Return to Main Menu

PROGRAMMING MANUAL

for

ALL MAZATROL TURNING CNC (including T-Plus)
MAZATROL FUSION 640T NEXUS

Programming EIA/ISO

MANUAL No. : C734PBT018E

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/Service 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/Service 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/Service 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/Service Center with the manual No. or the machine name, serial No. and manual name.**

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

08. 2002

Return to Main Menu

Return to Main Menu

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. 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 Main Menu

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.

Return to Main Menu

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. (This precaution applies only to the M640T, M640MT, M640T NEXUS and M640M Pro.)
- The system of G-code and M-code commands differs between the machines equipped with M640M Pro (e-Series such as the INTGEREX e-410, e-650 and e-1060) and the machines equipped with M640MT/T/T NEXUS (such as the INTGEREX non e-Series, the SQT Series, the MPX Series and the QTN Series).

  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	Machine with M640M Pro	Machine with M640MT/T/T NEXUS
`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}.

Return to Main Menu

- For the machines equipped with M640M Pro (e-Series such as the INTGEREX e-410, e-650 and e-1060), 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).

⚠ 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.

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 100 times and the handle operated inadvertently, axial movement will become faster than that expected.

Return to Main Menu

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: 5° to 40°C (41° to 104°F)

Note: When power is turned on, if the thermal sensor detects an ambient temperature under 5°C, the hard disk warm-up status indicator lamp will light up and the NC unit will not start operating at once. After automatic heating of the hard disk by its internal heater, the lamp will go out and the NC unit will start. It takes about 20 minutes for temperature to increase from 0 to 5°C in order to avoid condensation due to sudden changes in temperature.

2. Relative humidity

During machine operation: 30 to 75 % (without bedewing)

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

- NOTE -

CONTENTS

Return to Main Menu

Return to Main Menu

Return to Main Menu

Return to Main Menu

Return to Main Menu

- NOTE -

1 INTRODUCTION

EIA/ISO programs executed by the CNC unit include two modes: One is T32 compatible mode which is host-compatible with MAZATROL T32, and the other is standard mode which is compatible with standard EIA/ISO. The selection of mode can be made by changing the related parameter as described below.

This manual gives descriptions with G code series system A of the T32 compatible mode unless otherwise noted. The descriptions in this general-purpose manual may not all apply to the respective machine model (refer to the machine specifications).

P16 bit 3 = 0: T32 compatible mode
= 1: Standard EIA/ISO mode

Return to Main Menu

- NOTE -

Return to Main Menu

2 UNITS OF PROGRAM DATA INPUT

2-1 Overview

1. Units of Program Data Input

The distances through which the axes are to be moved in a program must be designated using a tape. The distance data is expressed in millimeters, inches, or degrees.

2. Units of Data Setting

Various data, such as offsetting data, must be set for the machine to perform an operation as desired. Data that you set will be shared by each axis.

2-2 Detailed description

The units of data setting for each axis and those of program data input shared by each axis are listed below and selected by parameters. (For details of data settings, see the operating manual.)

	Linear axis				Rotational axis (deg)
	Millimeter		Inch		
	Diametrical command	Radial command	Diametrical command	Radial command	
Units of program data input	0.001	0.001	0.0001	0.0001	0.001
Units of minimal movement	0.0005	0.001	0.00005	0.0001	0.001
Units of data setting	0.001	0.001	0.0001	0.0001	0.001

Note 1: Inch/metric selection can be freely made using parameter **P19** or G-code commands (G20, G21).
Parameter setting is validated through power-off and -on.
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 included together in the same program.

Return to Main Menu

- 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 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.

