing of variables

Under user macro specifications, variables can have their identifiers (identification numbers) either transformed into variables, which is referred to as multiplexing, or replaced with <expression>.
For <expression>, only one arithmetic expression (for either multiplication, division, addition, or subtraction) can be used.

Example 1: Multiplexing variables

`#1=10 #10=20 #20=30`	From #1 = 10, #[#[#1]] = #[#10] will result.
`#5=#[#[#1]]`	From #10 = 20, #[#10] = #20 will result. Therefore #5 = #20, i.e. #5 = 30 will result.
`#1=10 #10=20 #20=30`	From #1 = 10, #[#[#1]] = #[#10] will result.
`#5=1000`	From #10 = 20, #[#10] = #20 will result. Therefore #20 = #5, i.e.
`#[#[#1]]=#5`	#20 = 1000 will result.

Example 2: Replacing variables identifiers with <expression>

`#10=5`	
`#[#10+1]=1000`	#6 = 1000 will result.
`#[#10–1]=–1000`	#4 = –1000 will result.
`#[#10*3]=100`	#15 = 100 will result.
`#[#10/2]=100`	#2 = –100 will result.

2. Undefined variables

Under user macro specifications, variables remaining unused after power-on or local variables that are not argument-specified by G65, G66, or G66.1 can be used as <empty>. Also, variables can be forcibly made into <empty>.
Variable #0 is always used as <empty> one, and this variable cannot be defined on the left side of the expression.

A. Arithmetic expression

`#1=#0`.................#1 = <empty>

`#2=#0+1` ....#2 = 1

`#3=1+#0` ....#3 = 1

`#4=#0*10` ....#4 = 0

`#5=#0+#0` ....#5 = 0

Note: Be careful that <empty> is handled the same as 0 during processing of expressions.

<empty> + <empty> = 0

<empty> + <constant> = constant

<constant> + <empty> = constant

B. Applying variables

Application of an undefined variable alone causes even the address to be ignored.

If #1 = <empty>

`G0X#1Y1000` is equivalent to G0Y1000, and
`G0X[#1+10]Y1000` is equivalent to G0X10Y1000.

C. Conditional expression

Only for EQ and NE, does <empty> differ from 0 in meaning.

If #101 = <empty>		If #101 = 0	
`#101EQ#0`	<empty> = <empty> holds.	`#101EQ#0`	0 = <empty> does not hold.
`#101NE0`	<empty> ≠ 0 holds.	`#101NE0`	0 ≠ 0 does not hold.
`#101GE#0`	<empty> ≥ <empty> holds.	`#101GE#0`	0 ≥ <empty> holds.
`#101GT0`	<empty> > 0 does not hold.	`#101GT0`	0 > 0 does not hold.

Hold-conditions and not-hold-conditions list
(For conditional expressions including undefined variables)

Right side / Left side	EQ		NE		GT		LT		GE		LE	
	Empty	Constant	Empty	Constant	Empty	Constant	Empty	Constant	Empty	Constant	Empty	Constant
Empty	H			H				H	H		H	
Constant			H		H				H			

H: Holds (The conditional expression holds.)
Blank: The conditional expression does not hold.

Return to Library

14-14-4 Types of variables

1. Common variables (#100 to #199, and #500 to #999)

Common variables refer to the variables to be used in common at any position. The identifiers of common variables which can be used are from #100 to #199, or from #500 to #999.

2. Local variables (#1 to #33)

Local variables refer to variables that can be defined as <argument> when calling a macro subprogram, or those which can be used locally within the main program or a subprogram. There is no relationship between macros. Thus, these variables can be overlapped on each other, but up to a maximum of four levels of overlapping.

$\boxed{\text{G65P}p_1\text{L}\ell_1\ \textit{<argument>}}$

where p_1 : Program number

ℓ_1 : Number of repeat times

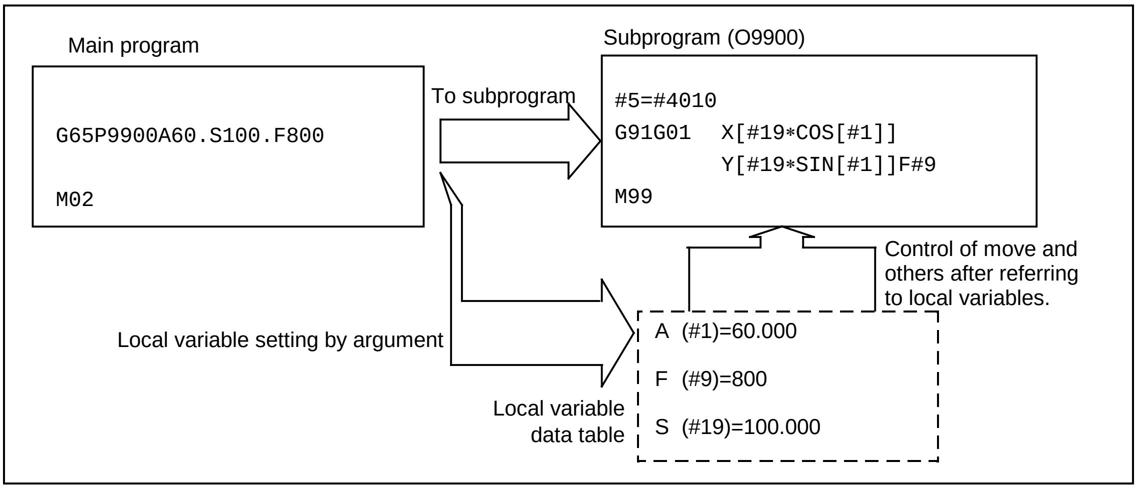
<Argument> must be: $Aa_1\ Bb_1\ Cc_1 \cdots Zz_1$.

The following represents the relationship between the address specified by <argument> and the local variables number used in the user macro unit:

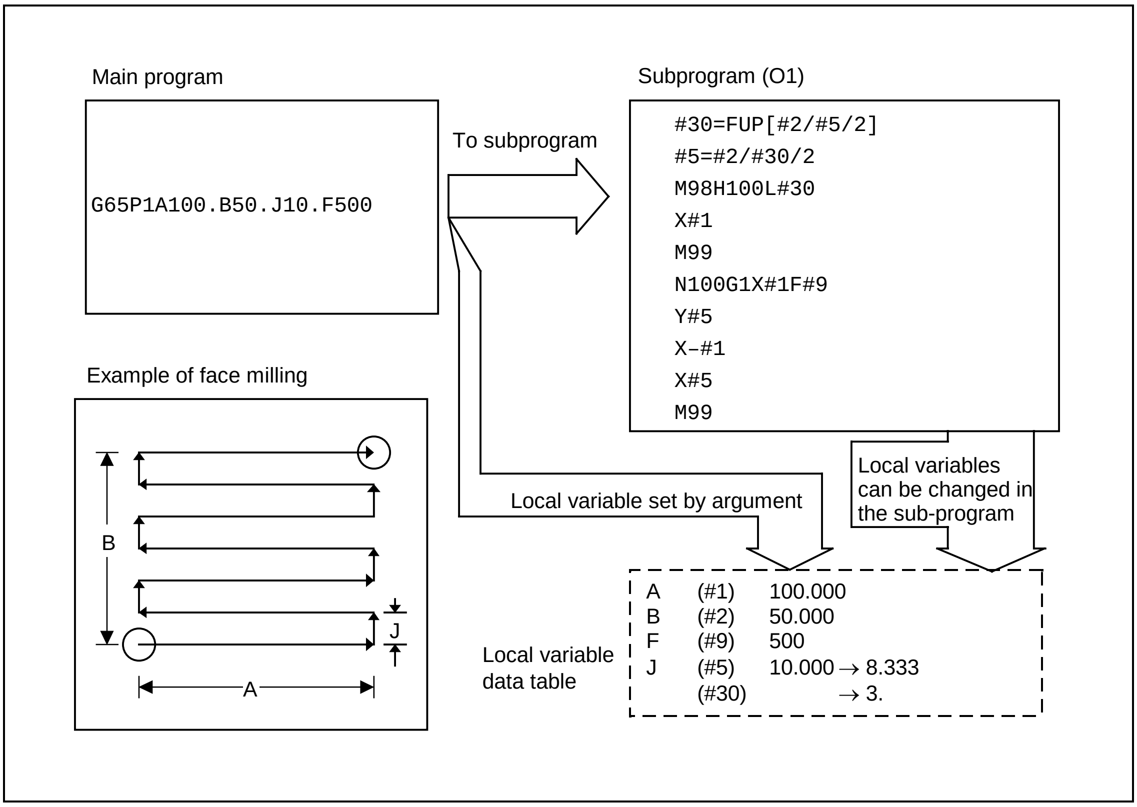
Call commands		Argument address	Local variable	Call commands		Argument address	Local variable
G65 G66	G66.1			G65 G66	G66.1		
○	○	A	#1	○	○	R	#18
○	○	B	#2	○	○	S	#19
○	○	C	#3	○	○	T	#20
○	○	D	#7	○	○	U	#21
○	○	E	#8	○	○	V	#22
○	○	F	#9	○	○	W	#23
×	×*	G	#10	○	○	X	#24
○	○	H	#11	○	○	Y	#25
○	○	I	#4	○	○	Z	#26
○	○	J	#5			–	#27
○	○	K	#6			–	#28
×	×*	L	#12			–	#29
○	○	M	#13			–	#30
×	×*	N	#14			–	#31
×	×	O	#15			–	#32
×	×*	P	#16			–	#33
○	○	Q	#17				

Argument addresses marked as × in the table above cannot be used. Only during the G66.1 mode, however, can argument addresses marked with an asterisk (*) in this table be additionally used. Also, the dash sign (–) indicates that no address is crosskeyed to the local variables number.

1. Local variables for a subprogram can be defined by specifying <argument> when calling a macro.



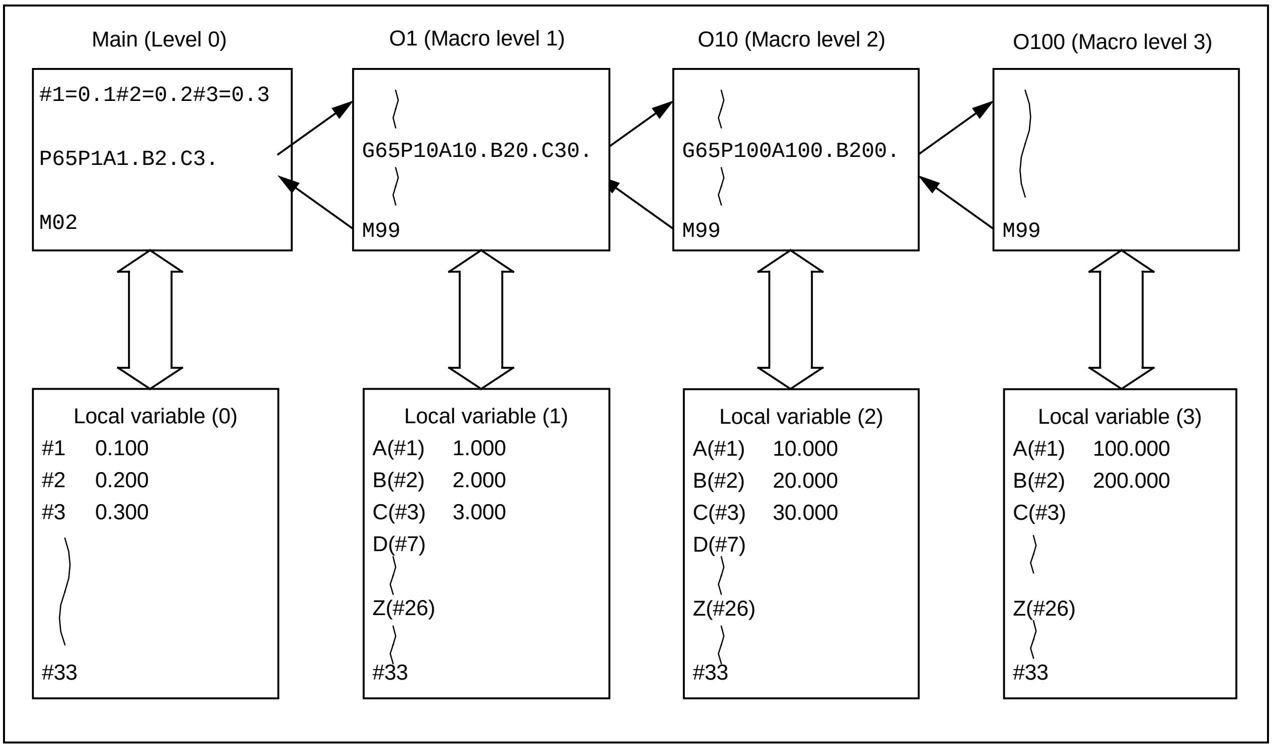
2. Within a subprogram, local variables can be freely used.



In the sample program for face-milling that is shown above, although the argument J has initially been programmed as a machining pitch of 10 mm, it has been changed into 8.333 mm to ensure equal-pitched machining.

Also, local variable #30 contains the calculated data about the number of times of reciprocal machining.

3. Local variables can be used for each of the four levels of macro call separately. For the main program (macro level 0), separate local variables are also provided. The local variables of level 0, however, cannot be designated with arguments.



How the local variables are currently being used is displayed on the screen.
For further details, refer to the Operating Manual.

3. Macro interface input system variables (#1000 to #1035)

You can check the status of an interface input signal by reading the value of the appropriate variables number (#1000 to #1035).

The read value of the variables number is either 1 (contact closed) or 0 (contact open). You can also check the status of all input signals of the variables from #1000 to #1031 by reading the value of variables number 1032. Variables from #1000 to #1035 can only be read; they cannot be placed on the left side of an arithmetic expression.

System variable	Points	Interface input signal	System variable	Points	Interface input signal
#1000	1	Register R72, bit 0	#1016	1	Register R73, bit 0
#1001	1	Register R72, bit 1	#1017	1	Register R73, bit 1
#1002	1	Register R72, bit 2	#1018	1	Register R73, bit 2
#1003	1	Register R72, bit 3	#1019	1	Register R73, bit 3
#1004	1	Register R72, bit 4	#1020	1	Register R73, bit 4
#1005	1	Register R72, bit 5	#1021	1	Register R73, bit 5
#1006	1	Register R72, bit 6	#1022	1	Register R73, bit 6
#1007	1	Register R72, bit 7	#1023	1	Register R73, bit 7
#1008	1	Register R72, bit 8	#1024	1	Register R73, bit 8
#1009	1	Register R72, bit 9	#1025	1	Register R73, bit 9
#1010	1	Register R72, bit 10	#1026	1	Register R73, bit 10
#1011	1	Register R72, bit 11	#1027	1	Register R73, bit 11
#1012	1	Register R72, bit 12	#1028	1	Register R73, bit 12
#1013	1	Register R72, bit 13	#1029	1	Register R73, bit 13
#1014	1	Register R72, bit 14	#1030	1	Register R73, bit 14
#1015	1	Register R72, bit 15	#1031	1	Register R73, bit 15

System variable	Points	Interface input signal
#1032	32	Register R72 and R73
#1033	32	Register R74 and R75
#1034	32	Register R76 and R77
#1035	32	Register R78 and R79

Note: The following interface input signals are used exclusively in the NC system operation (cannot be used for other purposes).

Interface input signal	Description
Register R72, bit 0	Touch sensor mounted in the spindle
Register R72, bit 4	X- and Y-axis machine lock ON
Register R72, bit 5	M-, S-, T-code lock ON
Register R72, bit 6	Z-axis machine lock ON

4. Macro interface output system variables (#1100 to #1135)

You can send an interface output signal by assigning a value to the appropriate variables number (#1100 to #1135).

All output signals can take either 0 or 1.

You can also send all output signals of the variables from #1100 to #1131 at the same time by assigning a value to variables number 1132. In addition to the data writing for offsetting the #1100 to #1135 output signals, the reading of the output signal status can be done.

System variable	Points	Interface output signal	System variable	Points	Interface output signal
#1100	1	Register R172, bit 0	#1116	1	Register R173, bit 0
#1101	1	Register R172, bit 1	#1117	1	Register R173, bit 1
#1102	1	Register R172, bit 2	#1118	1	Register R173, bit 2
#1103	1	Register R172, bit 3	#1119	1	Register R173, bit 3
#1104	1	Register R172, bit 4	#1120	1	Register R173, bit 4
#1105	1	Register R172, bit 5	#1121	1	Register R173, bit 5
#1106	1	Register R172, bit 6	#1122	1	Register R173, bit 6
#1107	1	Register R172, bit 7	#1123	1	Register R173, bit 7
#1108	1	Register R172, bit 8	#1124	1	Register R173, bit 8
#1109	1	Register R172, bit 9	#1125	1	Register R173, bit 9
#1110	1	Register R172, bit 10	#1126	1	Register R173, bit 10
#1111	1	Register R172, bit 11	#1127	1	Register R173, bit 11
#1112	1	Register R172, bit 12	#1128	1	Register R173, bit 12
#1113	1	Register R172, bit 13	#1129	1	Register R173, bit 13
#1114	1	Register R172, bit 14	#1130	1	Register R173, bit 14
#1115	1	Register R172, bit 15	#1131	1	Register R173, bit 15

System variable	Points	Interface output signal
#1132	32	Register R172 and R173
#1133	32	Register R174 and R175
#1134	32	Register R176 and R177
#1135	32	Register R178 and R179

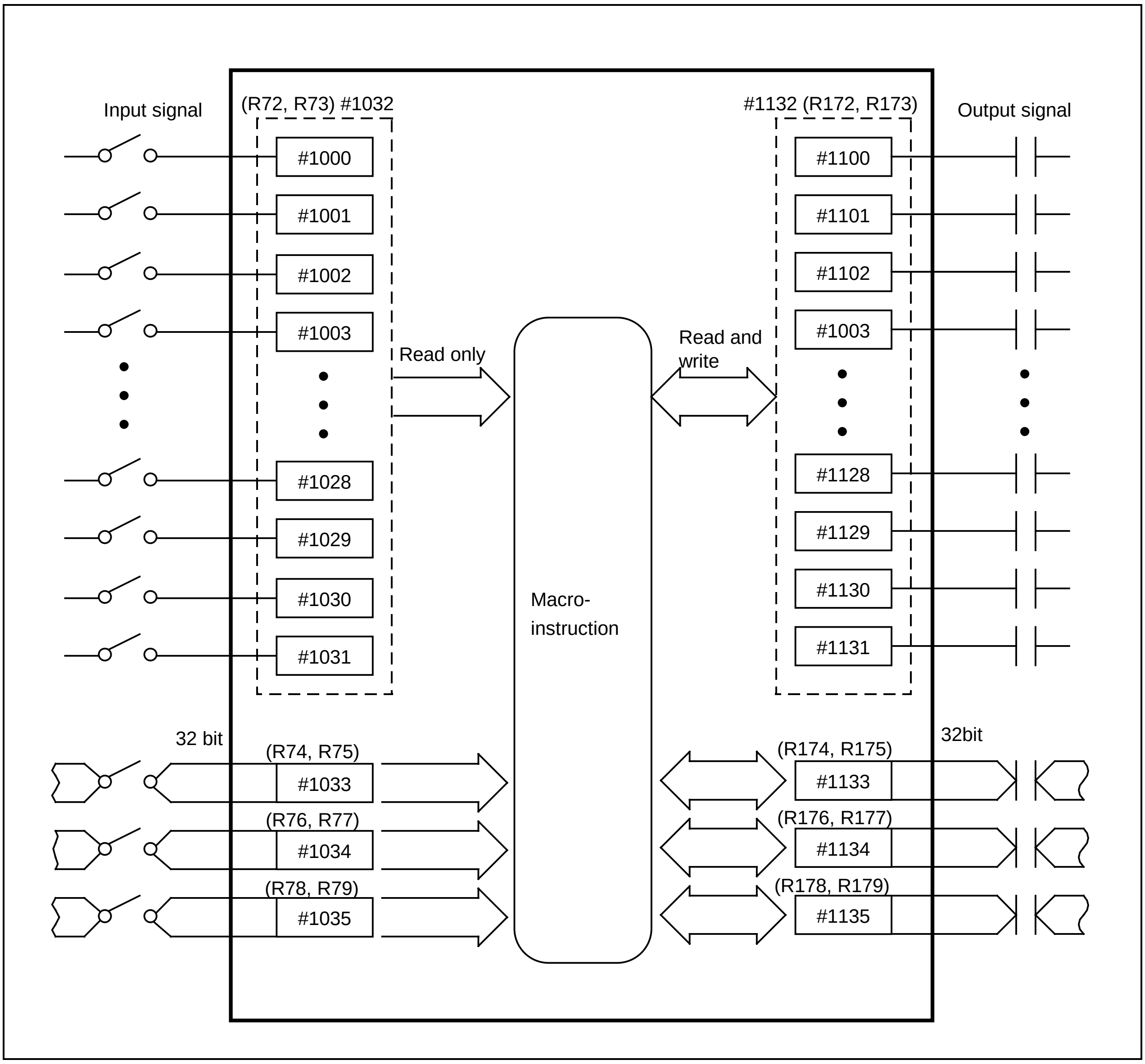
Note 1: Data of the system variables from #1100 to #1135 is saved according to the logical level (1 or 0) of the signal that has been lastly sent. The saved data is cleared by power-on/off automatically.

Note 2: The following applies if a data other than 1 or 0 is assigned to the variables from #1100 to #1131:

<empty> is regarded as equal to 0.

Data other than 0 and <empty> is regarded as equal to 1.

Data less than 0.00000001, however, is regarded as undefined.


Input signal
(R72, R73) #1032
#1132 (R172, R173)
Output signal
#1000
#1001
#1002
#1003
#1028
#1029
#1030
#1031
#1100
#1101
#1102
#1003
#1128
#1129
#1130
#1131
Read only
Read and write
Macro-instruction
32 bit
(R74, R75)
#1033
(R76, R77)
#1034
(R78, R79)
#1035
32bit
(R174, R175)
#1133
(R176, R177)
#1134
(R178, R179)
#1135

5. Tool offset

Standard 128 sets
Optional 512 sets

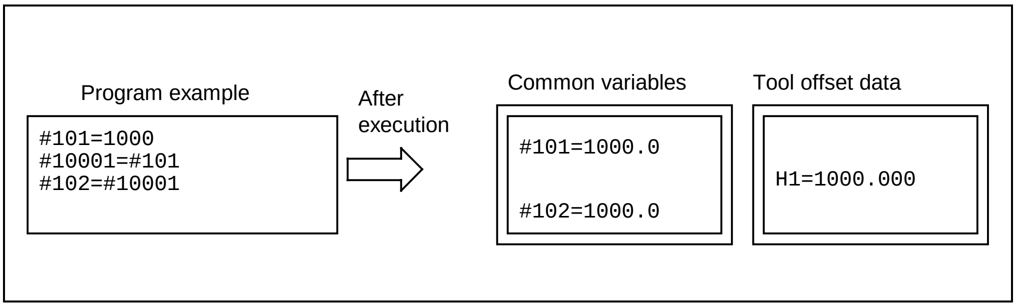
Range of variable Nos.		Type A	Type B	Type C
#10001 - #10000+n	#2001 - #2000+n	O	O Length Geom. offset	O Geom. offset Z
#11001 - #11000+n	#2201 - #2200+n	×	O Length Wear comp.	O Wear comp. Z
#16001 - #16000+n *(#12001 - #12000+n)	#2401 - #2400+n	×	O Dia. Geom. offset	O Geom. offset Nose-R
#17001 - #17000+n *(#13001 - #13000+n)	#2601 - #2600+n	×	O Dia. Wear comp.	O Wear comp. Nose-R
#12001 - #12000+n		×	×	O Geom. offset X
#13001 - #13000+n		×	×	O Wear comp. X
#14001 - #14000+n		×	×	O Geom. offset Y
#15001 - #15000+n		×	×	O Wear comp. Y

*: The numbers of variables used for tool offset depend upon a parameter:

F96 bit 0 = 0: #16001 to #16000+n, and #17001 to #17000+n
= 1: #12001 to #12000+n, and #13001 to #13000+n.

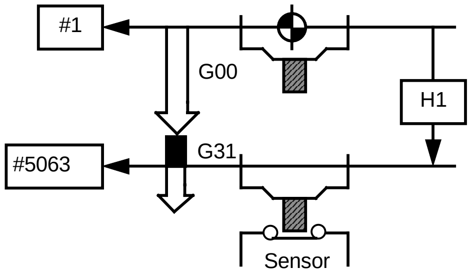
Note: Set bit 0 of parameter **F96** to "0" to use the TOOL OFFSET type C data.

Using variables numbers, you can read tool data or assign data.
Usable variables numbers are of the order of either #10000 or #2000. For the order of #2000, however, only up to 200 sets of tool offsets can be used.
The last three digits of a variables number denote a tool offset number.
As with other variables, tool offset data is to contain the decimal point. The decimal point must therefore be included if you want to set data that has decimal digits.



Example: Tool offset data measuring

```
G28Z0T01              Return to zero point
M06                   Tool change (Spindle T01)
#1=#5003              Starting point memory
G00Z-500.             Rapid feed to safe position
G31Z-100.F100         Skip measuring
#10001=#5063-#1       Measuring distance calculation
                      and tool offset data setting
  :
```



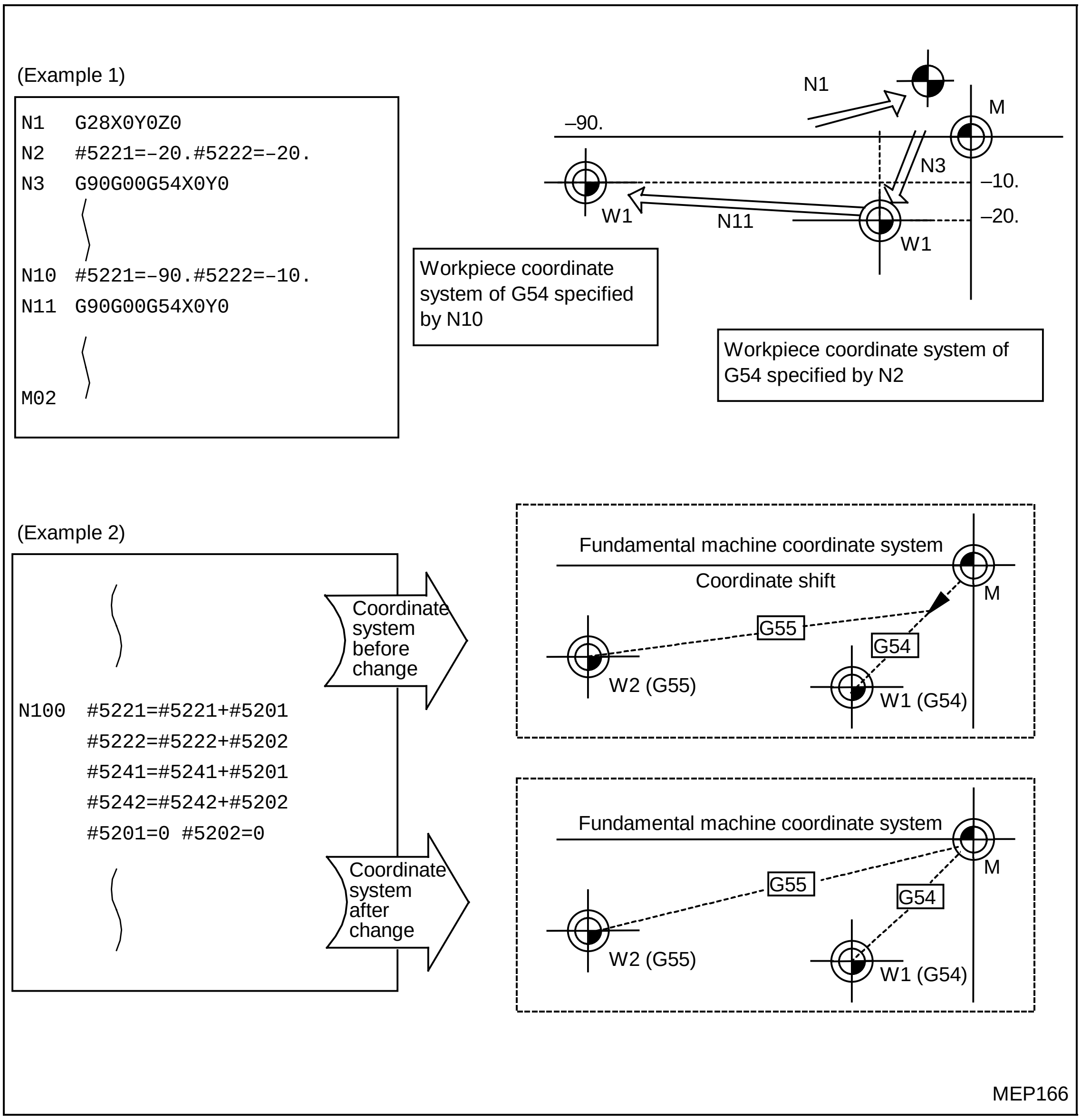
Note: The example shown above does not allow for any skip sensor signal delay. Also, #5003 denotes the position of the starting point of the Z-axis, and #5063 denotes the skip coordinate of the Z-axis, that is, the position at which a skip signal was input during execution of G31.

6. Workpiece coordinate system offset

Using variables numbers from 5201 to 5334, you can read workpiece coordinate system offset data or assign data.

Note: The number of controllable axes depends on the machine specifications.

Axis No. / Data name	1st axis	2nd axis	3rd axis		14th axis	Remarks
SHIFT	#5201	#5202	#5203		#5214	An external data input/output optional spec. is required.
G54	#5221	#5222	#5223		#5234	A workpiece coordinate system offset feature is required.
G55	#5241	#5242	#5243		#5254	
G56	#5261	#5262	#5263		#5274	
G57	#5281	#5282	#5283		#5294	
G58	#5301	#5302	#5303		#5314	
G59	#5321	#5322	#5323		#5334	



The example 2 shown above applies only when coordinate shift data is to be added to the offset data of a workpiece coordinate system (G54 or G55) without changing the position of the workpiece coordinate system.

[Additional workpiece coordinate system offset]

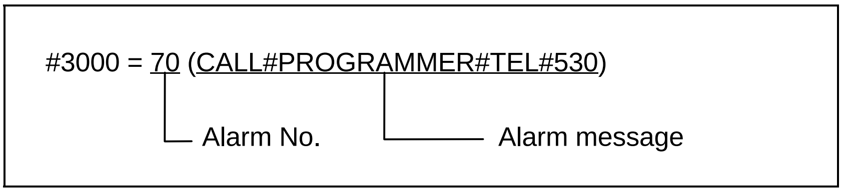
Variables numbered 7001 to 7954 can be used to read or assign additional workpiece coordinate system offsetting dimensions.

Note: The total number of controllable axes depends on the machine specifications.

Axis No. / Data name	1st axis	2nd axis	3rd axis	4th axis		14th axis	Remarks
G54.1P1	#7001	#7002	#7003	#7004		#7014	Only available with the optional function for additional coordinate system offset.
G54.1P2	#7021	#7022	#7023	#7024		#7034	
G54.1P3	#7041	#7042	#7043	#7044		#7054	
G54.1P48	#7941	#7942	#7943	#7944		#7954	

7. NC alarm (#3000)

The NC unit can be forced into an alarm status using variables number 3000.



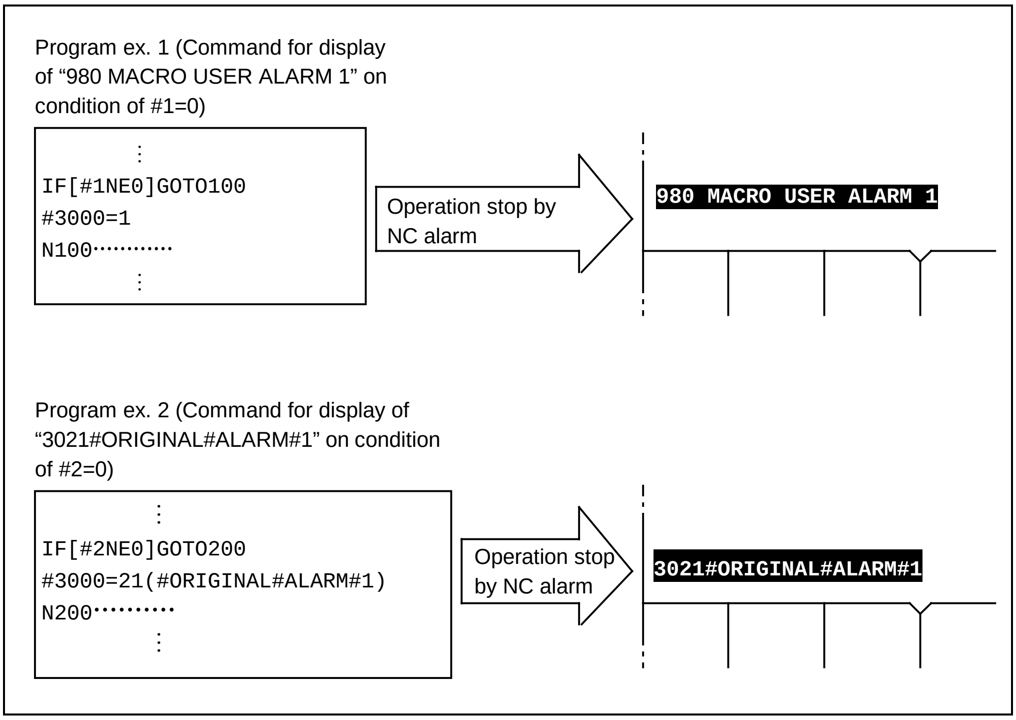
The setting range for the alarm No. is from 1 to 6999.
The maximum available length of the alarm message is 31 characters.

Note: The type of alarm message displayed on the screen depends on the designated alarm number, as indicated in the following table.

Designated alarm No.	Displayed alarm No.	Displayed alarm message
1 to 20	[Designated alarm No.] + 979	Message preset for the displayed alarm No. *1
21 to 6999	[Designated alarm No.] + 3000	Designated alarm message as it is *2

*1 Refers to alarm Nos. 980 to 999 whose messages are preset as indicated in Alarm List.

*2 Display of a message as it is set in the macro statement.

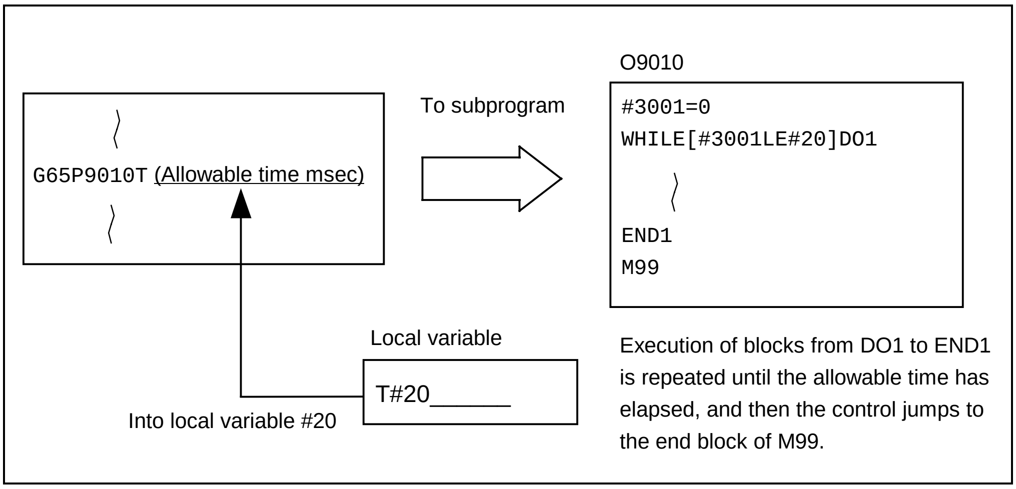


8. Integrated time (#3001, #3002)

Using variables #3001 and #3002, you can read the integrated time existing during automatic operation or assign data.

Type	Variable No.	Unit	Data at power-on	Initialization	Counting
Integrated time 1	3001	msec	Same as at power-off	Data is assigned in variables.	Always during power-on
Integrated time 2	3002	msec	Same as at power-off	Data is assigned in variables.	During auto-starting

The integrated time is cleared to 0 after having reached about 2.44×10^{11} msec (about 7.7 years).



9. Validation/invalidation of single-block stop or auxiliary-function finish signal wait (#3003)

Assigning one of the values listed in the table below to variables number 3003 allows single-block stop to be made invalid at subsequent blocks or the program to be advanced to the next block without ever having to wait for the arrival of an auxiliary-function code (M, S, T, or B) execution finish signal (FIN).

#3003	Single block stop	Auxiliary-function completion signal
0	Effective	Wait
1	Ineffective	Wait
2	Effective	No wait
3	Ineffective	No wait

Note: Variable #3003 is cleared to 0 by resetting.

10. Validation/invalidation of feed hold, feed rate override, or G09 (#3004)

Feed hold, feed rate override, or G09 can be made valid or invalid for subsequent blocks by assigning one of the values listed in the table below to variables number 3004.

#3004	Bit 0	Bit 1	Bit 2
Contents (Value)	Feed hold	Feed rate override	G09 check
0	Effective	Effective	Effective
1	Ineffective	Effective	Effective
2	Effective	Ineffective	Effective
3	Ineffective	Ineffective	Effective
4	Effective	Effective	Ineffective
5	Ineffective	Effective	Ineffective
6	Effective	Ineffective	Ineffective
7	Ineffective	Ineffective	Ineffective

Note 1: Variable #3004 is cleared to 0 by resetting.

Note 2: Each of the listed bits makes the function valid if 0, or invalid if 1.

11. Program stop (#3006)

Use of variables number 3006 allows the program to be stopped after execution of the immediately preceding block.

Format:
#3006 = 1 (CHECK OPERAT)
Character string to be displayed

Additional setting of a character string (in 29 characters at maximum) in parentheses allows the required stop message to be displayed on the monitor.

12. Mirror image (#3007)

The mirror image status of each axis at one particular time can be checked by reading variables number 3007.
Variable #3007 has its each bit crosskeyed to an axis, and these bits indicate that:
If 0, the mirror image is invalid.
If 1, the mirror image is valid.

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Axis no.											6	5	4	3	2	1

13. G-command modal status

The G-command modal status in a pre-read block can be checked using variables numbers from 4001 to 4021. For variables numbers from #4201 to #4221, the modal status of the block being executed can be checked in a similar manner to that described above.

Variable Nos.		Function	
Block pre-read	Block executed		
#4001	#4201	Interpolation mode	G00-G03:0-3, G2.1:2.1, G3.1:3.1, G33:33
#4002	#4202	Plane selection	G17:17, G18:18, G19:19
#4003	#4203	Programmed software limit	G22:22, G23:23
#4004	#4204	Feed specification	G98:98, G99:99
#4005	#4205	Inch/metric	G20:20, G21:21
#4006	#4206	Tool radius compensation	G40:40, G41:41, G42:42
#4007	#4207	Fixed cycle	G80:80, G73/74:73/74, G76:76, G81-G89:81-89
#4008	#4208	Workpiece coordinate system	G54-G59:54-59, G54.1:54.1
#4009	#4209	Acceleration/Deceleration	G61-G64:61-64
#4010	#4210	Macro modal call	G66:66, G66.1: 66.1, G67:67

Return to Library

14. Other modal information

Modal information about factors other than the G-command modal status in a pre-read block can be checked using variables numbers from 4101 to 4130. For variables numbers from #4301 to #4330, the modal information of the block being executed can be checked in a similar manner to that described above.

Variable Nos.		Modal information	Variable Nos.		Modal information
Preread	Execution		Preread	Execution	
#4101	#4301		#4112	#4312	
#4102	#4302	No. 2 miscellaneous function···B	#4113	#4313	Miscellaneous function···M
#4103	#4303		#4314	#4114	Sequence No.···N
#4104	#4304		#4115	#4315	Program No.···O
#4105	#4305		#4116	#4316	
#4106	#4306		#4117	#4317	
#4107	#4307	Tool diameter offset No···D	#4118	#4318	
#4108	#4308		#4119	#4319	Spindle function···S
#4109	#4309	Feed rate···F	#4120	#4320	Tool function···T
#4110	#4310		#4130	#4330	Addt. Workpiece coordinate system G54-G59:0, G54.1P1-P48:1-48
#4111	#4311	Tool length offset No···H			

15. Position information

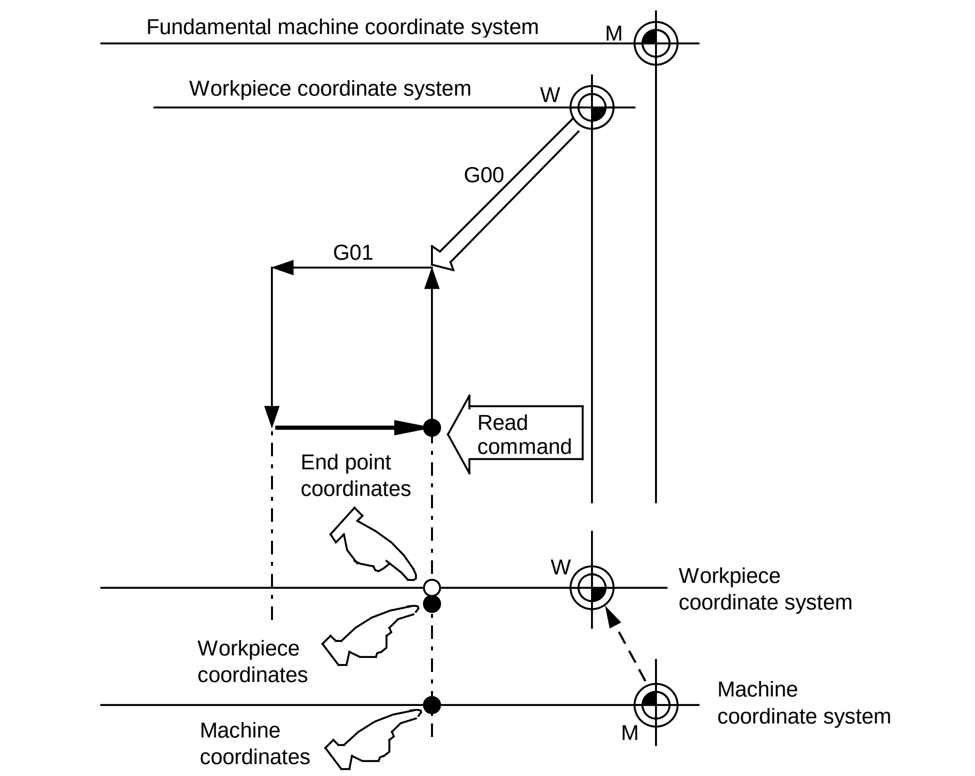
Using variables numbers from #5001 to #5114, you can check the ending-point coordinates of the previous block, machine coordinates, workpiece coordinates, skip coordinates, tool position offset coordinates, and servo deviations.

Position information / Axis. No.	End point coordinates of previous block	Machine coordinate	Workpiece coordinate	Skip coordinate	Tool position offset coordinates	Servo deviation amount
1	#5001	#5021	#5041	#5061	#5081	#5101
2	#5002	#5022	#5042	#5062	#5082	#5102
3	#5003	#5023	#5043	#5063	#5083	#5103
14	#5014	#5034	#5054	#5074	#5094	#5114
Remarks (Reading during move)	Possible	Impossible	Impossible	Possible	Impossible	Possible

Note: The number of controllable axes depends on the machine specifications.

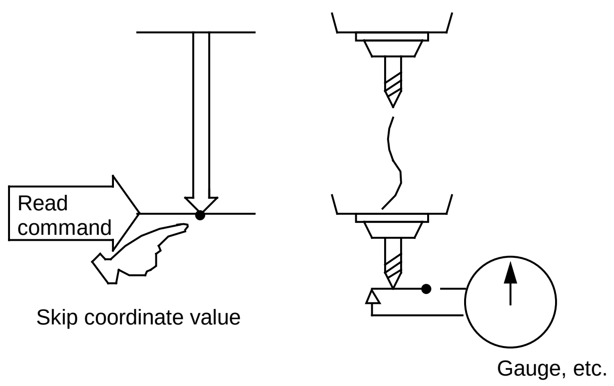
1. The ending-point coordinates and skip coordinates read will be those related to the workpiece coordinate system.

2. Ending-point coordinates, skip coordinates, and servo deviations can be checked even during movement. Machine coordinates, workpiece coordinates, and tool position offset coordinates must be checked only after movement has stopped.



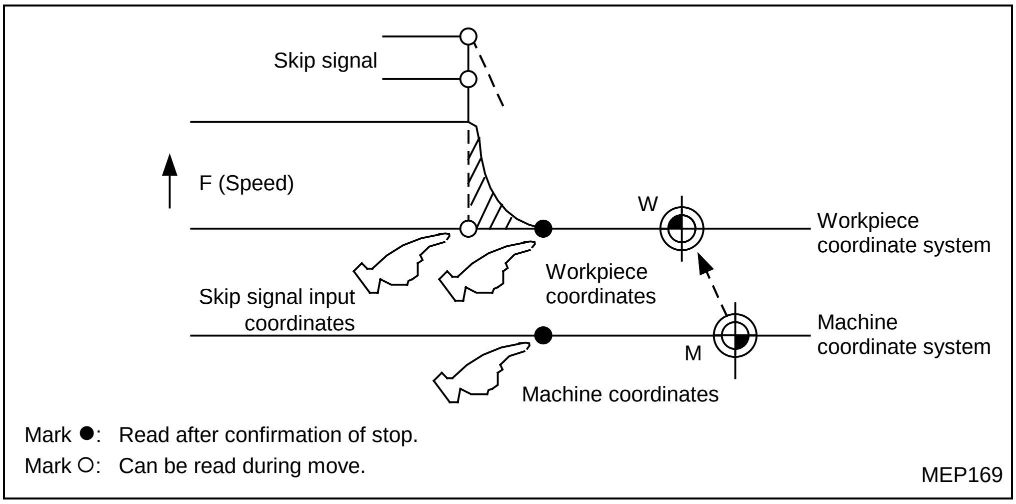
MEP167

3. Skip coordinates denote the position at which a skip signal has turned on at the block of G31. If the skip signal has not turned on, skip coordinates will denote the corresponding ending-point position.