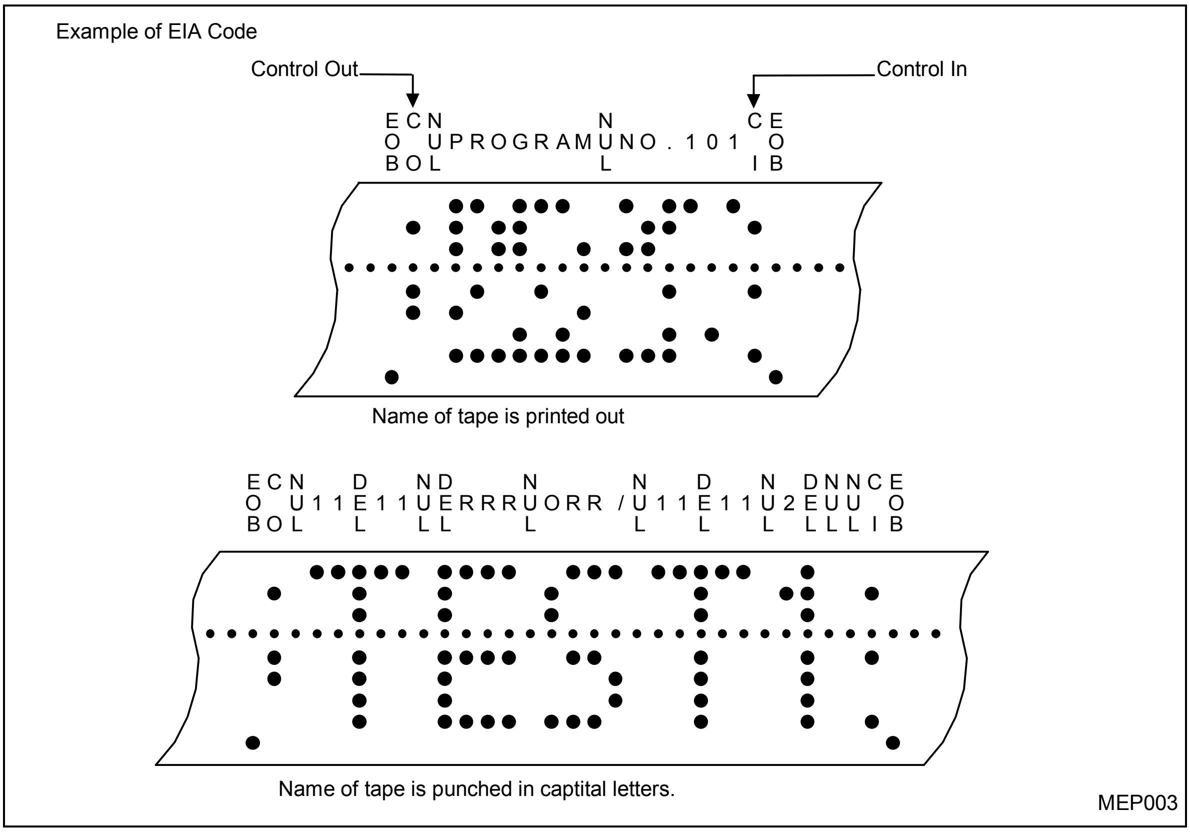
Note 3: EIA/ISO code identification is made automatically according to the first EOB/LF code appearing after the NC unit has been reset. (EOB: End Of Block, LF: Line Feed)

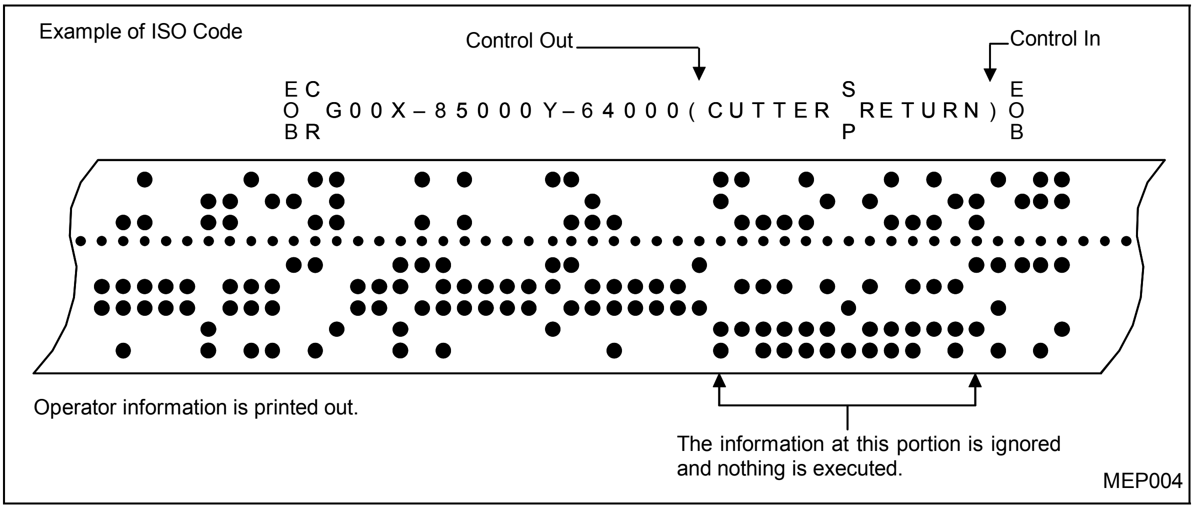
1. Significant information area (LABEL SKIP function)

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.