MEP168

4. The ending-point position denotes the tool tip position which does not allow for any tool offsets, whereas machine coordinates, workpiece coordinates, and skip coordinates denote the tool reference-point position which allows for tool offsets.



MEP169

The input coordinates of a skip signal denote the position within the workpiece coordinate system. The coordinates stored in variables from #5061 to #5066 are those existing when skip signals were input during movement of the machine. These coordinates can therefore be read at any time after that. See the section (Chapter 16) on skip functions for further details.

Example 1: Workpiece position measurement:

The following shows an example of measuring the distance from a reference measurement point to the workpiece end:

```
G65P9031X100.Y100.Z-10.F200
```

Argument (Local variable)

F	(#9)	200
X	(#24)	100.000
Y	(#25)	100.000
Z	(#26)	–10.000

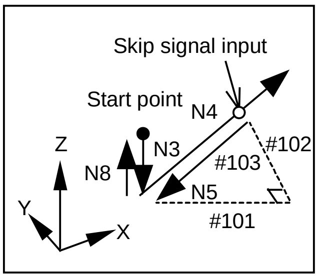
To subprogram

O9031

```
N1   #180=#4003
N2   #30=#5001#31=#5002
N3   G91G01Z#26F#9
N4   G31X#24Y#25F#9
N5   G90G00X#30Y#31
N6   #101=#30–#5061#102=#31–#5062
N7   #103=SQR[#101?#101+#102?#102]
N8   G91G01Z–#26
N9   IF[#180EQ91]GOTO11
N10  G90
N11  M99
```

(Common variable)

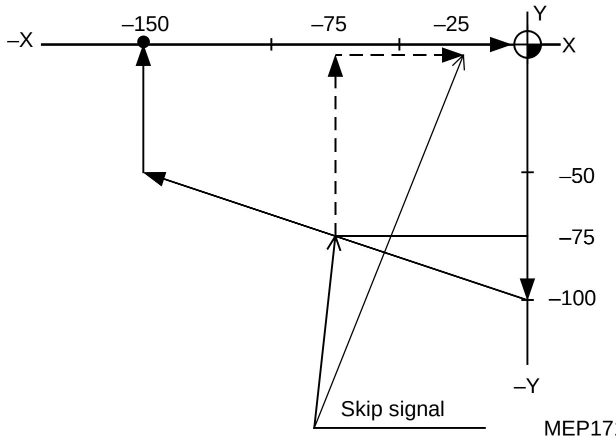
#101	87.245
#102	87.245
#103	123.383



#101	X-axis measuring amount	N1	Modal data storage of G90/G91
#102	Y-axis measuring amount	N2	X, Y starting point data storage
#103	Measuring line linear amount	N3	Z-axis entry
#5001	X-axis measuring start point	N4	X, Y measuring (Stop at skip input)
#5002	Y-axis measuring start point	N5	Return to X, Y starting point
#5061	X-axis skip input point	N6	X, Y measuring incremental data calculation
#5062	Y-axis skip input point	N7	Measuring line linear amount calculation
		N8	Z-axis escape
		N9, N10	Modal return of G90/G91
		N11	Return from subprogram

Example 2: Skip input coordinates reading:

```
N1   G91G28X0Y0
N2   G90G00X0Y0
N3   X0Y-100.
N4   G31X-150.Y-50.F80
N5   #111=#5061 #112=#5062
N6   G00Y0
N7   G31X0
N8   #121=#5061 #122=#5062
N9   M02
```



MEP171

#111 = –75. + ε   #112 = –75. + ε
#121 = –25. + ε   #122 = –75. + ε

where ε denotes an error due to response delay. (See Chapter 16 on skip functions for further details.) Variable #122 denotes the skip signal input coordinate of N4 since N7 does not have a Y-command code.

16. Tool No. (#51999) and data line No. (#3020) of the spindle tool

Using variables numbers 51999 and 3020, you can check the tool number and TOOL DATA line number of the tool mounted in the spindle.

System variable	Description
#51999	Tool number of the spindle tool
#3020	TOOL DATA line number of the spindle tool

Note: These system variables are read-only variables.

17. MAZATROL tool data

MAZATROL tool data can be checked (or assigned) using the following variables numbers:

Tool quantity (n): 960 (maximum)

$1 \le n \le 960$ (n = Sequence number of the tool data line)
(The maximum applicable tool quantity depends on the machine specifications.)

Usable variables numbers	MAZATROL tool data
#60001 to #60000 + n	Tool length (milling)/Length A (turning)
#61001 to #61000 + n	Tool diameter (milling)/Length B (turning)
#62001 to #62000 + n	Tool life flag
#63001 to #63000 + n	Tool damage flag
#64001 to #64000 + n	Wear compensation X
#65001 to #65000 + n	Wear compensation Y
#66001 to #66000 + n	Wear compensation Z
#67001 to #67000 + n	Group number

Note 1: During tool path check, tool data can be checked but cannot be assigned.

Note 2: Tool life flags (variables numbers of the order of #62000) and tool damage flags (likewise, the order of #63000) can take either 1 or 0 as their logical states (1 for ON, 0 for OFF).

18. EIA/ISO tool data

Using variables numbers tabulated below, EIA/ISO tool data (tool life management data) can be read or updated, as required.

Tool quantity (n): 960 (maximum)
1 ≤ n ≤ 960 (n = Sequence number of the tool data line)

System variables	Corresponding data
#40001 to #40000 + n	Tool length offset numbers or tool length offset amounts
#41001 to #41000 + n	Tool diameter offset numbers or tool diameter offset amounts
#42001 to #42000 + n	Tool life flags
#43001 to #43000 + n	Tool damage flags
#44001 to #44000 + n	Tool data flags
#45001 to #45000 + n	Tool operation time (sec)
#46001 to #46000 + n	Tool life time (sec)

Note 1: During tool path check, tool data can be checked but cannot be assigned.

Note 2: Tool life flags (variables numbers of the order of #42000) and tool damage flags (likewise, the order of #43000) can take either 1 or 0 as their logical states (1 for ON, 0 for OFF).

Note 3: The identification between number and amount of tool length or diameter offset is made by referring to the tool data flag.

Tool data flag	bit 0	bit 1	bit 2	bit 3
Length offset No.	0	0	–	–
Length offset amount	0	1	–	–
Diam. offset No.	–	–	0	0
Diam. offset amount	–	–	0	1

19. Date and time (Year-month-day and hour-minute-second)

Variables numbered 3011 and 3012 can be used to read date and time data.

Variable Nos.	Description
#3011	Date (Year-month-day)
#3012	Time (Hour-minute-second)

Example: If the date is December 15, 1995 and the time is 16:45:10, data is set as follows in the corresponding system variables:

#3011 = 951215
#3012 = 164510.

20. Total number of machined parts and the number of parts required

Variables numbered 3901 and 3902 can be used to read or assign the total number of machined parts and the number of parts required.

Variable Nos.	Description
#3901	Total number of machined parts
#3902	Number of parts required

Note 1: These variables must be integers from 0 to 9999.

Note 2: Data reading and writing by these variables is surely suppressed during tool path checking.

21. Setting and using variables names

Any variables name can be assigned to each of common variables #500 through #519. The variables name, however, must be of seven alphanumerics or less that begin with a letter of the alphabet.

Format:

SETVNn [*NAME1*, *NAME2*, ...]

- n: Starting number of the variable to be named
- *NAME1*: Name of #n (Variables name)
- *NAME2*: Name of #n + 1 (Varaibles name)

Each variables name must be separated using the comma (,).

Detailed description

- Once a variables name has been set, it remains valid even after power-off.
- Variables in a program can be called using the variables names. The variable to be called must, however, be enclosed in brackets ([ ]).

  Example:
  ```
  G01X[#POINT1]
  [#TIMES]=25
  ```

- Variables names can be checked on the **USER PARAMETER No. 1** display. The names assigned to variables #500 to #519 are displayed at F47 to F66.

  Example: Program `SETVN500[ABC,EFG]`

  On the display

F46	0	
F47	ABC	← Variables name assigned to #500
F48	EFG	← Variables name assigned to #501
F49		← Variables name assigned to #502
F50		

22. Tool data line number (#3022 and #3023)

Variables numbered 3022 and 3023 can be used to read the tool data line number of a particular tool.

Variable No.	Description
#3022	Designation of the required tool (for designation only). As is the case with a T-code, use the integral and decimal parts respectively for specifying the required tool with its number and suffix. #3022=○○○. △△ ○○○: Tool number △△: Suffix
#3023	Data line number of the specified tool (for reading only). Use this variable to read out the data line number of the tool specified by the variable #3022. The reading in #3023 is zero (0) if there is no corresponding tool registered in the memory.

Example:

TNo.		#3022 setting	Reading in #3023
1	A	1.01	1
1	B	1.02	2
1	C	1.03	3
2	A	2.61	4
2	B	2.62	5
2	C	2.63	6
3	H	3.08	7
3	V	3.22	8
3	Z	3.26	9
: :	: :	: :	: :
Failure		–	0

23. Positional information for the powered tailstock

Variables numbered 56154 and 56156 can be used to assign the particular positions as required for moving a powered tailstock.

Variable Nos.	Description
#56154	Tailstock position 1
#56156	Tailstock position 2

Note 1: The setting range is from –9999.999 to 0 for metric data input, or from –999.9999 to 0 for inch data input.

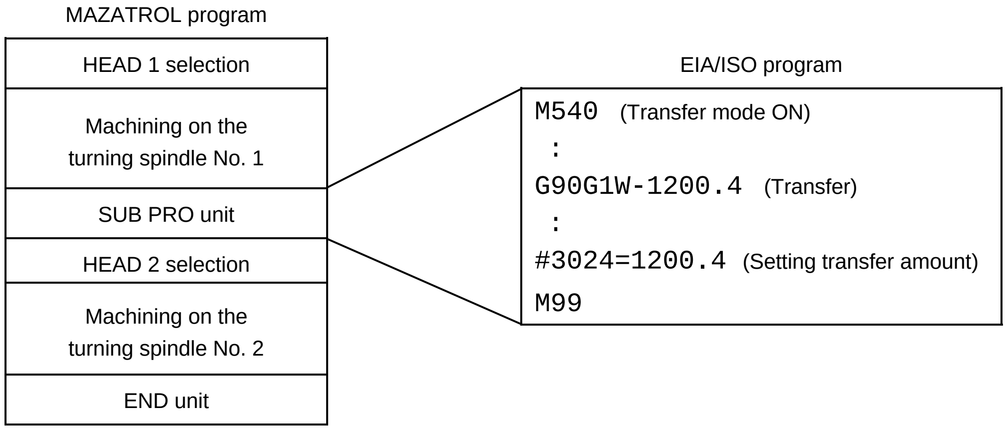
Note 2: “Position 1” and “Position 2” are the respective positions, to which the tailstock can be moved by the miscellaneous functions M841 and M842.

Return to Library

24. Amount of workpiece transfer (#3024)

Variable numbered 3024 can be used to set the amount of a workpiece transfer operation which is performed within an EIA/ISO program. The block of setting the variable #3024 does not cause any axis motion on the machine, but informs the NC unit of the workpiece being transferred so that a tool path avoiding collision with the shifted workpiece can be drawn timely for the succeeding process by a MAZATROL program on the side of the turning spindle No. 2.

<Example of programming>



Note: For a restart operation, based on a MAZATROL program of the above structure, from a block of machining on the side of the turning spindle No. 2, enter a block of “#3024 = 1200.4” in the MDI mode before starting the operation.

14-14-5 Arithmetic operation commands

Various operations can be carried out between variables using the following format.

#i = <expression>

where <expression> must consist of a constant(s), a variable(s), a function(s), or an operator(s). In the table given below, constants can be used instead of #j and/or #k.

[1] Definition/replacement of variables	`#i=#j`	Definition/replacement
[2] Additional-type operations	`#i=#j+#k`	Addition
	`#i=#j-#k`	Subtraction
	`#i=#jOR#k`	Logical additon (For each of 32 bits)
	`#i=#jXOR#k`	Exclusive OR (For each of 32 bits)
[3] Multiplicative-type operations	`#i=#j*#k`	Multiplication
	`#i=#j/#k`	Division
	`#i=#jMOD#k`	Surplus
	`#i=#jAND#k`	Logical product (For each of 32 bits)
[4] Functions	`#i=SIN[#k]`	Sine
	`#i=COS[#k]`	Cosine
	`#i=TAN[#k]`	Tangent (tanq is used as sinq/cosq.)
	`#i=ATAN[#j]`	Arc-tangent (Either ATAN or ATN can be used.)
	`#i=ACOS[#j]`	Arc-cosine
	`#i=SQRT[#k]`	Square root (Either SQRT or SQR is available.)
	`#i=ABS[#k]`	Absolute value
	`#i=BIN[#k]`	BINARY conversion from BCD
	`#i=BCD[#k]`	BCD conversion from BINARY
	`#i=ROUND[#k]`	Rounding to the nearest whole number (Either ROUND or RND is available.)
	`#i=FIX[#k]`	Cutting away any decimal digits
	`#i=FUP[#k]`	Counting any decimal digits as 1s
	`#i=LN[#k]`	Natural logarithm
	`#i=EXP[#k]`	Exponent with the base of e (= 2.718 ...)

Note 1: In principle, data without a decimal point is handled as data that has a decimal point. (Example: 1 = 1.000)

Note 2: Offsets from variable #10001, workpiece coordinate system offsets from variable #5201, and other data become data that has a decimal point. If data without a decimal point is defined using these variables numbers, therefore, a decimal point will also be assigned to the data.

Example:

```
#101=1000
#10001=#101
#102=#10001
```

Execution →

Common variable

```
#101   1000
#102  1.000
```

Note 3: The <expression> after a function must be enclosed in brackets ([ ]).

1. Operation priority

Higher priority is given to functions, multiplicative operations, and additive operations, in that order.

```
#101=#111+#112∗SIN[#113]
```

[1] Function
[2] Multiplicative
[3] Additional

2. Specifying an operational priority level

The part to which the first level of operation priority is to be given can be enclosed in brackets ([ ]). Up to five sets of brackets, including those of functions, can be used for one expression.

```
#101=SQRT[[[#111–#112]∗SIN[#113]+#114]∗#15]
```

One fold
Two fold
Three fold

3. Examples of operation instructions

[1] Main program and argument specification	G65 P100 A10 B20. #101=100.000 #102=200.000	#1 10.000 #2 20.000 #101 100.000 #102 200.000
[2] Definition, replacement =	#1=1000 #2=1000. #3=#101 #4=#102 #5=#5081	#1 1000.000 #2 1000.000 #3 100.000 } Data of common variables #4 200.000 } #5 –10.000 Offset amount
[3] Addition, subtraction + –	#11=#1+1000 #12=#2–50. #13=#101+#1 #14=#5081–3. #15=#5081+#102	#11 2000.000 #12 950.000 #13 1100.000 #14 –13.000 #15 190.000
[4] Logical addition OR	#3=100 #4=#3OR14	#3 = 01100100 14 = 00001110 #4 = 01101110 = 110
[5] Exclusive OR XOR	#3=100 #4=#3XOR14	#3 = 01100100 14 = 00001110 #4 = 01101010 = 106
[6] Multiplication, Division ∗ /	#21=100∗100 #22=100.∗100 #23=100∗100. #24=100.∗100. #25=100/100 #26=100./100 #27=100/100. #28=100./100. #29=#5081∗#101 #30=#5081/#102	#21 10000.000 #22 10000.000 #23 10000.000 #24 10000.000 #25 1.000 #26 1.000 #27 1.000 #28 1.000 #29 –1000.000 #30 –0.050
[7] Surplus MOD	#31=#19MOD#20	$\frac{\#19}{\#20} = \frac{48}{9}$ = 5 surplus 3
[8] Logical product AND	#9=100 #10=#9AND15	# 9 = 01100100 15 = 00001111 #10 = 00000100 = 4

[9] Sine SIN	#501=SIN[60] #502=SIN[60.] #503=1000*SIN[60] #504=1000*SIN[60.] #505=1000.*SIN[60] #506=1000.*SIN[60.] **Note:** SIN[60] is equal to SIN[60.].	#501 0.866 #502 0.866 #503 866.025 #504 866.025 #505 866.025 #506 866.025
[10] Cosine COS	#541=COS[45] #542=COS[45.] #543=1000*COS[45] #544=1000*COS[45.] #545=1000.*COS[45] #546=1000.*COS[45.] **Note:** COS[45] is equal to COS[45.].	#541 0.707 #542 0.707 #543 707.107 #544 707.107 #545 707.107 #546 707.107
[11] Tangent TAN	#551=TAN[60] #552=TAN[60.] #553=1000*TAN[60] #554=1000*TAN[60.] #555=1000.*TAN[60] #556=1000.*TAN[60.] **Note:** TAN[60] is equal to TAN[60.].	#551 1.732 #552 1.732 #553 1732.051 #554 1732.051 #555 1732.051 #556 1732.051
[12] Arc-tangent ATAN	#561=ATAN[173205/1000000] #562=ATAN[173.205/100.] #563=ATAN[1.732]	#561 60.000 #562 60.000 #563 59.999
[13] Arc-cosine ACOS	#521=ACOS[100000/141421] #522=ACOS[100./141.421] #523=ACOS[1000/1414.213] #524=ACOS[10./14.142] #525=ACOS[0.707]	#521 45.000 #522 45.000 #523 45.000 #524 44.999 #525 45.009
[14] Square root SQRT	#571=SQRT[1000] #572=SQRT[1000.] #573=SQRT[10.*10.+20.*20.] #574=SQRT[#14*#14+#15*#15] **Note:** For enhanced accuracy, perform operations within [ ] as far as possible.	#571 31.623 #572 31.623 #573 22.361 #574 190.444
[15] Absolute value ABS	#576=-1000 #577=ABS[#576] #3=70. #4=-50. #580= ABS[#4-#3]	#576 –1000.000 #577 1000.000 #580 120.000
[16] BIN, BCD	#1=100 #11=BIN[#1] #12=BCD[#1]	 #11 64 #12 256
[17] Rounding into the nearest whole number ROUND	#21=ROUND[14/3] #22=ROUND[14./3] #23=ROUND[14/3.] #24=ROUND[14./3.] #25=ROUND[-14/3] #26=ROUND[-14./3] #27=ROUND[-14/3.] #28=ROUND[-14./3.]	#21 5 #22 5 #23 5 #24 5 #25 –5 #26 –5 #27 –5 #28 –5
[18] Cutting away any decimal digits FIX	#21=FIX[14/3] #22=FIX[14./3] #23=FIX[14/3.] #24=FIX[14./3.] #25=FIX[-14/3] #26=FIX[-14./3] #27=FIX[-14/3.] #28=FIX[-14./3.]	#21 4.000 #22 4.000 #23 4.000 #24 4.000 #25 –4.000 #26 –4.000 #27 –4.000 #28 –4.000

[19] Counting any decimal digits as 1s FUP	#21=FUP[14/3] #22=FUP[14./3] #23=FUP[14/3.] #24=FUP[14./3.] #25=FUP[–14/3] #26=FUP[–14./3] #27=FUP[–14/3.] #28=FUP[–14./3.]	#21 5.000 #22 5.000 #23 5.000 #24 5.000 #25 –5.000 #26 –5.000 #27 –5.000 #28 –5.000
[20] Natural logarithm LN	#101=LN[5] #102=LN[0.5] #103=LN[–5]	#101 1.609 #102 –0.693 Alarm 860 CALCULATION IMPOSSIBLE
[21] Exponent EXP	#104=EXP[2] #105=EXP[1] #106=EXP[–2]	#104 7.389 #105 2.718 #106 0.135

4. Operation accuracy

The errors listed in the table below are generated by one arithmetic operation, and the error rate increases each time an operation is performed.

Operation format	Mean error	Max. error	Kind of error
a = b + c a = b – c	2.33×10^{-10}	5.32×10^{-10}	Min. $\left\|\frac{\varepsilon}{c}\right\|, \left\|\frac{\varepsilon}{b}\right\|$
a = b • c	1.55×10^{-10}	4.66×10^{-10}	Relative error $\left\|\frac{\varepsilon}{a}\right\|$
a = b/c	4.66×10^{-10}	1.86×10^{-9}	
$a = \sqrt{b}$	1.24×10^{-9}	3.73×10^{-9}	
a = sin b a = cos b	5.0×10^{-9}	1.0×10^{-8}	Absolute error $\|\varepsilon\|$ degree
$a = \tan^{-1} b/c$	1.8×10^{-6}	3.6×10^{-6}	

Note: The function TAN (Tangent) is calculated as SIN/COS (Sine/Cosine).

5. Notes on deterioration of accuracy

A. Addition/subtraction

As for additional-type operations, if an absolute value is subtracted from the other, the relative error cannot be reduced below 10^{-8}.

For example, when the true values (such values, by the way, cannot be substitued directly) of #10 and #20 are as follows:

#10 = 2345678988888.888
#20 = 2345678901234.567

then #10 – #20 = 87654.321 would not result from calculation of #10 – #20. This is because, since the effective number of digits of the variable is eight (decimal), the approximate values of #10 and #20 are:

#10 = 2345679000000.000
#20 = 2345678900000.000

More strictly, internal binary values slightly differ from these values. Actually therefore, a significant error results as follows:

#10 – #20 = 100000.000.

B. Logical relationship

As for EQ, NE, GT, LT, GE and LE, the processing is executed in a similar manner to addition and substraction, so be careful to errors. For example, to judge whether #10 is equal to #20 of the above example, the conditional expression

IF [#10EQ#20]

is not appropriate due to the errors. In such a case, therefore, give a macro-instruction as shown below to allow for an acceptable tolerance in the judgement on the equality of two values.

IF [ABS[#10 – #20] LT200000]

C. Trigonometric functions

For trigonometric functions, although the absolute error is guaranteed, the relative error is not below 10^{-8}. Be careful, therefore, when carrying out multiplication, or division after trigonometric function operations.

14-14-6 Control commands

The flow of a program can be controlled using IF ~ GOTO ~ and WHILE DO ~ commands.

1. Branching

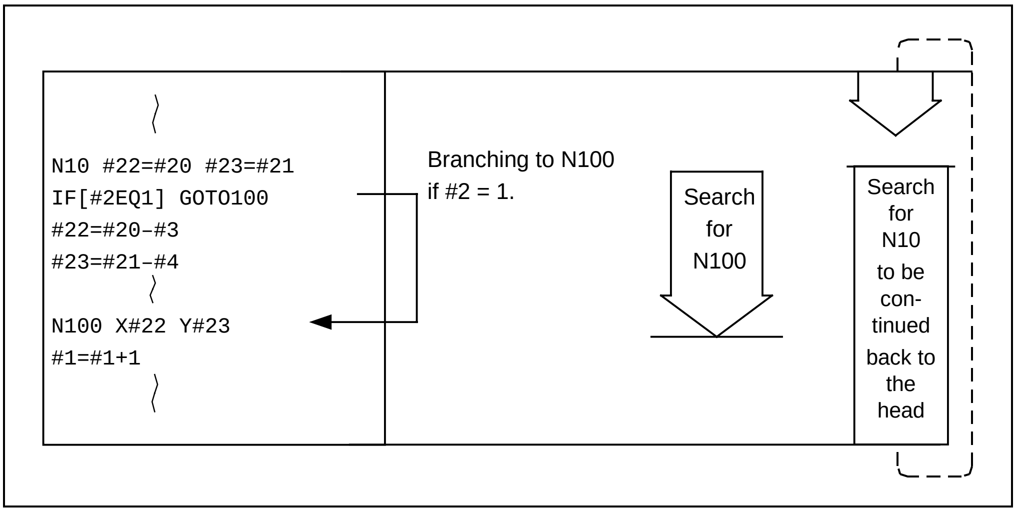
Format: IF [conditional expression] GOTO n

where n is a sequence number in the same program.
The branching will occur to the block headed by sequence number 'n' if the condition holds, or if the condition does not hold, the next block will be executed.
An independent setting of GOTO statement without IF [conditional expression] will perform unconditional branching to the specified block.

The [conditional expression] consists of the following six types:

#i EQ #j	=	(#i is equal to #j.)
#I NE #j	≠	(#i is not equal to #j.)
#i GT #j	>	(#i is larger than #j.)
#I LT #j	<	(#i is smaller than #j.)
#I GE #j	≥	(#i is equal to #j, or larger than #j.)
#I LE #j	≤	(#i is equal to #j, or smaller than #j.)

For GOTO n, n must be a sequence number within the same program. If the sequence number does not exist in that program, an alarm **843 DESIGNATED SNo. NOT FOUND** will occur. An expression or a variable can be used instead of #i, #j, or n.
Sequence number designation Nn must be set at the beginning of the destination block. Otherwise, an alarm **843 DESIGNATED SNo. NOT FOUND** will result. If, however, the block begins with "/" and Nn follows, the program can be branched into that sequence number.



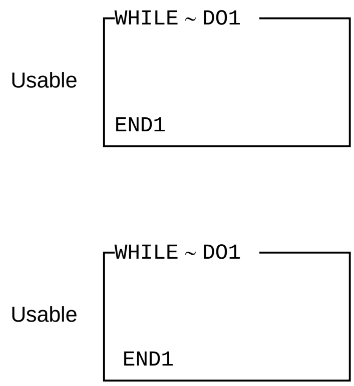
Note: During search for the branching destination sequence number, if the area from the block after "IF ..." to the program end (% code) is searched (forward search) in vain, then the area from the head down to the block before "IF ..." will be searched next (backward search). It will therefore take more time to execute backward search (searching in the opposite direction to the flow of the program) than to execute forward search.

2. Looping

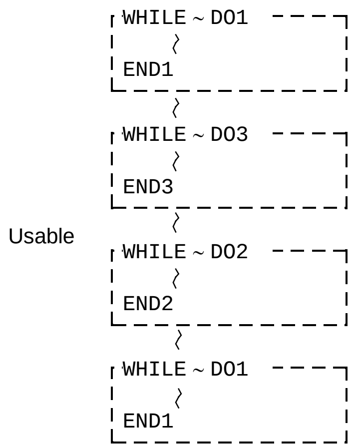
```
Format:
    WHILE [Condition expression] DOm   (m = 1, 2, 3 ··· 127)
      ⟩
      ⟨
    ENDm
```

The area from the next block to the ENDm block loops while the conditional expression holds. If the conditional expression does not hold, control will be tansferred to the block after ENDm. In the format shown above, DOm can precede WHILE.
You must always use WHILE [conditional expression] DOm and ENDm in pairs. If you omit WHILE [conditional expression], the area from DOm to ENDm will endlessly loop. In DOm, m (1 to 127) identifies the number of looping. (DO1, DO2, DO3, and so on up to DO127)
The maximum available number of degrees of multiplicity is 27.

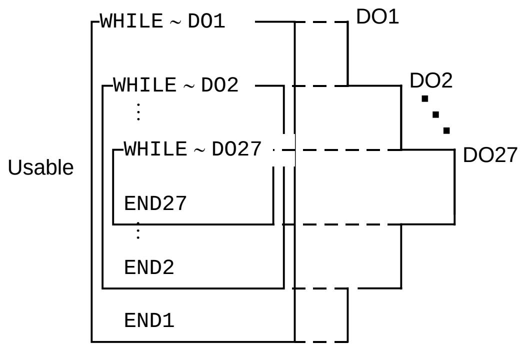
[1] Same identifying No. can be used repeatedly.



[2] The identifying No. of WHILE ~ DOm is arbitrary.

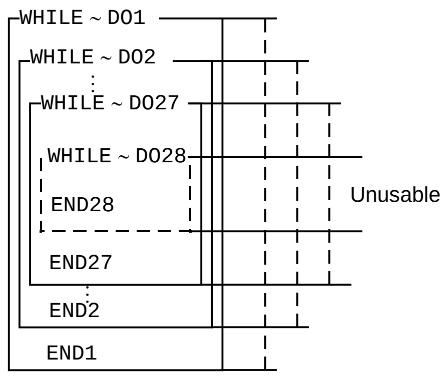


[3] Up to 27 levels of WHILE ~ DOm can be used.
m can be 1 to 127, independent of the depth of nesting.

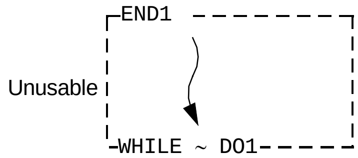


Note: For nesting, m once used cannot be used again.

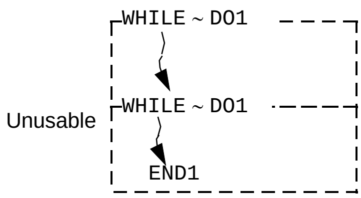
[4] The total number of levels of WHILE ~ DOm must not exceed 27.



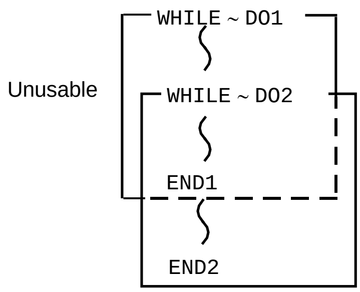
[5] WHILE ~ DOm must precede ENDm.



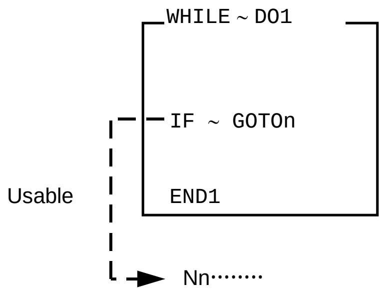
[6] WHILE ~ DOm must correspond to ENDm one-to-one in the same program.



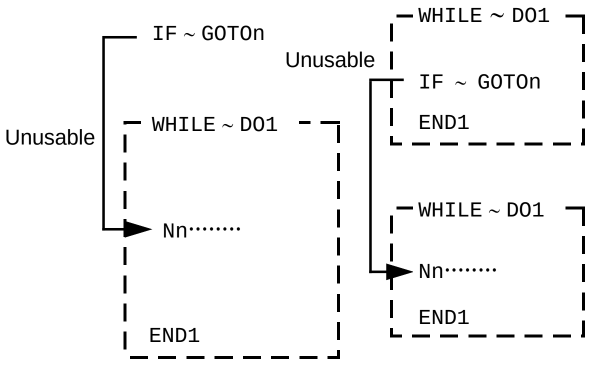
[7] WHILE ~ DOm must not overlap.



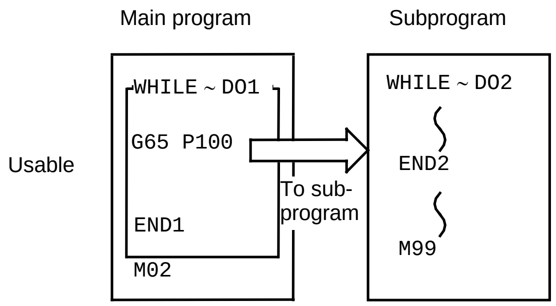
[8] Outward branching from the range of WHILE ~ DOm is possible.



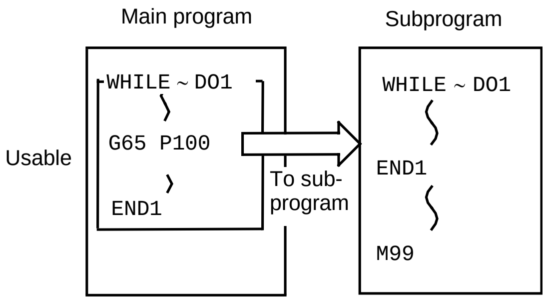
[9] Branching into WHILE ~ DOm is not allowed.



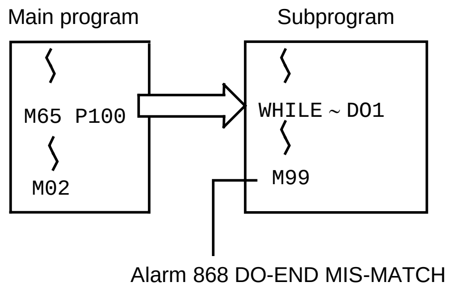
[10] Subprogram can be called using M98, G65, G66, etc. from the midway of WHILE ~ DOm.



[11] The looping can be independently programmed in a subprogram which is called using G65/G66 from the midway of WHILE ~ DOm. Up to 27 levels of nesting for both programs can be done.



[12] If WHILE and END are not included in pairs in subprogram (including macro subprogram), a program error will result at M99.



14-14-7 External output commands (Output via RS-232C)

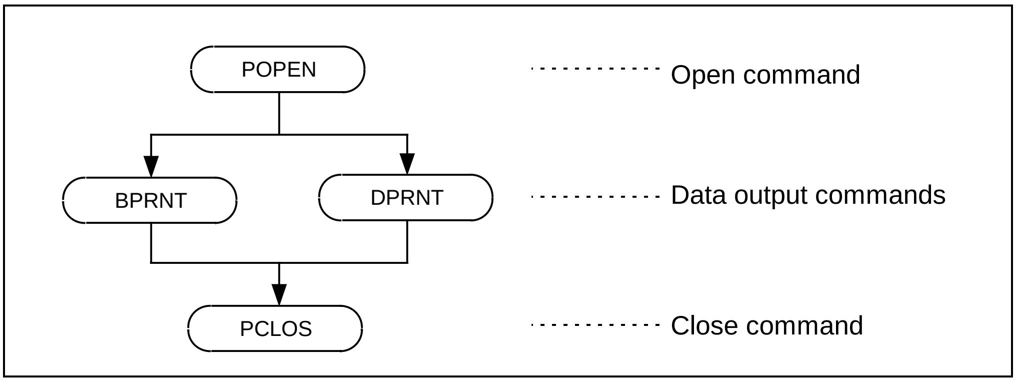
1. Overview

In addition to standard user macros, the types of macros listed below are provided as external output commands. These external output macros can be used to output character data or the numerical data in variables to an external unit via an RS-232C interface. The data are outputted in a data length of 7 bits with an even-parity bit added.

A. Types and functions of external output macros

POPEN	Setup processing for data output
PCLOS	Termination processing of data output
BPRNT	Printout of character data or binary printout of variable data
DPRNT	Printout of character data or numerical printout of variable data on a digit-by-digit basis

B. Programming order



2. Open command POPEN

Programming format: POPEN

<u>Detailed description</u>

- The command code POPEN must be included before a series of data output command codes.
- The control code for DC2 and the percentage code % are output from the NC unit to an external output unit.
- Once POPEN has been set, it will remain valid until PCLOS is set.

3. Close command PCLOS

Programming format: PCLOS

<u>Detailed description</u>

- The command code PCLOS must be included after all data output command codes.
- The control code for DC4 and the percentage code % are outputted from the NC unit to an external output unit.
- This command must be used together with POPEN. This command code must be included only after POPEN.
- This command must be set at the end of the program even after data output has been aborted using, for example, the NC reset switch.

4. Data output command BPRNT

Programming format:

BPRNT[ℓ1#v1[c1]ℓ2#v2[c2]·······]

- c1: Effective digits after decimal point
- v1: Variable number
- ℓ1: Character string

(Effective digits after decimal point and Variable number: Variable value × 10^c)

Detailed description

- The command BPRNT can be used to output characters or to output variable data in binary form.
- The designated character string is outputted directly in the ISO coded format. Alphanumerics (A to Z, and 0 to 9) and/or special characters (+, –, ∗, /) can be used. Of these characters, only the asterisk (∗) is outputted as a space code.
- Since all variables are saved as those having a decimal point, the necessary number of decimal digits must be enclosed in brackets ([ ]).
  All variables are handled as data of four bytes (32 bits), and each byte is outputted as binary data in the order of the most significant byte first. Minus data is processed as the complement for that data.

  Example 1: If three digits are specified for 12.3456, then
  [12.346 × 10^3] = 12346 (0000303A)
  will be outputted as binary data.

  Example 2: If no digits are specified for –100.0, then
  –100 (FFFFFF9C)
  will be outputted as binary data.

- After the specified data has been outputted, the EOB (End Of Block) code is outputted in the format of the appropriate ISO code.
- Variables containing <empty> are interpreted as 0s.

5. Data output command DPRNT

Programming format:

DPRNT[ℓ1#v1[d1 c1]ℓ2#v2[c2]······]

- c1: Effective digits below decimal point
- d1: Effective digits above decimal point
- v1: Variable number
- ℓ1: Character string

(Effective digits below and above decimal point: c + d ≤ 8)

Detailed description

- Output of character data or decimal output of variable data is performed in the format of ISO codes.
- The designated character string is outputted directly in the ISO coded format.
  Alphanumerics (A to Z, and 0 to 9) and/or special characters (+, –, ∗, /) can be used. Of these characters, only the asterisk (∗) is outputted as a space code.
- Of the data contained in a variable, the necessary number of digits above the decimal point and that of digits below the decimal point must each be enclosed in brackets ([ ]). The variable data will then have its total specified number of digits, including the decimal point, outputted in the ISO coded format in the order of the most significant digit first. No trailing zeros will be left out in that case.

14-14-8 External output command (Output onto the hard disk)

1. Overview

External output macros can also be used to output data in text file format into the predetermined directory on the hard disk.

2. Related parameters

- **DPR14**: Selection of an output destination port
  Set **DPR14** to “4” (Output onto the hard disk) under OTHER on the **DATA I/O PARAMETER** display.
- **DPR15**: Number of lines in feed section
  Set the required number of lines to be fed.
- **DPR8**: Output file size
  Use this parameter to specify the maximum permissible output file size.

  Maximum permissible file size: **DPR8** × 100 (KB)

  A command for outputting a greater file will cause a corresponding alarm.

  However, the file size limit is 100 KB if the value in **DPR8** is 0.

Note: Output of a file of smaller size than the limit, however, may not be possible due to a shortage of available area on the hard disk.

The **DATA I/O PARAMETER** display can be selected by pressing the **[DATA I/O PARAM.]** menu key on the **DATA I/O** display.

See the Parameter List for details of the parameters.

3. Output file

The text file will be automatically outputted with a particular file name into the predetermined directory.

Output directory: c:\MC_sdg\print\

Output file name: print.txt
(A file of this name will be automatically created, if required, or the text data will be added to the current contents of the file.)

File contents:

Given below on the right is an example of text file contents created by the execution of the program shown on the left under the particular parameter settings.

[Program]	[Output example]
``G28XYZ``` POPEN DPRNT[OOOOOOOOOOOO] DPRNT[XXXXXXXXXXXX] DPRNT[IIIIIIIIIIII] PCLOS G0X100.Y100.Z100. M30	print.txt % OOOOOOOOOOOO XXXXXXXXXXXX IIIIIIIIIIII %

[Parameter]
 DPR14: 4
 DPR15: *No setting*

4. Related alarms

The alarm given for text file output is described below.

No.	Message	Argument 1		Argument 2	Argument 3
887	TAPE I/O ERROR	–100	File open error	0	0
		–111	File write error	0	0
		–112	File size too great	0	0

14-14-9 Precautions

Use of user macro commands allows a machining program to be created by combining arithmetic operation, judgment, branching, or other macro commands with conventional NC commands such as move commands, M-, S-, T-commands, etc. The statement defined by these macro commands and that of conventional NC commands are taken as a macro statement and an NC execute statement, respectively. The treatment of a macro statement has no direct rerations with machine control. Its treatment as short as possible is effective for shortening machining time.
Parallel processing of the NC execute statement and the macro statement becomes possible according to the setting of bit 6 of parameter **F93.**
(It becomes possible to process all macro statements in batch form by setting the parameter bit to OFF when machining the workpiece, or to execute the macro statements block-by-block by setting the parameter bit to ON when checking the program. Therefore, set the parameter bit according to your requirements.)

<u>Sample program</u>

```
N1 G91G28X0Y0Z0
N2 G92X0Y0Z0
N3 G00X–100.Y–100.
N4 #101=100.*COS[210.]    <-- Macro statements
N5 #102=100.*SIN[210.]
N6 G01X#101Y#102F800
```

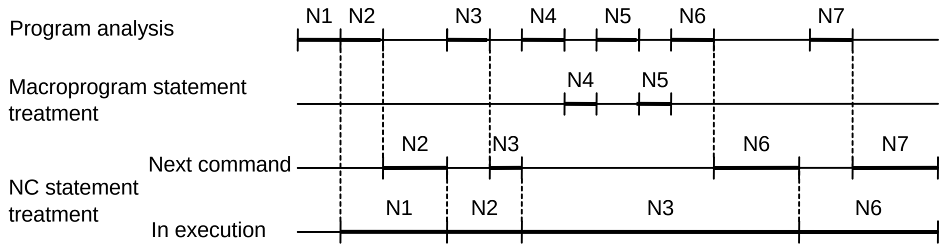
A macro statement refers to a statement that consists of the following blocks:

- Arithmetic operation command block (compassing the equal sign =)
- Control command block (compassing GOTO, DO ~ END, etc.)
- Macro call command block (compassing macro call or cancellation G-code commands G65, G66, G66.1, or G67)

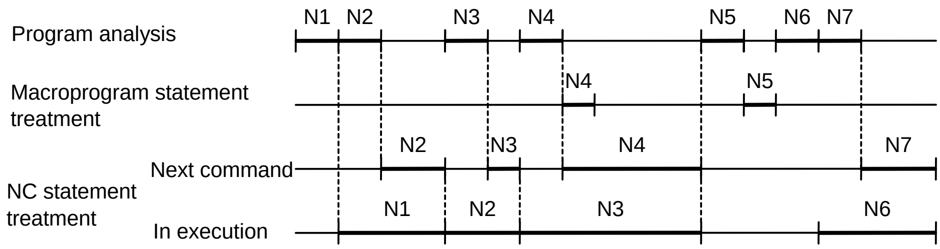
An NC execute statement refers to a non-macro statement.

The flow of processing of these two types of statements is shown below.

Parameter OFF



Parameter ON



Machining program data is displayed as follows:

Parameter OFF	(In execution) `N3 G00X–100.Y–100.` (Next command) `N6 G01X#101Y#102F800`	N4, N5 and N6 are treated in parallel with NC execution sentence of N3, and N6 is displayed as next command because it is NC execution sentence. When N4, N5, and N6 are analized during NC execution sentence of N3, machine control continues.
Parameter ON	(In execution) `N3 G00X–100.Y–100.` (Next command) `N4 #101=100.*COS[210.]`	N4 is treated in parallel with the control of NC execution sentence of N3, and is displayed as next command. After N3 is completed, N5 and N6 are analized so the machine control is forced to wait by the analyzing time of N5 and N6 before N6 can be executed.

14-14-10 Specific examples of programming using user macros

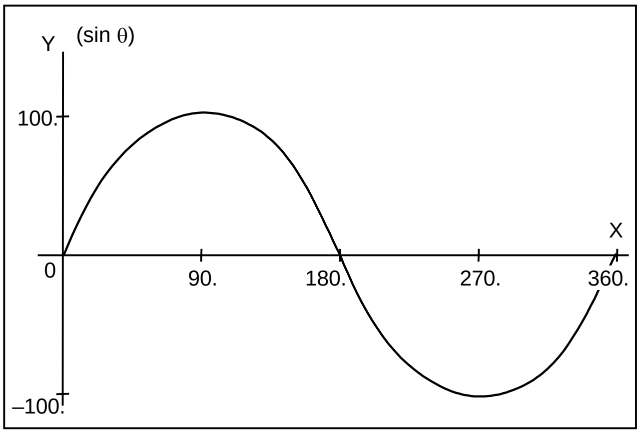
The following three examples of programming are shown here:

Example 1: SIN curve
Example 2: Bolt-hole circle
Example 3: Grid

Example 1: SIN curve

G65 Pp_1 Aa_1 Bb_1 Cc_1 Ff_1

A: Initial value 0°
B: Final value 360°
C: R of R∗SINθ
F: Feed rate



MEP172

Main program

```
      :
G65P9910A0B360.C100.F100
      :
```

To subprogram

O9910 (Subprogram)

```
WHILE[#1LE#2]DO1
#10=#3∗SIN[#1]
G90G1X#1Y#10F#9
#1=#1+10.
END1
M99
```

} Note

Local variable set in argument

#1 = 0
#2 = 360.000
#3 = 100.000
#9 = 100.000

Note: `G90G01X#1Y[#3∗SIN[#1]]F#9` makes one block command available.

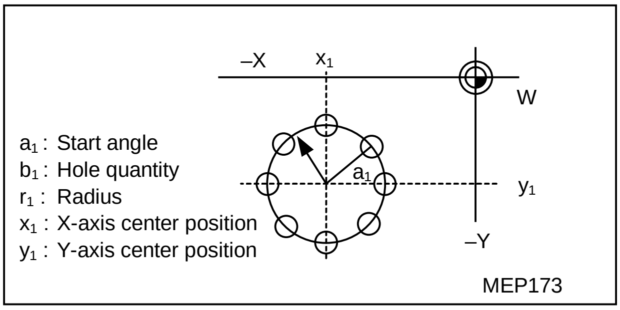
Example 2: Bolt-hole circle

After hole data has been defined using fixed-cycle machining commands G72 to G89, hole positions are to be designated using macro commands.