Example of EIA Code
Control Out
Control In
EOB CR NUL UPROGRAMNUL NO.101 CI EOB
Name of tape is printed out
EOB CR NUL U11 DEL E11 NUL U DEL E R R R NUL U O R R / NUL U11 DEL E11 NUL U2 DEL E NUL U NUL U CI EOB
Name of tape is punched in captital letters.
MEP003


Example of ISO Code
Control Out
Control In
EOB CR G00X−85000Y−64000(CUTTER SP RETURN) EOB
Operator information is printed out.
The information at this portion is ignored and nothing is executed.
MEP004

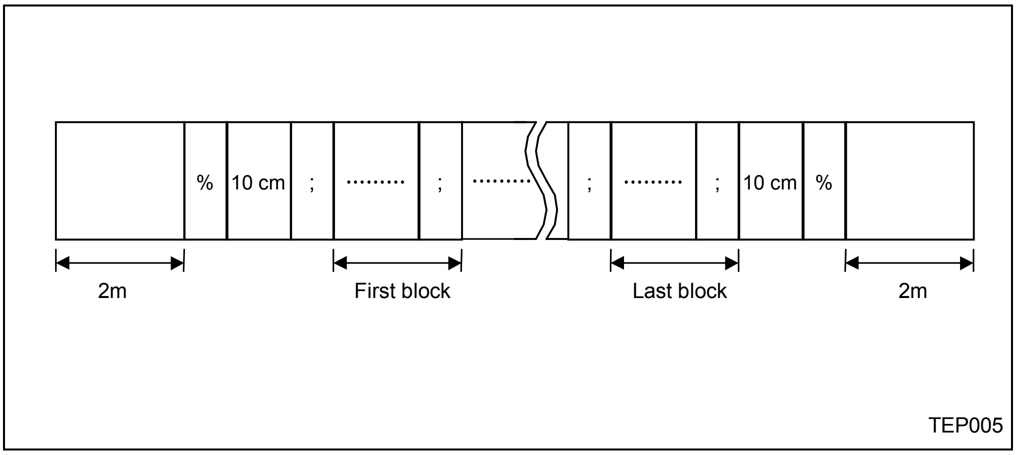
3. EOR code (%)

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

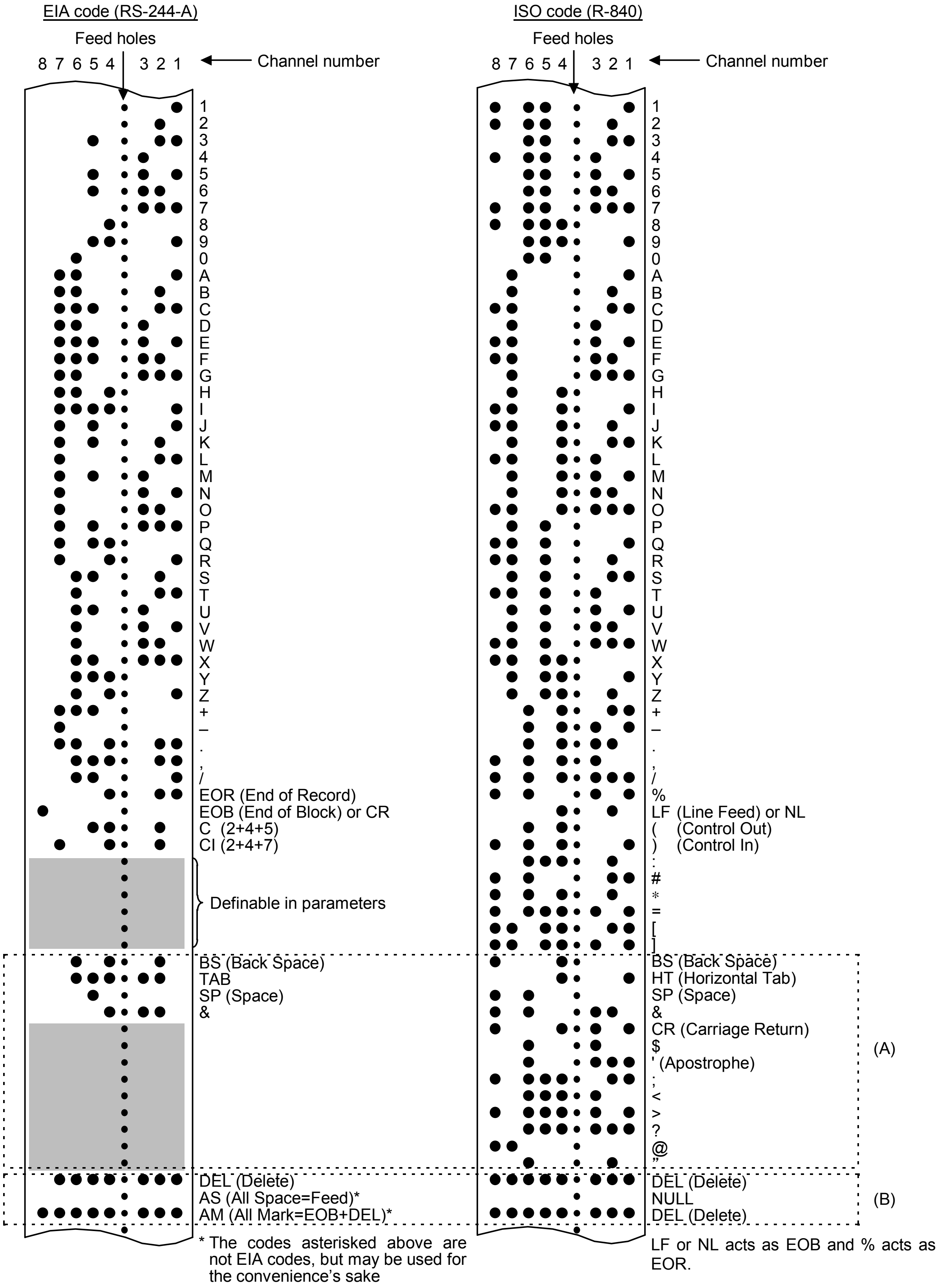
- To stop rewinding (only when a rewinding device is provided)
- To start rewinding during tape data search (only when a rewinding device is provided)
- To terminate the storage of tape data.

Return to Main Menu

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 not used.



Codes is Section (A) above will only be stored onto the tape when they are present in the comment field.
Codes is Section (B) above, which consists of no-operation codes, will always be ignored (but undergo the parity-V check).
For dotted sections, corresponding EIA standard codes do not exist.

MEP006

Fig. 3-1 Tape codes

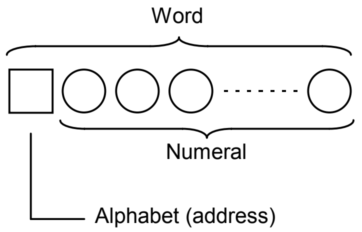
Return to Main Menu

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.

An outline of format details

Item		Metric command	Inch command
Program No.		O8	
Sequence No.		N5	
Preparatory function		G3 or G21	
Moving axis	Input unit: 0.001 mm (deg.), 0.0001 inch	X+53 Y+53 Z+53 α+53	X+44 Y+44 Z+44 α+44
Movement command, arc, cutter radius	Input unit: 0.001 mm (deg.), 0.0001 inch	I+53 J+53 K+53	I+44 J+44 K+44
Dwell	Input unit: 0.001 mm (deg.), 0.0001 inch	X53 P8 U53	
Feed	Input unit: 0.001 mm (deg.), 0.0001 inch	F53 (per minute) F33 (per revolution)	F44 (per minute) F24 (per revolution)
Fixed cycle	Input unit: 0.001 mm (deg.), 0.0001 inch	R+53 Q53 P8 L4	R+44 Q44 P8 L4
Tool offset		T1 or T2	
Miscellaneous function		M3 × 4	
Spindle function		S5	
Tool function		T4 or T6	
No. 2 miscellaneous function		A8, B8 or C8	
Subprogram call		P4 Q5 L4	
Variables number		#5	

1. O8 here indicates that the program number is set using an unsigned integer of eight digits following O and for X+53, "+" indicates that the value can be signed (negative) and the two-digit number (53) indicates that the decimal point can be used and that five digits before and three after the decimal point are effective (5 + 3 = 8 digits are effective for a designation without decimal point).