Main program

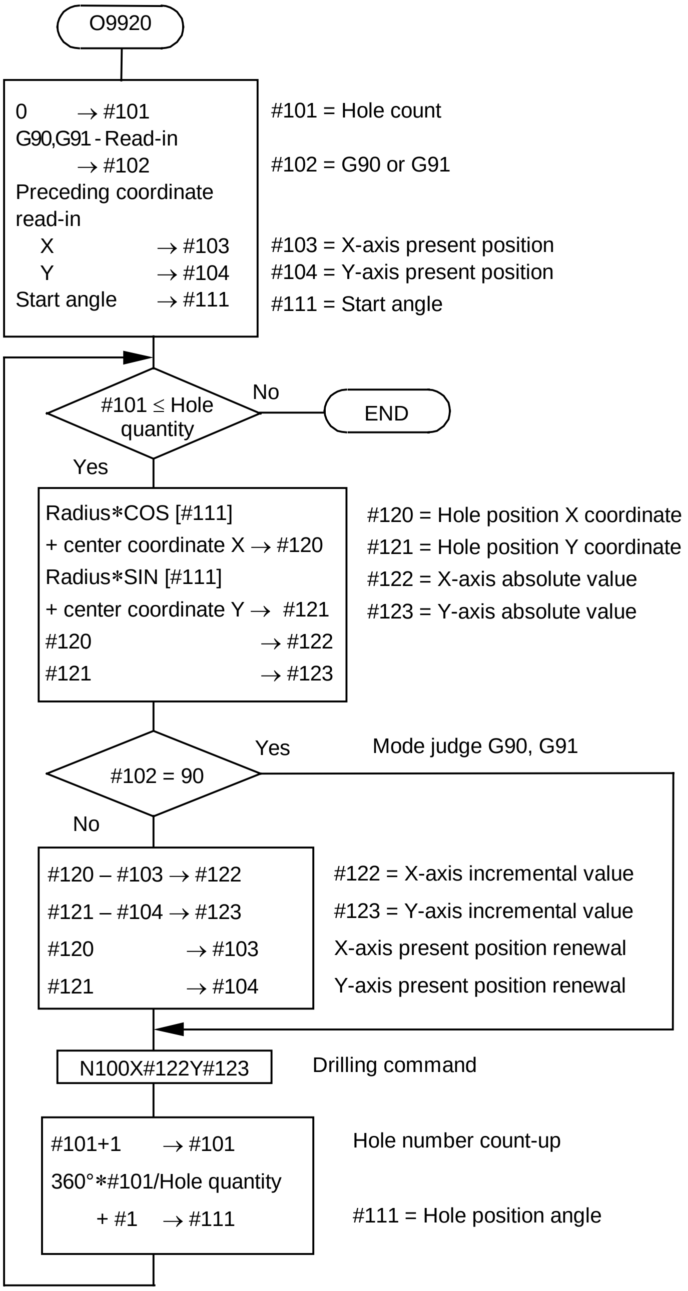
```
G81Z-100.R50.F300
G65P9920Aa1Bb1Rr1Xx1Yy1
```

To sub program



O9920 (Subprogram)

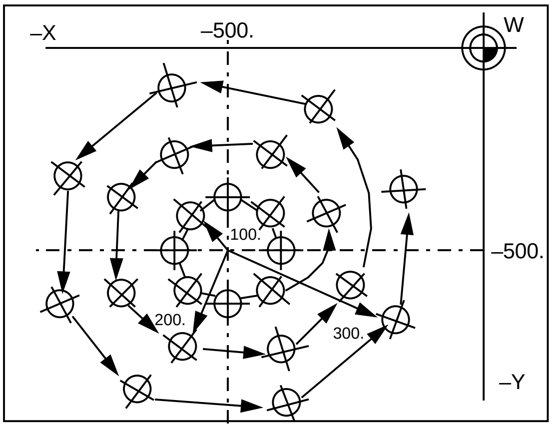
```
#101=0
#102=#4003
#103=#5001        Note
#104=#5002
#111=#1

WHILE[#101LT#2]DO1

#120=#24+#18*COS[#111]   Note
#121=#25+#18*SIN[#111]

#122=#120 #123=#121
IF[#102EQ90]GOTO100

#122=#120-#103    Note
#123=#121-#104

#103=#120
#104=#121

N100 X#122Y#123
#101=#101+1                Note
#111=#1+360.*#101/#2

END1
M99
```



Note:
The processing time can be reduced by one-block reduced programming.

```
G28X Y Z
T1 M06
G90 G43 Z100. H01
G54 G00 X0 Y0
G81 Z-100. R3. F100 M03
G65 P9920X-500. Y-500. A0  B8R100.
G65 P9920X-500. Y-500. A30. B8R200.
G65 P9920X-500. Y-500. A60. B8R300.
```

To subprogram



MEP174

Example 3: Grid

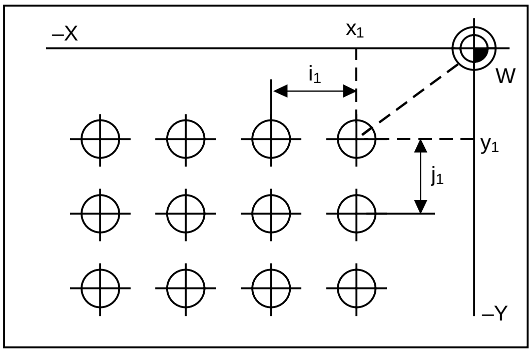
After hole data has been defined using fixed-cycle machining commands G72 to G89, hole positions are to be designated using macro call commands.

G81 Zz_1 Rr_1 Ff_1

G65Pp_1 Xx_1 Yy_1 Ii_1 Jj_1 Aa_1 Bb_1

- X : X-axis hole position
- Y : Y-axis hole position
- I : X-axis distance
- J : Y-axis distance
- A : X direction hole quantity
- B : Y direction hole quantity

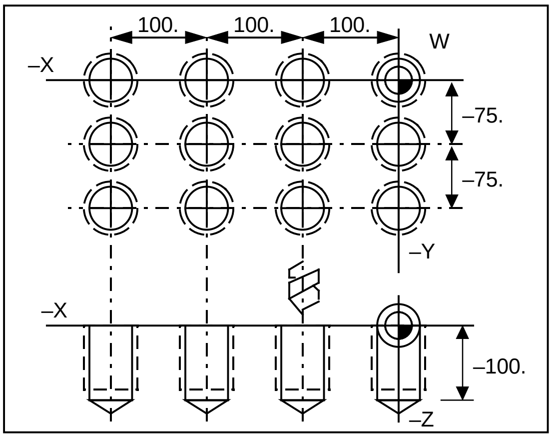
Subprogram is shown on the next page.



MEP175

```
G28X Y Z
T1  M06
G90 G43 Z100. H01
G54 G00 X0 Y0
G81 Z–100. R3. F100M03
G65P9930 X0 Y0 I100. J–75. A5B3


G84 Z–90. R3. F250M03
G65P9930 X0 Y0 I–100. J–75. A5B3
```

To subprogram

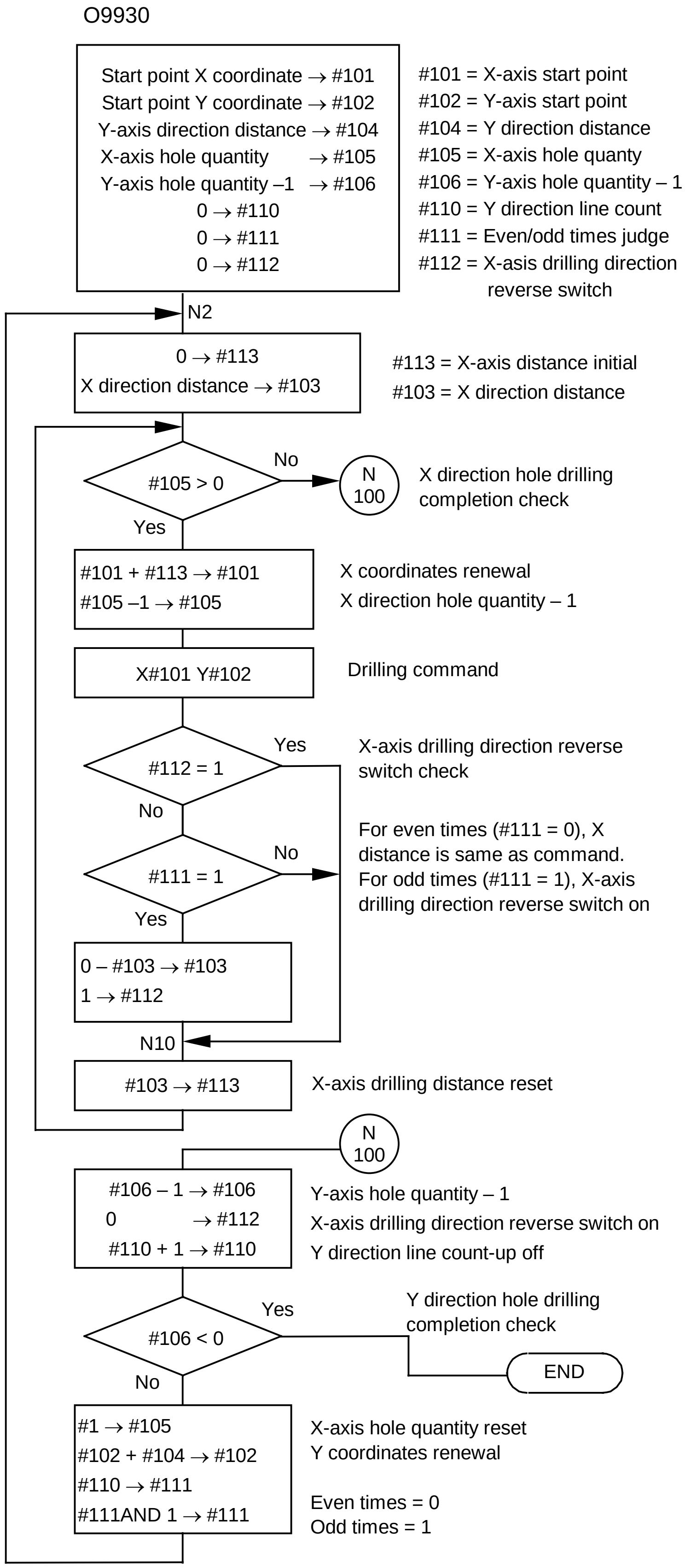


MEP176

O9930 (Subprogram)

```
#101=#24
#102=#25
#104=#10
#105=#1
#106=#2-1
#110=0
#111=0
#112=0                    } Note

N2 #113=0
#103=#9                   } Note

WHILE[#105GT0]DO1
#101=#101+#113
#105=#105-1
X#101Y#102                } Note

IF[#112EQ1]GOTO10
IF[#111NE1]GOTO10

#103=0-#103
#112=1                    } Note

N10 #113=#103
END1

N100 #106=#106-1
#112=0
#110=#110+1               } Note

IF[#106LT0]GOTO200

#105=#1
#102=#102+#104
#111=#110                 } Note

#111=#111AND1

GOTO2

N200 M99
```

Note:
The processing time can be reduced by one-block reduced programming.



Return to Library

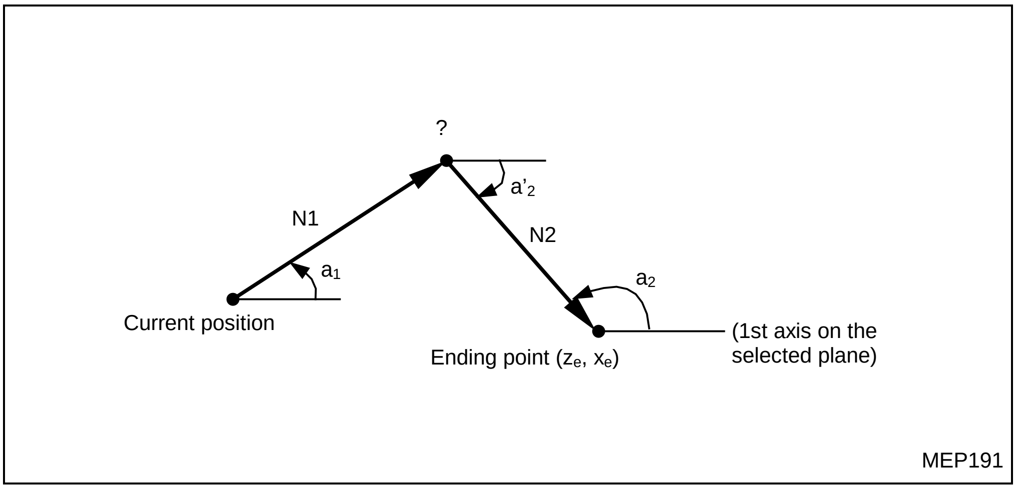
14-15 Geometric Commads (Option)

1. Function and purpose

Even if it is difficult to find the crossing point of two lines using linear interpolation commands, setting the slope of the first line and the absolute coordinates of the ending point of the second line and its slope will allow the NC unit to calculate the coordinates of the ending point of the first line and thus to control move commands.

2. Programming format

G18	Specify the intended plane using G17, G18, or G19.
N1 G01 Aa_1 Ff_1	Specify the angle and speed for the first block.
N2 Xx_e Zz_e Aa_2 (a'_2) Ff_2	Specify the absolute coordinates of the ending point of the next block, angles, and a speed.



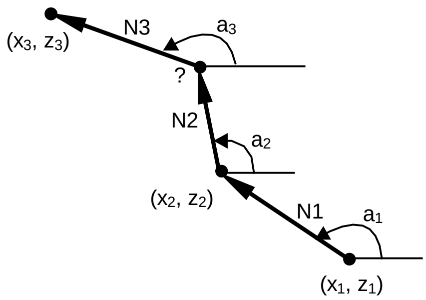
3. Detailed description

- The slope of a line denotes an angle relative to the plus (+) direction of the first axis (horizontal axis) on the selected plane. Assign the sign + for a counterclockwise direction (CCW), or the sign – for a clockwise direction (CW).
- The range of the slope a must be $-360.000° \le a \le +360.000°$.
- For the second block, the slope at either the starting point or the ending point can be set. The NC unit will identify whether the specified slope is for the starting point or for the ending point.
- The coordinates of the ending point of the second block must be specified using absolute data. Otherwise a program error will result.
- Any speed can be specified for each block.
- A program error will result if the angle of the crossing point of the two lines is 1 degree or less.
- A program error will result if the preselected plane for the first block is changed over at the second block.
- The geometric command function does not work if address A is to be used for an axis name or for the No. 2 auxiliary function.
- Single-block stop can be used at the ending point of the first block.
- A program error will result if the first block or the second block is not linear.

4. Correlationships to other functions

Geometric command can be set following a linear angle command.

N1Xx_2Aa_1
N2Aa_2
N3Xx_3Zz_3Aa_3



MEP192

15 COORDINATE SYSTEM SETTING FUNCTIONS

15-1 Coordinate System Setting Function: G50 [Series M: G92]

1. Function and purpose

A coordinate system can be set by commanding G50 wherever a tool is positioned. This coordinate system can be placed anywhere, but normally its X-, Y-axes zero points are on the workpiece center, and the Z-axis zero point on the workpiece end face.

2. Programming format

G50 Xx Zz αα; (α is additional axis.)

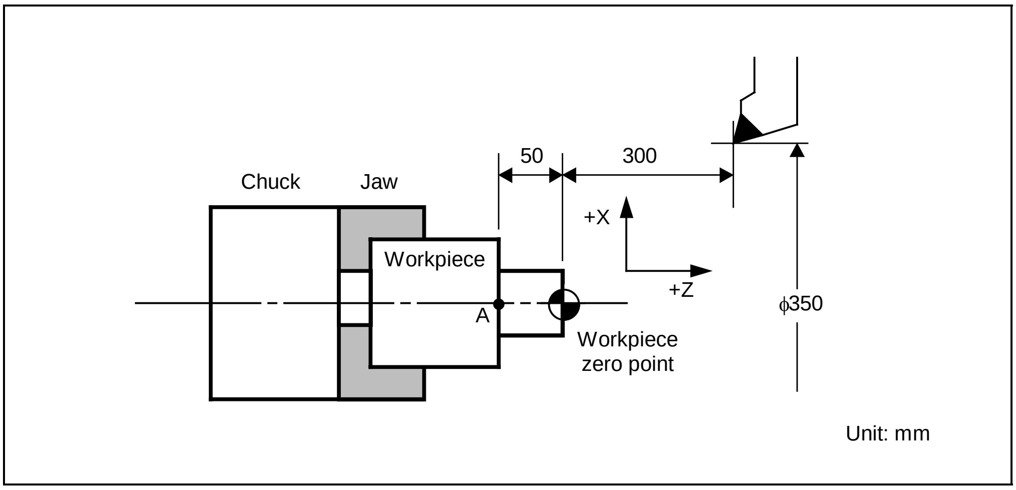
3. Detailed description

For moving the tool by absolute command, the coordinate system needs to be determined in advance. The coordinate system can be set by a command as below.

G50 X_ Z_ C_;

This command allows to set a coordinate system where a point on a tool, for example the tool tip position, can be represented with coordinates (X, Z). This coordinate system is called the workpiece coordinate system.
Once a coordinate system is set, coordinates by absolute command will represent the positions on this coordinate system.
The command has not to be used for all axes at the same time.
For changing coordinate systems in the midway on a program, command only the axis for which change is required.

Example 1:

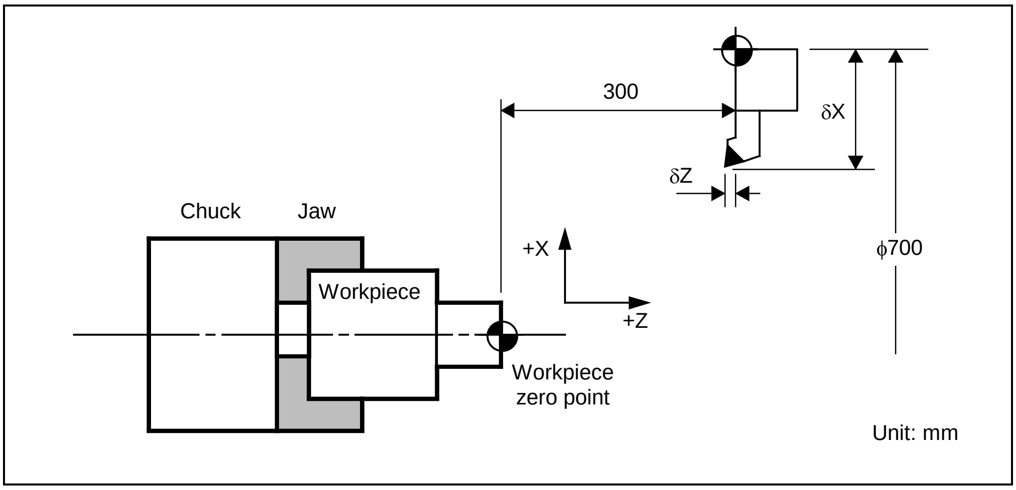



```
G50 X350. Z300.;
```

For setting a coordinate system with point A as zero point, command:

```
G50 X350. Z350.;
```

Example 2:



For setting at a reference point, command:

```
G50 X700. Z300.;
```

This coordinate system uses the center of turret rotation as a reference point.
Any point can be used as a reference. For δX and δZ, tool position compensation is used. For details, refer to Section 12-2.

Remarks

- The base machine coordinate system is shifted by G50 command to set an virtual machine coordinate system.
- Spindle clamp revolution speed is set by G50 with S or Q command. (Refer to the section for spindle clamp speed setting.)
- If the MAZATROL coordinate system is selected, validity or invalidity of G50 coordinate system can be selected by parameter setting.
- If a coordinate system is set by G50 during compensation, the coordinate system will be set on which the position specified by G50 is the position without compensation.
- Nose radius compensation is temporarily cancelled by G50.
- S data in a block with G50 will be regarded as spindle clamp revolution speed setting, but not as ordinary S data.

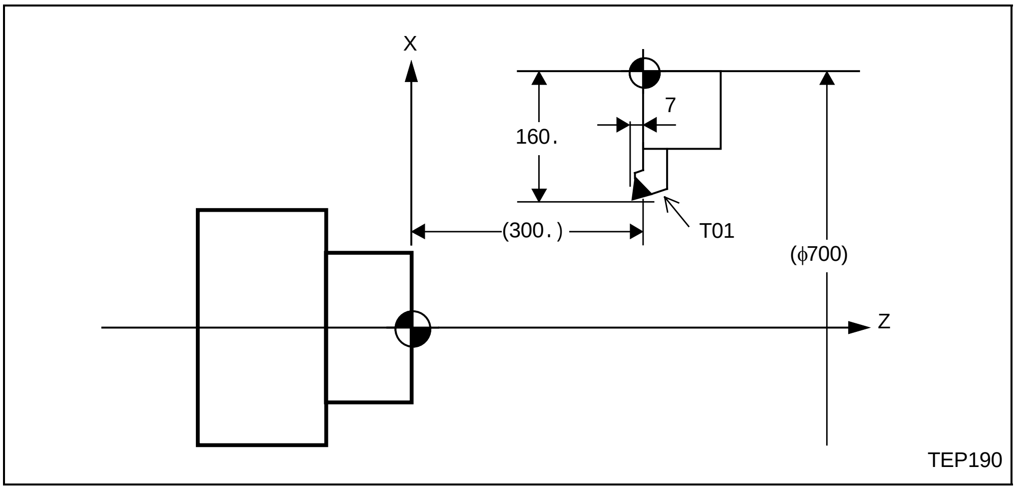
4. Coordinate system shift

A coordinate system can be shifted by a command as below:

G50 U_ W_ H_ ;

This command will create a new coordinate system where a point on a tool, for example, the tool tip position represented by (X, Z) in the preceding coordinate system will be represented by (X + U, Z + W). In other words, it is equivalent to the following:
G50X (present position X + U), Z (present position Z + W) and C (present position C + H);

Example:



TEP190

(G50 X700. Z300. ;)
(T001T000M06)
G50 U–320. W–7. ;

In the above example, T01 coordinate system is corrected by shift command.

5. G50 command and indications on POSITION and MACHINE counters

Example 1:

Offset data (–10.0, –10.0)

Program	Parameter **K92** bit 2 = 0		Parameter **K92** bit 2 = 1	
	POSITION	MACHINE	POSITION	MACHINE
`N001 G28 U0 W0;`	(0, 0)	(0, 0)	(0, 0)	(0, 0)
`N002 G50 X0. Z0.;`	(0, 0)	(0, 0)	(0, 0)	(0, 0)
`N003 T001T000M06D001;`	(0, 0)	(0, 0)	(–10, –10)	(–10, –10)
`N004 G00 X50. Z50.;`	(40, 40)	(40, 40)	(40, 40)	(40, 40)
`N005 G50 X0. Z0.;`	(–10, –10)	(40, 40)	(–10, –10)	(40, 40)
`N006 G00 X50. Z50.;`	(40, 40)	(90, 90)	(40, 40)	(90, 90)
`N007 T001T000M06D000;`	(40, 40)	(90, 90)	(50, 50)	(100, 100)
`N008 G00 X0. Z0.;`	(0, 0)	(50, 50)	(0, 0)	(50, 50)
`N009 G28 U0 W0;`	(–50, –50)	(0, 0)	(–50, 50)	(0, 0)
`N010 M02;`	(–50, –50)	(0, 0)	(–50, –50)	(0, 0)

Return to Library

Example 2:

Offset data (–10.0, –10.0)

Program	Parameter **K92** bit 2 = 0		Parameter **K92** bit 2 = 1	
	POSITION	MACHINE	POSITION	MACHINE
`N001 G28 U0 W0;`	(0, 0)	(0, 0)	(0, 0)	(0, 0)
`N002 G50 X0. Z0.;`	(0, 0)	(0, 0)	(0, 0)	(0, 0)
`N003 G00 X50. Z50.;`	(50, 50)	(50, 50)	(50, 50)	(50, 50)
`N004 G50 X0. Z0. T001T000M06D001;`	(0, 0)	(50, 50)	(–10, –10)	(40, 40)
`N005 G00 X50. Z50.;`	(40, 40)	(90, 90)	(40, 40)	(90, 90)
`N006 T001T000M06D000;`	(40, 40)	(90, 90)	(50, 50)	(100, 100)
`N007 G00 X0. Z0.;`	(0, 0)	(50, 50)	(0, 0)	(50, 50)
`N008 G28 U0 W0;`	(–50, –50)	(0, 0)	(–50, –50)	(0, 0)
`N009 M02;`	(–50, –50)	(0, 0)	(–50, –50)	(0, 0)

Example 3:

Offset data (–10.0, –10.0)

Program	Parameter **K92** bit 2 = 0		Parameter **K92** bit 2 = 1	
	POSITION	MACHINE	POSITION	MACHINE
`N001 G28 U0 W0;`	(0, 0)	(0, 0)	(0, 0)	(0, 0)
`N002 G50 X0.;`	(0, 0)	(0, 0)	(0, 0)	(0, 0)
`N003 G00 X50. Z50.;`	(50, 50)	(50, 50)	(50, 50)	(50, 50)
`N004 G50 X0. Z0. T001T000M06D001;`	(0, 0)	(50, 50)	(–10, –10)	(40, 40)
`N005 G00 X50. Z50.;`	(40, 40)	(90, 90)	(40, 40)	(90, 90)
`N006 G28 U0 W0;`	(–50, –50)	(0, 0)	(–50, –50)	(0, 0)
`N007 M02;`	(–50, –50)	(0, 0)	(–50, –50)	(0, 0)

Note: Bit 2 of parameter **K92** setting selects whether compensation movement is made on T command (Yes = 1, No = 0).

15-2 MAZATROL Coordinate System Cancellation: G52.5 (Series T)

1. Function and purpose

It is a function to select ordinary workpiece coordinate system (G54 to G59). If MAZATROL coordinate system is selected, it is cancelled by this function.

- Command of G52.5 is ignored in G52.5 mode.
- When G52.5 is commanded in G53.5 mode, the MAZATROL coordinate system currently provided is cancelled. The workpiece coordinate system that has been provided before G53.5 was commanded is reset. Therefore, display of current position counter is changed.

2. Detailed description

1. When G52.5 and other move command are given in the same block, the coordinate system is changed independently of the order of programs after the move command is executed.
2. When G52.5 and G50 are commanded in the same block, G50 is executed first and G52.5 is executed next independently of the order of programs.
3. G52.5 should be commanded in an independent block, if possible.

   Program example

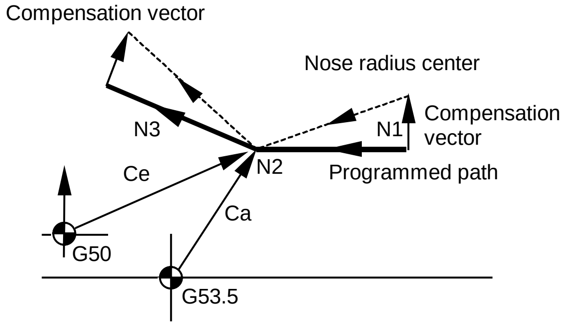
   ```
   G50  X__ Z__;
     :
   G00  X__ Z__;     POSITION counter with respect to the G50 system
   G53.5;            POSITION counter changed for the G53.5 system
   G00  X__ Z__;
     :
   G00  X__ Z__;     POSITION counter with respect to the G53.5 system
   G52.5;            POSITION counter changed for the preceding G50 system
   ```

4. It depends on parameter (**F114** bit 6) whether G52.5 or G53.5 is selected when turning the power on or resetting.

   F114 bit 6 = 0 : Initial selection G52.5
   F114 bit 6 = 1 : Initial selection G53.5

   Here, “reset” refers to the following:
   - When the reset key is pressed in G53.5 mode.
   - When the program is ended in G53.5 mode. (M30, M998, M999, %)

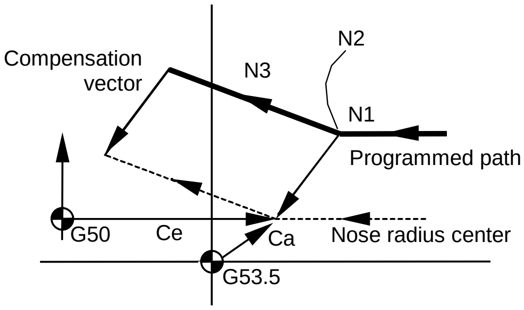
5. When G52.5 is commanded during nose radius compensation, compensation data will be temporarily cancelled in the movement block just before G52.5 command. At the time of G52.5 command, the programmed position and virtual tool tip point are identical.




```
G53.5;
G41;
 :
N1  G00 Z10.;
N2  G52.5;
N3  G01 X20. Z20.;
```

TEP183

6. If G52.5 is commanded during tool position compensation, compensation data will not be cancelled.




```
G53.5;
 :
N1  G00 Z100.;
N2  G52.5;
N3  G01 X100. Z100.;
```

TEP184

Return to Library

15-3 Selection of MAZATROL Coordinate System: G53.5 (Series T)

1. Function and purpose

G53.5 Z_ C_;

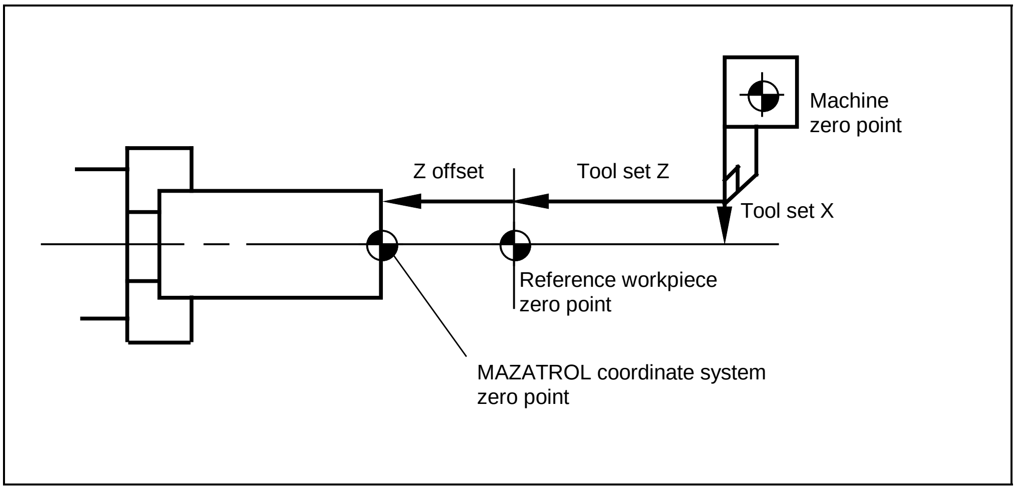
Z_: Z-offset value (The corresponding setting on the **SETUP INFORMATION** display will be used when argument Z is omitted.)

C_: C-offset value (The corresponding setting on the **SETUP INFORMATION** display will be used when argument C is omitted.)

Workpiece coordinate system (G54 to G59) is changed to MAZATROL coordinate system. MAZATROL coordinate system is a system set by machine position, *tool set value and *Z-offset value established beforehand. Use of the function eliminates the need for complex coordinate handling. And programs can be constracted with the same image as MAZATROL.

* Tool set: It is the distance of tool tip movement from the machine zero point to the reference workpiece zero point.

* Z-offset: Cutting workpieces different in length with reference to the reference workpiece zero point provided as a reference to set a tool set requires the shift of the reference workpiece zero point in Z-axis direction. Z-offset is the difference between MAZATROL coordinate system zero point agreed with the workpiece and reference workpiece zero point.



As shown above, a position where the tool tip is moved on the workpiece end to the center of rotation is set to the zero point of MAZATROL coordinate system. In other words, it is equivalent to giving “G50X0Z0 ;” command at the position.

- Command of G53.5 is ignored in G53.5 mode.
- When G53.5 is commanded in G52.5 mode, the workpiece coordinate system currently provided is cancelled, and the tool coordinate system for a tool currently selected is set. Current position display is also changed to a value of new coordinate system then.
- When tool change command (T command) is executed in G53.5, the tool coordinate system is automatically changed.
- When T command and move command are executed to the same block in G53.5 mode, the coordinate system is changed independently of the order of programs to the coordinate system of the tool selected after the move command is executed.
- When G53.5 and other move command are given to the same block, the coordinate system is changed independently of the order of programs after the move command is executed.

2. Detailed description

1. It depends upon the setting of the parameter concerned (G50 “valid/invalid” in G53.5 mode) whether the coordinate system is changed to the coordinate system of G50 when G50 is commanded in G53.5 mode.
   - When G50 “invalid” is set, the coordinate system is not changed except “G50S__;”.
   - When G50 “valid” is set, the coordinate system of G53.5 is ignored, and the coordinate system is changed to the workpiece coordinate system of G50.
2. When MAZATROL coordinate system is cancelled by commanding G52.5 in G53.5 mode, the coordinate system is returned to the workpiece coordinate system of G54 to G59. The coordinate system to be restored is a coordinate system just before G53.5 mode is established, and it is not a coordinate system set by G50 in G53.5 mode.
   For example, in the first (commanded first in the program) G52.5 at initial selection G53.5, thellows parameter P1 coordinate system of G54 is selected.
3. G53.5, T command in G53.5 mode and G52.5 are divided into two blocks. The first block does not make a single block stop.

Command	Output block 1	Output block 2
G53.5	Commands other than G53.5	G50
T command in G53.5 mode	T command, movement command, etc.	G50 by T command
G52.5	Commands other than G52.5	G50

3. Supplement

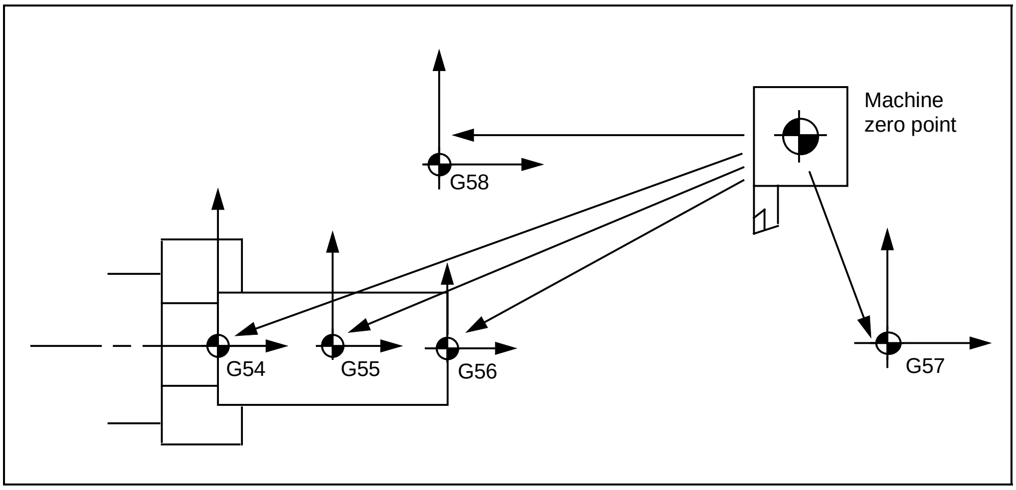
1. Selection of workpiece coordinate system (G54 to G59) is ignored in G53.5 mode.
2. Selection of local coordinate system (G52) is ignored in G53.5 mode.
3. When all the axes other than C-axis have not finished zero point return using watchdog method, G53.5 command will cause an alarm. However, when G53.5 command and G28 command are given in the same block and when the zero point return of all the axes are finished by executing G28, an alarm does not occur.
4. When G53.5 and G50 are commanded in the same block, the command is executed in order of G50 to G53.5 independently of the order of programs.
5. When two or more pieces of G50 exist in a change to G53.5 mode, the coordinate system of the last G50 is valid.
6. When T command and G50 are commanded in the same block in G53.5 mode, the coordinate system of a tool selected after execution of G50 is selected provided that G50 is valid in G53.5 mode.
7. When T command and G52.5 are given in the same block in G53.5 mode, the coordinate system is not changed by T command, and G52.5 is only executed.

15-4 Selection of Workpiece Coordinate System: G54 to G59

1. Function and purpose

G54; Workpiece coordinate system 1
G55; Workpiece coordinate system 2
G56; Workpiece coordinate system 3
G57; Workpiece coordinate system 4
G58; Workpiece coordinate system 5
G59; Workpiece coordinate system 6

Commanding the above permits selection/change of one of six coordinate systems specified beforehand that are agreed with the machine. (For G53.5 MAZATROL coordinate system, it is ignored.) By this command, subsequent axis commands are used as positioning at selected workpiece coordinate system until the reset key is pressed.



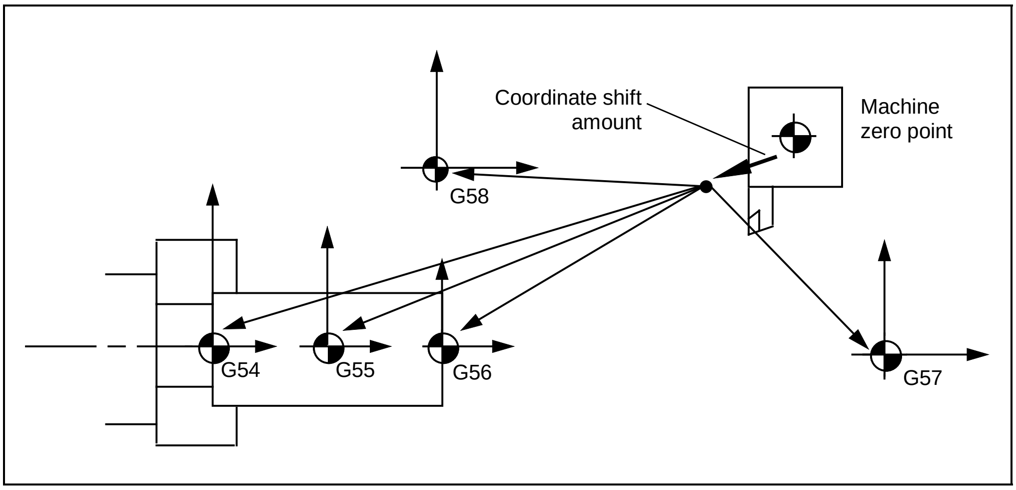
For the six workpiece coordinate systems, set the distance of each axis from the machine zero point to the zero point of each coordinate system on the **WORK OFFSET** display.

2. Remarks

1. When G54 to G59 and move command are given in the same block, the coordinate system is changed to the specified one to move to the specified position on a new coordinate system.
2. When G54 to G59 are changed independently, the counter display of current position changes to a value on the specified coordinate system. (Machine does not move.)
3. Workpiece coordinate systems 1 to 6 are correctly established after reference point return after the power is turned on.
4. When the power is turned on or when the reset key is pressed, G54 is selected.
5. The distance when coordinate system is moved by G50 is added subsequently to all workpiece zero point offset values. For example, when coordinate system is moved by “G50 U_ W_ ” command in the selection of G54, G55 to G59 also move by the same distance. Therefore, take care when having changed to G55.
6. The coordinate system cannot be established exactly for the C-axis by a command of G54 to G59 if it is given with the C-axis not being connected. Do not fail, therefore, to select the milling mode (for C-axis connection) before entering G54 to G59 as required for the C-axis.

15-5 Workpiece Coordinate System Shift

Difference may be caused between the workpiece coordinate system considered in programming and the coordinate system specified actually by G50 command or G54 to G59 command. The coordinate system being specified can be shifted then. The amount to be shifted is specified at the **SHIFT** item on the **WORK OFFSET** display.



All the six workpiece coordinate systems are shifted by the coordinate shift amount.

- When the shift amount is specified, the workpiece coordinate system is shifted immediately. (The shift amount is reflected in the current position counter.)
- When "G50 X_ Z_" is specified after the shift amount has been specified, the shift amount is ignored.

15-6 Change of Workpiece Coordinate System by Program Command

G10 L2 P_ X(U)_ Z(W)_ C(H)_ ;

P = 0:	Coordinate shift amount is specified.
P = 1 to 6:	Specified to workpiece coordinate system 1 to 6
X, Z, C:	Workpiece zero point offset value of each axis

The above command permits rewriting a workpiece offset value to change the position of workpiece coordinate system. To move the workpiece coordinate system for each program, it is commanded at the head of a program.

Note: The timing at which the rewritten value becomes effective is after G54 to G59 command is subsequently executed.

15-7 Selection of Machine Coordinate System: G53

1. Function and purpose

G53 X_ Z_ C_ ;

The above command permits moving the tool to the commanded position in the machine coordinate system at rapid feed. G53 is valid only for the commanded block.
To move a tool to the position specifically set for the machine including tool change position, command G53 using the machine coordinate system.
A base point on the machine is referred to as the machine zero point. Machine zero point depends on machine specifications.
A coordinate system using machine zero point as the zero point of coordinate system is referred to as machine coordinate system.
The tool cannot always move to the machine zero point. In some cases, machine zero point is set at a position to which the tool cannot move.
Machine coordinate system is established when the reference point return is executed after the power is turned on.
Once the machine coordinate system is established, it is not changed by reset, workpiece coordinate system setting (G50), local coordinate system setting (G52) and other operation unless the power is turned off.
Stored stroke limit (G22, G23), which specifies the stroke of the machine, must be set using the coordinate value of the machine coordinate system.

2. Remarks

1. When G53 is commanded, tool offset and tool nose radius compensation must be cancelled. (Because they are not incorporated when G53 is commanded.)
2. Since the machine coordinate system must be set before G53 is commanded, at least one manual reference point return or automatic reference point return by G28 should be executed after the power is turned on.
3. G53 with incremental command can be commanded, but it is meaningless.
4. Virtual axes such as Y-axis cannot be commanded. (The execution gives an alarm.)
5. This command is valid even in MAZATROL coordinate system (G53.5).

15-8 Selection of Local Coordinate System: G52

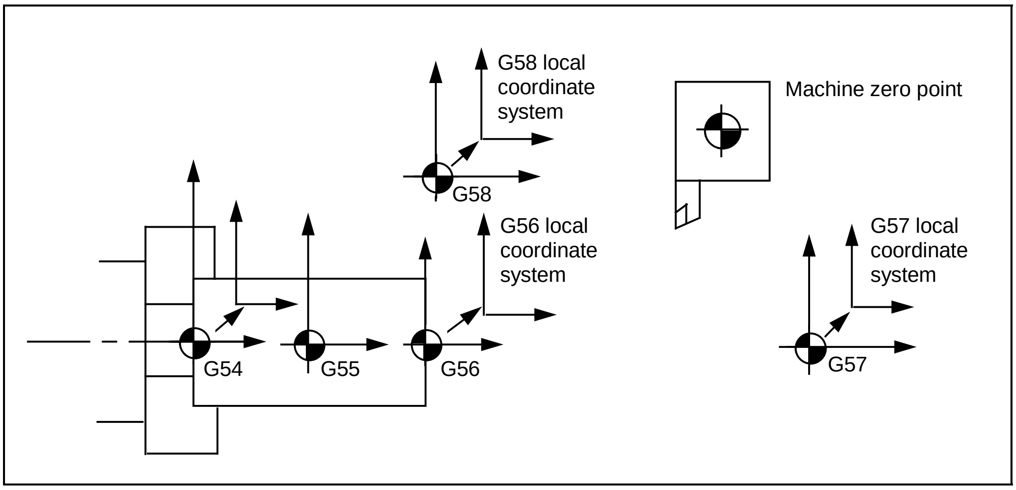
G52 X(U)_ Z(W)_ C(H)_ ;

The above command permits specifying further each coordinate system (local coordinate system) in each workpiece coordinate system (G54 to G59).

When local coordinate system is specified, the move commands given subsequently provide coordinate values in the local coordinate system.

To change local coordinate system, the zero point position of a new local coordinate system is commanded using workpiece coordinate system together with G52.

To cancell the local coordinate system, to align the zero point of coordinate system with that of workpiece coordinate system. That is, “G52 X0Z0 ;” must be commanded.

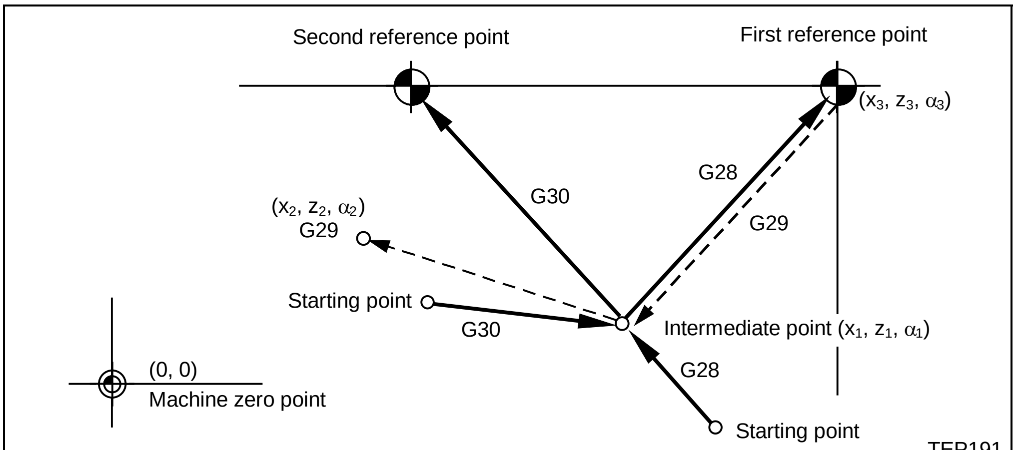


Note: G52 can be used in place of G50 command. However, the distance through which coordinate system is shifted by G52 is not added to other workpiece zero point offset values.

15-9 Automatic Return to Reference Point (Zero Point): G28, G29

1. Function and purpose

- Setting of command G28 allows each of the designated axes to be returned separetely to the first reference point (zero point) at a rapid feed rate after all those designated axes have been positioned in G00 mode.
- Setting of command G29 allows each axis to be placed sepatately at the G28 or G30 specified intermediate point at high speed and then placed at the designated position in G00 mode.



TEP191

2. Programming format

G28 Xx_1 Zz_1 $\alpha\alpha_1$; (α: Additional axis) Automatic return to reference point
G29 Xx_2 Zz_2 $\alpha\alpha_2$; (α: Additional axis) Return to start point

3. Detailed description

1. Command G28 is equivalent to the following commands:

   G00 Xx_1 Zz_1 $\alpha\alpha_1$;
   G00 Xx_3 Zz_3 $\alpha\alpha_3$;

   Where x_3, z_3 and a_3 denote the coordinates of the appropriate reference point, determined by the parameter as the distance from the zero point of the base machine coordinate system.