2. The alpha sign (α) denotes additional axis address. + 43 will be used when α is specified for rotational axis.

Return to Main Menu

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, M198, M199 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 G90 X120. Z100.;`	1234	100	0
`G94 S1000;`	1234	100	1
`N102 G71 P210 Q220 I0.2 K0.2 D0.5 F600;`	1234	102	0
`N200 G94 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

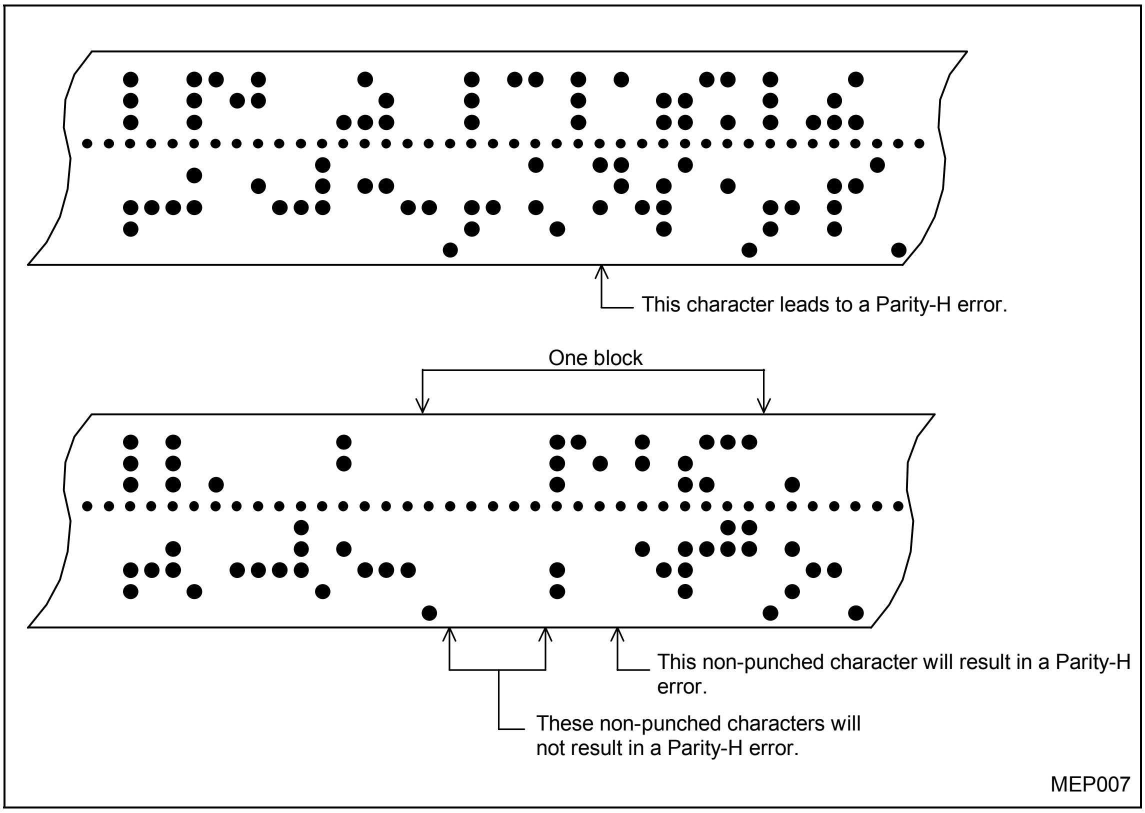
One method of checking if the tape is correctly created is by parity checks. Parity checks are performed to check a tape for errors in punched codes, that is, for punching errors. There are two types of parity checks: parity-H and parity-V.

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)

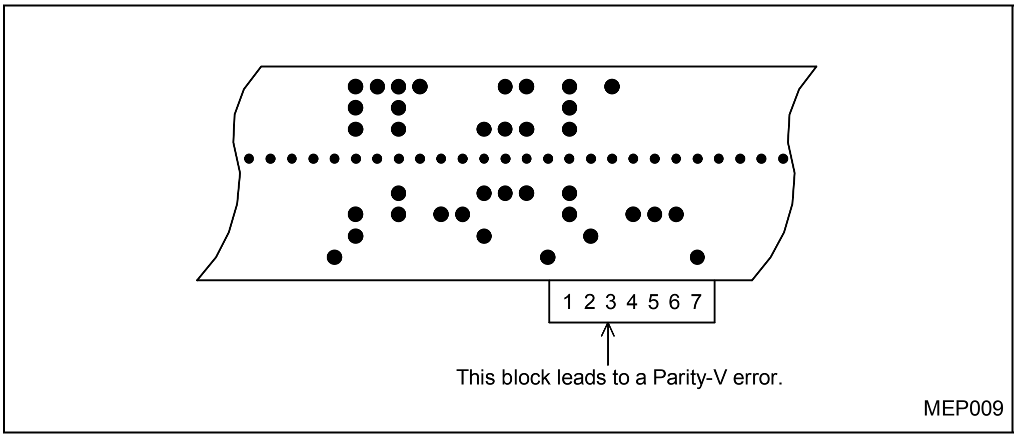

This character leads to a Parity-H error.
One block
This non-punched character will result in a Parity-H error.
These non-punched characters will not result in a Parity-H error.
MEP007

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 3: 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.