2. Axes that have not been returned to the reference point (zero point) in manual mode after power-on are returned using the watchdog method. In that case, the direction of return is regarded as the same as the designated direction. For the second time onward, the axes are returned at high speed to the reference point that was stored into the memory by execution of the first return command. (The return using the watchdog method can also be parameter-set for the second time onward.)

3. When return to reference point (zero point) is completed, a return complete output signal will be outputted and the monitor of the operation panel will display “#1” in the display field of the axis name.

4. Command G29 is equivalent to the following commands:

   G00 Xx_1 Zz_1 $\alpha\alpha_1$;
   G00 Xx_2 Zz_2 $\alpha\alpha_2$; } This results in independent rapid feed of each axis.

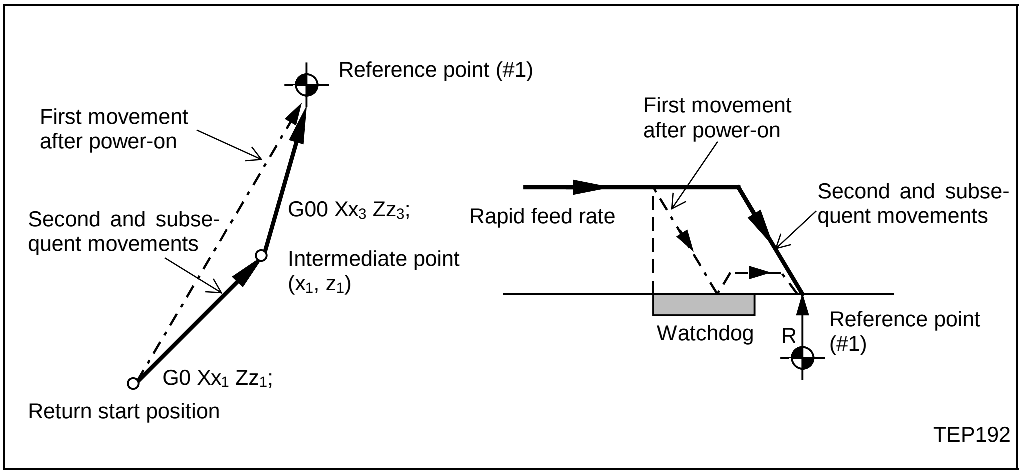
   Where x_1, z_1 and a_1 are the coordinates of the intermediate point specified by G28 or G30.

Return to Library

5. A program error will result if G29 is commanded without G28 (automatic return to reference point command) or manual return to zero point being executed after the power has been turned on.
6. The intermediate point coordinates (x_1, z_1, α_1) for the positioning are set by absolute/incremental value.
7. G29 is valid for either G28 or G30, but the commanded axes are positioned after a return has been made to the latest intermediate point.
8. The tool offset will be temporarily cancelled during return to reference point unless it is already cancelled but the intermediate point will be the offset position.
9. During return to reference point under a machine lock status, movement from the intermediate point to the reference point is omitted. The next block is executed after the designated axis has arrived at the intermediate point.
10. During return to reference point in the mirror image mode, the mirror image is valid for movement from the starting point to the intermediate point and the axis moves in an opposite direction to the corresponding point. For movement from that point to the reference point, however, the mirror image become invalid and thus the axis moves to the reference point.
11. In case of cycle start in the single step mode, stop will be made at the intermediate point.

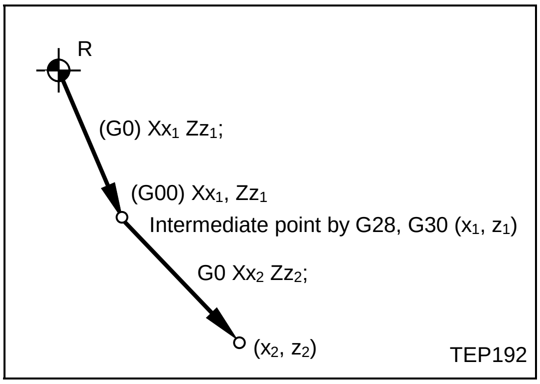
4. Sample programs

Example 1: G28 Xx_1 Zz_1;



TEP192

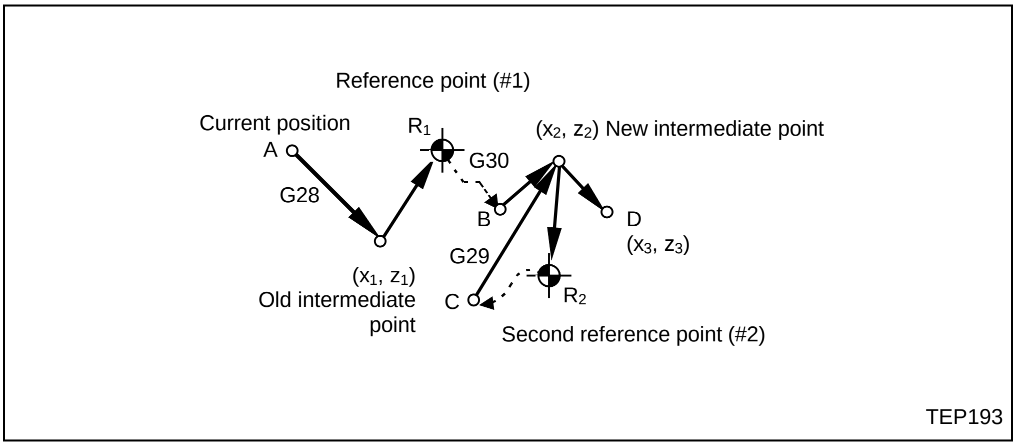
Example 2: G29 Xx_2 Zz_2;



TEP192

Return to Library

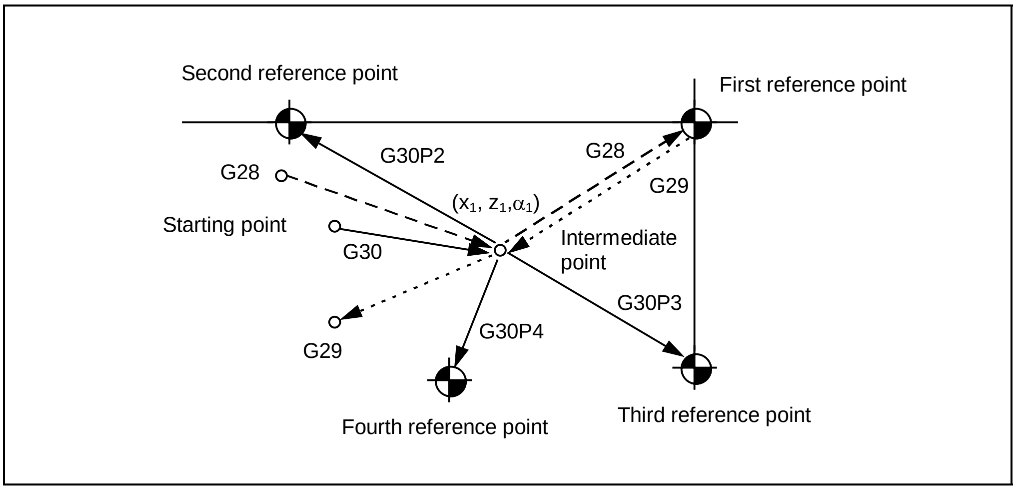
Example 3: G28 Xx_1 Zz_1;
⋮ From point A to reference point
⋮
G30 Xx_2 Zz_2;
⋮ From point B to second reference point
⋮
G29 Xx_3 Zz_3;
From point C to point D



15-10 Return to Second Reference Point (Zero Point): G30

1. Function and purpose

The designated axis can be returned to the second, third, or fourth reference point (zero point) by commanding G30 P2 (P3, P4).



2. Programming format

G30 P2 (P3, P4) Xx_1 Zz_1 $\alpha\alpha_1$; (α: Addition axis.)

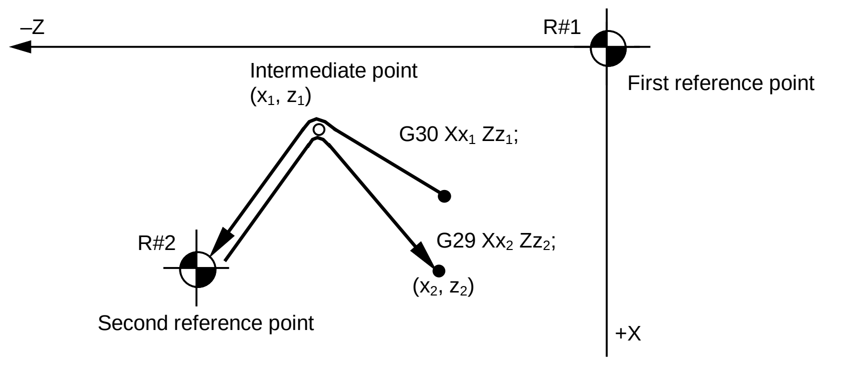
3. Detailed description

1. Command P2, P3, or P4 can be used to designate returning to the second, third, or fourth reference point (zero point). However, the return to the second reference point is automatically selected if P-command is omitted or zero, one, five or a greater integer is set at address P.

2. When return to second, third, or fourth reference point is specified, it will be executed to the position of the second, third, or fourth reference point via the intermediate point specified by G30 like the return to the first reference point.

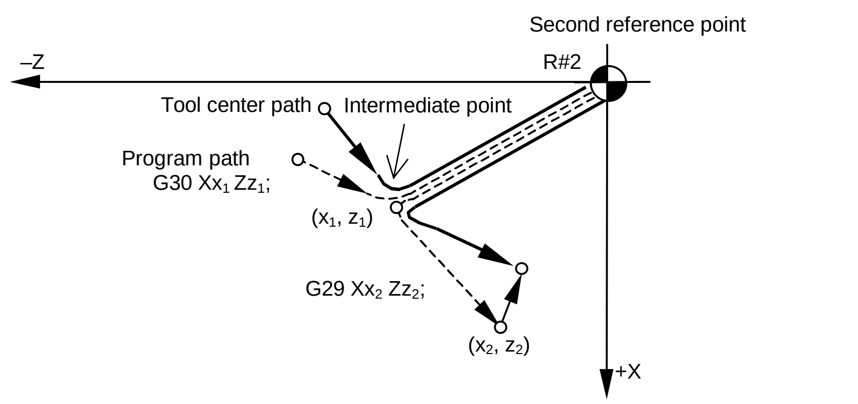
3. The coordinates of the second, third, or fourth reference point represent the position specific to the machine. The coordinates can be checked by using the machine parameter **M5**, **M6**, or **M7**.

4. If the return to second, third, or fourth reference point command is followed by command G29, the intermediate point of return by G29 will be set at that of the return to reference point operation lastly performed.



TEP196

5. If the plane containing the designated reference point is currently undergoing tool nose radius compensation, the designated axis will move to the intermediate point according to the tool nose radius compensation data. The movement of the axis from the intermediate point to the second reference point will become free from that compensation data. For the next G29 command, movement from the reference point to the intermediate point will be based without the tool nose radius compensation data which will apply to the movement from the intermediate point to the point specified with G29 command.



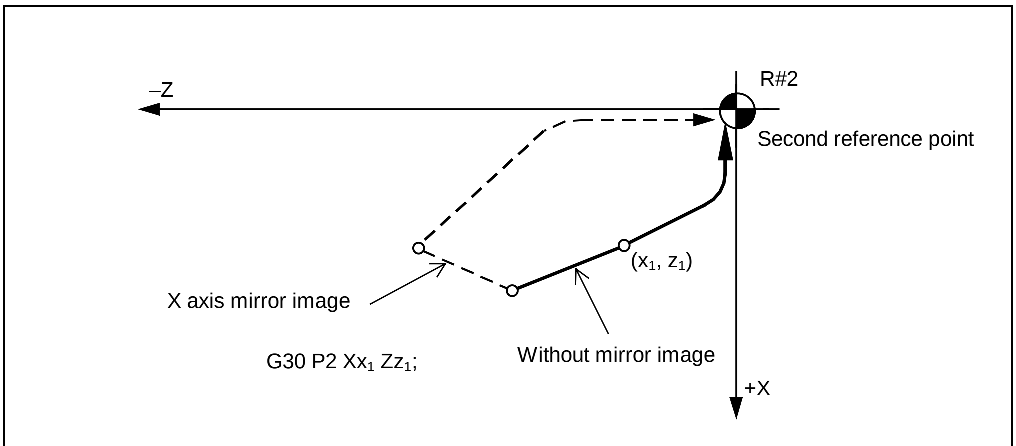
TEP197

6. After return to second reference point, the tool nose radius compensation data for the next movement is cancelled automatically.

7. During return to second reference point under a mahcine lock status, movement from the intermediate point to the reference point is omitted. The next block is executed after the designated axis has arrived at the intermediate point.

Return to Library

8. During return to second reference point in the mirror image mode, the mirror image is valid for movement from the starting point to the intermediate point and the movement is effectuated in an opposite direction to the corresponding point. For movement from that point to the reference point, however, the mirror image becomes invalid and thus the axis moves to that reference point.



TEP198

9. In case of cycle start in single step mode, no stop will be made at the intermediate point.

15-11 Return to Reference Point Check Command: G27

1. Function and purpose

As with command G28, execution of command G27 will output a return to reference point return complete signal to the machine if the point at which the designated axis has been positioned by the program is the first reference point. Thus, if the axis is programmed to start moving from the first reference point and then returns to that reference point, you can check whether the axis has returned to the reference point after execution of the program.

2. Programming format

G27 Xx_1 Zz_1 $\alpha\alpha_1$ Pp_1 ;

- Pp_1: Check No.
  - P1 : First reference point check
  - P2 : Second reference point check
  - P3 : Third reference point check
  - P4 : Fourth reference point check
- Xx_1 Zz_1 $\alpha\alpha_1$: Return control axis
- G27: Check command

3. Detailed description

- The first reference point check will occur if the P command is omitted.
- The number of axes for which reference point checks can be done at the same time depends on the number of simultaneously controllable axes.
- An alarm will result if the axis has not arrived at the designated reference point on completion of this command.

15-12 Programmed Coordinate Conversion ON/OFF: G68.5/G69.5 [Series M: G68/G69]

1. Outline

This command is used to determine a new coordinate system through the translation of the origin of the currently active workpiece coordinate system and the rotation on an axis of coordinate. Use this command to specify freely a plane in space which is convenient for programming.

2. Programming format

G68.5 Xx_0 Zz_0 Yy_0 Ii Jj Kk Rr ; .... Program coordinate system rotation ON

G69.5 ; .... Program coordinate system rotation OFF

Xx_0 Zz_0 Yy_0 : Coordinates of the center of rotation
Specify in absolute dimensions the translation of the workpiece origin.

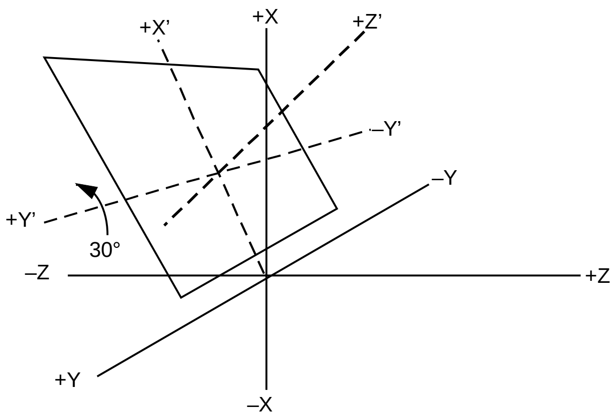
i, j, k : Designation of rotational axis (1: valid, 0: invalid)
I : X-axis
J : Y-axis
K : Z-axis

r : Angle and direction of rotation on the rotational axis
A positive value of angle refers to the left turn when seen from the positive side of the rotational axis.

3. Detailed description

- It is impossible to change the coordinate system in the G68.5 mode.
- The coordinate system set by a command of G68.5 is as indicated here below.

```
G17  G68.5 X10. Z–5. Y0 I0 J1 K0 R30.;
```

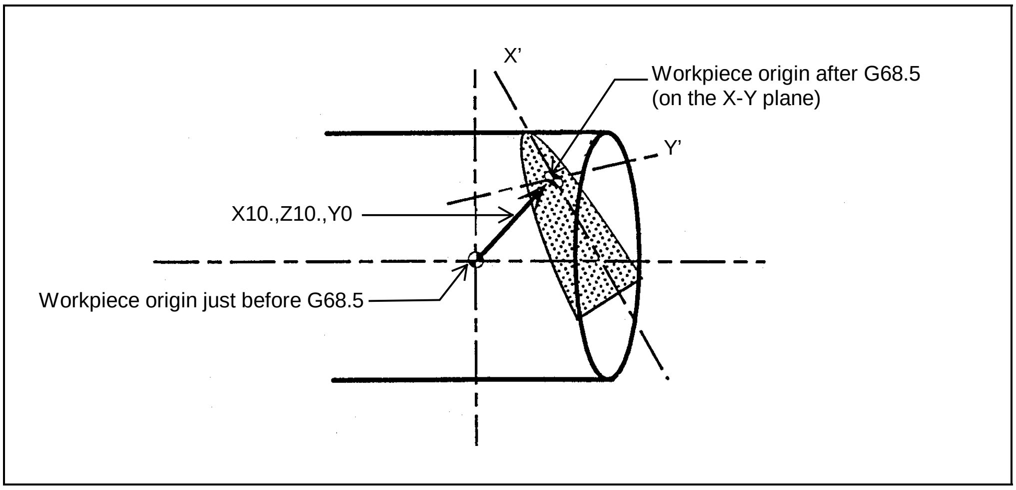


After the selection of the G17 (X-Y) plane, the workpiece origin is shifted to the point
(X, Z, Y) = (10, –5, 0)
and the plane is rotated by 30 degrees on the Y'-axis. The new coordinate system (X', Y', Z') has thus been established.

- The cancel command G69.5 will set again the coordinate system subjected to the translation and rotation by the preceding G68.5 command.
- In the G68.5 mode all the dimensions must be entered in radius values.

4. Sample program

```
G53.5;
N100 T003T000M06;
G00 B30.; ......................... Positioning on the B-axis
#100=200; ......................... Distance btw. B-axis rotat. center and tool ref. position
#1=-SIN[30.]*#100; ................ X-axial variation due to the B-axial rotation
#2=#100-COS[30.]*#100; ............ Z-axial variation due to the B-axial rotation
G50 X[#1+#5041] Z[#2+#5042]; ...... Correction of coordinate system against B-axial rotation
G68.5 X10.Z10.Y0 I0 J1 K0 R30.; ... Definition of coordinate system by translation of origin to
                                    (X10. Z10. Y0) and rotation on Y'-axis by 30°
G17; .............................. Selection of the X-Y plane
G00 X0 Y0 Z20.;
G41;
G01 Z-5. F50;
X10.Y-10.;
G03 XY10.R30.;
 ⋮
G40;
G69.5; ............................ Cancellation of program coord. system rotation mode
N200 T005T000M06; ................. Tool change, which is inhibited in the G68.5 mode
G00 B30.;
G50 X[#1+#5041] Z[#2+#5042]; ...... Correction of coordinate system against B-axial rotation
G68.5 X10.Z10.Y0 I0 J1 K0 R30.;
G17;
G00 Z10.;
G83 X40.Y-30.Z-30.R5.P100 F80; .... Positioning on G17 plane and hole machining on Z-axis
G80;
G00 Z10.;
G69.5;
 ⋮
M30;
%
```



5. Restrictions

1. The G68.5 command cannot be given in the following modes:

   - Tool nose radius compensation (G40 mode not selected)
   - Fixed cycle (G80 not selected in the G-code group 09)
   - Opposite turret mirror image (G68 mode)

2. The G68.5 command cannot be given in the mode of cross machining.

3. No tool change commands by T-code can be given in the G68.5 mode. A T-code in this mode will be processed as a programming error.

4. A block in the G68.5 mode cannot be designated as restart position. The search for such a block as restart position will cause an alarm.

5. Certain G-codes cannot be given in the G68.5 mode. Refer to the table "Usable G-codes in the G68.5 mode" that follows. An alarm will be caused if an unavailable G-code is given.

6. If the addresses X, Y and Z are all omitted, no translation of the origin will occur and the rotation will be performed on an existing axis of coordinate.

7. All the arguments I, J and K must be specified in general as required. If one of the arguments is omitted, such a block of G68.5 will be processed as a programming error.

   Example 1: `G68.5 X10.Z0 Y0 I0 J1 R30.;` Format error

   If, in particular, all the arguments are omitted, then the axis perpendicular to the currently selected plane will be regarded as the axis of rotation.

   Example 2: `G17;`
   `G68.5 X10.Z0 Y0 R30.;` Equiv. to G68.5 X10. Z0 Y0 I0 J1 K0 R30.;

8. A block of G68.5 will be processed as a programming error if all the arguments I, J and K are specified with zero (0).

   Example: `G68.5 X10.Z0 Y0 I0 J0 K0 R30.;` Format error

9. The codes G68.5 and G69.5 are not available for a system without the optional function of coordinate system rotation.

10. A MAZATROL program cannot be called up as subprogram in the G68.5 mode.

Usable G-codes in the G68.5 mode

G-code series T	Group	Function
G00	01	Rapid positioning
G01	01	Linear interpolation
G02	01	Circular interpolation, CW
G03	01	Circular interpolation, CCW
G02	01	Helical interpolation, CW
G03	01	Helical interpolation, CCW
G04	00	Dwell
G09	00	Exact stop
G10	00	Data setting mode
G11	00	Data setting mode cancel
G17	02	X-Y plane section
G18	02	Z-X plane section
G19	02	Y-Z plane section
G20	06	Inch command
G21	06	Metric command
G22	04	Stored stroke check ON
G23	04	Stored stroke check OFF
G32	01	Thread cutting
G34	01	Variable lead thread cutting
G40	07	Tool radius/nose radius compensation OFF
G41	07	Tool radius/nose radius compensation, left
G42	07	Tool radius/nose radius compensation, right
G60	00	Unidirectional positioning
G61	13	Exact stop mode
G62	13	Automatic corner override mode
G64	13	Cutting mode
G65	00	Macro call
G66	14	Modal macro call
G67	14	Modal macro call cancel
G69.5	16	Program coordinate system rotation mode cancel
G80	09	Hole machining cycle cancel
G83	09	Face drilling cycle
G84	09	Face tapping cycle
G84.2	09	Face synchronous tapping cycle
G85	09	Face boring cycle
G87	09	Outside drilling cycle
G88	09	Outside tapping cycle
G88.2	09	Outside synchronous tapping cycle
G89	09	Outside boring cycle
G90	09	Longitudinal turning cycle
G92	09	Thread cutting cycle
G94	09	Transverse turning cycle
G96	17	Constant peripheral speed control ON
G97	17	Constant peripheral speed control OFF
G98	05	Feed per minute
G99	05	Feed per revolution

15-13 Workpiece Coordinate System Rotation (Series M)

1. Function and purpose

The function refers to rotating the workpiece coordinate system around the position of the specified machine coordinates. The machining program can be rotated in its entirety as required for the actual inclination of the workpiece.

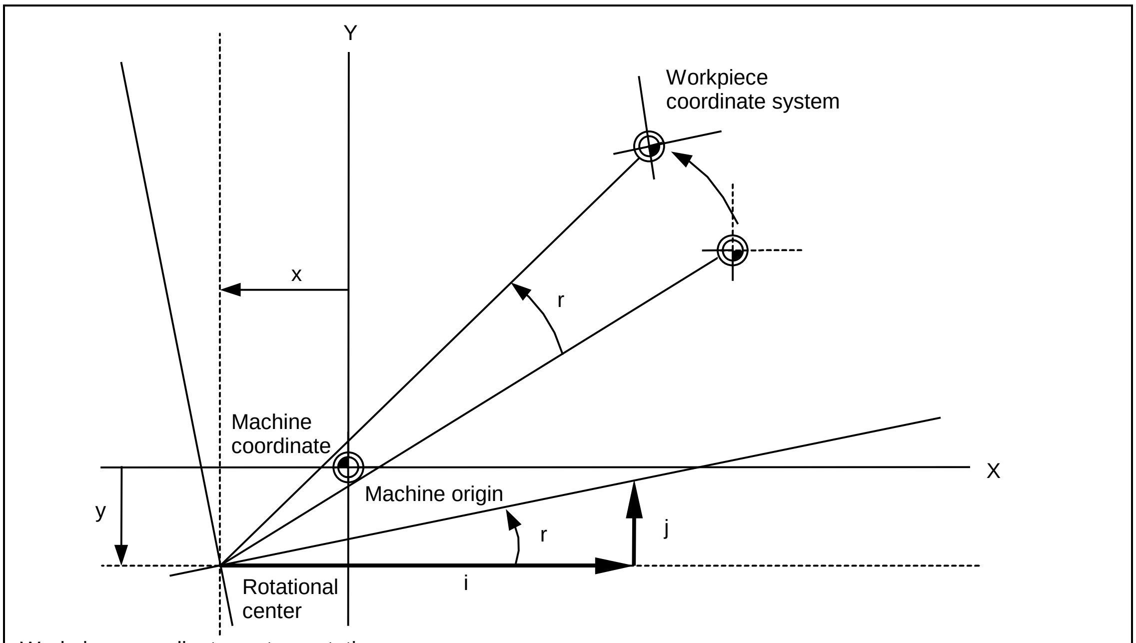
2. Programming format

(G17) G92.5 Xx Yy Rr ....X-Y plane
(G18) G92.5 Zz Xx Rr ....Z-X plane
(G19) G92.5 Yy Zz Rr ....Y-Z plane
or
(G17) G92.5 Xx Yy Ii Jj ....X-Y plane
(G18) G92.5 Zz Xx Kk Ii ....Z-X plane
(G19) G92.5 Yy Zz Jj Kk ....Y-Z plane

x, y, z : Coordinates of the rotational center.
The position along the two axes of the previously selected X-Y, Z-X, or Y-Z plane must be designated in machine coordinates.
The designation for an axis not corresponding to the plane will be ignored.

r : Angle of rotation.
Designate the rotational angle for the coordinate system. A positive value refers to a counterclockwise rotation.

i, j, k : Axial component vectors.
The angle for coordinate system rotation can also be designated in axial component vectors corresponding to the previously selected plane.
The designation for an axis not corresponding to the plane will be ignored.



Workpiece coordinate system rotation

Range and unit for angle data setting

Setting method		Setting range	Setting unit
Axial component vectors (i, j, k)	Metric system	0 to ±99999.999	0.001 mm
	Inch system	0 to ±9999.9999	0.0001 in.
Angle of rotation (r)	Metric system	0 to ±99999.999°	0.001°
	Inch system		

3. Detailed description

1. Irrespective of the actual mode for incremental or absolute data input, the values at addesses X, Y, Z, or I, J, K as well as R are always referred to the machine coordinate system.
2. Two methods are available to designate a rotational angle:
   (a) Designation in rotational angle (r), or
   (b) Designation in axial component vectors (i, j, k).
3. If angle data are entered using both methods (a) and (b) above, the rotational angle (at address R) will govern.
4. If, during rotation of the workpiece coordinate system, a rotational angle of zero degrees is designated (by setting G92.5 R0, for example), the coordinate system rotation will be cancelled, irrespective of the data input mode of G90 (absolute) or G91 (incremental). The next move command will then be executed for the ending point in the original (not rotated) workpiece coordinate system (refer to Article 1 in Item 5. Precautions).
5. The rotational center coordinates will be retained and automatically applied for a succeeding rotation command without data designation at addresses X, Y, and/or Z.

Example:

`N1 G17`	Selection of the X-Y plane
`N2 G92.5X100.Y100.R45.` ⋮	Rotation of the workpiece coordinate system through 45 deg around the point of (X, Y) = (100, 100)
`N3 G92.5R0` ⋮	Cancellation of the workpiece coordinate system rotation
`N4 G17G92.5R90.` ⋮ `%`	Rotation of the workpiece coordinate system through 90 deg around the center last programmed (X100, Y100)

6. Omission of addresses R and I, J , K is regarded as a rotational angle designation of zero degrees.

Example: "G92.5 X0. Y0." is equivalent to "G92.5 X0. Y0. R0".

7. Alarm No. **809 ILLEGAL NUMBER INPUT** will be displayed if the specified axial component vectors (i, j, k) or rotational angle (r) oversteps the effective setting range.
8. Plane selection (by codes G17, G18, and G19) need not be included in the block of G92.5, if the rotation shall be performed on the currently active plane.
9. The designation for an axis not corresponding to the selected plane will be ignored. The designations at addresses Z and K in a block of G92.5, for example, will be ignored in the mode of G17 (X-Y plane).

Example: The second block shown below rotates the workpiece coordinate system through 63.435 , calculation from $\tan^{-1}(2/1)$, around the point of (X, Y) = (10, 20) on the X-Y plane, and the values at Z and K are ignored for the rotation.

```
G17
G92.5X10.Y20.Z30.I1.J2.K3.
```

Even the ignored axial values at X, Y, and Z in a G92.5 block are retained as well as the values actually used (see Article 5 above) and, for example, if the G92.5 block shown above is followed by

```
G19
G92.5J2.K3.
```

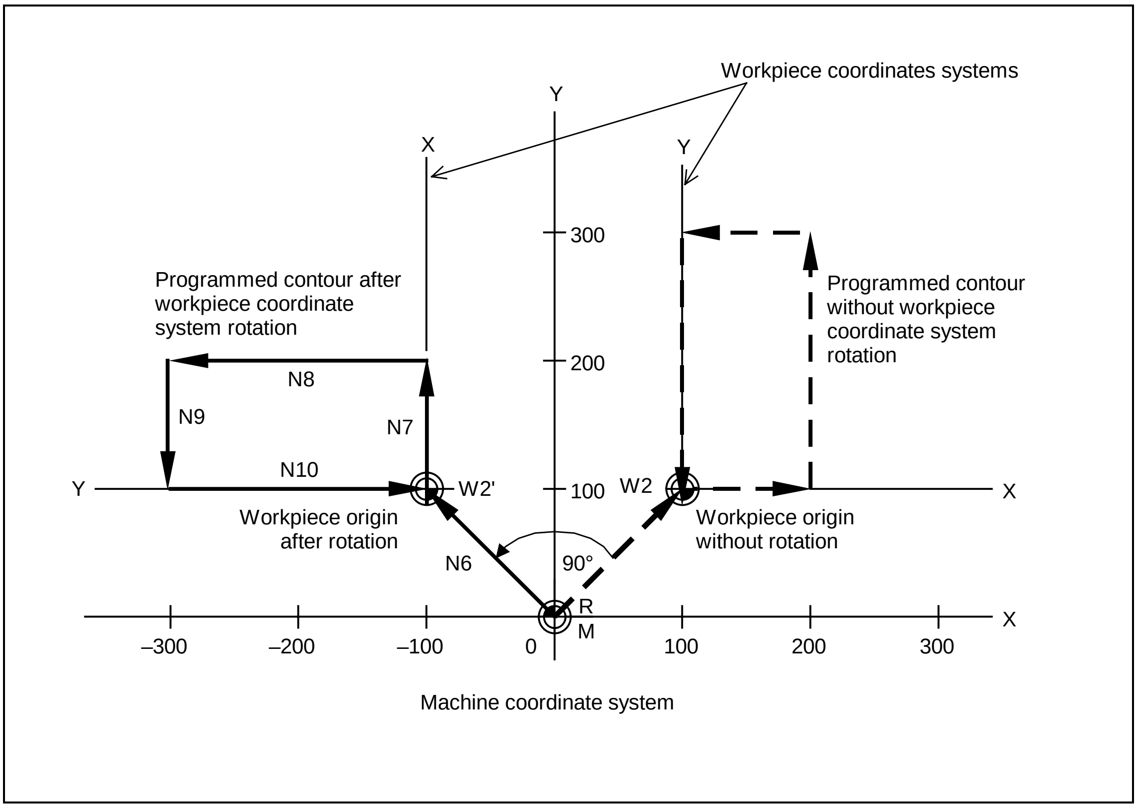
then the workpiece coordinate system will be rotated around the point of (Y, Z) = (20, 30) through 56.301°, calculation from $\tan^{-1}(3/2)$, on the Y-Z plane (G19).

4. Examples of operation

1. Rotation around the machine origin.

```
N1   G28X0Y0
N2   G17
N3   G90
N4   G55
N5   G92.5X0Y0R90. ................................ Rotation around the machine origin
     (or G92.5X0Y0I0J1.)                            through 90 deg
N6   G0X0Y0
N7   G1X100.F1000.
N8   Y200.
N9   X0.
N10  Y0.
N11  M30
%
```

G55 (Work Offset)
X100.
Y100.



- The block of G92.5 under N5 rotates the workpiece coordinate system through 90 degrees around the origin of the machine coordinate system. For N6 onward, the machine operates according to the rotated workpiece coordinate system.

- The above example of the vector setting method for the same 90-deg rotation is based on the following calculation:

  $\theta = \tan^{-1}(J/I) = \tan^{-1}(1/0) = 90°$.

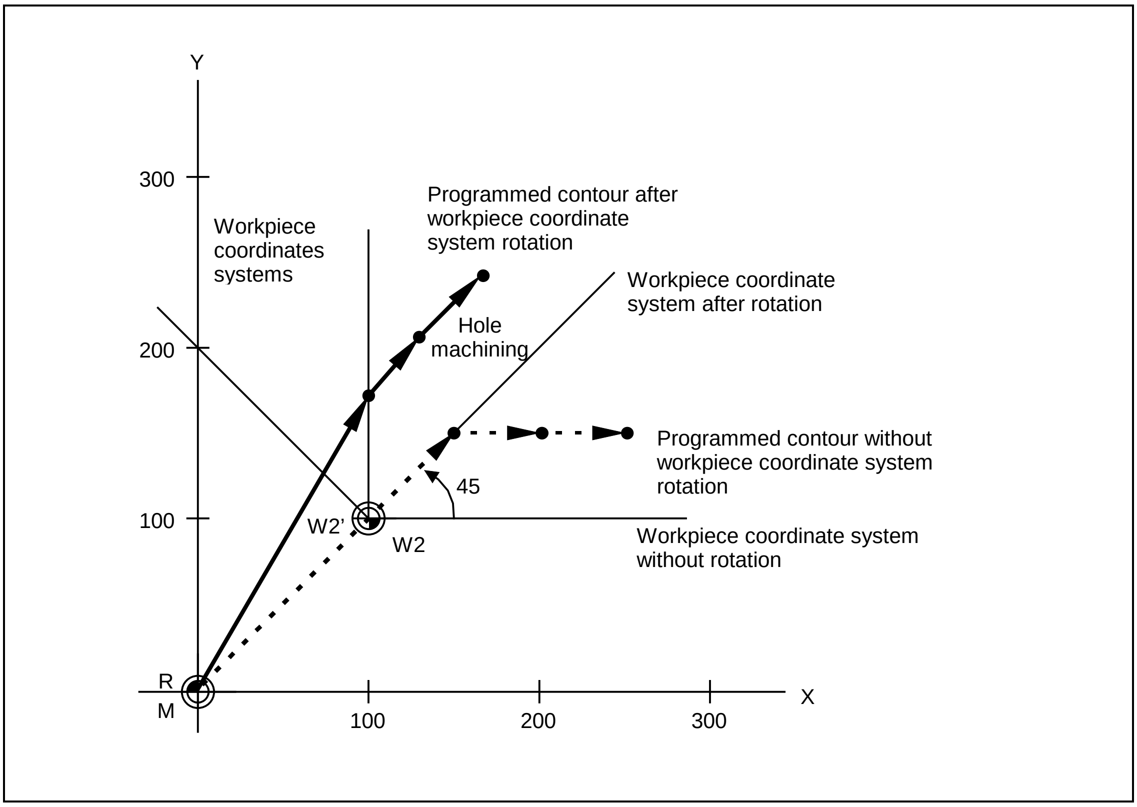
Return to Library

2. Rotation around the workpiece origin.

```
N1 G28X0Y0Z0
N2 G17
N3 G55
N4 G90
N5 G92.5X100.Y100.R45. ....................
N6 G81X50.Y50.Z-25.R-5.F500
N7 X100.
N8 X150.
N9 M30
%
```

G55 (Work Offset)
X100.
Y100.

Rotation through 45 deg around the point of machine coordinates X=100 and Y=100 (that is, the origin of the G55 workpiece coordinate system).

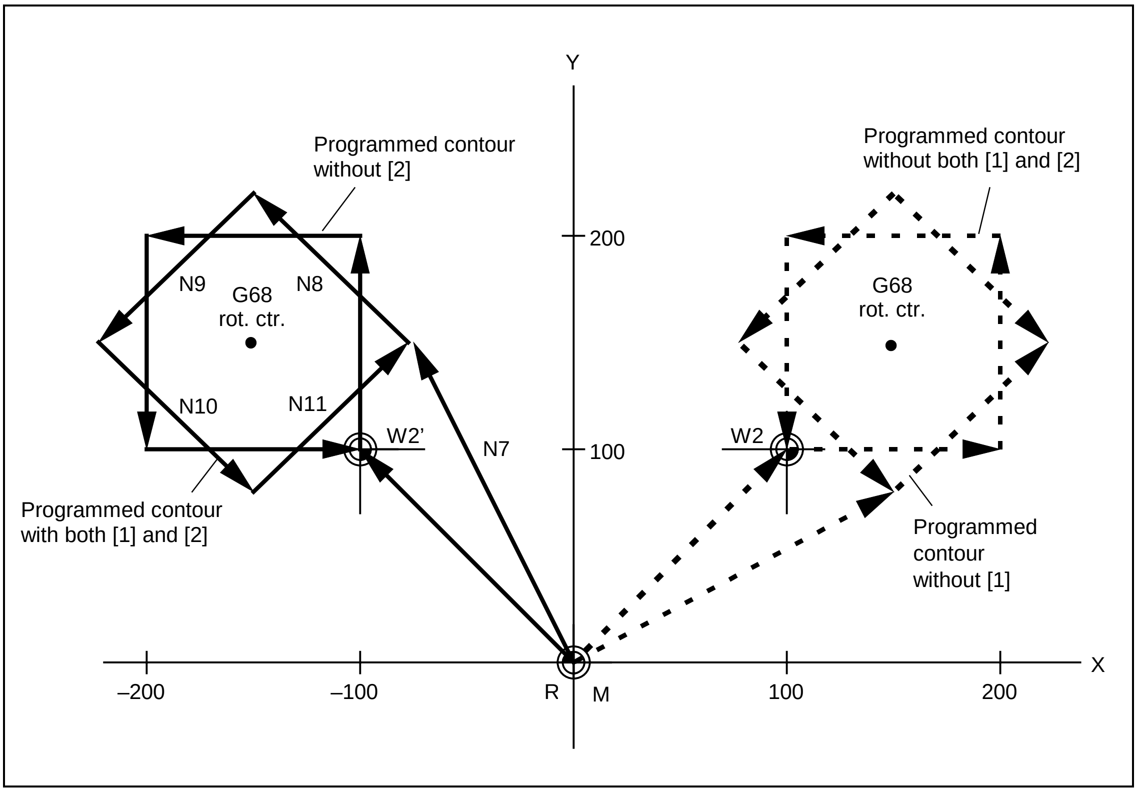


- The block of G92.5 under N5 rotates the workpiece coordinate system around its own origin through 45 degrees. For N6 onward, the machine operates according to the rotated workpiece coordinate system.
- Set the rotational center on the workpiece origin, as shown in this example, to rotate the current workpiece coordinate system around its own origin.

3. Programmed coordinate rotation (G68) in the mode of G92.5

```
N1  G28X0Y0
N2  G17
N3  G55
N4  G90
N5  G92.5X0Y0R90. ...............................  [1]
N6  G68X50.Y50.R45. ...........................  [2]
N7  G0X0Y0
N8  G1X100.F500
N9  Y100.
N10 X0
N11 Y0
N12 M30
%
```

G55 (Work Offset)
X100.
Y100.



In a combined use with G92.5, the center of programmed coordinate rotation by G68 will be a position which corresponds with the workpiece coordinate system rotation designated by the G92.5 command.

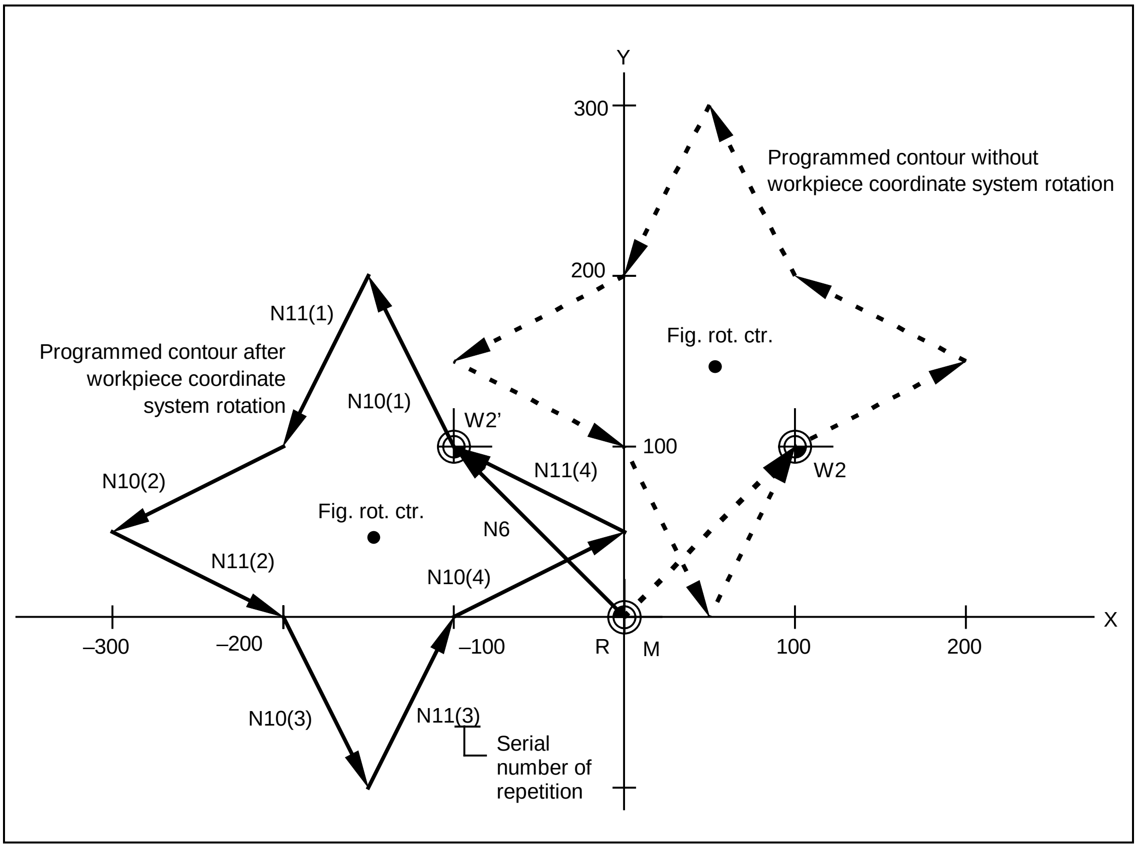
It will not affect operation even if the order of the program blocks marked [1] and [2] above is reversed.

4. Figure rotation (M98) in the mode of G92.5

```
N1   G28X0Y0
N2   G17
N3   G55
N4   G90
N5   G92.5X0Y0R90. ..........................
N6   G0X0Y0
N7   M98H10I–50.J50.L4
N8   M30
N9
N10  G1X100.Y50.F500
N11  X0Y100.
N12  M99
%
```

G55 (Work Offset)
X100.
Y100.

Rotation through 90 deg around the origin of the machine coordinate system

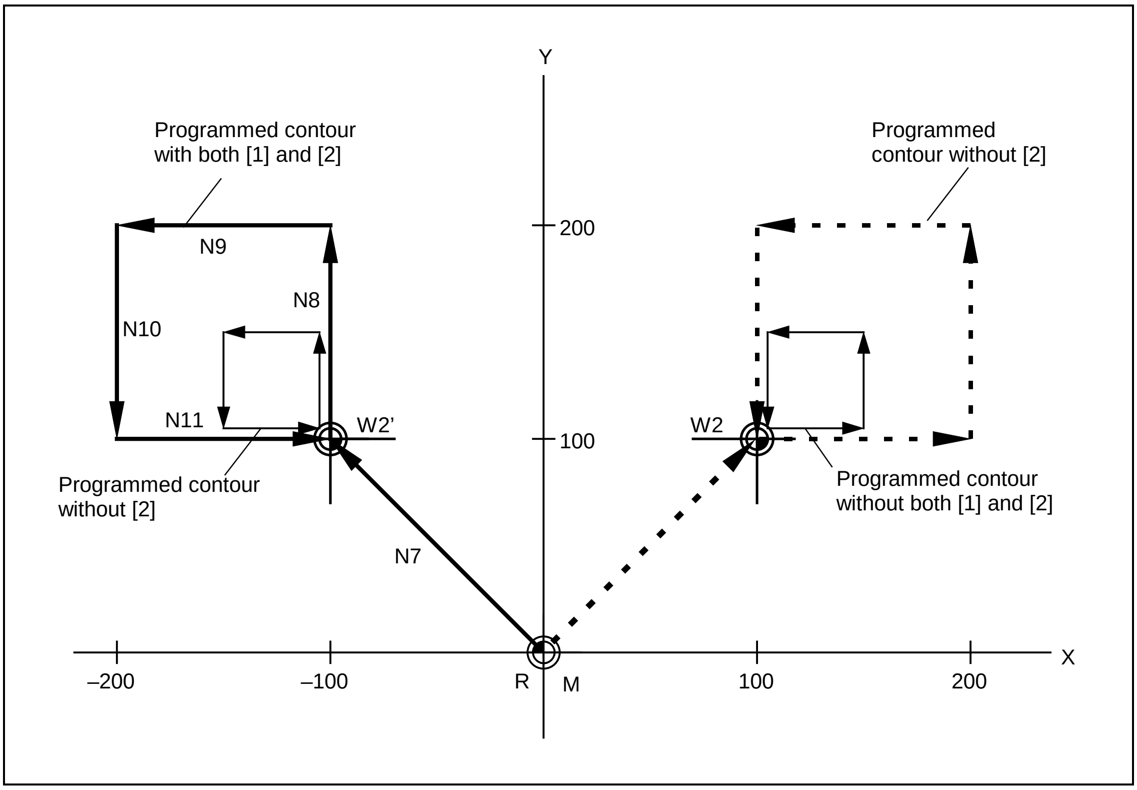


In a combined use with G92.5, the center of figure rotation by M98 will be a position which corresponds with the workpiece coordinate system rotation designated by the G92.5 command.

5. Scaling (G51) in the mode of G92.5

```
N1  G28X0Y0
N2  G17
N3  G55
N4  G90
N5  G92.5X0Y0R90. .................................. [1]
N6  G51X0Y0P2.......................................... [2]
N7  G0X0Y0
N8  G1X50.F500
N9  Y50.
N10 X0
N11 Y0
N12 M30
%
```

G55 (Work Offset)
X100.
Y100.



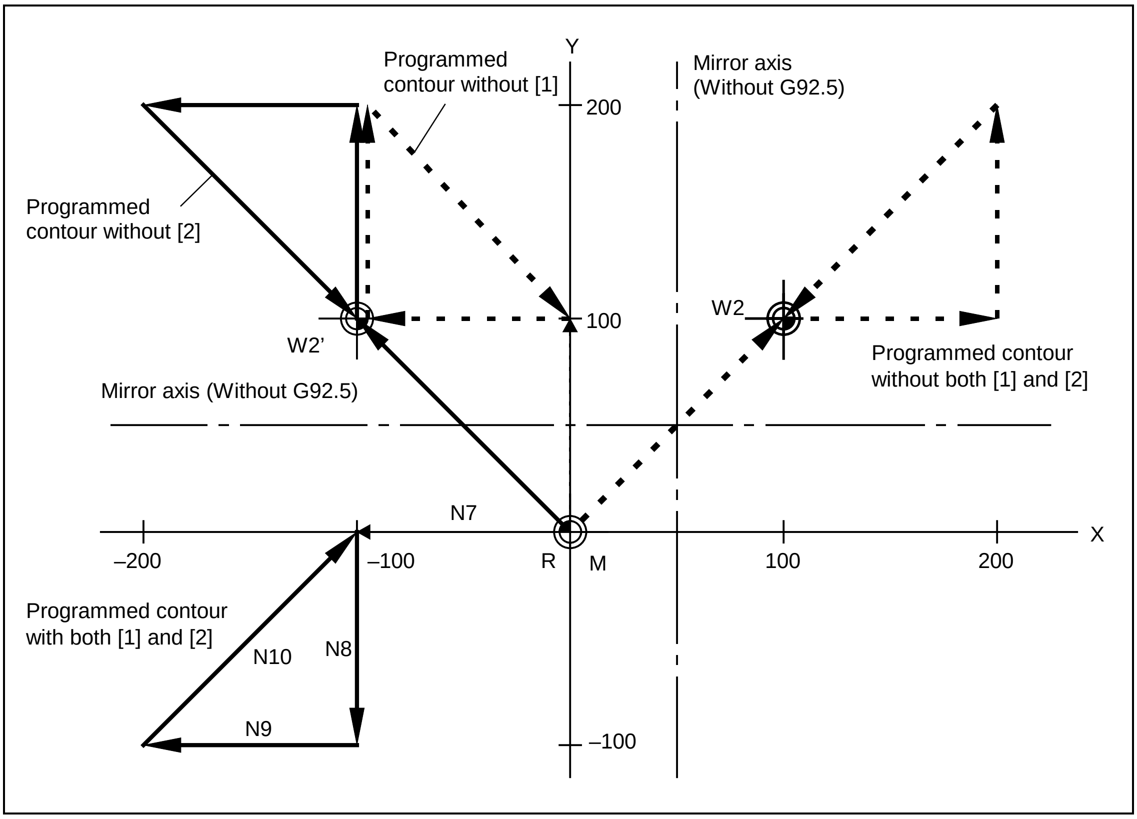
In a combined use with G92.5, the scaling center will be a position which corresponds with the workpiece coordinate system rotation designated by the G92.5 command.

6. Mirror image in the mode of G92.5

a) G-code mirror image

```
N1   G28X0Y0
N2   G17
N3   G55
N4   G90
N5   G92.5X0Y0R90. ..............................  [1]
N6   G51.1X-50. ....................................  [2]
N7   G0X0Y0
N8   G1X100.F500
N9   Y100.
N10  X0Y0
N11  M30
%
```

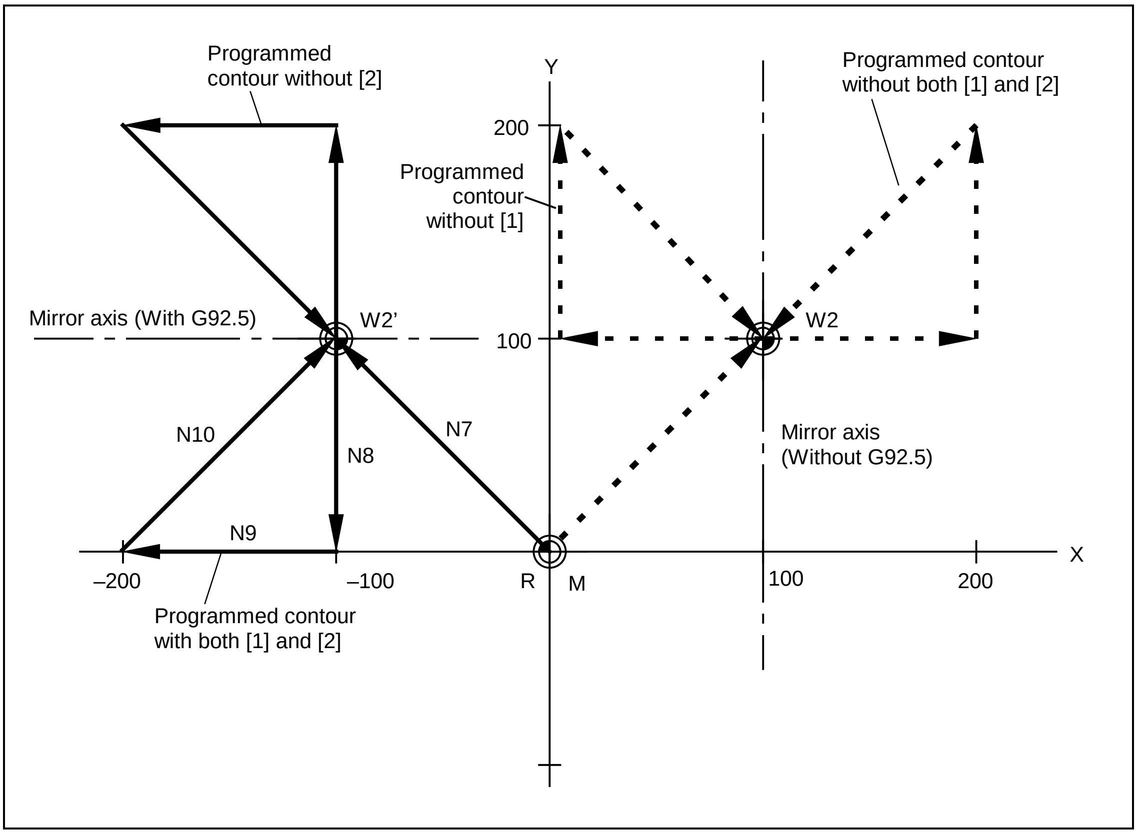
G55 (Work Offset)
X100.
Y100.



b) M-code mirror image

```
N1   G28X0Y0
N2   G17
N3   G55
N4   G90
N5   G92.5X0Y0R90. ................................ [1]
N6   M91. ......................................... [2]
N7   G0X0Y0
N8   G1X100.F500
N9   Y100.
N10  X0Y0
N11  M30
%
```

G55 (Work Offset)
X100.
Y100.

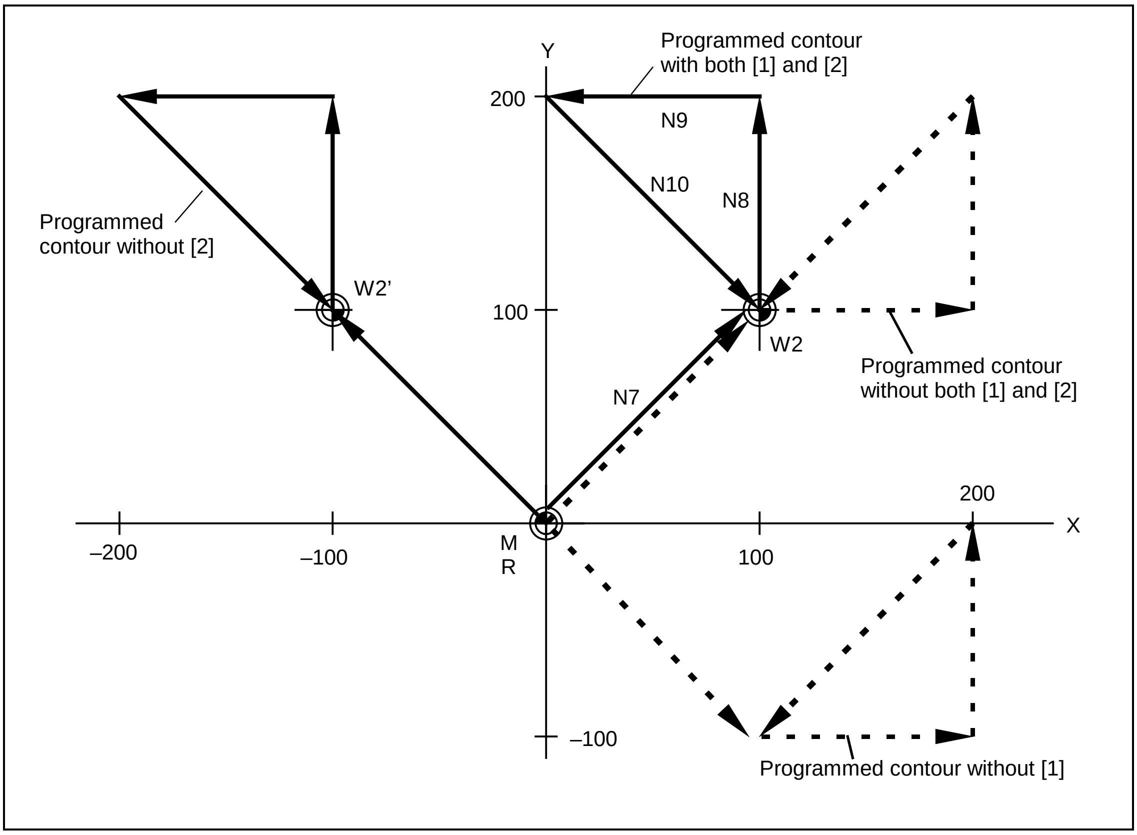


In a combined use with G92.5, the axis of symmetry for G-code or M-code mirror image will be set in accordance with the workpiece coordinate system rotation designated by the G92.5 command.

7. Coordinate system setting (G92) in the mode of G92.5

```
N1   G28X0Y0
N2   G17
N3   G55
N4   G90
N5   G92.5X0Y0R90. ............................ [1]
N6   G92X–100.Y100. .......................... [2]
N7   G0X0Y0
N8   G1X100.F500
N9   Y100.
N10  X0Y0
N11  M30
%
```

G55 (Work Offset)
X100.
Y100.



Coordinate system setting by a G92 block after G92.5 will be performed in reference to the coordinate system rotation designated by the G92.5 command.

5. Precautions

1. If, during rotation of the workpiece coordinate system, a rotational angle of zero degrees is designated (by setting G92.5 R0, for example), the coordinate system rotation will be cancelled, irrespective of the data input mode of G90 (absolute) or G91 (incremental). The next move command will then be executed for the ending point in the original (not rotated) workpiece coordinate system.

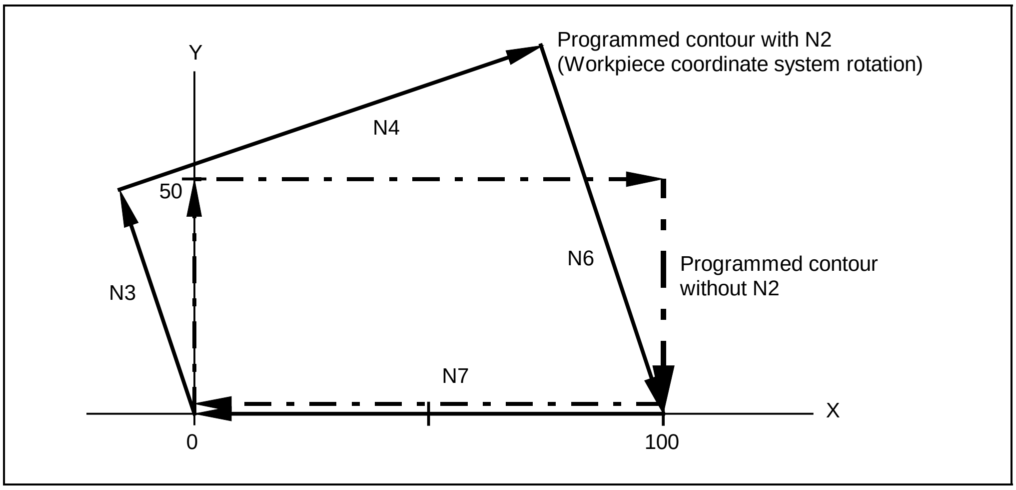
   Example 1: For incremental data input

```
N1  G28X0Y0
N2  G17G92.5X0Y0R20.
N3  G91G01Y50.F1000.
N4  X100.
N5  G92.5R0.......................................  Command for 0-deg rotation
N6  Y–50. ...........................................  Motion to (X100, Y0)
N7  X–100.
N8  M30
%
```

   Example 2: For absolute data input

```
N1  G28X0Y0
N2  G17G92.5X0Y0R20.
N3  G90G01Y50.F1000.
N4  X100.
N5  G92.5R0.......................................  Command for 0-deg rotation
N6  Y0..................................................  Motion to (X100, Y0)
N7  X0
N8  M30
%
```

   Programmed contour for Examples 1 and 2 above



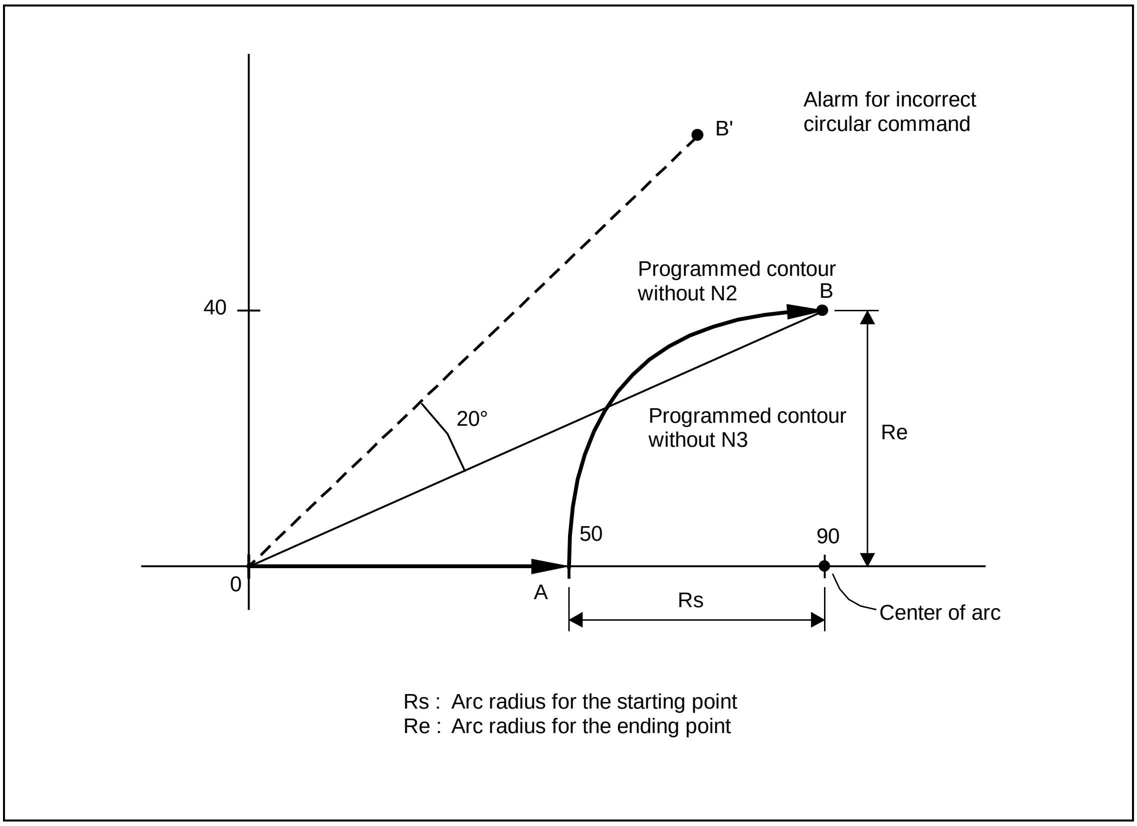
Return to Library

2. Use a linear motion command (with G00 or G01) for the first movement to be executed after G92.5 command.
   Circular interpolation in such a case, as shown below, would have to take place from the current position A, which refers to the original workpiece coordinate system, to the ending point B’ to which the point B should be shifted in accordance with the rotation. As a result, the radii of the starting and ending points would differ too significantly and the alarm No. **817 INCORRECT ARC DATA** would be caused.

   Example:

```
N1   G28X0Y0
N2   G91G01X50.F1000.
N3   G17G92.5X0Y0R20.
N4   G02X40.Y40.I40.
N5   M30
%
```

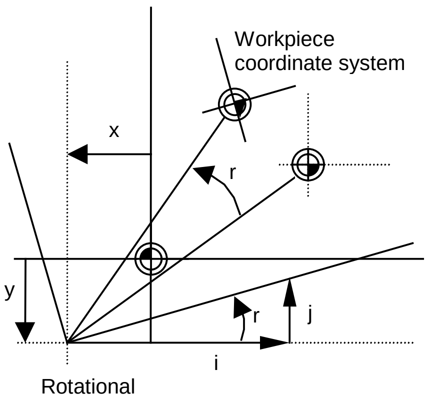
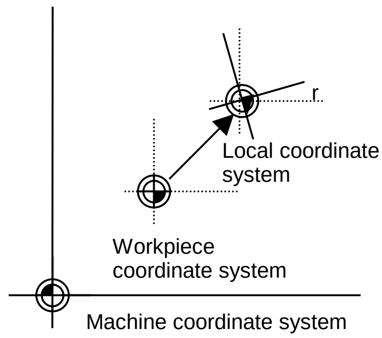
Circular interpolation as the first motion after G92.5



3. Set a G92.5 command in the mode of G40.
4. The machine will operate on the rotated coordinate system for an MDI interruption during the mode of G92.5.
5. For a manual interruption during the mode of G92.5 using the JOG or handle feed mode, the machine will operate independently of the coordinate system rotation.

Return to Library

6. Differences between workpiece coordinate system rotation and programmed coordinate rotation.

<table>
<tr><th colspan="2">Function name</th><th>Workpiece coordinate system rotation</th><th>Programmed coordinate rotation</th></tr>
<tr><td colspan="2">System to be rotated</td><td>Workpiece coordinate system</td><td>Local coordinate system</td></tr>
<tr><td colspan="2">Programming format</td><td>(G17) G92.5 Xx Yy Rr
(G18) G92.5 Yy Zz Rr (Angle)
(G19) G92.5 Zz Xx Rr
or
(G17) G92.5 Xx Yy Ii Jj
(G18) G92.5 Yy Zz Jj Kk (Vector comp.)
(G19) G92.5 Zz Xx Kk Ii</td><td>(G17) G68 Xx Yy Rr
(G18) G68 Yy Zz Rr
(G19) G68 Zz Xx Rr</td></tr>
<tr><td colspan="2">Operation</td><td>
</td><td>
</td></tr>
<tr><td colspan="2">Rotational center coordinates</td><td>Designation at addresses X, Y, Z</td><td>Designation at addresses X, Y, Z</td></tr>
<tr><td colspan="2">Angle of rotation</td><td>Designation at R (angle) or at I, J, K (vector components)</td><td>Designation at R (angle)</td></tr>
<tr><td rowspan="4">Information on center and angle of rotation cleared?</td><td>Power-off → on</td><td>Retained</td><td>Cleared</td></tr>
<tr><td>M02/M30</td><td>Retained</td><td>Cleared</td></tr>
<tr><td>Reset key</td><td>Retained</td><td>Cleared</td></tr>
<tr><td>Resumption of readiness after emergency stop</td><td>Retained</td><td>Cleared</td></tr>
</table>

Note: Resetting or M02/M30 cancels the G92.5 mode itself, while the information on the rotational center, etc., at related addresses is retained as indicated above.

Return to Library

16 MEASUREMENT SUPPORT FUNCTIONS

Measurement by EIA/ISO is basically the same as that by MAZATROL. Information given by MAZATROL may be executed by preparation function below.

G31: Skip function

16-1 Skip Function: G31

16-1-1 Function description

1. Overview

During linear interpolation by G31, when an external skip signal is inputted, the feed will stop, all remaining commands will be cancelled and then the program will skip to the next block.

2. Programming format

G31 Xx/Uu Zz/Ww Yy/Vv Ff ;

x, z, y, u, w, v : The coordinates of respective axes. These coordinates are designated using absolute or incremental data.

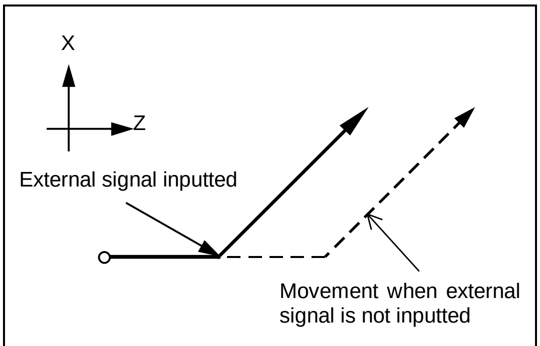
f: Feed rate (mm/min)

3. Detailed description

1. An asynchronous feed rate commanded previously will be used as feed rate. If an asynchronous feed command is not made previously and if Ff is not commanded, the alarm **SKIP SPEED ZERO** will be caused. F-modal command data will not be updated by the F-command given in the G31 block.
2. Automatic acceleration/deceleration is not applied to command block G31.
3. If feed rate is specified per minute, override, dry run and automatic acceleration/de-celeration will not be allowed. They will be effective when feed rate is specified per revolution.
4. Command G31 is unmodal, and thus set it each time.
5. The execution of command G31 will immediately terminate if a skip signal is inputted at the beginning.
   Also, if a skip signal is not inputted until the end of command block G31, execution of this command will terminate on completion of execution of move commands.
6. Setting this command code during tool nose radius compensation results in a program error.
7. Under a machine lock status the skip signals will be valid.

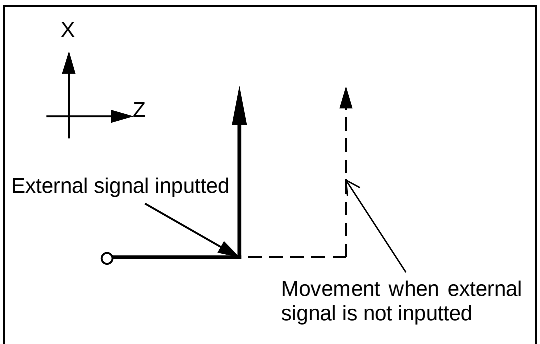
4. Execution of G31

Example 1: When the next block is an incremental value command
G31 Z1000 F100;
G01 U2000 W1000;



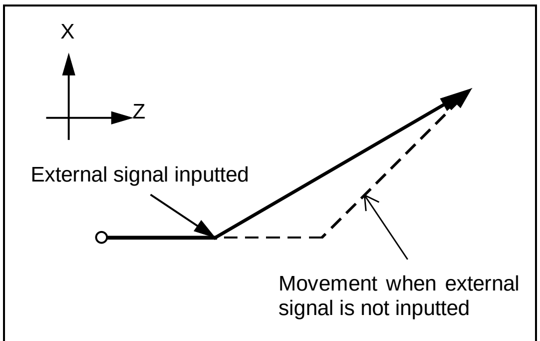
TEP199

Example 2: When the next block is a one axis move command with absolute value
G31 Z1000 F100;
G01 X1000;



TEP200

Example 3: When the next block is a two axes move command with absolute value
G31 Z1000 F100;
G01 X1000 Z2000;



TEP199

Return to Library

16-1-2 Amount of coasting

The amount of coasting of the machine from the time a skip signal is inputted during G31 command to the time the machine stops differs according to the G31-defined feed rate or the F command data contained in G31.

Accurate machine stop with a minimum amount of coasting is possible because of a short time from the beginning of response to a skip signal to the stop with deceleration.

The amount of coasting is calculated as follows:

$$\delta_0 = \frac{F}{60} \times Tp + \frac{F}{60}(t_1 \pm t_2) = \underbrace{\frac{F}{60} \times (Tp + t_1)}_{\delta_1} \pm \underbrace{\frac{F}{60} \times t_2}_{\delta_2}$$

δ_0 : Amount of coasting (mim)

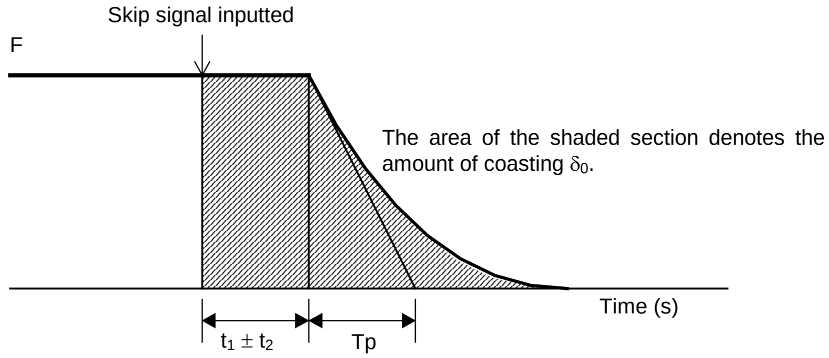
F : G31 skip rate (mm/min)

Tp : Position loop time constant (sec) = (Position loop gain)$^{-1}$

t_1 : Response delay time (sec) = (The time from skip signal detection until arrival at NC through PC)

t_2 : Response error time = 0.001 (sec)

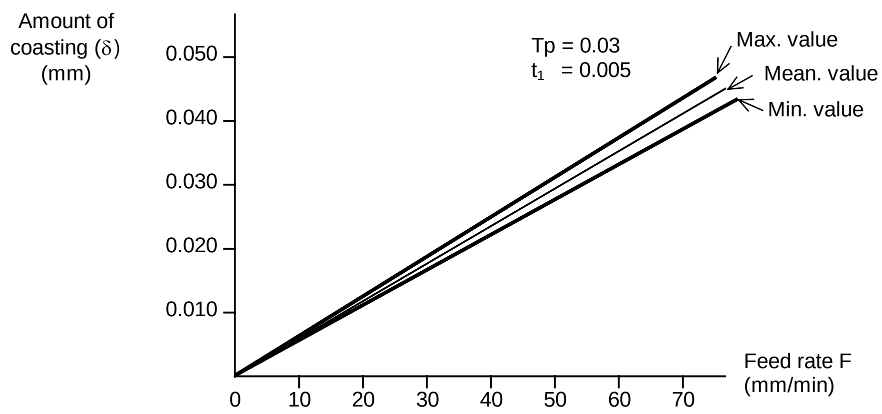
When using command G31 for measurement purposes, measured data δ_1 can be corrected. Such corrections, however, cannot be performed for δ_2.



Stop pattern during skip signal inputted

TEP202

The diagram shown below represents the relationship between the feed rate and the amount of coasting that will be established if Tp is set equal to 30 msec and, t_1 to 5 msec.



Relationship between the amount of coasting and the feed rate (Example)

TEP203

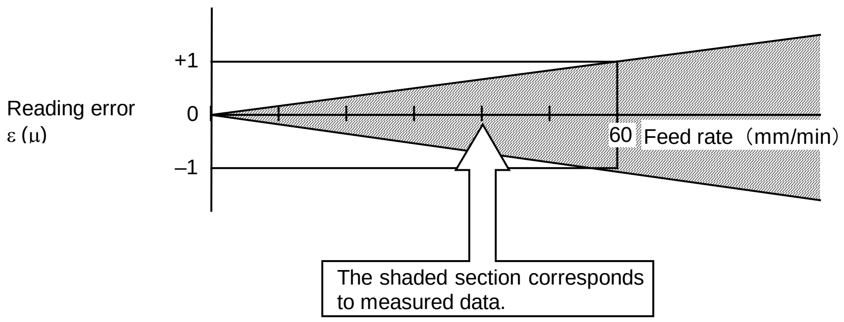
16-1-3 Skip coordinate reading error

1. Reading the skip signal input coordinates

Skip signal input coordinate data does not include the amounts of coasting defined by position loop time constant Tp and cutting feed time constant Ts. Thus skip signal input coordinates can be checked by reading within the error range shown in the diagram below the workpiece coordinates existing when skip signals were intputted. The amount of coasting that is defined by response delay time t_1, however, must be corrected to prevent a measurement error from occurring.

$$\varepsilon = \pm \frac{F}{60} \times t_2$$

ε : Reading error (mm)
F : Feed rate (mm/min)
t_2 : Response delay time 0.001(sec)



Skip signal input coordinate reading error

The reading error at a feed rate of 60 mm/min

$$\varepsilon = \pm \frac{60}{60} \times 0.001$$
$$= \pm 0.001 \text{ (mm)}$$

and measured data stays within the reading error range of ± 1μ.

TEP204

2. Reading coordinates other than those of skip signal inputs

Coordinate data that has been read includes an amount of coasting. If, therefore, you are to check the coordinate data existing when skip signals were inputted, perform corrections as directed above. If, however, the particular amount of coasting defined by response delay time t_2 cannot be calculated, then a measurement error will occur.

Return to Library

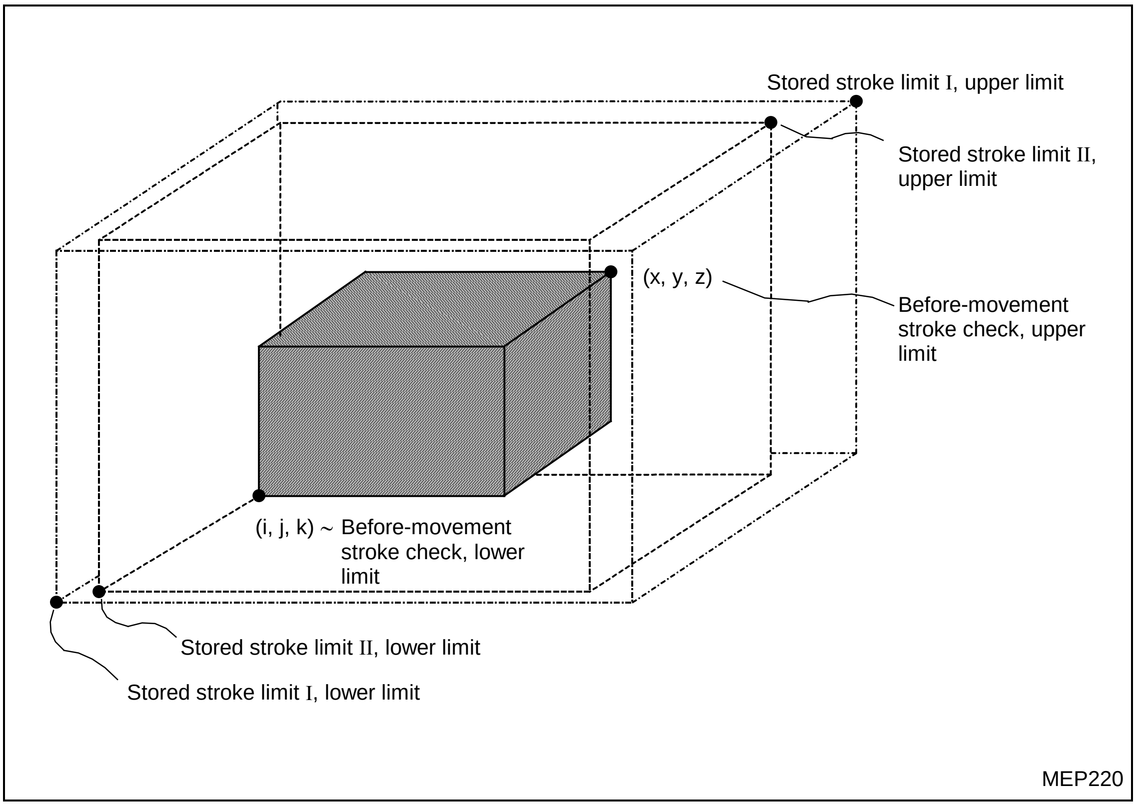
17 PROTECTIVE FUNCTIONS

17-1 Stored Stroke Limit ON/OFF: G22/G23

1. Function and purpose

While stored stroke limit check generates an outside machining prohibit area, before-movement stroke limit check generates an inside machining prohibit area (shaded section in the diagram below).

An alarm will result if you set a move command code that brings an axis into contact with (or moves it through) the shaded section.



2. Programming format

G22 X_ Y_ Z_ I_ J_ K_ (Inside machining prohibitarea specification)

I_ J_ K_: Lower limit specification
X_ Y_ Z_: Upper limit specification

G23 (Cancel)

3. Detailed description

1. Both upper-limit and lower-limit values must be data present on the machine coordinate system.
2. Use X, Y, Z to set the upper limit of the prohibit area, and I, J, K to set the lower limit. If the value of X, Y, Z is smaller than that of I, J, K, then the former will become the lower-limit value and, the latter, the upper-limit value.

3. No stroke limit checks will be performed if the upper- and lower-limit values that have been assigned to the axis are identical.

```
G22X200.Y250.Z100.I200.J-200.K0
```

The X-axis does not undergo the stroke check.

4. The before-movement stroke limit check function will be cancelled if you set G22.

5. If, for example, G23 X_Y_Z_ is set, it will be regarded as G23 X_Y_. After cancellation of before-movement stroke limit check, therefore, X_Y_ will be executed according to the modal move command code last set.

Note: Before setting G22, move the machine to a position outside the prohibit area.

18 TWO-SYSTEM CONTROL FUNCTION

18-1 Two-Process Control by One Program: G109

1. Outline

When machining of different processes are performed by respective systems on a machine with two systems of headstock (HD1 and HD2), or turret (TR1 and TR 2), the two systems can be controlled by a single program.

The program section from “G109L○;” to “%” or to “G109L*;” is used for controlling the ○-system.

2. Programming format

G109 L_;

L = 1 : HD1 (or TR1)
2 : HD2 (or TR2)

The system number is to be specified by a value following the address L.

3. Notes

1. Even if a value following L includes a decimal point or negative sign (–), it is ignored.
2. In the mode of single-block operation, the stop can be performed after execution of G109 block. However, when the number specified by L belongs to the other system such as L2 in HD1 operation, the single-block stop does not occur.
3. G109 can be specified in the same block as G-codes other than of group 0. When specified in the same block as another G-code of group 0, the G-code specified later is effective.
4. The section from the head of a program to the place where G109 is commanded is common to HD1 and HD2, or TR1 and TR2.

Example:

```
G28 U W;     ] Common to HD1 and HD2 (or TR1 and TR2)
G109 L1;     ]
 :           ] HD1 (or TR1)
G109 L2;     ]
 :           ] HD2 (or TR2)
M30;         ]
%            ] Common to HD1 and HD2 (or TR1 and TR2)
```

5. One block including more than 128 characters causes an alarm (**707 ILLEGAL FORMAT**).
6. In the remainder of this chapter, “HD1” and “HD2” generally refer to “TR1” and “TR2”, respectively, at once.
7. The peripheral speed (cutting speed) of the respective turning spindles is to be specified, with reference to the G109 condition, as follows:

G96S__.................to specify the peripheral speed for the 1st spindle under “G109L1”
G96G112S__........to specify the peripheral speed for the 2nd spindle under “G109L1”
G96G112S__........to specify the peripheral speed for the 1st spindle under “G109L2”
G96S__.................to specify the peripheral speed for the 2nd spindle under “G109L2”

18-2 Specifying/Cancelling Cross Machining Control Axis: G110/G111

1. Outline

Axis control of HD2 side by HD1 side or that of HD1 side by HD2 side is referred to as cross machining control. Cross machining control axis is specified by G110 and G111.
Specify after G110 an axis address and the HD number controlling the axis.

2. Programming format

G110 X_ Z_ C_; .... Cross machining control axis and HD number are specified.

1: Axis controlled by HD1
2: Axis controlled by HD2

G111;.. Cross machining control axis specified by G110 is returned to normal control (not cross machining).

Example: Operation at HD1 side

G110 X2; ——— Changed to X-axis of HD2
G00 X10. Z10.; ——— X of HD2 moves to 10, Z of HD1 moves to 10.
G110 Z2; ——— Changed to Z-axis of HD2
G00 X20. Z20.; ——— X of HD2 moves to 20, Z of HD2 moves to 20.
G110 X1 Z1; ——— Changed to X-axis and Z-axis of HD1
G00 X30. Z30.; ——— X of HD1 moves to 30, Z of HD1 moves to 30.
G110 Z3; ——— Changed to B-axis of HD2
G00 Z40.; ——— B of HD2 moves to 40.

Use the incremental data input method for the W-axis on the 2. headstock side (INTEGREX-IV) as follows:

Example:

G110 Z[B]2; ——— Selection of the 2nd headstock's W-axis
G00 Z–100.; ——— The command with address Z is given for the W-axis movement to an absolute position of –100 on the 2nd headstock side.
G00 W–10.; ——— The command with address W is given for an incremental W-axis movement by –10 on the 2nd headstock side.
⋮
G111; ——— Cancellation of G110

Specify the Z-axis for the lower turret as follows:

Example:

G110 Z2; ——— Selection of the lower turret's Z-axis
G00 Z100.; ——— All the Z-axial commands between G110 and G111 are processed as those for the lower turret.
⋮
G111; ——— Cancellation of G110

Specify the C-axis on the 2nd headstock side as follows:

Example:

G110 C2; ——— Selection of the 2nd headstock's C-axis
G00 C45.123.; ——— All the C-axial commands between G110 and G111 are processed as those for the 2nd headstock side.
⋮
G111; ——— Cancellation of G110

Return to Library

Prepare a program as follows to use the C-axis settings on the **WORK OFFSET** display for the 2nd spindle:

Example:

```
G52.5;          ———— MAZATROL coordinate system cancellation
M200;           ———— Milling mode selection for 1st spindle
G28UWH;
T001T000M6;
G54;            ———— Origin data of the G54 system: C = 30°
G00 C150.;      ———— HD1 C-axis motion to 150° (POSITION) or 180° (MACHINE)
M202;           ———— Milling mode cancellation for 1st spindle
M902;           ———— 2nd spindle selection
M300;           ———— Milling mode selection for 2nd spindle
G110 C2;        ———— Selection of 2nd spindle C-axis
G00 C150.;      ———— HD2 C-axis motion to 150° (POSITION) or 180° (MACHINE)
G55;            ———— Origin data of the G55 system: C = 50°
G00 C150.;      ———— HD2 C-axis motion to 150° (POSITION) or 200° (MACHINE)
G56;            ———— Origin data of the G56 system: C = 100°
G00 C150.;      ———— HD2 C-axis motion to 150° (POSITION) or 250° (MACHINE)
G111;           ———— Cancellation of G110
M302;           ———— Milling mode cancellation for 2nd spindle
```

Prepare a program as follows to use a fixed cycle for hole machining on the 2nd spindle side:

Example:

```
M902;                   ———— 2nd spindle selection
M300;                   ———— Milling mode selection for 2nd spindle
G110 C2;                ———— Selection of 2nd spindle C-axis
G00 C0.;                ———— HD2 C-axis positioning
G87Z–5.0X5.0P0.2M310;   ———— Clamping; Deep-hole drilling cycle
C45.;                   ———— Unclamping, positioning, clamping; Deep-hole drilling cycle
C90.;                   ———— Unclamping, positioning, clamping; Deep-hole drilling cycle
M312;                   ———— Unclamping on the 2nd spindle side
G80;                    ———— Fixed cycle cancellation
G111;                   ———— Cancellation of G110
M30;                    ———— Program end
```

3. Sample programs

Examples of programming for the machine specifications with the secondary spindle

The major sections of a sample program for machines equipped with the secondary spindle are shown below.

```
O1234
G53.5                       MAZATROL coordinate system establishment
#101=124.750 (SP1 COF)      1st spindle side C-axis offset
#102=10.664 (SP2 COF)       2nd spindle side C-axis offset
```

(MAIN SPINDLE SIDE)	1st spindle side machining program
M901	1st spindle select mode (enter for machining at the 1st spindle side)
G50S3000	Spindle clamping speed setting
M202	1st spindle turning mode
G110Z2	2nd spindle side Z-axis selection
G00Z0.	2nd spindle side Z-axis positioning
G111	2nd spindle side Z-axis selection revoking
G00G28U0V0W0	1st spindle return to zero point (X, Y, Z)
T001T000M6	Tool selection
N101(EDG-R)	Edge machining with 1st spindle
G96S200	Peripheral speed setting
G00X110.0Z0.1	Positioning
G99G01X22.0F0.3	Cutting feed
G00Z0.8	Positioning
N102(OUT-R)	O.D. machining with 1st spindle
	(Machining program omitted for convenience's sake.)
(TRS CHK)	Transfer program
G28U0V0W0	1st spindle return to zero point (X, Y, Z)
M902	2nd spindle selection
M302	2nd spindle turning mode
M200 (MAIN C-ON)	1st spindle mill-point machining mode
G00C#101	1st spindle C-axis positioning (angle indexing)
M300(SUB C-ON)	2nd spindle mill-point machining mode
G110C2	2nd spindle C-axis selection
G00C#102	2nd spindle C-axis positioning (angle indexing)
G111	2nd spindle C-axis selection revoking (G110 cancellation)
M306	2nd spindle chuck open
M540	TRS-CHK mode
G110Z2	2nd spindle side Z-axis selection
G00Z-686.	2nd spindle side Z-axis positioning
M508	Start of pressing action on the 2nd spindle side
G31W-1.1F50	2nd spindle side Z-axis positioning for pressing
M202	1st spindle turning mode
M509	2nd spindle M508 cancellation
G111	2nd spindle side Z-axis selection revoking
M541	TRS-CHK mode cancellation
M307	2nd spindle chuck close
M206	1st spindle chuck open
M302	2nd spindle turning mode
G110Z2	2nd spindle side Z-axis selection
G00Z-80.	2nd spindle side Z-axis positioning
G111	2nd spindle side Z-axis selection revoking
(SUB SPINDLE SIDE)	2nd spindle machining program
N301(SP2 DRL)	
M902	2nd spindle selection (enter for machining at the 2nd spindle side)
T003T000M6	Tool selection
G98G97	Feed per minute and cancellation of constant peripheral speed control
M300	2nd spindle mill-point machining mode
M203S3184	Milling speed selection and milling spindle normal rotation
G110C2	2nd spindle C-axis selection
G0C#102	2nd spindle C-axis positioning (angle indexing)
M310	2nd spindle C-axis clamping
G00X25.Z-5.	Positioning
G87Z-5.X5.Q5000P0.2F200	Longitudinal deep-hole drilling cycle
M312	2nd spindle C-axis unclamping
G80	Cancellation of fixed hole-drilling cycle
G00C[#102+180.]	2nd spindle C-axis positioning (angle indexing)
M310	2nd spindle C-axis clamping
G87Z-5.X5.Q5000P0.2F200	Longitudinal deep-hole drilling cycle
M312	2nd spindle C-axis unclamping
G80	Cancellation of fixed hole-drilling cycle
G111	2nd spindle C-axis selection revoking
G28U0V0W0	Return to zero point (X, Y, Z)
M30	End of program

4. Notes

1. After the axis is changed by G110 or G111, always specify the coordinate system by G50.
2. G110 and G111 must always be given in a single-command block.
3. When axis address is commanded by G110 in increment, (for example, U and W are used) it causes an alarm. And when a value following the axis address includes a decimal point or negative sign, it is ignored.
4. In the single-block operation mode, the stop is performed after execution of G110 and G111 blocks.
5. The tool information to be used in tool offsetting does not automatically change for the other system on the occasion of designating for cross machining control an axis which is in direct relation to tool movement. Use, therefore, a G53 command (for positioning in the machine coordinate system) as required.
6. As long as an axis in direct relation to tool movement is controlled for cross machining, do not change tools (by M6).
7. When the axis is changed by G110, the counterpart system must be in a state of automatic starting and standby.

   <u>State of standby</u>
   M-codes from M950 to M997 are used for waiting. When both HD1 and HD2 are operated and when machining is performed with HD1 and HD2 synchronized, M950 to M997 is used. A state of standby refers to the time before the same waiting M-code is outputted from the counterpart.
   For example, when M950 is outputted from HD1, HD1 is in a state of standby until M950 is outputted from HD2. (HD1 does not execute blocks subsequent to M950.) When M950 is outputted from HD2, HD1 executes the block following M950.