Return to Main Menu

3-7 List of G-Codes

G functions are described in the list below.

Function	T32 compatible mode G-code series			Standard mode G-code series			Group
	A	B	C	A	B	C	
Positioning	■G00	■G00	■G00	■G00	■G00	■G00	01
Linear interpolation	■G01	■G01	■G01	■G01	■G01	■G01	01
Threading with C-axis interpolation	G01.1	G01.1	G01.1	G01.1	G01.1	G01.1	01
Circular interpolation CW	G02	G02	G02	G02	G02	G02	01
Circular interpolation CCW	G03	G03	G03	G03	G03	G03	01
Dwell	G04	G04	G04	G04	G04	G04	00
Cylindrical interpolation	–	–	–	G07.1	G07.1	G07.1	19
Exact stop check	G09	G09	G09	G09	G09	G09	00
Data setting	G10	G10	G10	G10	G10	G10	00
Data setting cancel	G11	G11	G11	G11	G11	G11	00
Polar coordinate interpolation	–	–	–	G12.1	G12.1	G12.1	19
Polar coordinate interpolation cancel	–	–	–	●G13.1	●G13.1	●G13.1	19
Milling mode	G12.1	G12.1	G12.1	–	–	–	19
Milling mode cancel	●G13.1	●G13.1	●G13.1	–	–	–	19
Y-Z cylindrical plane selection	G16	G16	G16	–	–	–	02
X-Y plane selection	G17	G17	G17	G17	G17	G17	02
Z-X plane selection	●G18	●G18	●G18	●G18	●G18	●G18	02
Y-Z plane selection	G19	G19	G19	G19	G19	G19	02
Inch command	■G20	■G20	■G70	■G20	■G20	■G70	06
Metric command	■G21	■G21	■G71	■G21	■G21	■G71	06
Stored stroke ON	G22	G22	G22	G22	G22	G22	04
Stored stroke OFF	●G23	●G23	●G23	●G23	●G23	●G23	04
Reference point return check	G27	G27	G27	G27	G27	G27	00
Reference point return	G28	G28	G28	G28	G28	G28	00
Return from reference point	G29	G29	G29	G29	G29	G29	00
Return to 2nd, 3rd and 4th reference points	G30	G30	G30	G30	G30	G30	00
Return to floating reference point	G30.1	G30.1	G30.1	G30.1	G30.1	G30.1	00
Skip function	G31	G31	G31	G31	G31	G31	00
Thread cutting (straight, taper)	G32	G33	G33	G32	G33	G33	01
Variable lead thread cutting	G34	G34	G34	G34	G34	G34	01
Automatic tool compensation X	–	–	–	G36	G36	G36	00
Automatic tool compensation Z	–	–	–	G37	G37	G37	00
Preparation for measurement	G36	G36	G36	G36.5	G36.5	G36.5	00
Measurement computing	G37	G37	G37	G37.5	G37.5	G37.5	00
Tool nose radius compensation cancel	●G40	●G40	●G40	●G40	●G40	●G40	07
Tool nose radius compensation left	G41	G41	G41	G41	G41	G41	07
Tool nose radius compensation right	G42	G42	G42	G42	G42	G42	07
Tool nose radius compensation (automatic direction selection) ON	G46	G46	G46	G46	G46	G46	07
Coordinate system setting/spindle clamp speed setting	G50	G92	G92	G50	G92	G92	00
Polygonal machining cancel	●G50.2	●G50.2	●G50.2	●G50.2	●G50.2	●G50.2	23