   Program example

```
HD1                          HD2
M950;                        M950;
G110 X2;  ┐                  M951;
X....     │
X.....Z...├ *  Indicates the waiting time for which HD2 is in a state of standby when
   ⋮      │    X-axis of HD2 is controlled by HD1.
M951;     ┘
```

8. Give a command of G111 as required at the end of machining section in an EIA/ISO program which is to be called from a MAZATROL program as a subprogram for point machining.
9. The axis being under cross machining control in automatic mode of operation cannot be controlled in manual mode. An attempt to do so will only result in the alarm **ILLEGAL COMMAND CROSS MACHINING**.
10. Barrier is effective also during axis change. In other words, barrier is checked in the region of HD1 side for the axis of HD1 side and in that of HD2 for the axis of HD2 independently of the axis change by G110.
11. Synchronous feed with, or control of feed per, revolution of the milling spindle is not available during cross machining control.
12. The alarm **CROSS MACHINING IMPOSSIBLE** will be caused when a command for cross machining control is given under one of the following incompatible modal conditions:
    - Nose R/Tool radius compensation
    - Polar coordinate interpolation
    - Cylindrical interpolation

- Fixed cycle
- 3-D coordinate conversion
- Mirror image
- Tool tip point control

13. C-axis commands in the cross machining mode can only be given for the preparatory functions (G-codes) enumerated below.

Usable G-codes for C-axis commands in the cross machining mode

G-code series T	Group	Function
G00	01	Rapid positioning
G01	01	Linear interpolation
G02	01	Circular interpolation CW
G03	01	Circular interpolation CCW
G10	00	Data setting/change
G27	00	Reference point return check
G28	00	Reference point return
G29	00	Return from reference point
G30	00	Return to 2nd/3rd/4th reference point
G30.1	00	Return to floating reference point
G36 (G36.5)	00	Measurement target data setting
G50	00	Coordinate system setting/Spindle limit speed setting
G53	00	MAZATROL coordinate system selection
G65	00	Macro call
G66	14	Macro modal call
G83	09	Face drilling cycle
G84	09	Face tapping cycle
G84.2	09	Face synchronous tapping cycle
G85	09	Face boring cycle
G87	09	Outside drilling cycle
G88	09	Outside tapping cycle
G88.2	09	Outside synchronous tapping cycle
G89	09	Outside boring cycle
G110	00	Cross machining control axis selection
G111	00	Cross machining control axis cancellation
G112	00	M-, S-, T-, and B-code output to counterpart

14. Even in the mode of “G110B2”, commands with address Z can only cause a linear motion of the W-axis when they are given under G0 or G1. Z-values given in the G2 or G3 mode will always be processed for an circular interpolation with the control of HD1’s Z-axis.
15. When the axis (normally the X-axis) relevant to the constant peripheral speed control is designated for cross machining control, the speed of the turning spindle may change steeply in accordance with the change in positional information to be used in the calculation of spindle speed for a particular peripheral speed.
16. The inclined Y-axis cannot be controlled for cross machining.

18-3 M, S, T, B Output Function to Counterpart: G112

1. Outline

The function outputs M-, S-, T- and B-codes (second miscellaneous function) commanded after G112 to the counterpart system.

2. Programming format

G112 L_ M_ M_ M_ M_ S_ T_ T_ B_;

Example 1: With an argument L specified.
<Given in a section for System 2>
`G109L2;`
`:`
`G112 L1M203S1000;`........Normal rotation of the upper turret's milling spindle.

Example 2: With an argument L omitted.
<Given in a section for System 1, with **BA71** = 1 for that system>
`G109L1;`
`:`
`G112 M203S1000;`............Normal rotation of the lower turret's milling spindle.

3. Notes

1. Do not give any other G-code in one block with a G112 command; otherwise the alarm **ILLEGAL FORMAT** wil be caused.
2. Do not enter any codes concerned (M, S, T or second miscellaneous function) before G112L_ within a block; otherwise the alarm **ILLEGAL FORMAT** wil be caused.
3. Entering values with any other address than N, M, S, T, and that for second miscellaneous function in one block with a G112 command will lead to the alarm **ILLEGAL ADDRESS**.
4. The alarm **ILLEGAL NUMBER INPUT** will be caused if any of the following commands is given in one block with a G112 command:

   M0, M1, M2, M30, M99, M-codes for waiting, and M-, S-, T- or second miscellaneous function code for macroprogram call.
5. Entering a number for the self-system or non-existent system with address L as well as in parameter **BA71** will lead to the alarm **ILLEGAL NUMBER INPUT**.
6. An attempt to specify an offset number in the T-code format for turning machines will lead to the alarm **ILLEGAL NUMBER INPUT**.
7. The T-code in a G112 block will only cause the corresponding code for tool designation to be outputted (without information of tool offsetting).
8. The number of the codes concerned to be entered in a G112 block is limited as follows:

   4 for M, 1 for S, 2 for T, and 1 for the second miscellaneous function.

   Entering codes in excess will only result in the last ones within the limit being outputted.
9. The single-block stop can occur after the execution of a G112 block.
10. Use waiting M-codes so as to output the codes concerned (M, S, T or second miscellaneous function) to one and the same system at one time from multiple systems.

- NOTE -

19 COMPOUND MACHINING FUNCTIONS

This chapter describes the functions proper to the machines equipped with two turrets (upper and lower) which can be operated independently from each other.

19-1 Programming for Compound Machining

1. Outline

The movement of the upper and lower turrets is to be controlled in a single program as follows:

G109 L1;.................................. Selection of the upper turret

Commands for the upper turret

M30;

G109 L2;.................................. Selection of the lower turret

Commands for the lower turret

M30;

2. Remarks

1. If an argument L includes a decimal point or negative sign (–), a programming error will result.
2. In the mode of single-block operation, the stop can be performed after execution of G109 block. However, when the number specified by L belongs to another system, the single-block stop does not occur.
3. G109 must be given in an independent block. If any other command is given in the same block, a programming error will result.
4. Note that the program section under no specification by the G109 command is used for all the systems without distinction.
5. The restart position for the **[RESTART 2 NONMODAL]** menu function must be set within a program section which is prepared commonly for all the systems.
6. The control for a constant peripheral speed (by G96) is always conducted with reference to that tool tip's position of either turret which is nearer to the axis of turning.
7. The call command for a MAZATROL program must be given in program sections of both turrets for one and the same program. If it is given for either turret only, the flow of the called MAZATROL program will enter in a waiting state which cannot be cleared and, as a result, stop the machine operation.

19-2 Waiting Command: M950 to M997, P1 to P99999999

1. Outline

Waiting commands are used to time the operation of the upper and lower turrets as required. Two types of waiting command are provided: M-code and P-code, which can be used freely and even mixedly.

2. Detailed description

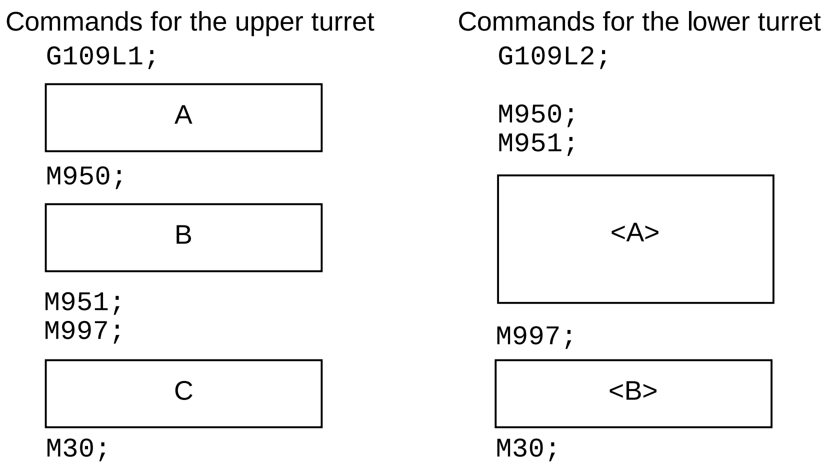
A. M-codes for waiting

The execution of the commands for turret A will be stopped at the position of a waiting M-code with some number until the program flow for turret B reaches a waiting M-code with the same number.

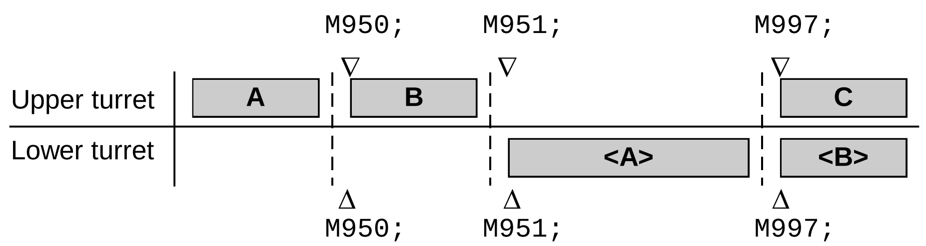
Programming format

M∗∗∗; (∗∗∗ denotes a number from 950 to 997.)

Program structure



Operation



Note: A waiting M-code must be given in a single-command block. It may not function as waiting command if it is entered in the same block together with other instructions.

B. P-codes for waiting

The execution of the commands for turret A will be stopped at the position of a waiting P-code with some number until the program flow for turret B reaches a waiting P-code with the same or a larger number.

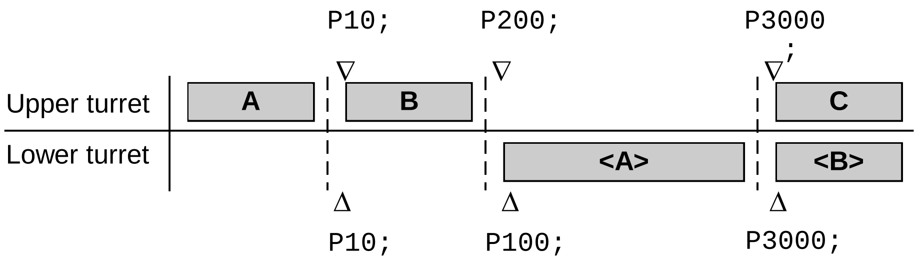
<u>Programming format</u>

P∗∗∗∗∗∗∗∗; (∗∗∗∗∗∗∗∗ denotes a number from 1 to 99999999.)

<u>Program structure</u>

Commands for the upper turret	Commands for the lower turret
`G109L1;`	`G109L2;`
A	`P10;`
`P10;`	`P100;`
B	<A>
`P200;`	
`P3000;`	`P3000;`
C	<B>
`M30;`	`M30;`

<u>Operation</u>



Note 1: A waiting P-code must be given in a single-command block. It may not function as waiting command if it is entered in the same block together with other instructions.

Note 2: Use the waiting P-codes in the ascending order of their number, since one turret cannot be released from the wait state until the program flow for the other turret reaches a waiting P-code with the same or a larger number.

19-3 Balanced Cutting

1. Outline

Balanced cutting is achieved through the symmetrical movement of the upper and lower turrets. It helps the reduction in the vibration of a long workpiece and permits the cutting speed to be doubled for the saving of the machining time.
During the balanced cutting one turret acts as the main turret (master turret) and the other as the subordinate turret (servant turret). Enter the movement commands for the balanced cutting in a program section for the main turret.

2. Programming method

The balanced cutting can be achieved by combining the following three commands:

- Waiting command (M950 to M997 or P1 to P99999999)
- M562;..........Coupling command for the two turrets
- M563;..........Coupling cancellation command

The main points of programming the balanced cutting are the following:

1) Enter the waiting command just before the balanced cutting in order to synchronize the movement of both turrets.

2) Enter the command M562 for the main turret in order to couple both turrets. The subordinate turret must have been set in wait state.

3) Enter the movement commands for the main turret. The subordinate turret will be moved symmetrically during the balanced cutting.

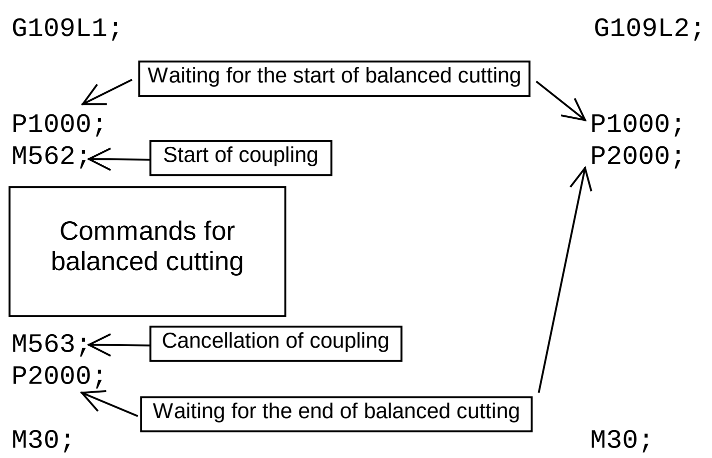
4) Enter the command M563 after the movement commands for the master turret to cancel the coupling.

5) Enter the waiting command for the main turret to release the subordinate turret from the wait state.

3. Program structure

Given below is an example of program structure with the upper turret as the master.

Commands for the upper turret Commands for the lower turret



4. Sample program

```
N000 G109 L1;
M901;
N001 G00 X800.Z70.;
P10;
M03 S250;
T001T000M06D001;
N002 X132.Z60.M08;
M950;
M562;
N003 G01 X78.F0.35;
N004 G00 X156.Z63.;
N005 Z29.;
N006 G01 X150.;
N007 X148.Z30.;
N008 X128.;
N009 G00 X800.Z70.;
N010 X112.Z63.T0202;
N011 G01 X120.Z59.F0.4;
N012 Z30.;
N013 X130.;
N014 G00 X800.Z70.;
M563;
P20;
N015 M09 M05;
N100 G109 L2;
M901;
N101 G00 X800.Z200.;
P10;
M03 S250;
T001001;
N102 X92.Z65.M08;
M950;
P20;
```

G109 L1: Upper turret selection

P10: Single command for waiting

M950 for waiting for the start of cutting

Movement commands for balanced cutting

G109 L2: Lower turret selection

P20: Single command for waiting. Required for both turrets

19-4 Milling with the Lower Turret

1. Programming format

The basic format of programming for milling with the lower turret is an application of the preparatory function G109 (Two processes in one program; see Section 19-1).

G109 L_;
 L = 1: HD1 (TR1)
  2: HD2 (TR2)

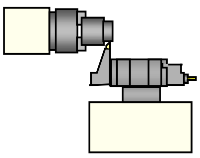
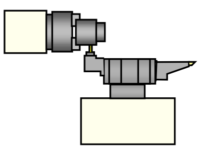
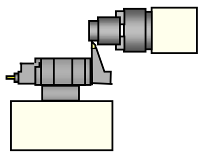
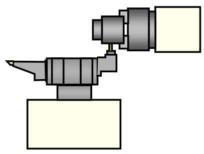
Example:

```
G28 U W;  .......  Common to both spindles (both turrets)
G109 L1;
:          .......  Commands for Upper turret
:
G109 L2;  .......  Selection of 2nd spindle (Lower turret)
M200;
M203;
:                   Commands for Lower turret
:                   Fixed cycle for hole machining
M210;
M30
%  ..............  Common to both spindles (both turrets)
```

<Usable machining patterns>

As shown in the table below, not only for turning can the lower turret be used, but also for milling.

Table 19-1 Machining patterns

	1st spindle		2nd spindle	
Lower turret	Turning 	Milling 	Turning 	Milling 

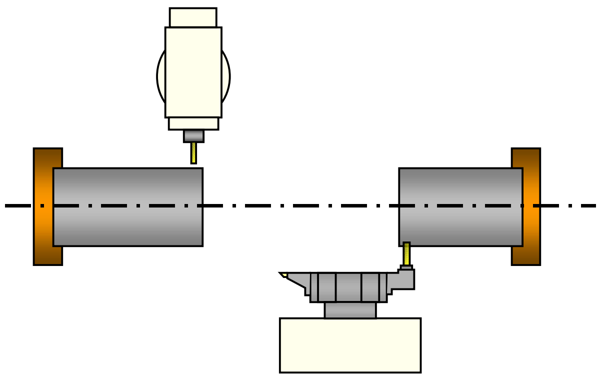
2. G-codes for milling

The G-codes of fixed cycle for hole machining are available for milling with the lower turret.

(See Section 14-3 for more information on the above G-codes.)

3. Sample program

`N000 G00 G97 G98;`	
`N001 G28 U W;`	
`N100 G109 L1;`	Upper turret selection
`N101 T001T000M6D001;`	
`N102 M901;`	1st spindle selection
`N103 M200;`	Point milling mode
`N104 M203 S800;`	Normal rotation of the milling spindle
`N105 X102.Z-50.C0.;`	
`N106 G87 Z-50.H30.X70.R5.Q5000 P.2 F200 M210;`	Fixed cycle for hole machin-ing with Upper turret
`N107 G80;`	
`N108 M950;`	M950 for waiting
`N109 M30;`	
`N200 G109 L2;`	Lower turret selection
`N201 T102022;`	
`N202 M902;`	2nd spindle selection
`N203 M300;`	
`N204 M203 S800;`	
`N205 X-102.Z-30.C180.;`	
`N206 G87 Z-30.H30.X70.R5.Q5000 P.2 F200 M210;`	Fixed cycle for hole machin-ing with Lower turret
`N207 G80;`	
`N208 M950;`	M950 for waiting
`N209 M30;`	

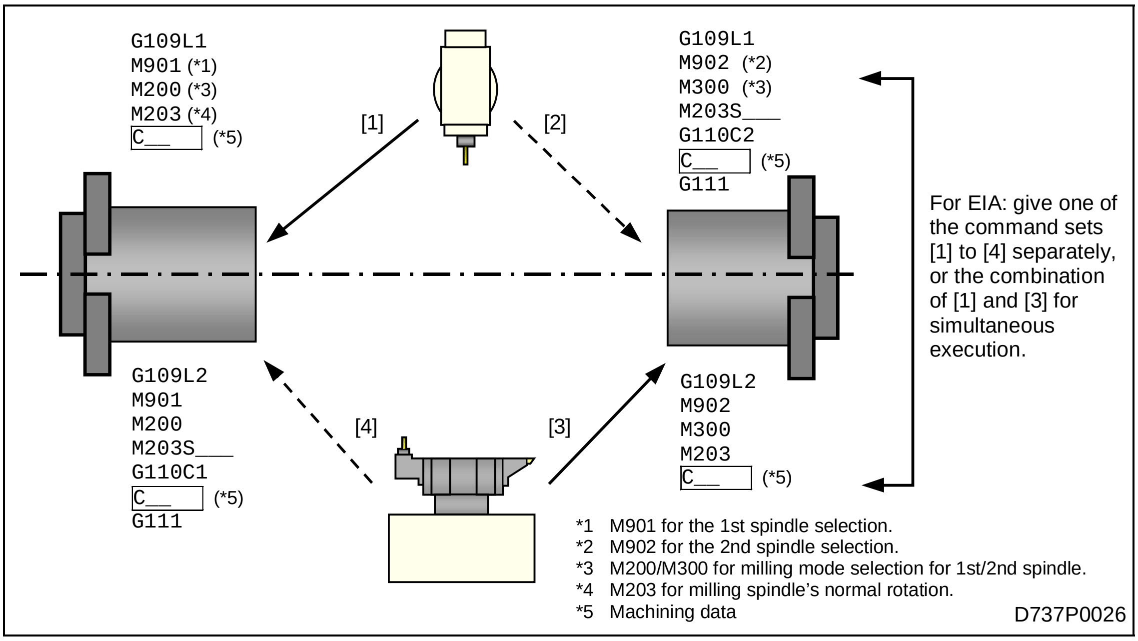


D737P0025

Return to Library

19-5 Compound Machining Patterns

1. Overview



2. Machining pattern list

<Single workpiece>

- Separate machining (with either turret)

○: Possible

EIA	1st spindle		2nd spindle	
	Turning	Milling	Turning	Milling
Upper turret	○	○	○	○
Lower turret	○	○	○	○

- Parallel machining (on either spindle side with both turrets)

○: Possible —: Inapplicable

EIA			Upper turret			
			1st spindle		2nd spindle	
			Turning	Milling	Turning	Milling
Lower turret	1st spindle	Turning	○	—	—	—
		Milling	—	○ (**Note**)	—	—
	2nd spindle	Turning	—	—	○	—
		Milling	—	—	—	○ (**Note**)

Note: Simultaneous milling is possible in the EIA programming format, indeed, but take care of a phase difference occurring for machine structural reasons.

Return to Library

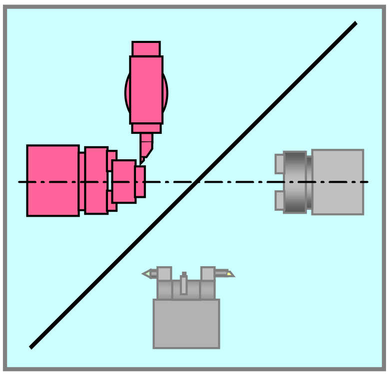
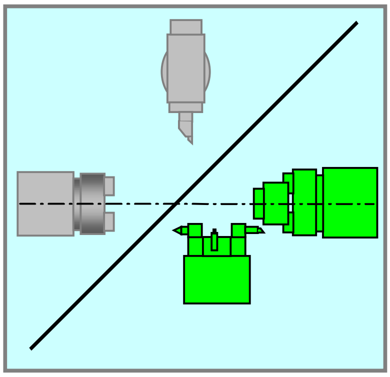
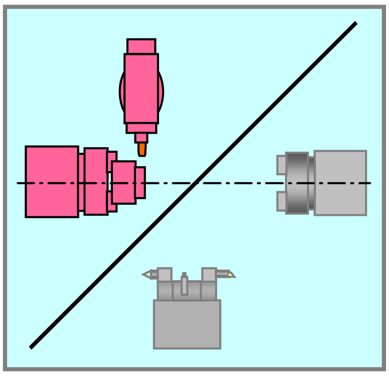
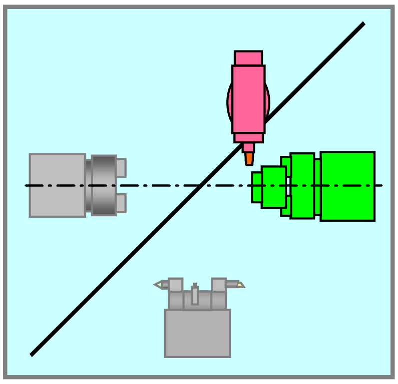
<Dual workpiece>

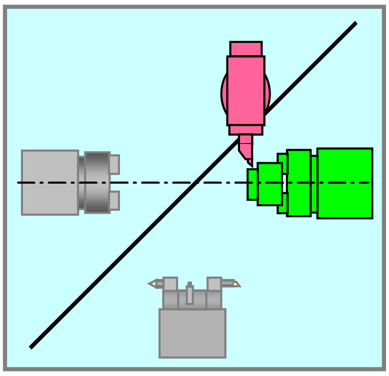
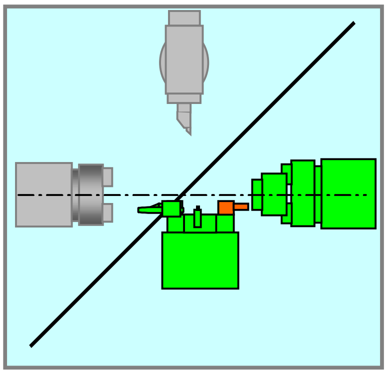
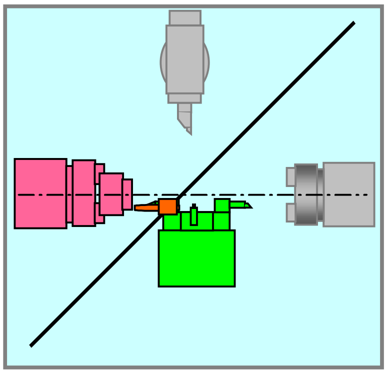
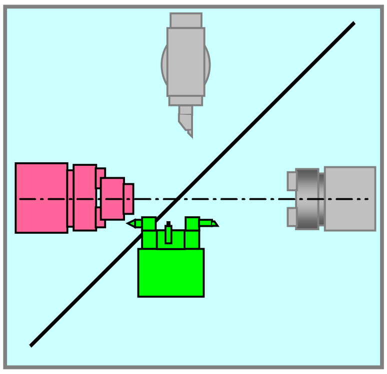
Parallel machining (on both spindle sides with each turret)

○: Possible —: Inapplicable

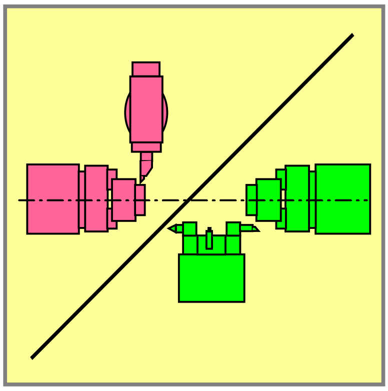
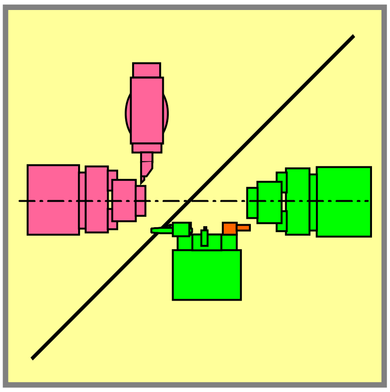
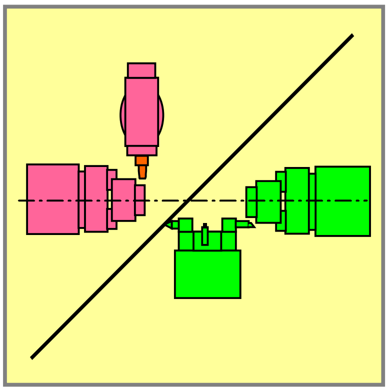
EIA			Upper turret			
			1st spindle		2nd spindle	
			Turning	Milling	Turning	Milling
Lower turret	1st spindle	Turning	—	—	○	○
		Milling	—	—	○	○
	2nd spindle	Turning	○	○	—	—
		Milling	○	○	—	—

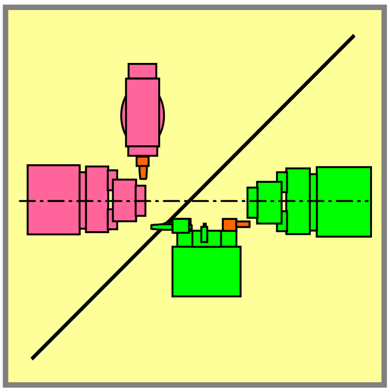
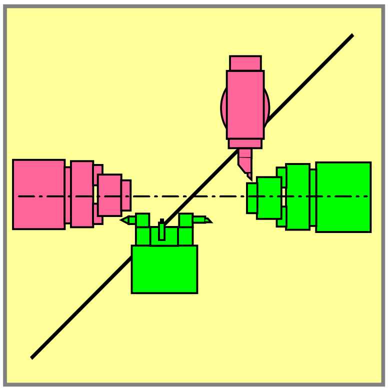
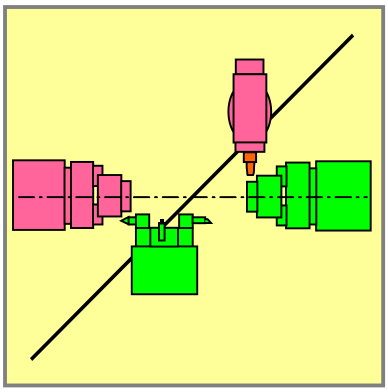
Return to Library

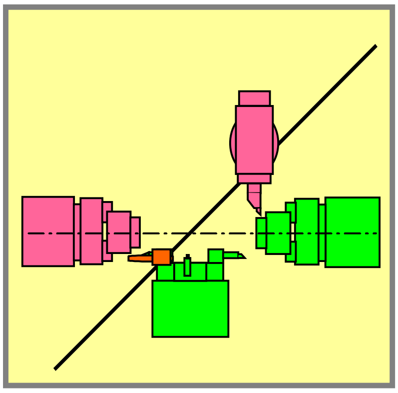
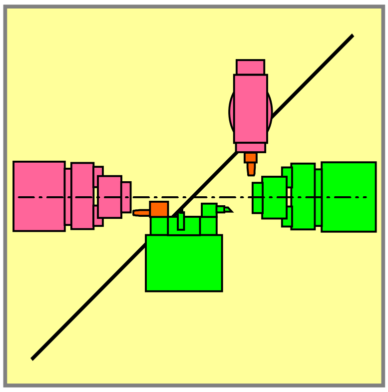
No.	Machining pattern	Programming example
1	Upper turret — 1st spindle; Turning, Separate 	G109L1 M901 M202 M3 S○○○ : Machining data M5 M950 M30 G109L2 M950 M30
2	Lower turret — 2nd spindle; Turning, Separate 	G109L1 M950 M30 G109L2 M902 M302 M303 S○○○ : Machining data M305 M950 M30
3	Upper turret — 1st spindle; Milling, Separate 	G109L1 M901 M200 M203 S○○○ : Machining data M205 M202 M950 M30 G109L2 M950 M30
4	Upper turret — 2nd spindle; Milling, Separate 	G109L1 M902 M300 M203 S○○○ : Machining data M205 M302 M950 M30 G109L2 M950 M30

No.	Machining pattern	Programming example
5	Upper turret — 2nd spindle; Turning, Separate 	G109L1 M902 M302 M303 S○○○ : Machining data M305 M950 M30 G109L2 M950 M30
6	Lower turret — 2nd spindle; Milling, Separate 	G109L1 M950 M30 G109L2 M902 M300 M203 S○○○ : Machining data M205 M302 M950 M30
7	Lower turret — 1st spindle; Milling, Separate 	G109L1 M950 M30 G109L2 M901 M200 M203 S○○○ : Machining data M205 M202 M950 M30
8	Lower turret — 1st spindle; Turning, Separate 	G109L1 M950 M30 G109L2 M901 M202 M3 S○○○ : Machining data M5 M950 M30

Return to Library

No.	Machining pattern	Programming example	
9	Upper turret — 1st spindle; Turning, Lower turret — 2nd spindle; Turning. 	G109L1 M901 M202 M3 S○○○ : Machining data M5 M950 M30	G109L2 M902 M302 M303 S○○○ : Machining data M305 M950 M30
10	Upper turret — 1st spindle; Turning, Lower turret — 2nd spindle; Milling. 	G109L1 M901 M202 M3 S○○○ : Machining data M5 M950 M30	G109L2 M902 M300 M203 S○○○ : Machining data M205 M302 M950 M30
11	Upper turret — 1st spindle; Milling, Lower turret — 2nd spindle; Turning. 	G109L1 M901 M200 M203 S○○○ : Machining data M205 M202 M950 M30	G109L2 M902 M302 M303 S○○○ : Machining data M305 M950 M30

No.	Machining pattern	Programming example	
12	Upper turret — 1st spindle; Milling, Lower turret — 2nd spindle; Milling. 	G109L1 M901 M200 M203 S○○○ : Machining data M205 M202 M950 M30	G109L2 M902 M300 M203 S○○○ : Machining data M205 M302 M950 M30
13	Upper turret — 2nd spindle; Turning, Lower turret — 1st spindle; Turning. 	G109L1 M902 M302 M303 S○○○ : Machining data M305 M950 M30	G109L2 M901 M202 M3 S○○○ : Machining data M5 M950 M30
14	Upper turret — 2nd spindle; Milling, Lower turret — 1st spindle; Turning. 	G109L1 M902 M300 M203 S○○○ : Machining data M205 M302 M950 M30	G109L2 M901 M202 M3 S○○○ : Machining data M5 M950 M30

<table>
<tr><th>No.</th><th>Machining pattern</th><th colspan="2">Programming example</th></tr>
<tr><td>15</td><td>Upper turret — 2nd spindle; Turning,
Lower turret — 1st spindle; Milling.
</td><td>G109L1
M902
M302
M303 S○○○
: Machining data
M305
M950
M30</td><td>G109L2
M901
M200
M203 S○○○
: Machining data
M205
M202
M950
M30</td></tr>
<tr><td>16</td><td>Upper turret — 2nd spindle; Milling,
Lower turret — 1st spindle; Milling.
</td><td>G109L1
M902
G28UW
T014000T0 M6
M300
M203 S○○○
M950
G110C2M951 (Note)
M952
G00C90.
G111

G00X100.Z0.
G01Z-50.F100 (*)
G00X120.
Z0.
:

M205
M202
M953
M30</td><td>G109L2
M901
T003000

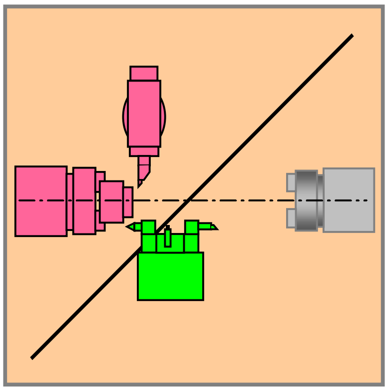
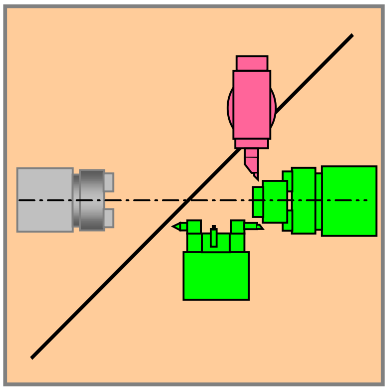
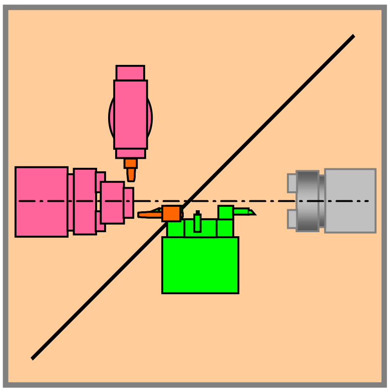
M200
M203 S○○○
M950
M951 (Note)
G110C1M952
G00C0.
G111

G00X100.Z-10.
G01X50.F100 (*)
G00Z10.
X100.
:

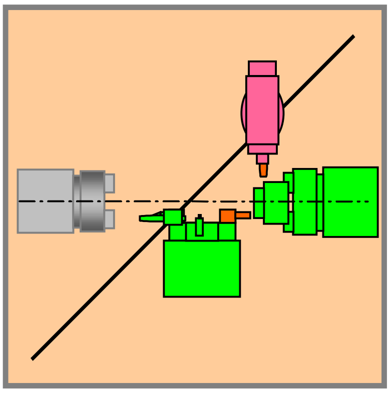
M205
M202
M953
M30

(*) Machining data</td></tr>
</table>

Note: Give commands of cross machining control (G110) successively for the 1st and 2nd spindles.

No.	Machining pattern	Programming example	
17	Upper turret — 1st spindle; Turning, Lower turret — 1st spindle; Turning. 	G109L1 M901 M202 M3 S○○○ M950 : Machining data M951 M5 M952 M30	G109L2 M901 M950 : Machining data M951 M952 M30
18	Upper turret — 2nd spindle; Turning, Lower turret — 2nd spindle; Turning. 	G109L1 M902 M950 : Machining data M951 M952 M30	G109L2 M902 M302 M303 S○○○ M950 : Machining data M951 M305 M952 M30
19	Upper turret — 1st spindle; Milling, Lower turret — 1st spindle; Milling. 	G109L1 M901 M950 M200 M951 M203 S○○○ : Machining data M205 M202 M952 M30	G109L2 M901 M950 M951 M203 S○○○ : Machining data M205 M952 M30

Return to Library

No.	Machining pattern	Programming example	
20	Upper turret — 2nd spindle; Milling, Lower turret — 2nd spindle; Milling. 	G109L1 M902 M950 M300 M951 M203 S○○○ : Machining data M205 M302 M952 M30	G109L2 M902 M950 M951 M203 S○○○ : Machining data M205 M952 M30

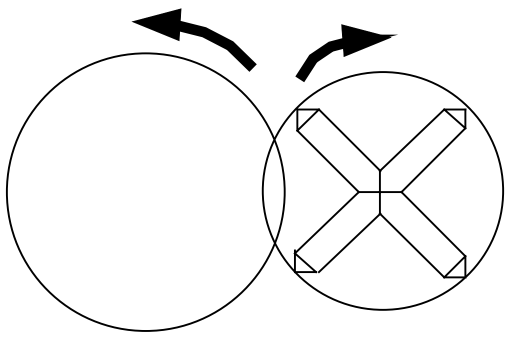
20 POLYGONAL MACHINING AND HOBBING (OPTION)

20-1 Polygonal Machining ON/OFF: G51.2/G50.2

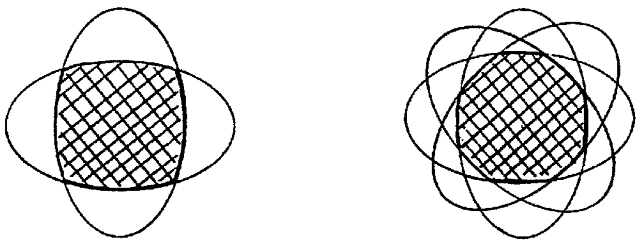
1. Function and purpose

A workpiece is machined in a polygonal shape by turning the rotary tool at a constant rate to the workpiece at a given rotating speed.
The shape to be machined depends on the following conditions:

- The number of the edges of a rotary tool
- The ratio of the rotating speed of a workpiece to that of a rotary tool



Polygonal machining has an advantage of machining polygonal workpieces in shorter time than polar coordinate interpolation. However, it has a disadvantage of not giving an accurate polygon. As a result, polygonal machining is usually used to machine bolt heads and nuts not requiring an accurate polygon.



Machining shape by polygonal machining (hatched section)

D732S0036

2. Programming format

Starting polygonal machining

G51.2 P_ Q_ ;

- Give a command so that addresses P and Q provide the following:
  (Address P): (Address Q) = (Workpiece rotational speed) : (Rotary tool speed)
- Command the rotational direction of rotary tool with the sign of address Q as follows.
  When the sign of Q is “+”, positive direction is selected.
  When the sign of Q is “–”, negative direction is selected.
- The command range of addresses P and Q is as follows:
  Command addresses P and Q with integers.
  They cannot be commanded with a value including decimal fraction.

Address	Command range
P	1 to 9
Q	–9 to –1, 1 to 9

- When commanding G51.2
  When the signal per revolution of position coder mounted on the spindle is sent, the rotary tool starts turning, synchronizing with the spindle used for the workpiece.
  Move command cannot be given to the rotary tool axis except the command of reference point return.
  The above two facts prove that the tool and the workpiece are always placed at the same position when the rotary tool starts turning. This reveals that intermittent polygonal machining does not impair the shape of a workpiece.

Canceling polygonal machining

G50.2;

3. Sample program

```
G28 U0 W0;
T11T00 M06;       Selection of tool No. 11 for polygonal machining
G98;              Mode of feed per minute
M260;             Polygonal machining mode selection
M3 S250;          Normal rotation of spindle at 250 rpm
G51.2 P1 Q-2;     Reversed rotation of milling spindle at 500 rpm
G0 X100.Z30.;   ⎫
G0 X46.6 Z3.;   ⎪
G1 Z-20.F50;    ⎪
G1 X60.F100;    ⎪
G0 Z3.;         ⎬ Machining
G0 X46.0;       ⎪
G1 Z-20.F30;    ⎪
G1 X60.F100;    ⎪
G0 X100.Z30.;   ⎭
G50.2;            Polygonal machining mode cancellation
M261;             Polygonal machining mode cancellation
M205;             Milling spindle stop
M5;               Spindle stop
M30;              End
```

Return to Library

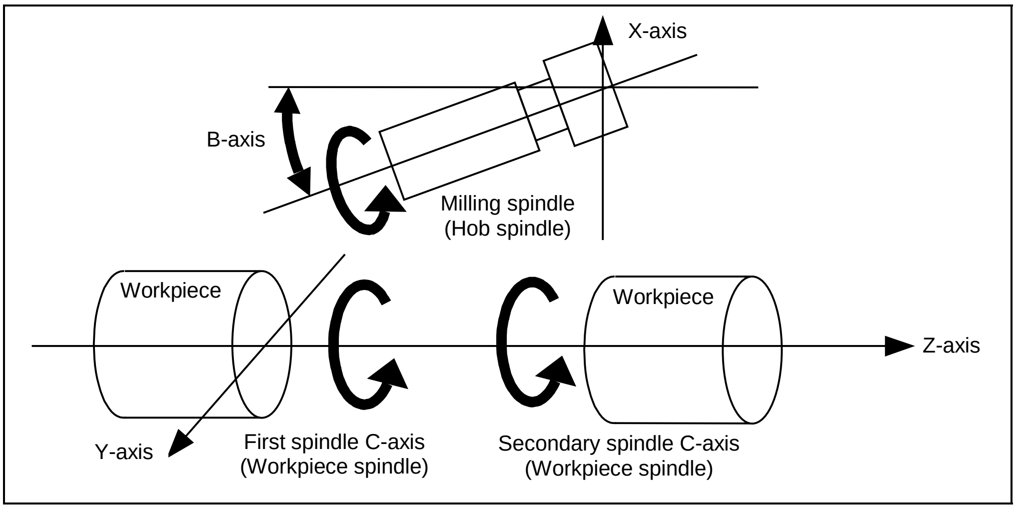
4. Notes

1. G50.2 and G51.2 must be commanded independently.
2. Command a proper workpiece rotating speed and ratio of such workpiece rotating speed to the rotary tool speed so that the maximum rotating speed of rotary tool cannot be exceeded.
3. Move command such as one for general control axis cannot be given to the rotary tool axis except the command of reference point return.
4. A mahine coordinate value of rotary tool axis is displayed within a range from 0 to “movement distance per rotation”. Relative coordinates and absolute coordinates are not renewed.
5. An absolute position detector cannot be mounted on the rotary tool axis.
6. Jogging feed and handle feed for the rotary tool axis are ineffective during polygonal machining.
7. Peforming thread cutting during polygonal maching makes the start point of thread cutting be shifted. Therefore, cancel the polygonal machining before thread cutting.
8. Rotary tool axis during polygonal machining is not counted as a synchronous control axis.
9. During polygonal machining, it is possible, indeed, but not advisable at all to apply feed hold or to change the override value for fear of deformation of the workpiece.
10. The milling spindle speed is not indicated on the **POSITION** display during polygonal machining.
11. The gear for rotary tool, if provided, must be taken into account in setting the ratio of milling spindle speed to spindle speed (Q : P).
12. Polygonal machining with the milling spindle can only be executed in combination with the first or main spindle (not possible, therefore, with the second or sub-spindle).

20-2 Selection/Cancellation of Hob Milling Mode: G114.3/G113

1. Outline

A synchronization control of the milling spindle and the C-axis allows them to be used as the hob spindle and the workpiece spindle, respectively, and thus enables the turning machine to generate spur and helical gears on a level with a hob milling machine.
The hob milling function, however, is only available to machines equipped with the control functions of the C-, B- and Y-axis.



2. Programming format

G114.3 H_D±_E_L_P_R_; Start of hobbing

H ....Selection of hob spindle (1: Selection of the milling spindle as hob spindle)

D ....Selection of workpiece spindle and its rotational direction
±1: C-axis of the first spindle
±2: C-axis of the secondary spindle
"+" for a rotation of the workpiece spindle in the same direction as the hob spindle.
"–" for a rotation of the workpiece spindle in the reverse direction to the hob spindle.

E ....Number of threads of the hob

L........Number of teeth on the gear

P ....Helix angle
Specify the desired helix angle for a helical gear.
Omit the argument, or specify 0 (degree) for a spur gear.

Q.......Module or Diametral pitch
Specify the normal module, or diametral pitch, for a helical gear. Set a negative value (with a minus sign) to use a hob cutter with left-hand teeth.
Enter the module for metric specification.
Enter the diametral pitch for inch specification.

R ....Angle of phase shift
Specify the angle for phase matching between the hob spindle (milling spindle) and the workpiece spindle (C-axis).
The specified angle refers to the initial rotation (angular positioning) of the hob spindle after completion of the zero-point return of the hob and workpiece spindles as a preparation for the synchronization control.

G113; Cancellation of the hob milling mode
The synchronization control of the hob spindle and the workpiece spindle is canceled.

- The setting range and default value for each argument are as follows:

Address	Setting range	Default value
H	1	1
D	±1, ±2	+1
E	0 to 20	1
L	1 to 9999	1
P	–90.000 to 90.000 [deg]	0 (Spur gear)
Q	±100 to ±25000 [0.001 mm or 0.0001 inch^{-1}]	Ommision of Q causes an alarm if a significant argument P is specified in the same block.
R	0 to 359.999 [deg]	No phase matching

- The arguments H and D lead to an alarm if a value outside the setting range is specified.
- The workpiece spindle does not rotate wit the argument E (Number of hob threads) set to "0". Accordingly, the designation of argument R for phase matching is not effective.
- The argument Q is ignored if the argument P is not specified in the same block.

3. Sample program

A. Generating a spur gear (without phase matching)

`M200;`	Selection of the milling mode.
`M203S0;`	Start of milling spindle normal rotation at a speed of zero.
`M250;`	Unclamping of the B-axis.
`G00B92.8;`	B-axis rotation through the lead angle (92.8°) of the hob cutter.
`M251;`	Clamping of the B-axis.
`G00X40.Z-5.;`	
`G114.3H1D+1E1L10;`	Selection of the hob milling mode. Positive value of D for the same rotational direction (normal in this case) of the workpiece spindle as the hob spindle.
`S50;`	Specification of the hob spindle rotation at 50 min^{-1}.
`G00X18.;`	
`G01Z20.F10;`	
`G00X40.;`	
`   Z-5.;`	
`G113;`	Cancellation of the hob milling mode.
`M205;`	Milling spindle stop.
`M202;`	Cancellation of the milling mode.

B. Generating a helical gear (with phase matching)

`G98;`	Selection of asynchronous feed mode.
`M200;`	Selection of the milling mode.
`M203S0;`	Start of milling spindle normal rotation at a speed of zero.
`M250;`	Unclamping of the B-axis.
`G00B92.8;`	B-axis rotation through the lead angle (92.8°) of the hob cutter.
`G00X40.Z-5.;`	
`G114.3H1D-1E1L10P45`	Selection of the hob milling mode (with phase matching for zero shift angle).
`Q2.5R0;`	Helix angle 45° (for B-axis rotation), Module 2.5 (mm). Negative value of D for the reverse rotational direction of the workpiece spindle to the hob spindle.
`M251;`	Clamping of the B-axis.
`S50;`	Specification of the hob spindle rotation at 50 min^{-1}.
`G00X18.;`	
`G01Z20.F10;`	
`G00X40.;`	
`   Z-5.;`	
`G113;`	Cancellation of the hob milling mode.
`M205;`	Milling spindle stop.
`M202;`	Cancellation of the milling mode.

C. Gear cutting on the secondary spindle

`M902;`	Selection of the the 2nd spindle side.
`M300;`	Selection of the milling mode for the 2nd spindle.
`M203S0;`	Start of milling spindle normal rotation at a speed of zero.
`M250;`	Unclamping of the B-axis.
`G00B92.8;`	B-axis rotation through the lead angle (92.8°) of the hob cutter.
`M251;`	Clamping of the B-axis.
`G00X40.Z-5.;`	
`G114.3H1D+2E1L10;`	Selection of the hob milling mode. Positive value of D for the same rotational direction (normal in this case) of the workpiece spindle as the hob spindle.
`S50;`	Specification of the hob spindle rotation at 50 min^{-1}.
`G00X18.;`	
`G01Z20.F10;`	
`G00X40.;`	
`  Z-5.;`	
`G113;`	Cancellation of the hob milling mode.
`M205;`	Milling spindle stop for the 2nd spindle.
`M302;`	Cancellation of the milling mode for the 2nd spindle.

4. Detailed description

1. The selection of the milling mode (M200) includes a zero-point return of the workpiece spindle (C-axis).
2. Give an S-code and M-code, respectively, to specify the rotational speed and direction of the spindle selected as the hob spindle.
3. The block of G114.3 must be preceded by a command of “0” speed and a selection of the rotational direction of the hob spindle. The synchronization cannot be established if a command of G114.3 is given with the hob spindle already rotating or without its rotational direction specified.
4. The rotational speed of the workpiece spindle is determined by the number of hob threads and that of gear teeth, both specified in the block of G114.3.

   Sw = Sh ∗ E/L

   where Sh: Rotational speed of the hob spindle
   Sw: Rotational speed of the workpiece spindle
   E: Rotational ratio of the hob spindle (Number of hob threads)
   L: Rotational ratio of the workpiece spindle (Number of gear teeth)
5. Once determined by the hob milling command (G114.3), the rotational relationship between the workpiece spindle and the hob spindle is maintained in all operation modes until a hob milling cancel command (G113) or spindle synchronization cancel command is given.
6. The synchronization of the workpiece spindle with the hob spindle is started by the hob milling command (G114.3) at a speed of 0 revolutions per minute.
7. In the mode of hob milling the C-axis counter on the **POSITION** display does not work as the indicator of actual motion.
8. Do not fail to give a milling mode cancel command (M202) after cancellation of the hob milling mode by G113.
9. Use the preparatory function for asynchronous feed (G98) to cut a helical gear.

5. Remarks

1. Gear cutting accuracy cannot be guaranteed if the milling spindle speed is changed by operating the override keys during execution of a feed block in the hob milling mode.
2. If a motion command for the C-axis (workpiece spindle) is given in the middle of the hob milling mode by a manual or MDI interruption, or even in the program, such a shifting motion will be superimposed on the synchronized C-axis movement. In this case, however, the synchronization between the C-axis and the milling spindle cannot be guaranteed.
3. The selection of the hob milling mode (G114.3) in the mode of polygonal machining (G52.1) will result in an alarm. The polygonal machining cannot be selected in the hob milling mode, either.
4. The designation of the secondary spindle by D±2 does not have any effect if it is not provided with the optional C-axis control function.
5. A faulty machining could occur if the axis movement should come to a stop in the hob milling mode by the activation of the single-block operation mode or the feed hold function.
6. A phase mismatching or an excessive error could occur if the milling spindle should be stopped in the hob milling mode by a command of M205, M00, or M01.
7. The C-axis offset settings are ignored appropriately in the hob milling mode.
8. If the specified speed of the milling spindle is in excess of its upper limit, the milling spindle speed will be set to that limit and the C-axis will rotate in accordance with the milling spindle limit sped and the rotational ratio.
9. If the calculated speed of the C-axis rotation exceeds its upper limit, the C-axis speed will be set to that limit and the milling spindle will rotate in accordance with the C-axis speed limit and the rotational ratio.
10. The hob milling function is not compatible with the geometry compensation function (G61.1). Cancel the geometry compensation mode as required to use the hob milling function.

- NOTE -

Return to Library

21 TORNADO TAPPING (G130)

1. Function and purpose

Tornado tapping cycle is provided to machine a tapped hole by one axial cutting motion with the aid of a special tool. While usual tapping cycles require multiple tools to be used in sequence, use of this cycle function spares tool change time as well as repetitive cutting motion in order to enhance the machining efficiency.

This cycle function is only available on machines equipped with the Y-axis control facility.

Note: Tornado tapping function requires the following parameter settings for macro-call G-codes:

J37 = 100009401 (Fixed value for the number of the macroprogram to be called for tornado tapping)

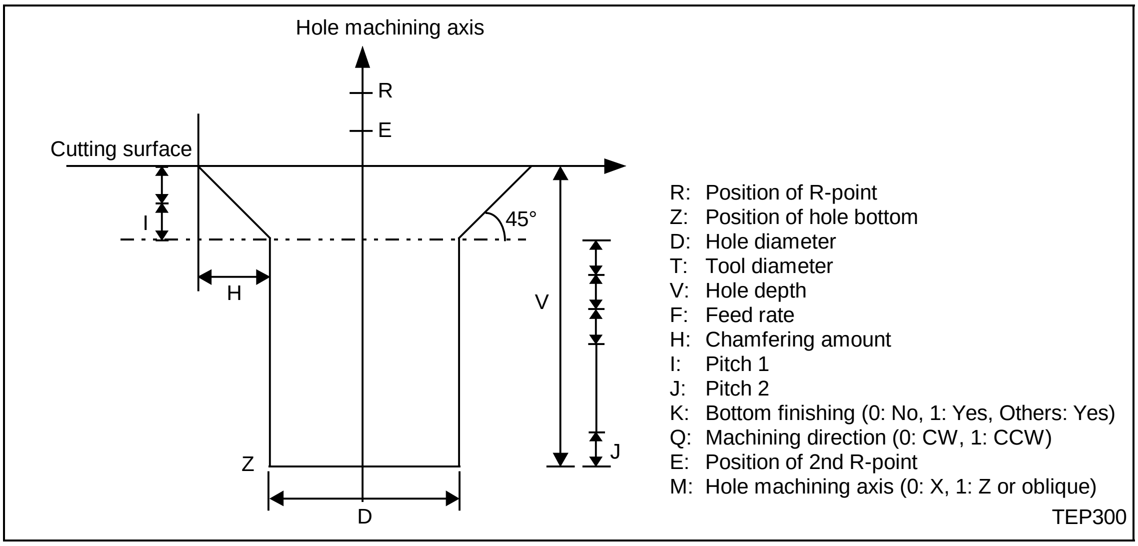
J38 = 130 (Fixed value for the number of the G-code to be used for macro call)

J39 = 2 (Fixed value for the type of macro call)

2. Programming format

The following format refers to hole machining on the face [or O. D. surface].

G17 [or G19];
G130 R_Z_D_T_V_F_H_I_J_K_Q_E_M1 [or M0];
X [or Z] _Y_; (Setting of hole position)
G67;



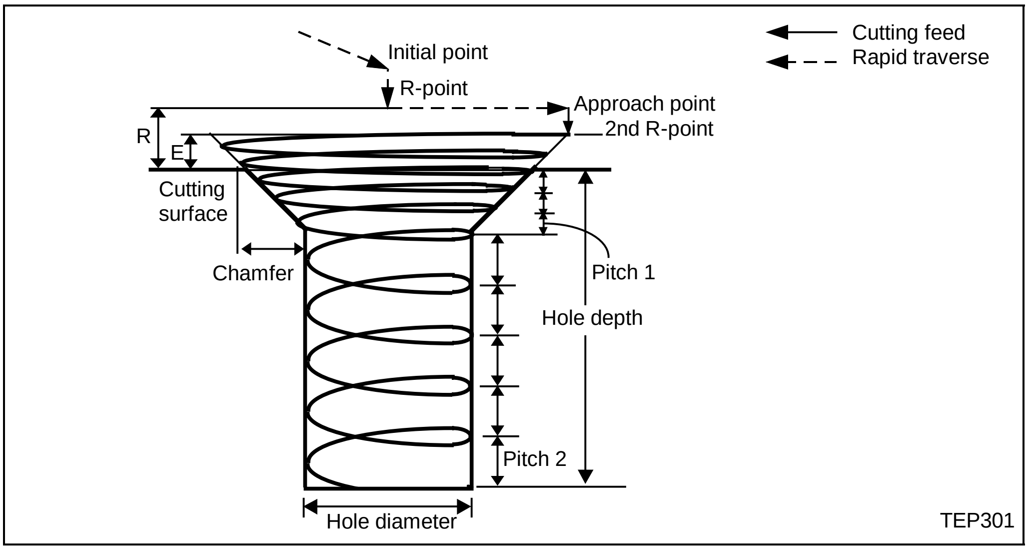
- The chamfering angle is fixed at 45°.
- Arguments D (hole diameter) and T (tool diameter) must satisfy the following condition:
  D ≥ T ≥ D/2.
- Argument K is used to select whether finishing is to be (K1) or not to be (K0) executed on the bottom of the hole.
- Set the hole position separately from the macro-call G-code (G130).
- As is the case with usual fixed cycles, actual machining with axial movement can only be executed for a block containing the hole position data.
- Do not fail to set the code G67 as required to cancel the modal call.
- Set the code G122.1 (Radius data input for the X-axis) as required beforehand to use the tornado tapping function.

3. Description of movement

A. Hole machining

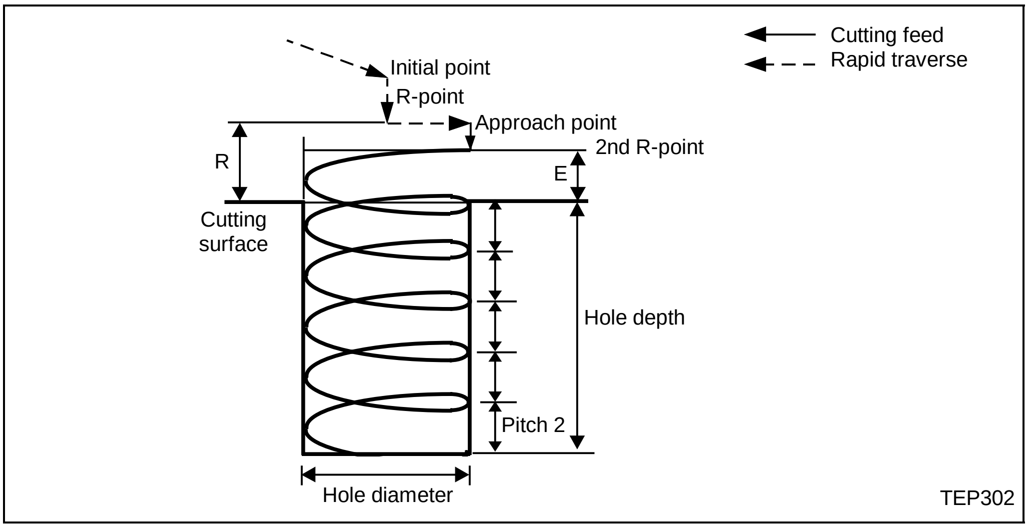
1. With chamfering

After moving from the current position to the R-point on the hole axis and then approaching to a point on the 2nd R-point level, chamfering is performed by a spiral-helical interpolation first, and then cylindrical machining is carried out to the bottom by a circular-helical interpolation.



2. Without chamfering

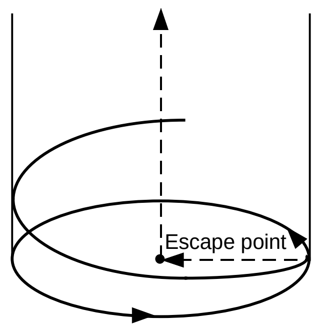
After moving from the current position to the R-point on the hole axis and then approaching through the hole radius and to a point on the 2nd R-point level, cylindrical machining is carried out from the top to the bottom by a circular-helical interpolation.



B. Movement on the bottom

1. With bottom finishing

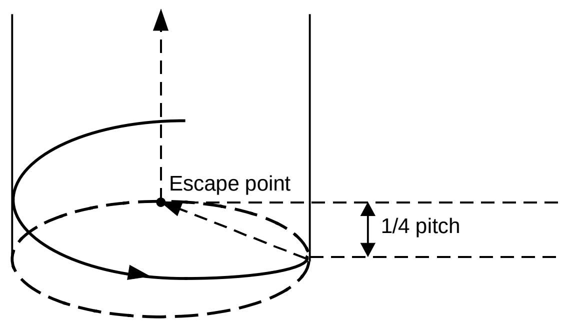
After cutting down to the bottom of the hole by helical interpolation, the tool performs a circular interpolation for full circle, and then escapes radially to the axis of the hole before returning in the axial direction to the initial point or R-point at the rapid traverse.



TEP303

2. Without bottom finishing

After cutting down to the bottom of the hole by helical interpolation, the tool escapes radially to the axis of the hole while axially returning through quarter the pitch, and then returns in the axial direction to the initial point or R-point at the rapid traverse.



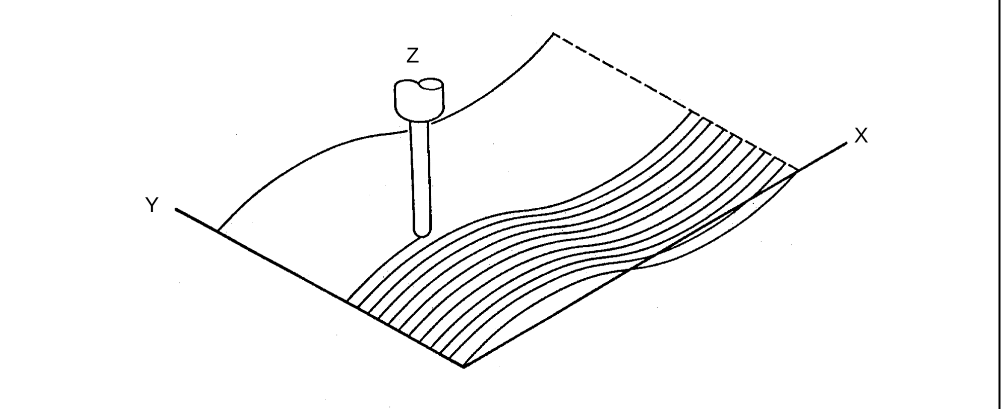
TEP304

- NOTE -

22 HIGH-SPEED MACHINING MODE FEATURE (OPTION)

1. Function and purpose

The high-speed machining mode feature allows high-speed execution of programs used for the machining of free-curved surfaces that have been approximated using very small lines.
In high-speed machining mode, microsegment machining capabilities improve by several times, compared with conventional capabilities. This allows the same machining program to be executed at several times the original feed rate, and thus the machining time to be reduced significantly.
Conversely, a machining program that has been approximated using lines of several fractions of the original segment length, can also be executed at the same feed rate, so more accurate machining is possible.
Combined use of the high-speed machining mode and the shape correction function allows more accurate machining to be implemented.
If, moreover, a protruding section exists in the microsegment machining program, smooth interpolation can be conducted automatically by removing this illegal path.



73129977

High-speed machining is available in the automatic operation modes: Memory, HD (Hard Disk), IC card and Ethernet.
Even in the high-speed machining mode can be applied various operational functions: override functions, cutting feed rate limit function, single-block operation function, dry run function, graphic trace function and high-precision control function.

The microsegment machining capability in the high-speed machining mode is as follows:

Operation mode	Max. speed	Conditions required
Memory operation	135 m/min (5315 IPM)	None
HD operation	67 m/min (2638 IPM)	With the **POSITION** display selected on the screen (see **Note 2**)
Ethernet operation	135 m/min (5315 IPM)	Avoid unusual key operations (see **Note 3**)
IC card operation	135 m/min (5315 IPM)	None

The microsegment machining capability is restricted further by the functions used in, or applied to, the program as shown below:

Preparatory functions		Fairing function	
		Not applied	Applied
G01	Linear interpolation only	135 m/min (5315 IPM)	84 m/min (3307 IPM)
G02/G03	Circular interpolation included	33 m/min (1299 IPM)	
G6.1	Fine-spline interpolation included	101 m/min (3976 IPM)	50 m/min (1969 IPM)

Note 1: The microsegment machining capabilities shown above refer to the case where three-axis simultaneous motion commands consist of 32 characters per block for a segment length of 1 mm.

Note 2: If the **POSITION** display should be changed to any other display during operation, program reading from the hard disk may be aborted to damage the surface to be machined.

Note 3: If unusual operations, such as holding down any cursor/page key or a mouse button, are performed, program reading from the network may be aborted to damage the surface to be machined.

Note 4: Before executing a microsegment machining program for hard disk operation or Ethernet operation, terminate the commercially available software if it is being used.

Note 5: Since optimum corner deceleration occurs during the shape correction mode, the machining time may be longer than in other modes.

2. Programming format

G5 P2 High-speed machining mode ON
G5 P0 High-speed machining mode OFF

Note: Both commands must be given in a single-command block.

3. Commands available in the high-speed machining mode

Only axis motion commands with the corresponding preparatory functions (G-codes) and feed functions (F-codes), and designation of sequence number are available in the high-speed machining mode. Setting data of any other type will result in an alarm (**807 ILLEGAL FORMAT**).

1. G-codes

   The available preparatory functions are G00, G01, G02 and G03.
   The circular interpolation can be programmed with R (radius designation) as well as with I and J (center designation). If the machining program includes circular commands, however, make bit 2 of the **F96** parameter valid.

   F96 bit 2: Type of control for circular commands in the high-speed machining mode:
   0: Control for the specified speed (with acceleration/deceleration)
   1: Control for a uniform feed

2. Axis motion commands

   The three linear axes (X, Y, Z) can be specified.
   Absolute data input as well as incremental data input is applicable, indeed, but the former input mode requires the validation of bit 5 of the **F84** parameter.

   F84 bit 5: Type of position data input in the high-speed machining mode:
   0: Always incremental data input
   1: According to the input mode before selection of the high-speed machining mode

3. Feed functions

   Feed rate can be specified with address F.

4. Sequence number

   Sequence number can be specified with address N. This number, however, is skipped as a meaningless code during reading.

5. Sample program

```
G28 X0 Y0 Z0
G90 G0X-100.Y-100.
G43 Z-5.H03
G01 F3000
G05 P2          ———— High-speed machining mode ON
X0.1                 When F84 bit 5 = 0:
X0.1 Y0.001              Incremental motion under G01
X0.1 Y0.002          When F84 bit 5 = 1:
      :                  Absolute motion under G01
X0.1 F200
G05 P0          ———— High-speed machining mode OFF
G49 Z0
M02
```

Note 1: Either 0 or 2 is to be set with address P (P0 or P2). Setting any other value will result in an alarm (**807 ILLEGAL FORMAT**).

Note 2: No other addresses than P and N must be set in the same block with G05.

Note 3: A decimal point must not be appended to address P.

Note 4: The maximum permissible length of one block is 30 characters.

4. Additional functions in the high-speed machining mode

A. Fairing function

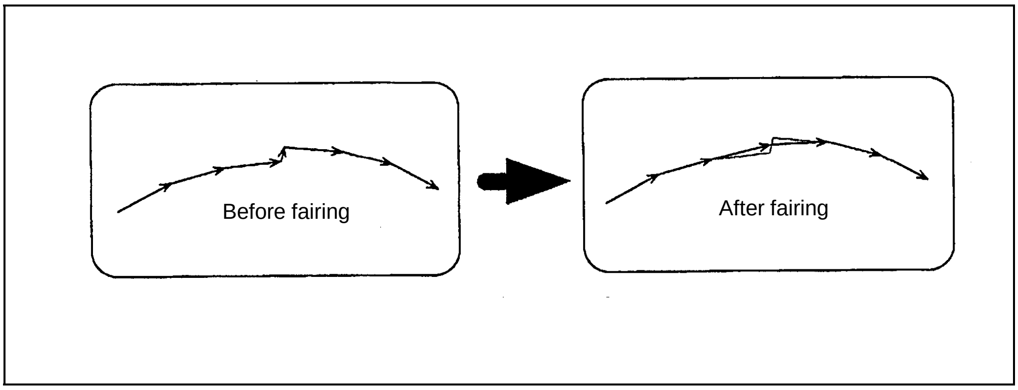
If, in a series of linear paths, a protruding section exists in the CAM-created microsegment machining program, this protruding path can be removed and the preceding and following paths connected smoothly by setting parameter **F96** bit 1 to “1”.

F96 bit 1: Fairing function for the microsegment machining program

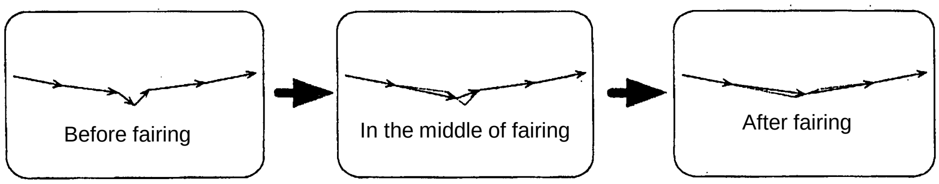
0: No fairing

1: Fairing for a protruding path

F103: Maximum length of a block to be removed for fairing



Fairing is also valid for a succession of protruding paths as shown below:



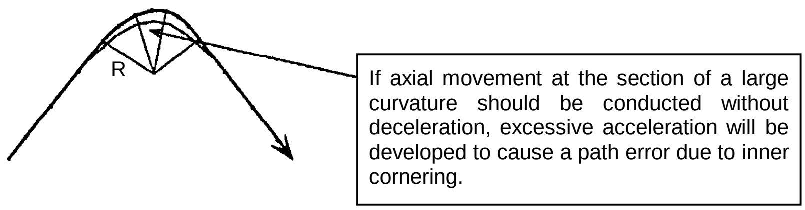
B. Cutting feed limiting speed

In shape correction mode, the minimum of the cutting feed limiting speeds of the movable axes is set as the cutting feed limiting speed in the high-speed machining mode. Setting parameter **F96** bit 5 to “1”, however, allows the curvature of every curved section to be judged for limiting the speed so as not to exceed the maximum available acceleration.

F96 bit 5: Type of cutting feed limiting speed for the high-speed machining mode

0: Minimum of the cutting feed limiting speeds of the movable axes

1: Limiting speed based on the radius of curvature



C. Deceleration at corners in the high-speed machining mode

In shape correction mode, automatic deceleration at corners of significantly large angle is provided in general to ensure that the acceleration developed during cornering shall fall within the predetermined tolerance.

A micro-length block between relatively longer blocks intersecting each other in a large angle in CAM-created microsegment machining programs, in particular, may cause the cornering speed to mismatch the surroundings and thus affect surface quality.

Setting parameter **F96** bit 4 to “1” will now allow corner judgment and deceleration without suffering any effects of such a microblock.

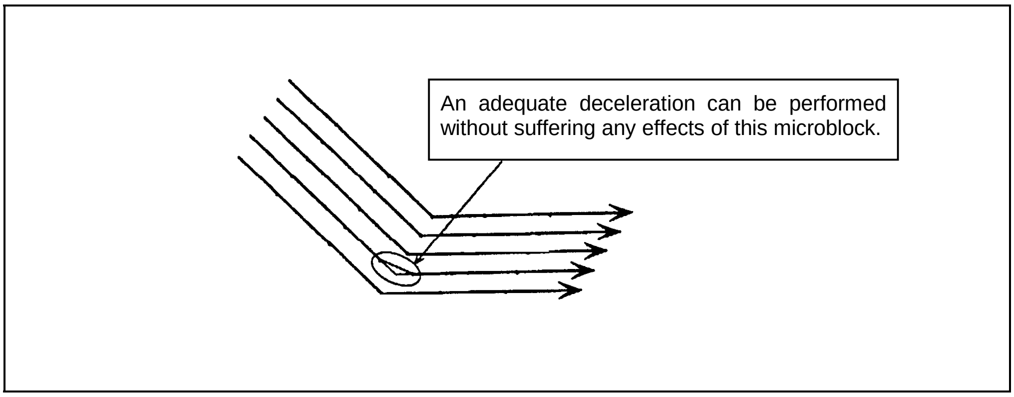
To use this function, however, the high-accuracy control option is required in addition to the optional high-speed machining function.

F96 bit 4: Type of corner judgment in the high-speed machining mode

0: Always judging from the angle between adjacent blocks

1: Judging after removing any microblock (if present between large-angle blocks)

F107: Reference length for microblock judgement



5. Restrictions

1. The modal functions other than that of G-code group 01 will be saved during, and restored upon cancellation of, the high-speed machining mode, indeed; but the modal functions for tool diameter offset, mirror image, scaling, coordinate system rotation, virtual axis interpolation and three-dimensional diameter offset should have been cancelled beforehand to give a G05 P2 command. Otherwise, an alarm may be caused or the modal function unexpectedly cancelled.

Example: Main program

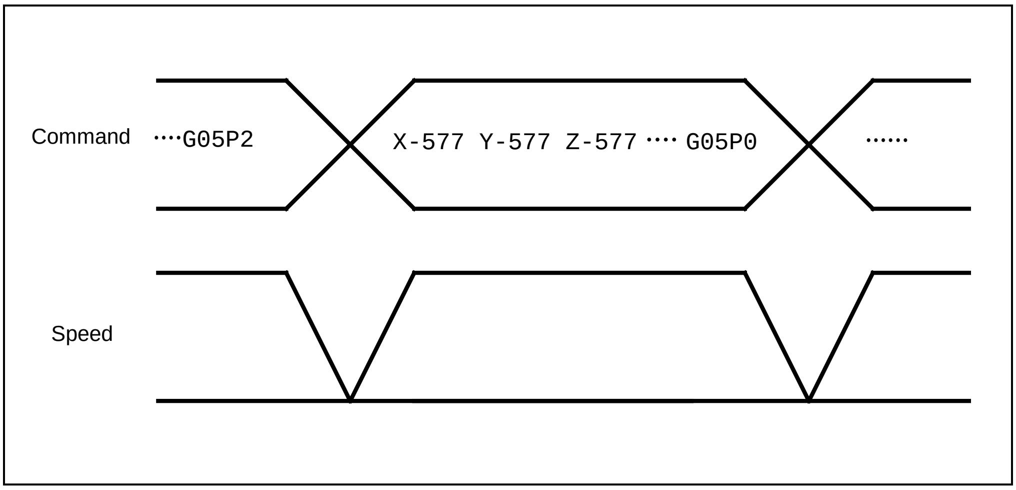
```
G28 X0 Y0 Z0
G90 G92 X0 Y0 Z100.
G00 X-100.Y-100.
G43 Z-10.H001      Movement under the conditions of G90, G00 and G43
M98 H001
G49 Z0             Movement under the conditions of G90 and G01
G28 X0 Y0 Z0
M02
```

Subprogram (O001)

```
N001 F3000
G05 P2             High-speed machining mode ON
G01 X0.1              When F84 bit 5 = 0:
X-0.1 Y-0.001             Incremental motion under G01
X-0.1 Y-0.002         When F84 bit 5 = 1:
        ⋮                 Absolute motion under G01
X0.1
G05 P0             High-speed machining mode OFF
M99
```

2. In the high-speed machining mode there may occur a delay in display response since priority is always given to the processing for the automatic operation.

3. The high-speed machining mode should be selected and cancelled by using commands of G05 P2 and G05 P0, respectively, with the tool sufficiently cleared from the workpiece since the selection and cancellation always cause a deceleration of feed motions as shown below:



4. Restrictions on programming and machine operation are listed in the following table:

○: Valid, –: Invalid, err: Error

Specification		Standard Mode	High-speed mode (Designation in the mode)
Classification	Subclassification		
Control axes	Maximum controllable axis quantity	14	14
	Effective controllable axis quantity	14	7
	Simultaneously controllable axis quantity	5	5
	Axis name	○	○ (○)
	CT axis	○	○ (○)
Units of control	Unit of input	ABC	ABC
	Unit of programming	○	○
	Unit-of-programming × 10	○	○
Input formats	Tape code	ISO/EIA	ISO/EIA
	Label skip	○	– (–)
	ISO/EIA automatic identification	○	○ (○)
	Parity H	○	○ (○)
	Parity V	○	○ (○)
	Tape format	○	Refer to the programming format.
	Program number	○	○ (err)
	Sequence number	○	○ (○)
	Control IN/OUT	○	○ (err)
	Optimal block skip	○	○ (err)
Buffers	Tape input buffer	○	○ (○)
	Pre-read buffer	○	○ (○)
Position commands	Absolute/inrcemental data input	○	○ (err)
	Inch/metric selection	○	○ (err)
	Decimal point input	○	○ (○)

○: Valid, –: Invalid, err: Error

Classification	Subclassification	Standard Mode	High-speed mode	(Designation in the mode)
Interpolation functions	Positioning	○	○	(○)
	One-way positioning	○	–	(err)
	Linear interpolation	○	○	(○)
	Circular interpolation	○	○	(○)
	Helical cutting	○	–	(err)
	Spiral interpolation	○	–	(err)
	Virtual-axis interpolation	○	–	(err)
	Threading	○	–	(err)
	Plane selection	○	○	(err)
	Fine-Spline interpolation	○	○	(err)
	NURBS interpolation	○	–	(err)
Feed functions	Rapid feed rate	○	○	(○)
	Cutting feed rate	○	○	(○)
	Synchronous feed	○	○	(err)
	Automatic acceleration/deceleration	○	○	(○)
	Linear acceleration/deceleration before cutting interpolation	○	○	(err)
	Cutting feed rate limitation	Limitation in cutting direction	Minimum limiting speed of feed axes/ According to curvature	
	Rapid feed override	○	○	(○)
	No. 1 cutting feed override	○	○	(○)
	No. 2 cutting feed override	○	○	(○)
	Exact-stop mode	○	–	(err)
	Cutting mode	○	○	(err)
	Tapping mode	○	–	(err)
	Automatic corner override	○	–	
	Error detection	○	○	(○)
	Override cancellation	○	○	
Dwell	Dwell in time	○	–	(err)
	Dwell in number of revolutions	○	–	(err)
Miscellaneous function	M-command	○	○	(err)
	M independent output command	○	–	(err)
	Optional stop	○	–	(err)
	No. 2 miscellaneous functions	○	○	(err)
Spindle functions	S-command	○	○	(err)
Tool functions	T-command	○	○	(err)
	Tool operation time integration	○	○	(○)
	Spare-tool selection	○	○	(–)
Tool offset functions	Tool-length offset	○	○	(err)
	Tool-position offset	○	–	(err)
	Tool-diameter offset	○	–	(err)
	3D-tool-diameter offset	○	–	(err)
	Tool-offset memory	○	○	(○)
	Number of tool offset data sets	○	○	(○)
	Programmed tool-offset input	○	–	(err)
	Tool-offset number auto selection	○	○	(err)

○: Valid, –: Invalid, err: Error

Specification		Standard Mode	High-speed mode (Designation in the mode)	
Classification	Subclassification			
Program auxiliary functions	Fixed cycle for drilling	○	–	(err)
	Pattern cycle	–	–	(–)
	Subprogram control	○	○	(err)
	Variable command	○	–	(err)
	Figure rotation	○	–	(err)
	Coordinate rotation	○	–	(err)
	User macro	○	○	(err)
	User macro interruption	○	○	(err)
	Scaling	○	–	(err)
	Mirror image	○	–	(err)
	Geometric function	○	–	(err)
	Geometric function	○	–	(err)
	Programmed parameter setting	○	err	(err)
Coordinate system setting	Watchdog-based reference-point return	○	○	(–)
	Memory-based reference-point return	○	○	(–)
	Automatic reference-point return	○	–	(err)
	#2/#3/#4 reference-point return	○	–	(err)
	Reference-point check	○	–	(err)
	Machine corrdinate system offset	○	–	(err)
	Workpiece coordinate system offset	○	–	(err)
	Local coordinate system offset	○	–	(err)
	Coordinate system setting	○	–	(err)
	Corrdinate system rotation setting	○	–	(err)
	Program restart	○	○	(err)
	Absolute data detection	○	○	(○)
Machine error correction	Backlash correction	○	○	(○)
	Lost-motion correction	○	○	(○)
	Memory-based pitch error correction	○	○	(○)
	Memory-based relative position correction	○	○	(○)
	Machine coordinate system correction	○	○	(○)
Protection functions	Emergency stop	○	○	(○)
	Stroke end	○	○	(○)
	Software limit	○	○	(○)
	Programmed software limit	○	–	(err)
	Interlock	○	○	(○)
	External deceleration	○	○	(○)
	Data protection	○	○	(○)

O: Valid, –: Invalid, err: Error

Specification		Standard Mode	High-speed mode (Designation in the mode)	
Classification	Subclassification			
Operation modes	Tape operation	O	O	(–)
	Memory operation	O	O	(–)
	MDI operation	O	O	(O)
	Jog feed	O	–	(O)
	Incremental feed	O	–	(O)
	Handle feed	O	–	(O)
	Manual rapid feed	O	–	(O)
	Handle interruption	O	O	(O)
	Auto/manual simultaneous	O	O	(O)
	HD operation	O	O	(–)
	IC card operation	O	O	(–)
	Ethernet operation	O	O	(–)
External control signals	Automatic-operation start	O	O	(O)
	Automatic-operation halt	O	O	(O)
	Single-block stop	O	O	(O)
	NC reset	O	–	(O)
	External reset	O	–	(O)
	All-axis machine lock	O	O	(O)
	Axis-by-axis machine lock	O	O	(O)
	Dry run	O	O	(O)
	Miscellaneous-function lock	O	O	(O)
	Manual-absolute selection	O	O	(–)
Status output signals	Control-unit ready	O	O	(O)
	Servo-unit ready	O	O	(O)
	Auto-run mode	O	O	(O)
	Auto-run in progress	O	O	(O)
	Auto-run halted	O	O	(O)
	Cutting feed in progress	O	O	(O)
	Tapping in progress	O	–	(–)
	Threading in progress	O	–	(–)
	Axis selected	O	O	(O)
	Axis-movement direction	O	O	(O)
	Rapid feed in progress	O	O	(O)
	Rewind		O	(O)
	NC alarm	O	O	(O)
	Reset	O	O	(O)
	Movement-command completed	O	O	(O)
Measurement aid functions	Manual tool-length measurement	O	–	(–)
	Automatic tool-length measurement	O	–	(err)
	Skip	O	–	(err)
	Multi-step skip	O	–	(err)
	Manual skip	O	–	(err)
Axis control functions	Servo off	O	O	(O)
	Follow-up	O	O	(O)
	Control-axis removal	O	O	(O)

O: Valid, –: Invalid, err: Error

Specification		Standard Mode	High-speed mode (Designation in the mode)	
Classification	Subclassification			
Data input/output	External data input I/F	O	O	(O)
	External data output I/F	O	O	(O)
	External data input/output	O	O	(O)
Setting/display functions	Setting/Display unit	O	O	(O)
	Settings display	O	O	(O)
	Search	O	O	(err)
	Check-and-stop	O	–	(–)
	MDI	O	O	(O)
	Program restart	O	O	(err)
	Machining-time calculation	O	O	(O)
	PC opening	O	O	(O)
	Program-status display	O	O	(O)
	Integrated-time display	O	O	(O)
	Graphics display	O	O	(O)
Program creation	Multi-step skip	O	–	(err)
	Graphics check	O	O	(O)
Self-diagnostics	Program-error display	O	O	(O)
	Operation-error display	O	O	(O)
	Servo-error display	O	O	(O)
	Operation-stop-cause display	O	O	(O)
	Servo monitor display	O	O	(O)
	NC-PC I/O signal display	O	O	(O)
	DIO display	O	O	(O)
	Keyboard-operation record	O	O	(O)

23 AUTOMATIC TOOL LENGTH MEASUREMENT: G37 (OPTION FOR SERIES M)

1. Function and purpose

When the tool for which command data has been assigned moves to a programmed measurement position, the NC system will measure and calculate any differential data between the coordinates at that time and those of the programmed measurement position. Data thus obtained will become offset data for that tool.
Also, if offsetting has already been performed for the tool, the current offset data will be further offset, provided that after movement of that tool under an offset status to the required measurement position, the measurements and calculations of any differential coordinates show some data to be further offset.
At this time, further offsetting will occur for the tool offset data if only one type of offset data exists, or for the tool wear offset data if two types of offset data exist (tool length offsets and tool wear offsets).

2. Programming format

G37 Z_ (X_, Y_) R_ D_ F_

X, Y, Z: Address of the measurement axis and the coordinate of the measurement position

R: Distance from the starting point of movement at a measurement feed rate, to the measurement position

D: The area where the tool is to stop moving

F: Measurement feed rate

If R, D, or F is omitted, respective parameter values will become valid.

3. Description of parameters

Parameter	Description
F42	R-code command. Deceleration area
F43	D-code command. Measurement area
F44	F-code command. Measurement feed rate f
F72	Conditions for skipping based on EIA G37

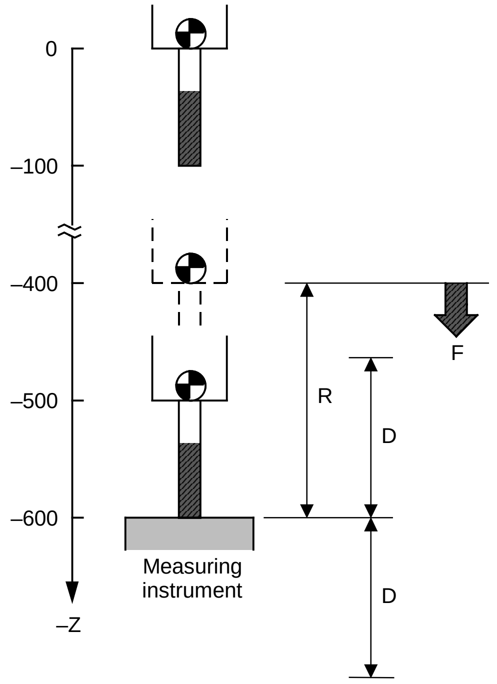
See the Parameter List for further details.

4. Example of execution

If H01 = 0

```
T01T00M06
G90G00G43Z0H01
G37Z-600.R200.D150.F300
```

Coordinate to reach the measurement position
= –500.01
–500.01 – (–600) = 99.99
0 + 99.99 = 99.99
Thus, H01 = 99.99



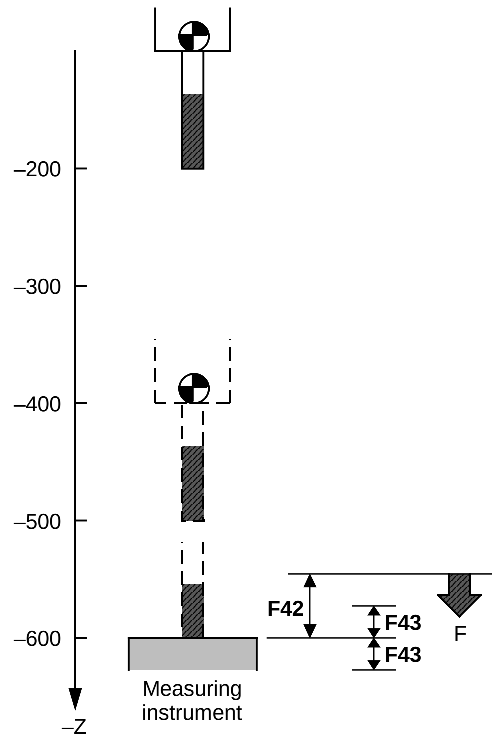
MEP229

If H01 = 100

```
T01T00M06
G90G00G43Z-200.H01
G37Z-600.F300
```

Coordinate to reach the measurement position
= –600.01
–600.01 – (–600) = –0.01
100 + (–0.01) = 99.99
Thus, H01 = 99.99

<Supplement>
When the program shown above is executed, parameter **F42** and **F43** are set as follows:
F42 (R-code command) : 25000 (25 mm)
F43 (D-code command) : 2000 (2 mm)



MEP230

5. Detailed description

1. Machine action based on command G37

G28X0Y0Z0
G90G0G43Zz_1Hh_0 .... [1]
G37Zz_0Rr_0Dd_0Ff_0 .... [2], [3]
G0G90Zz_1 .... [4]
G28X0Y0Z0 .... [5]

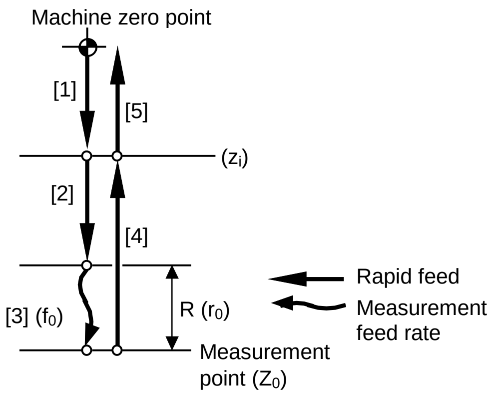
h_0 : Offset number

z_0 : Coordinate of the measurement point (measurement position)

r_0 : Starting point of movement at measurement feed rate

d_0 : The area where the tool is to stop moving

f_0 : Measurement feed rate



MEP231

2. Sensor signals (Measurement Position Reached) also act as skip signals.
3. If the F-code value is 0, the feed rate becomes 1 mm/min.
4. Update offset data becomes valid from the Z-axis (measurement axis) command codes that succeed the block of G37.
5. The delay and dispersion in processing of sensor signals, except for the PLC side, is from 0 to 0.2 msec for the NC side alone. Accordingly, the following measurement error may occur:

Maximum measurement error [mm]

$$= \text{Measurement feed rate [mm/min]} \times \frac{1}{60} \times \frac{0.2\ [\text{ms}]}{1000}$$

6. When a sensor signal is detected, although the coordinates of the machine position at that time will be read, the machine will stop only after overruning through the distance equivalent to a servo droop.

Maximum amount of overrun [mm]

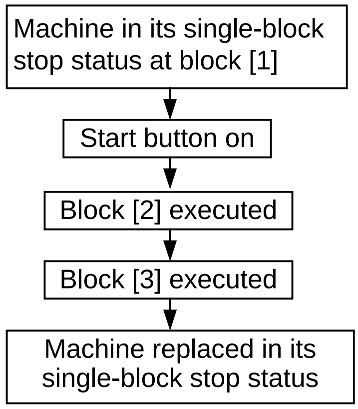
$$= \text{Measurement feed rate [mm/min]} \times \frac{1}{60} \times \frac{30.3\ [\text{ms}]}{1000}$$

30.3 [msec] if the position loop gain is 33.

7. If command G37 is executed in the single-block operation mode, the machine will come to a single block stop after execution of the block that immediately succeeds the G37-containing block.

Example:

```
G0G90G43Z-200.H01      [1]
G37Z-600.R25.D2.F10    [2]
G0G90Z-200.            [3]
```



6. Precautions

1. Alarm **889 G37 OPTION NOT FOUND** will result if G37 is set for a machine that does not have a mounted option for automatic tool length measurement.
2. Alarm **923 ILLEGAL COMMAND G37 AXIS** will result if the block of G37 does not contain axis data or contains data of two or more axes.
3. Alarm **924 G37, H COMMANDS SAME BLOCK** will result if an H code exists in the block of G37.
4. Alarm **925 H CODE REQUIRED** will result if G43 H_ does not exist before the block of G37.
5. Alarm **926 ILLEGAL G37 SIGNAL** will result if input sensor signals occur outside a predetermined allowable measurement range or if a sensor signal is not detected on arrival of the tool at the ending point of movement.
6. If a manual interruption operation has been carried out during movement of the tool at a measurement feed rate, the program must be restarted only after returning that tool to the position existing when the interruption operation was carried out.
7. Set G37 data or parameter data so that the following condition is satisfied:

   Measurement point – Staring point > R-code value or parameter r > D-code value or parameter d

8. If the R-code value, the D-code value and parameter d, mentioned in Item G above, are all 0s, the program will come to a normal end only when the designated measurement point and the sensor signal detection point agree. Alarm **926 ILLEGAL G37 SIGNAL** will result in all other cases.
9. If the R-code value, the D-code value, parameter r, and parameter d, mentioned in Item G above, are all 0s, alarm **926 ILLEGAL G37 SIGNAL** will result after the tool has been positioned at the designated measurement point, irrespective of whether a sensor signal is detected.
10. Set G37 (automatic tool length measurement code) together with G43 H_ (offset number assignment code).

    G43 H_
    G37 Z_R_D_F_

Return to Library

11. If the offset data is tool offsets of type A, then automatic correction of tool data occurs, or if the offset data is tool offsets of type B, then automatic correction of tool wear offsetting data occurs.