Return to Main Menu

Function	T32 compatible mode G-code series			Standard mode G-code series			Group
	A	B	C	A	B	C	
Polygonal machining	G51.2	G51.2	G51.2	G51.2	G51.2	G51.2	23
Local coordinate system selection	–	–	–	G52	G52	G52	00
Machine coordinate system selection	–	–	–	G53	G53	G53	00
MAZATROL coordinate system cancel	■G52	■G52	■G52	■G52.5	■G52.5	■G52.5	00
MAZATROL coordinate system selection	■G53	■G53	■G53	■G53.5	■G53.5	■G53.5	00
Workpiece coordinate system 1 selection	●G54	●G54	●G54	●G54	●G54	●G54	12
Workpiece coordinate system 2 selection	G55	G55	G55	G55	G55	G55	12
Workpiece coordinate system 3 selection	G56	G56	G56	G56	G56	G56	12
Workpiece coordinate system 4 selection	G57	G57	G57	G57	G57	G57	12
Workpiece coordinate system 5 selection	G58	G58	G58	G58	G58	G58	12
Workpiece coordinate system 6 selection	G59	G59	G59	G59	G59	G59	12
Exact stop mode	G61	G61	G61	G61	G61	G61	13
Geometry compensation	G61.1	G61.1	G61.1	G61.1	G61.1	G61.1	13
Automatic corner override	G62	G62	G62	G62	G62	G62	13
Cutting mode	●G64	●G64	●G64	●G64	●G64	●G64	13
Macroprogram call	G65	G65	G65	G65	G65	G65	00
Macro modal call	G66	G66	G66	G66	G66	G66	14
Macro modal call cancel	●G67	●G67	●G67	●G67	●G67	●G67	14
Opposite turret mirror image ON	G68	G68	G68	G68	G68	G68	15
Opposite turret mirror image OFF	●G69	●G69	●G69	●G69	●G69	●G69	15
Program coordinate system rotation ON	G68.5	G68.5	G68.5	G68.5	G68.5	G68.5	16
Program coordinate system rotation OFF	●G69.5	●G69.5	●G69.5	●G69.5	●G69.5	●G69.5	16
Finishing cycle	G70	G70	G70	G70	G70	G72	09
Longitudinal roughing cycle	G71	G71	G73	G71	G71	G73	09
End face roughing cycle	G72	G72	G74	G72	G72	G74	09
Forming material roughing cycle	G73	G73	G75	G73	G73	G75	09
End face cut-off cycle	G74	G74	G76	G74	G74	G76	09
Longitudinal cut-off cycle	G75	G75	G77	G75	G75	G77	09
Compound thread-cutting cycle	G76	G76	G78	G76	G76	G78	09
Drilling cycle cancel	●G80	●G80	●G80	●G80	●G80	●G80	09
Front drill cycle	G83	G83	G83	G83	G83	G83	09
Front tap cycle	G84	G84	G84	G84	G84	G84	09
Front synchronous tap cycle	G84.2	G84.2	G84.2	G84.2	G84.2	G84.2	09
Front boring cycle	G85	G85	G85	G85	G85	G85	09
Side drill cycle	G87	G87	G87	G87	G87	G87	09
Side tap cycle	G88	G88	G88	G88	G88	G88	09
Side synchronous tap cycle	G88.2	G88.2	G88.2	G88.2	G88.2	G88.2	09
Side boring cycle	G89	G89	G89	G89	G89	G89	09
Outside/inside diameter cutting cycle	G90	G77	G20	G90	G77	G20	09
Thread cutting cycle	G92	G78	G21	G92	G78	G21	09
End face cutting cycle	G94	G79	G24	G94	G79	G24	09
Constant peripheral speed control	■G96	■G96	■G96	■G96	■G96	■G96	17
Constant peripheral speed control cancel	■G97	■G97	■G97	■G97	■G97	■G97	17
Feed per minute (asynchronous)	■G98	■G94	■G94	■G98	■G94	■G94	05
Feed per revolution (synchronous)	■G99	■G95	■G95	■G99	■G95	■G95	05
Absolute command	–	■G90	■G90	–	■G90	■G90	03
Incremental command	–	■G91	■G91	–	■G91	■G91	03
Return to initial level	–	●G98	●G98	–	●G98	●G98	10

Function	T32 compatible mode G-code series			Standard mode G-code series			Group
	A	B	C	A	B	C	
Return to R-point level	–	G99	G99	–	G99	G99	10
2 process 1 program	G109	G109	G109	G109	G109	G109	00
Cross machining control axis selection	G110	G110	G110	G110	G110	G110	00
Cross machining control axis selection cancel	G111	G111	G111	G111	G111	G111	00
M, S, T, B output command to opposite system	G112	G112	G112	G112	G112	G112	00
Hob milling mode cancel	G113	G113	G113	G113	G113	G113	23
Hob milling mode	G114.3	G114.3	G114.3	G114.3	G114.3	G114.3	23
Swiss type machining mode	G120	G120	G120	G120	G120	G120	20
Swiss type machining mode cancel	●G121	●G121	●G121	●G121	●G121	●G121	20
Polar coordinate command ON	G122	G122	G122	G122	G122	G122	21
X-axis radial command ON	G122.1	G122.1	G122.1	G122.1	G122.1	G122.1	25
Polar coordinate command OFF	●G123	●G123	●G123	●G123	●G123	●G123	21
X-axis radial command OFF	●G123.1	●G123.1	●G123.1	●G123.1	●G123.1	●G123.1	25