Example: The **TOOL OFFSET** displays in both cases after offsetting of H1 = 100

	TOOL OFFSET (Type A)				TOOL OFFSET (Type B)		
						TOOL LENGTH	
Before measurement	No.	OFFSET	No.	OFFSET	No.	GEOMETRY	WEAR
	1	$\boxed{100}$	17	0	1	100	$\boxed{0}$
	2	0	18	0	2	0	0
	3	0	19	0	3	0	0
						TOOL LENGTH	
After measurement	No.	OFFSET	No.	OFFSET	No.	GEOMETRY	WEAR
	1	$\boxed{110}$	17	0	1	100	$\boxed{10}$
	2	0	18	0	2	0	0
	3	0	19	0	3	0	0

12. The distance from the machine zero point to the measurement point (skip sensor) is preset in register R2392 or R2393. Use this value as reference to set a coordinate using Z-, X-, or Y-code command.
13. When this function is used for tool offsets of type B, the correct data will not be displayed if the wear offset value exceeds 100.
14. When executing this function in the presence of offset data, set the value of a D code to 2mm or less to prevent damaging the measuring instrument.
15. When executing this function in the absence of offset data (offset data = 0), set the values of an R code and a D code to those larger than the tool length of the tool to be measured. Also, in that case, before executing this function, make sure that the skip sensor in the measuring instrument correctly operates.

- NOTE -

24 DYNAMIC OFFSETTING II: G54.2P0, G54.2P1 - G54.2P8 (OPTION FOR SERIES M)

1. Function and purpose

When a workpiece fixed on the turntable is to be machined with the rotation of the table, mis-matching between the workpiece reference position (program origin) and the origin of workpiece coordinates (center of rotation of the table) leads to an error in machining contour. Provided that the vector of a particular deviation from the center of rotation to the workpiece reference position is given as a “reference”, the “Dynamic Offsetting II” function will calculate for each command of rotation the deviation vector for the designated angular motion in order to control the linear axes for an adequate movement to the ending point as programmed with respect to the ideal workpiece origin, and thus to prevent the above-mentioned faulty machining from occurring.

2. Programming format

G54.2 Pn;

n: Dynamic offset number (1 to 8)
Give a “G54.2 P0” command (n = 0) to cancel the dynamic offsetting function.

Cancellation is the initial state of the function (upon turning-on).

3. Definitions of terms

A. Deviation vector

The vector of a deviation from the center of rotation of the table (Wo: presupposed position of the workpiece origin) to the actual origin of coordinates of the workpiece mounted on the table.

B. Dynamic offset

The offsetting vector (= deviation vector; whose direction depends upon the angular position of the table) for the ending point of each block containing a command of rotation.

C. Reference dynamic offset

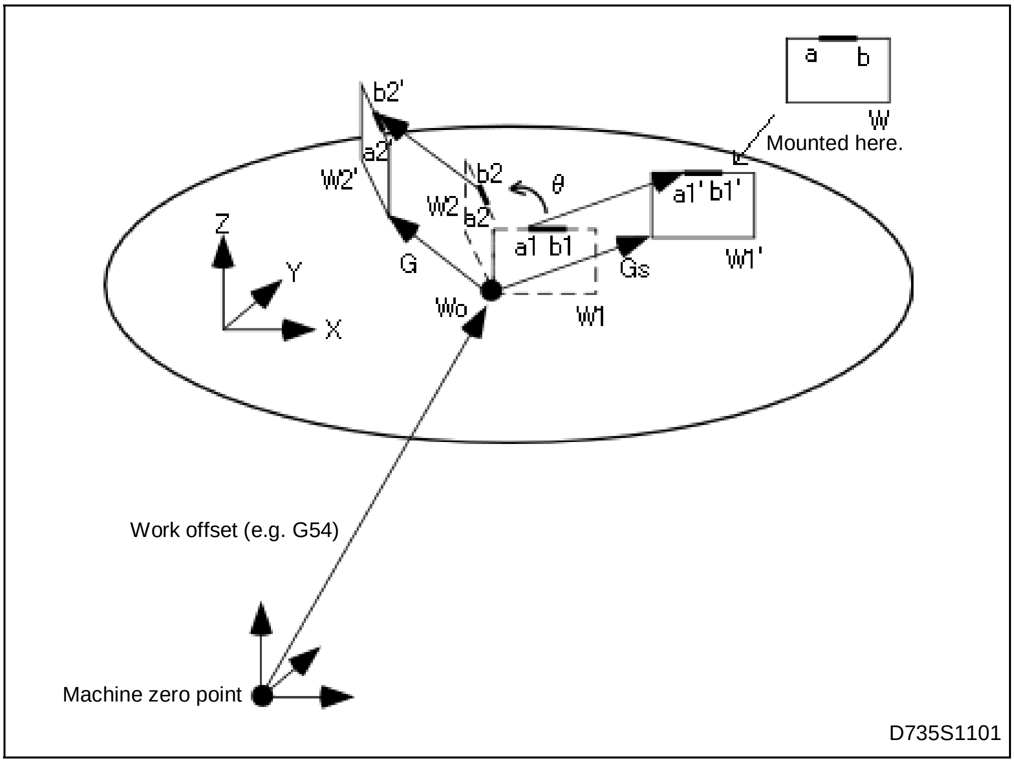
A particular deviation vector entered as the reference for the calculation of dynamic offsets. Consists of the vector proper (measured and entered in three-axis component vectors) and the positions (in machine coordinates) of the rotational and tilting axis for the measurement.

Return to Library

4. Operation description

A. Operation by a command of rotation in the G54.2 mode

In the G54.2 mode (modal group 23), which is selected by a "G54.2Pn" command, deviation vector (to be used in a vector addition for offsetting) is re-calculated for each command of table rotation beforehand in order to create an adequate tool path for the block's ending point as programmed with respect to the ideal workpiece origin.



[Legend]

W1:	The ideal workpiece mounting position (the workpiece origin set on to the center of rotation of the table)
W1':	The actual workpiece mounting position (vector Gs denotes the deviation from the ideal position)
W2':	The position of the actual workpiece W1' after a table rotation by θ
W2:	The position of the ideally mounted workpiece W1 after a table rotation by θ
Wo:	The origin of workpiece coordinates (given by a corresponding preparatory function, such as G54)
Gs:	The reference deviation vector (to be registered in the NC unit as a reference dynamic offset.)
G:	The deviation vector for the rotation of the rotational axis by θ
a (a1, a1'):	The starting point of the G1 (linear interpolation) microsegment command
b (b1, b2'):	The ending point of the G1 (linear interpolation) microsegment command

With the measurement results of the reference dynamic offset (Gs) registered for workpiece W fixed on the turntable, the selection (activation) of the G54.2 mode causes the tool to be shifted by the deviation vector Gs from the current position, point **a1** for example, to point **a1'** (if bit 0 of the **F87** parameter described later is set to "0").

A succeeding command of "G1b1" (b1 = designation of a point with X-, Y-, and Z-coordinates) feeds the tool from **a1'** to **b1'** in the G1 mode (linearly). If, however, simultaneous motion of the rotational axis is designated in the same block, "G1b1Cθ" for example, the tool is also fed linearly from the current position **a1'** to the offset position **b2'** which is obtained by adding the deviation vector G internally calculated for the θ rotation to point **b2**, the ending point on the ideally mounted workpiece.

B. On-reset operation

When the system is reset, the dynamic offsetting mode is normally canceled.

It depends, however, on the setting of parameter **F95** bit 7 whether or not dynamic offsetting is canceled on system reset operations.

F95 bit 7 = 0: The dynamic offset is cleared and the G54.2 mode is also canceled.

= 1: The existing dynamic offset is held along with the G54.2 mode. When the automatic operation is started again after resetting, the dynamic offsetting mode is active from the beginning of the program.

Note: When the dynamic offset is cleared by resetting, the tool will not move on the path corresponding to the cleared vector (even if bit 0 of the **F87** parameter described later is set to “0”).

C. Operation by the selection and cancellation of the G54.2 mode

When a G54.2Pn command is given, the deviation vector for the current position of the rotational axis is calculated and an offsetting movement is carried out on the linear axes by their respective components of the computed vector (dynamic offset). If an axis motion command is given in the same block, the deviation vector for the ending point of that block is calculated and the corresponding motion is performed from the current point to the dynamically offset ending point.

The cancellation command (G54.2P0) moves the tool by a vector reverse to the current dynamic offset. If an axis motion command is given in the same block, the corresponding motion is performed from the current point to the ending point as designated with workpiece coordinates (a movement including the cancellation of the dynamic offsetting).

The axis motion occurs according to the current modal function concerned (of G-code group 1).

D. Manual interruption in the G54.2 mode

The deviation vector does not change if automatic operation is stopped in the G54.2 mode (by single-block stop, etc.) and then a movement on the rotational axis carried out in manual mode. The re-calculation of the deviation vector for dynamic offsetting will not occur until a rotational axis motion command or another G54.2 command is given after setting the MDI or automatic operation mode.

5. Input and output of the reference dynamic offset

A. Setting the reference dynamic offset by G10

G10 L21 Pn Xx Yy ······$\alpha\alpha$;

Use this format of programmed parameter input. Argument P (n) denotes a dynamic offset number (1 to 8).
According to the data input mode, absolute (G90) or incremental (G91), the designated axis value overwrites, or is added to, the current one.

B. Reading/writing the reference dynamic offset with system variables

System variable number = 5500 + 20 × n + m

n: Dynamic offset number (1 to 8)
m: Axis number (1 to 6)

Use system variable #5510 to read the selected dynamic offset number (1 to 8).

C. Reading the machine coordinates of the center of table rotation with system variables

#5141: X-coordinate of the center of table rotation (Machine parameter **S5** X)

#5142: Y-coordinate of the center of table rotation (Machine parameter **S5** Y)

#5143: Z-coordinate of the center of table rotation (Machine parameter **S5** Z)

6. Other detailed precautions

1. When the related parameters and reference dynamic offset are modified in the G54.2 mode, the modifications will become valid for the next G54.2Pn command onward.
2. The following describes how some specific commands are executed in the G54.2 mode.
   (a) Machine coordinate system selection (G53)
   A G53 command temporarily suppresses the dynamic offset and the axis motion is performed to the ending point as designated in machine coordinates. The deviation vector is not re-calculated even when a value for the rotational axis is specified. The dynamic offsetting function will not be recovered until a motion command is given with workpiece coordinates.

   (b) Workpiece coordinate system change (G54 to G59, G54.1, G92, G52)
   Even when the workpiece coordinate system is changed in the G54.2 mode, the reference dynamic offset is not re-calculated and dynamic offsets are calculated according to the existing reference dynamic offset. The axis motion is carried out to the position obtained by adding the deviation vector to the ending point specified in the new workpiece coordinate system.

   (c) Commands related to zero point return (G27, G28, G29, G30, G30.n)
   The dynamic offsetting function is temporarily canceled for the path from the intermediate point to the reference point and recovered for the movement from there to a position specified in the workpiece coordinate system. (Similar to the processing of the commands related to zero point return in the tool length offset mode)
3. When the work offset data (workpiece origin) being used is modified by a G10 command in the G54.2 mode, the new work offset data will be valid for the next block onward.
4. As for the tool motion caused by a change only in the deviation vector, it is executed in the current mode of G-code group 1 and at the current rate of feed. If, however, the mode concerned is other than that of G0 or G1, e.g. a mode of circular interpolation (G2, G3, etc.), the tool is temporarily moved in the mode of linear interpolation (G1).
5. The type of the control axis for the turntable must be specified as “rotational”. The dynamic offsetting function II cannot be used for the C-axis specified as “linear type”.
6. The polar coordinate interpolation with the rotational axis cannot be executed properly in the G54.2 mode.
7. The following function commands cannot be executed in the G54.2 mode:
   - Restarting the program
   - Mirror image (by G51.1 or control signal)
   - Scaling (G51)
   - Figure rotation (M98)
   - Coordinates rotation (G68)
   - G61.1, G61.2, G5P0, G5P2
8. The workpiece coordinates read with system variables include dynamic offsets.
9. The component vectors of the current dynamic offset can be read using system variables #5121 (X-axis), #5122 (Y-axis) and #5123 (Z-axis).

7. Related alarms

936 OPTION NOT FOUND
The dynamic offset II option is not installed.

959 WORKPIECE COORDINATE ERROR
The origin of workpiece coordinates does not match the center of rotation of the turntable.

807 ILLEGAL FORMAT
Argument P is missing in the block of G54.2.
An incompatible G-code is used in the G54.2 mode or G54.2 is given in the mode of an incompatible G-code.

809 ILLEGAL NUMBER INPUT
The value of P in the block of G54.2 is not proper.

8. Related parameters

A. Rotational axis configuration

Specify the type of rotational axis configuration of the machine to be operated.

L81 = 0: Makes the dynamic offsetting function invalid.
= 1: Two rotational axes (C-axis on A-axis)
= 2: One rotational axis (A-axis)
= 3: One rotational axis (C-axis)
= 4: One rotational axis (B-axis)

Specify “1” for the VARIAXIS series, and “4” for the FH/PFH series with an NC rotary table.

B. Dynamic offset type

Specify whether or not the tool is to be offset by each change only in the deviation vector.

F87 bit 0 = 0: Offset (the indication of both workpiece and machine coordinates changes.)
= 1: Not offset (no change in the position indication at all)

Normally set this parameter to “0”.

C. Center of table rotation

Specify the center of rotation of the table in machine coordinates. These parameters are also used in the VARIAXIS control for MAZATROL programs.

The preset values refer to the factory adjustment at Mazak.

S5 X, Y Center of rotation of the turntable (Machine coordinates)
S12 Y, Z Axis of rotation of the tilting table (Machine coordinates)
S11 Z Distance (length) from the tilting axis to the turntable surface
(The turntable center must be in the direction of –Z from the tilting axis.)

Note: When **L81** = 2, 3, or 4, the **S11** and **S12** settings are not required.

D. Workpiece origin mismatch check

The origin of the selected workpiece coordinate system must correspond to the center of table rotation in order that the dynamic offsetting may effectively function. The following parameter is provided to check the condition in question for each G54.2 command.

F87 bit 1 = 0: The mismatch check is conducted.
= 1: The mismatch check is not conducted.

Normally set this parameter to “0”.

Return to Library

9. Mechanical requirements

The dynamic offsetting function requires the following conditions to be satisfied:

1. The machine is equipped with a table of either two-axis rotational control (construction of a turntable on the tilting axis) or of a single rotational axis control (turntable or tilting table). The tilting and rotational axis must refer to rotating around the X- and Z-axis, respectively. Moreover, the construction must not be of the tilting axis mounted on the turntable.
2. The workpiece coordinate origin corresponds to the center of table rotation, and the X-, Y-, and Z-axes of workpiece coordinates are in parallel with, and the same direction as, the corresponding axes of machine coordinates.
3. The requirements for machining with table rotation: The machining contour is described using a workpiece coordinate system fixed in parallel with the machine coordinate system (not rotated with the table rotation) and microsegment command blocks of G1.

10. Operation description using a sample program

The following describes the operation using a sample program (created for explanation only).

A. Settings on the related displays

WORK OFFSET (G54) X = –315.0, Y = –315.0, Z = 0.0, A = 0.0, C = 0.0

DYNAMIC OFFSET (P1) X = –1.0, Y=0.0, Z = 0.0, A = 0.0, C = 90.0

Parameters **L81** = 1 (Rotational axis configuration: Two rotational axes; C-axis on A-axis)
F87 bit 0 = 0 (Dynamic offset type: Offset)
S5 X = –315000
S5 Y = –315000

B. Sample program (for explanation of operation)

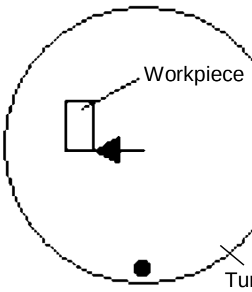
```
N1 G91 G28 X0 Y0 Z0 A0 C0
N2 G54
N3 G90 G00 X0 Y0 Z0 A0 C0
N4 G54.2P1
N5 G01 C180.0 F1000
N6 G01 X10.0
N7 G03 X0 Y10.0 R10.0
N8 G01 C240.0
```

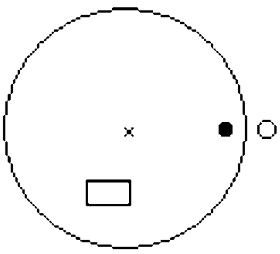
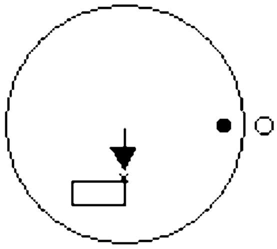
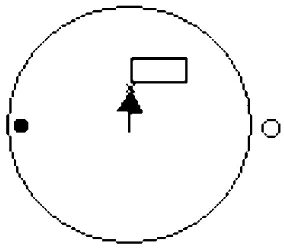
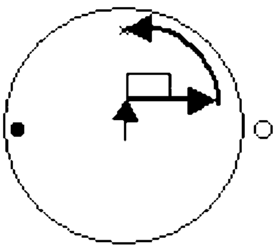
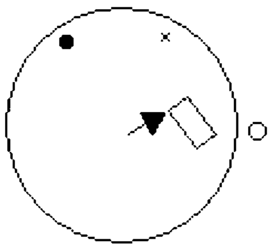
C. Position indication and dynamic offset for each line of the program

N-No.	POSITION (workpiece coordinates)					MACHINE (machine coordinates)					Dynamic offset		
	X	Y	Z	A	C	X	Y	Z	A	C	X	Y	Z
N1	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
N2	315.000	315.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
N3	0.000	0.000	0.000	0.000	0.000	–315.000	–315.000	0.000	0.000	0.000	0.000	0.000	0.000
N4	0.000	–1.000	0.000	0.000	0.000	0.000	–316.000	0.000	0.000	0.000	0.000	–1.000	0.000
N5	0.000	1.000	0.000	0.000	180.000	0.000	–314.000	0.000	0.000	180.000	0.000	1.000	0.000
N6	10.000	1.000	0.000	0.000	180.000	–305.000	–314.000	0.000	0.000	180.000	0.000	1.000	0.000
N7	0.000	11.000	0.000	0.000	180.000	–315.000	–304.000	0.000	0.000	180.000	0.000	1.000	0.000
N8	0.866	10.500	0.000	0.000	240.000	–314.134	–325.500	0.000	0.000	240.000	0.866	0.500	0.000

Return to Library

D. Illustration of the sample program

Measurement of the reference dynamic offset	
 	Let the position where the ● mark on the table is aligned with the fixed position marked with ○ be the zero point of the C-axis. The reference dynamic offset (arrow) = (–1, 0, 0) was measured with the table positioned at C = 90.0, as shown on the left.

N-No.	N3	N4	N5
Illustration	1 	2 	3 
N-No.	N6, N7	N8	N5
Illustration	4 	5 	

<Explanation>

1. N3 turns the table on the C-axis to ● (C = 0) and positions the tool tip to the × point (X, Y, Z = 0, 0, 0).
2. N4 causes the tool tip to be shifted by the dynamic offset (arrow) for an angular position of C = 0 to the × point (X, Y, Z = 0, –1, 0).
3. N5 turns the table on the C-axis to ● (C = 180) and causes the tool tip to be shifted by linear interpolation to the × point (X, Y, Z = 0, 1, 0) determined by the dynamic offset (arrow) for an angular position of C = 180.
4. N6 and N7 interpolate the linear and circular paths to the × point.
5. N8 turns the table on the C-axis to ● and causes the tool tip to be shifted by linear interpolation to the × point.

- NOTE -

25 EIA/ISO PROGRAM DISPLAY

This chapter describes general procedures for and notes on constructing an EIA/ISO program newly, and then editing functions.

25-1 Procedures for Constructing an EIA/ISO Program

(1) Press the display selector key.

(2) Press the **[PROGRAM]** menu key.

➔ The **PROGRAM** display will be selected.

(3) Press the **[WORK No.]** menu key.

➔ WORK No. is displayed in reverse to show the window of work number list.

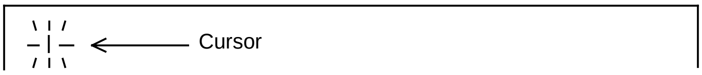
Remark: Refer to the Operating Manual for the window of work number list.

(4) Enter the new work number of a program to be constructed.

- Specifying a work number of a program registered already in NC unit allows the program to be displayed on the screen. Therefore, constructing a new program requires specifying a work number which has not been used. The conditions how work numbers are used are displayed on the window of work number list.

(5) Press the **[EIA/ISO PROGRAM]** menu key.

- Press the **[PROGRAM EDIT]** menu key instead of **[EIA/ISO PROGRAM]** if a work number of the program already registered has been set in Step (4).



(6) Enter the required programming data.

Set data using alphabetic keys, numeric keys and INPUT key .

- When INPUT key is pressed, the cursor is moved to the top of the next line, and then the data of the next block can be entered.

(7) Press the **[PROGRAM COMPLETE]** menu key to end the editing.

25-2 Editing Function of EIA/ISO PROGRAM Display

25-2-1 General

Establishing a constructing mode on the **PROGRAM (EIA/ISO)** display allows the following menu to be displayed as an initial one.

PROGRAM COMPLETE	SEARCH	COPY	ALTER	ERASE	MOVE	FIND & REPLACE	CHANGE PROGRAM	MACRO INPUT	MACRO VARIABLE
	[1]	[2]	[3]	[4]	[5]	[6]			

Terms [1] to [6] represent functions related to the program editing. Use of the functions permits the following operations:

- Inserting and altering data at any position
  Data can be inserted and altered at any position on the display.
- Erasing the data
  Data displayed on the display can be erased.
- Searching for the data
  Data can be searched in the following four ways.
  1) Searching for the top line of the program
  2) Searching for the bottom line of the program
  3) Searching for any required line of the program
  4) Searching for any character string
- Copying the data
  Other EIA/ISO programs registered in the NC unit can be copied into the selected program, or any data character string in the selected program can be copied into a given position of the selecting program or a new EIA/ISO program.
- Moving the data
  Any data character string can be moved to a given position of the selecting program or a new EIA/ISO program.
- Replacing the data
  Any data character string can be replaced by another character string.

25-2-2 Operation procedure

The procedure for each operation is described below.
(Given that EIA/ISO program, in which several lines of data are already provided, is selected, and editing mode is established, and also that ALTER menu item is not displayed in the reverse status in the operations 3 and onward.)

1. Inserting the data

(1) Press the **[ALTER]** menu key as required to obtain the display status ALTER.

- When ALTER is displayed, press the menu key to cancel the reverse-display status.

(2) Move the cursor to the position where data must be inserted.

- The cursor can be moved to any direction (vertically and horizontally).

(3) Enter the required data.

➔ Data is inserted in sequence into the position where the cursor is placed.

➔ Data previously set behind the cursor position are moved behind the inserted data.

2. Altering the data

(1) Press **[ALTER]** menu key to display ALTER.

- When ALTER is displayed, press the menu key to reverse the display status.

(2) Move the cursor to the position where data must be altered.

- The cursor can be moved to any direction (vertically and horizontally).

(3) Enter the required data.

➔ Data is altered in sequence from the position where the cursor is placed.

➔ The character previously set at the cursor position is replaced in sequence by the new data.

3. Erasing the data

(1) Move the cursor to the head of the character string to be erased.

(2) Press the **[ERASE]** menu key.

➔ The character at the cursor position is displayed in reverse and the **[ERASE]** menu item is also displayed in reverse.

(3) Move the cursor to the position next to the end of the character string to be erased.

➔ The portion from the head of the character string specified in (1) to the position before the cursor is displayed in reverse, which indicates that the reversed portion provides the object of erasure.

Example:

```
N001  G00 X10. Z10.;
      G00 X100.
      G00 Z20.          Cursor position
N002  M08               in (1)
      M03               Cursor
```

(4) Press the input key.

➔ The character string displayed in reverse in (3) is erased.

Example:

```
N001  G00 X10.
N002  M08
      M03
```

4. Searching for the data

A. Searching for the top line of the program

(1) Press the **[SEARCH]** menu key.

(2) Press the **[PROG HEAD]** menu key.

➔ The cursor moves to the top line.

B. Searching for the bottom line of the program

(1) Press the **[SEARCH]** menu key.

(2) Press the **[PROG END]** menu key.

➔ The cursor moves to the bottom line.

C. Searching for any required line of the program

(1) Press the **[SEARCH]** menu key.

(2) Press the **[SEARCH LINE No.]** menu key.

➔ SEARCH LINE No. is displayed in reverse.

(3) Set the line No. of the line to be searched for.

- Enter the line No. with numeric keys, and press the input key.

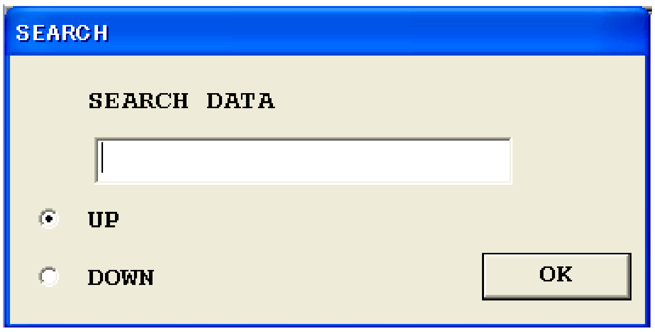
➔ The cursor moves to the specified line.

D. Searching for any character string

(1) Press the **[SEARCH]** menu key.

(2) Press the **[SEARCH FORWARD]** menu key or **[SEARCH BACKWARD]** menu key.

➔ SEARCH FORWARD or SEARCH BACKWARD is displayed in reverse.



- To search for a character string in the area before the cursor position, press the **[SEARCH FORWARD]** menu key, and for the area after the cursor position, press **[SEARCH BACKWARD]** menu key.

(3) Set the character string to be searched for and press the input key.

➔ The cursor moves to the head of the character string which has been found first.

- Press the data cancellation key (CANCEL) to stop halfway the searching operation, whose running state is indicated by the message **CNC BUSY** on the display.

Remark: Pressing the input key in sequence allows the cursor to move to the character string which has been found next.

5. Copying the data

A. Copying a program

(1) Move the cursor to the position where the program is to be copied.

(2) Press the **[COPY]** menu key.

(3) Press the **[PROGRAM COPY]** menu key.

➔ The window of work number list is displayed and the **[PROGRAM COPY]** menu item is displayed in reverse.

(4) Set the work number of the program to be copied and press the input key.

➔ The program is inserted into the cursor position.

Note: MAZATROL programs cannot be copied.

B. Copying any character string into the selected program

(1) Move the cursor to the head of the character string to be copied.

(2) Press the **[COPY]** menu key.

(3) Press the **[LINE(S) COPY]** menu key.

➔ The character at the cursor position is displayed in reverse and the **[LINE(S) COPY]** menu item is also displayed in reverse.

(4) Move the cursor to the position next to the end of the character string to be copied.

➔ The portion from the head of the character string specified in (1) to the position before the cursor is displayed in reverse, which indicates that the reversed portion provides the object of copying.

Example:

```
N001    G00 X10. Z10.
        G00 X100.
        G00 Z2
                          Cursor position in (1)
N002  M08
      M03         Cursor
```

(5) Press the input key.

➔ The area displayed in reverse is established as the object to be copied.

(6) Move the cursor to the position where the character string is to be copied.

➔ The cursor only moves, and the area displayed in reverse does not change.

Example:

```
N001   G00 X10. Z10.
       G00 X100.
       G00 Z20.
N002   M08
       M03
                Cursor
```

(7) Press the input key.

➔ The character string displayed in reverse is copied at the cursor position.

Example: (Continued)

```
N001  G00 X10.Z10.
      G00 X100.
      G00 Z20.
N002  M08
Z10.
      G00 X100.
      G00 Z20.
      M03
```

C. Copying any character string into a new program

(6) First, carry out Steps (1) to (5) of **B**. Set the workpiece number of a new program where the character string is to be copied and press the input key.

➔ The character string is copied in the new program, and the area displayed in reverse is returned to normal display.

Remark: Pressing the **[PROGRAM FILE]** menu key allows the window of program list to be displayed.

6. Moving the data

A. Moving the selected program to any position

(1) Move the cursor to the head of the character string to be moved.

(2) Press the **[MOVE]** menu key.

➔ The character at the cursor position and the **[MOVE]** menu item is also displayed in reverse.

(3) Move the cursor to the position next to the end of the character string to be moved.

➔ The portion from the head of the character string specified in (1) to the position before the cursor is displayed in reverse, which indicates that the reversed portion provides the object of moving.

```
N001   G00 X10. Z10.
       G00 X100.
       G00 Z20.

N002   M08
       M03
```

Cursor position in (1)

Cursor

(4) Press the input key.

➔ The area displayed in reverse is established as the object to be moved.

(5) Move the cursor to the position where the character string is to be moved.

- The cursor only moves, and the area displayed in reverse does not change.

Example: (Continued)

```
N001   G00 X10. Z10.
       G00 X100.
       G00 Z20.
N002   M08
       M03
```

Cursor

(6) Press the input key.

➔ The character string displayed in reverse is moved to the cursor position.

Example: (Continued)

```
N001  G00 X10.
N002  M08
Z10.
      G00 X100.
      G00 Z20. M03
```

B. Movement to a new program

(5) First, carry out Steps (1) to (4) of **A**. Set the work number of a new program where the character string is to be moved and press the input key.

➔ The character string is moved to the new program.

Remark: Pressing the **[PROGRAM FILE]** menu key allows the window of program list to be displayed.

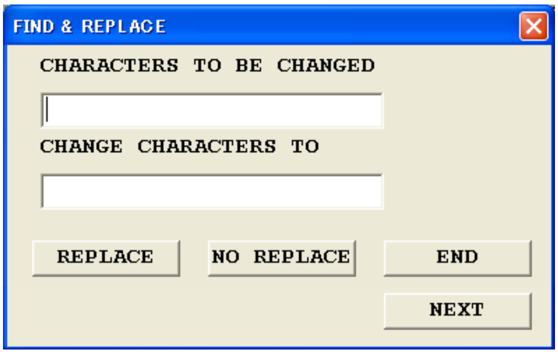
7. Replacing the data

(1) Move the cursor to the starting position of data replacement.

- Replacement is made downward from the cursor position. To make replacement throughout the program, therefore, move the cursor to the first character of the top line.

(2) Press the **[FIND & REPLACE]** menu key.

➔ **FIND & REPLACE** is displayed in reverse.



(3) Set the character string before replacement.

- Enter the character string to be replaced using alphanumeric keys, and press the tab key .

(4) Set the new character string after replacement using alphanumeric keys, and press the input key.

➔ The cursor moves to the head of the character string before replacement that has been found first after the cursor position specified in (1).

(5) Press the **[REPLACE]** menu key.

➔ The character string before replacement at the cursor position is replaced by the character string after replacement, and the cursor moves to the head of the next character string before replacement. Pressing the **[REPLACE]** menu key in sequence allows the character string before replacement to be replaced in order of being found.

When replacing the special character string at the cursor position is not required, press the **[NO REPLACE]** menu key in place of **[REPLACE]** menu key.

Remark 1: To stop the replacement, press the **[END]** menu key.

Remark 2: To replace all the character strings in the program, press the **[NEXT]** menu key.

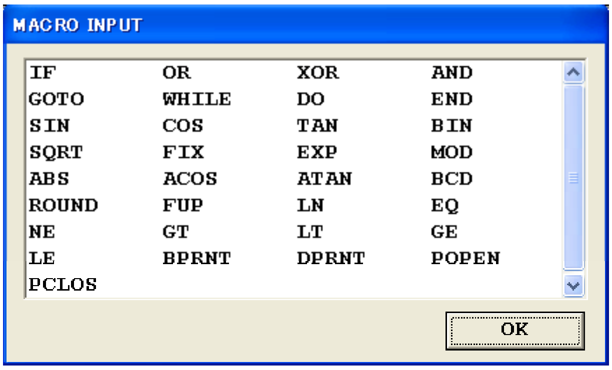
Remark 3: Press the data cancellation key (CANCEL) to stop halfway the total replacement by the NEXT menu function, whose running state is indicated by the message **CNC BUSY** on the display.

25-3 Macro-Instruction Input

This function permits entering the macro-instruction word by word for editing the EIA/ISO program efficiently.

(1) Press the **[MACRO INPUT]** menu key.

➔ The **MACRO INPUT** window will be opened.



- The character string selected with the cursor is usable.

(2) Move the cursor to the characters corresponding to the required macro-instruction and press the input key.

➔ The macro-instruction is entered in the editing zone of the program.

(3) Press the menu selector key to display the menu for normal data input, and continue program editing.

Return to Library

25-4 Division of Display (Split Screen)

1. Dividing the screen (vertically)

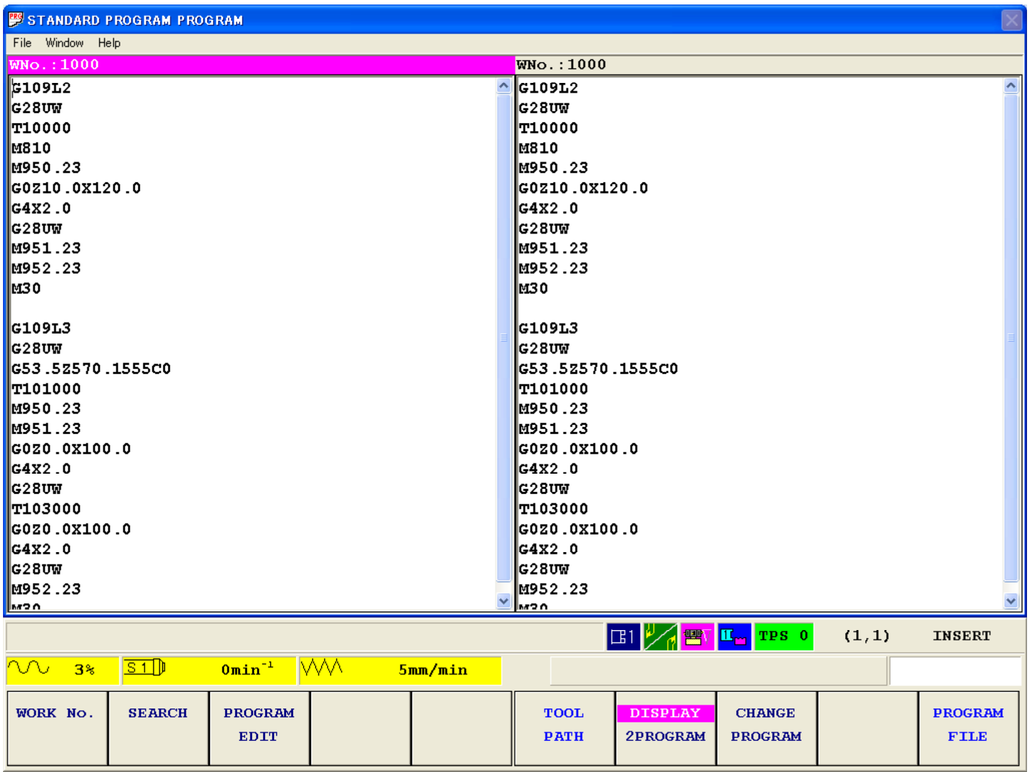
(1) Temporarily cancel the editing mode, if selected, by pressing the **[PROGRAM COMPLETE]** menu key.

(2) Press the **[DISPLAY 2 PROGRAM]** menu key.

➔ The display of the menu item will be highlighted and the work number listing window will appear.

(3) Select the work number of the program to be displayed.

➔ The screen will be divided into the left and right part. One and the same section of the program is initially displayed in both parts.



D740PB002E

- The editing operation can only be carried out in the part the title (WNo.) of which is highlighted.
- The display contents in the other part will remain unchanged even after the editing in the active part. Press the **[CHANGE PROGRAM]** menu key to change the display in the other part according to the editing operation.

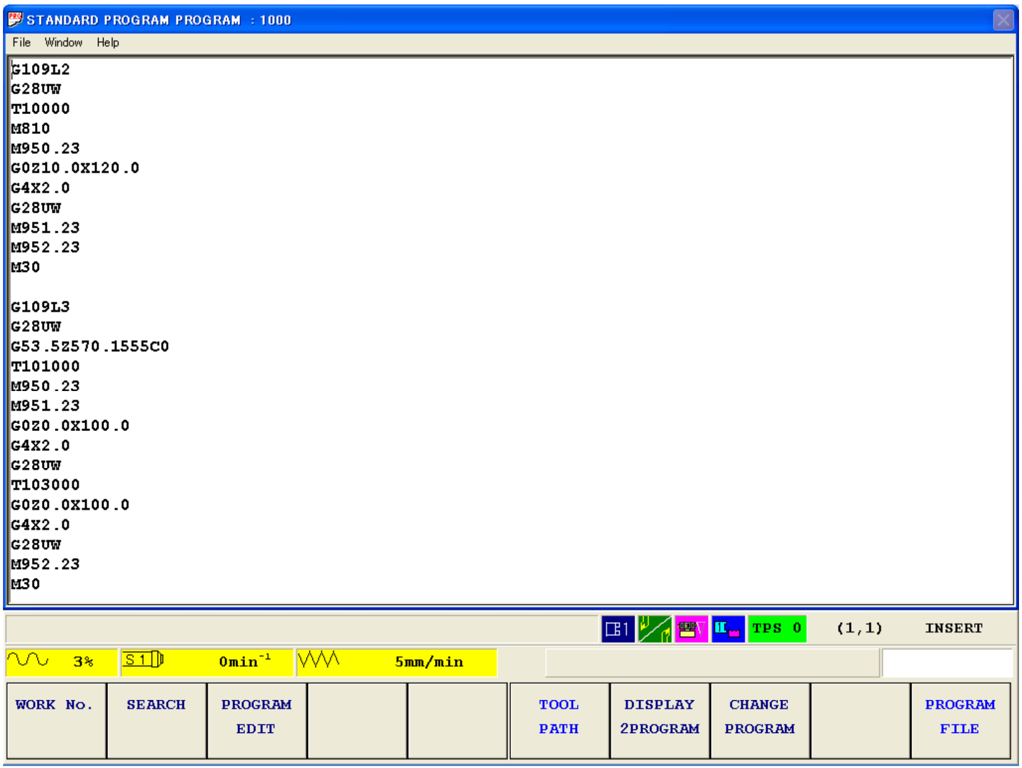
Return to Library

2. Cancelling the division

(1) Temporarily cancel the editing mode, if selected, by pressing the **[PROGRAM COMPLETE]** menu key.

(2) Press anew the **[DISPLAY 2 PROGRAM]** menu key.

➔ The highlighted display of the menu item will be released and the division of the screen cancelled.



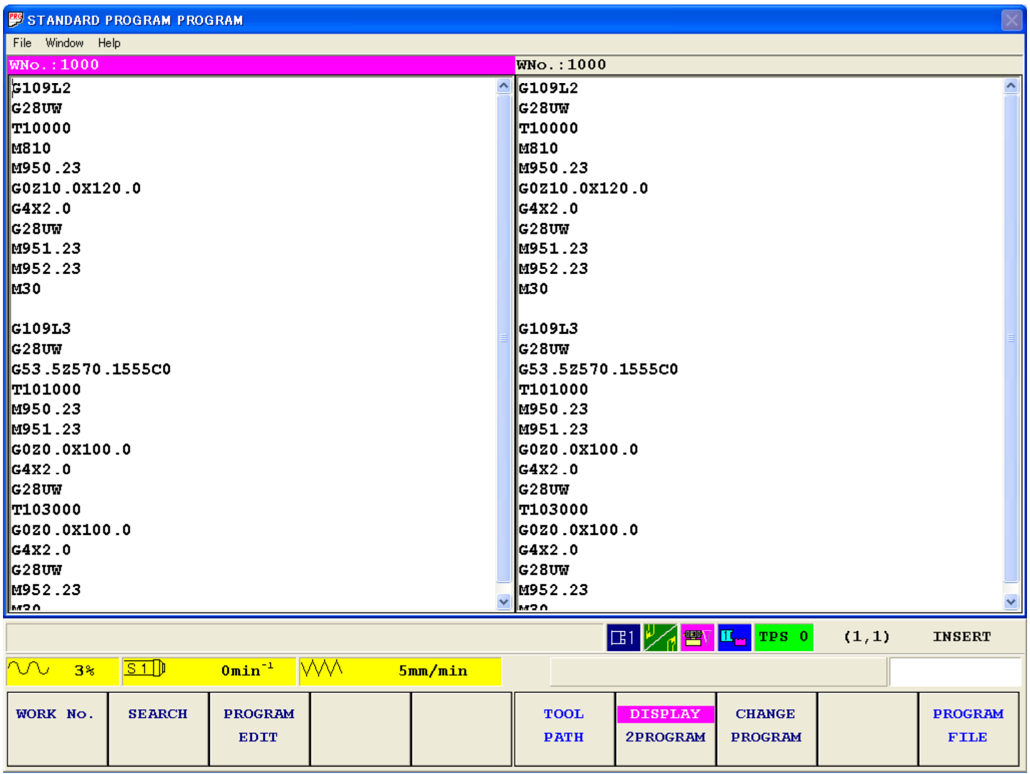
D740PB003E

3. Changing the active part

The editing is only possible for the part whose title (WNo.) is currently highlighted. The method to change the active part is indicated below.

The data after the editing will not be displayed in the other part (of the same WNo.) unless this changing operation is carried out.

In the example below, the left-hand part is currently active.

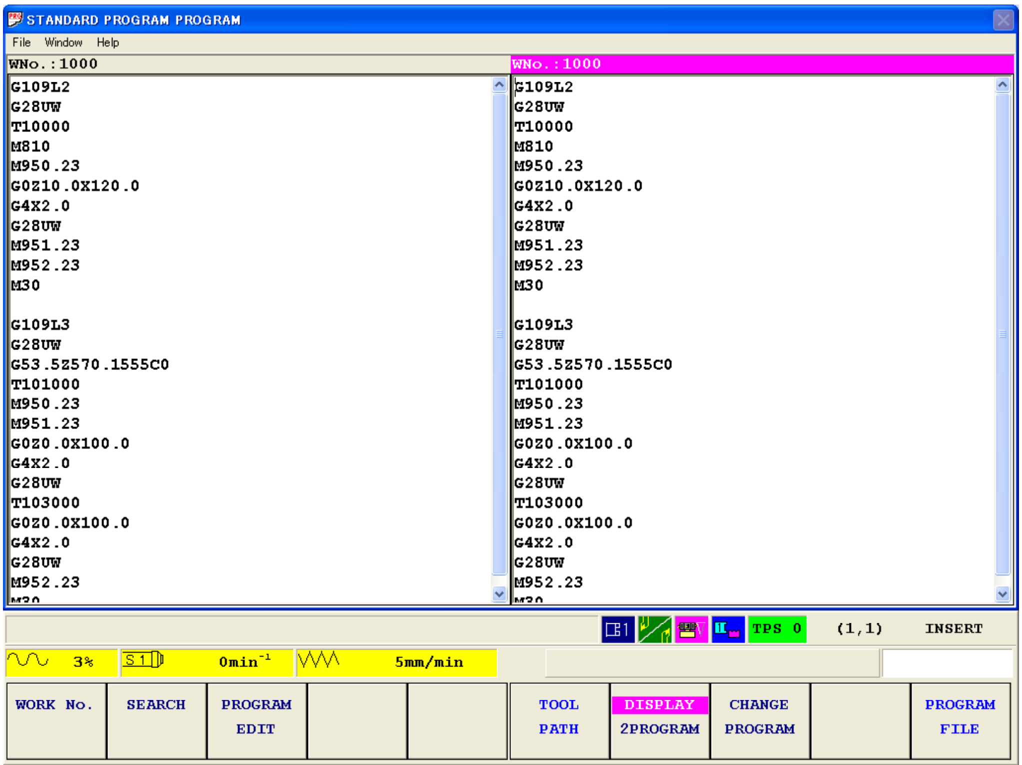


D740PB002E

(1) Press the **[CHANGE PROGRAM]** menu key.

➔ The highlighting of the title will be transferred from the left-hand to the right-hand part to indicate that the latter has been made active.

- The contents in the right-hand part will have been modified at the same time according to the editing operation performed for the left-hand part (of the same WNo.).

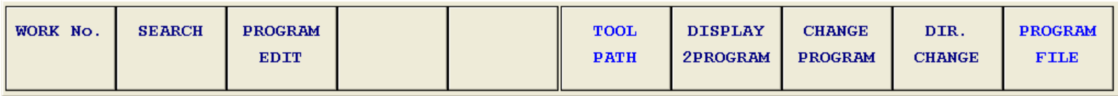


D740PB004E

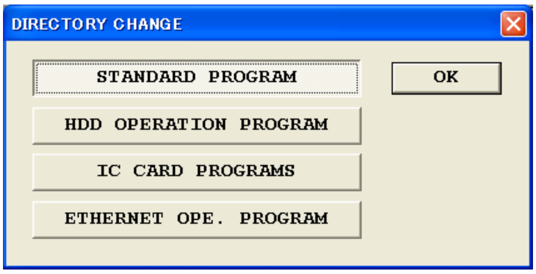
25-5 Editing Programs Stored in External Memory Areas

Follow the procedure below to edit machining programs (to be used for Hard Disk, IC Memory Card, and Ethernet operation) which are created in the EIA/ISO format and stored in external memory areas. The functions for IC Memory Card and Ethernet operation, however, are optional.

(1) Select **[DIR. CHANGE]** from the initial menu of the **PROGRAM** display for EIA/ISO programs.



➔ The menu item is highlighted and the **DIRECTORY CHANGE** window appears on the screen.



- The options **IC CARD PROGRAMS** and **ETHERNET OPE. PROGRAM** will only be presented for machines equipped with the corresponding optional functions.

(2) Use the mouse, or the cursor keys, to select the desired storage area.

(3) Click the **[OK]** button, or press the INPUT key.

➔ With a memory area other than that of STANDARD PROGRAM being selected, the color of the background of the **PROGRAM** display changes to yellow. Follow the same creating and editing procedure, however, as for programs in the STANDARD PROGRAM area to prepare a new program, or edit an existing one, for the selected memory area.

- The area selection made from this window will be maintained till turning off the NC power.
- The title bar displays the current selection of the memory area.