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. T32 compatible mode G-code series or standard mode G-code series can be selected by the parameter **P16** bit 3.

   P16 bit 3 = 0 : T32 compatible mode G-code series is selected.
   1 : Standard mode G-code series is selected.
4. G-codes of group 00 are those which are not modal, and they are valid only for commanded blocks.
5. If a G-code not given in the G-code list is commanded, an alarm is displayed (708 “ILLEGAL G-CODE”). And if a G-code without corresponding option is commanded, an alarm is displayed (708 “ILLEGAL G-CODE”).
6. 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.
7. Standard G-code (system A)/special G-code (system B)/special G-code (system C) are changed by user parameter **P9** bits 2 and 3.

User parameter **P9**		Selective G-code system
Bit 2	Bit 3	
0	0	B
1	0	A
0	1	B
1	1	C

Return to Main Menu

4 BUFFER REGISTERS

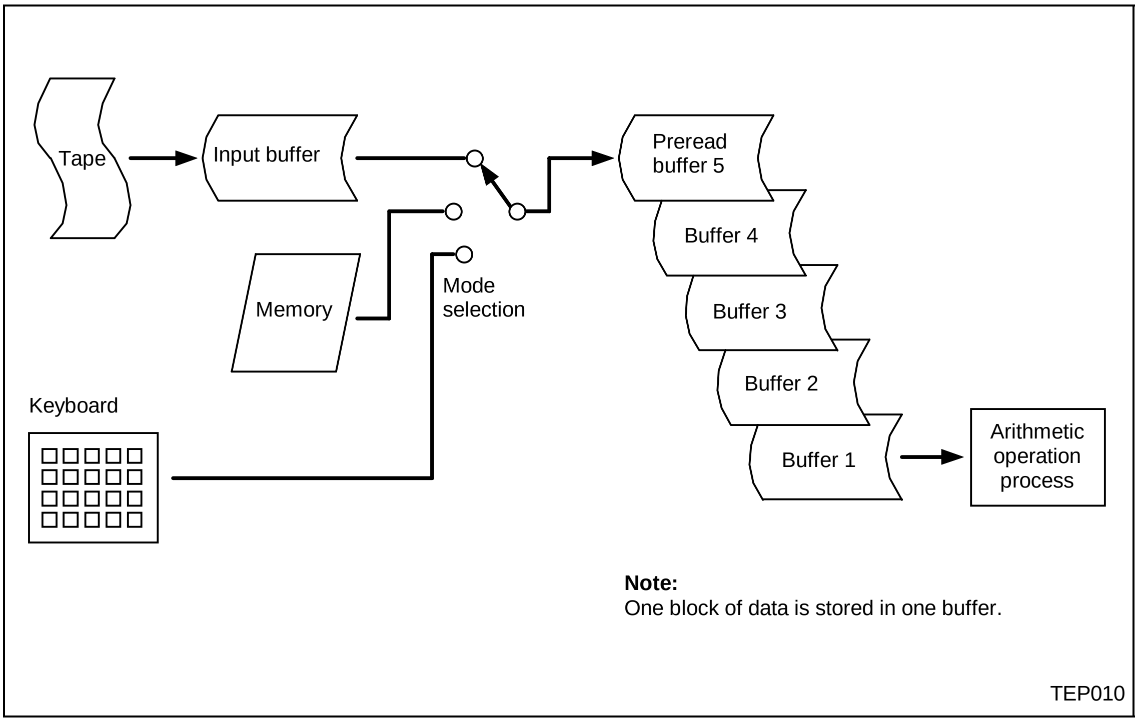
4-1 Input Buffer

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 diminuishes 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.



Note:
One block of data is stored in one buffer.

TEP010

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.

Return to Main Menu

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 SINGLE BLOCK is turned ON during continuous operation, processing will stop after pre-reading the next block data.

5 POSITION PROGRAMMING

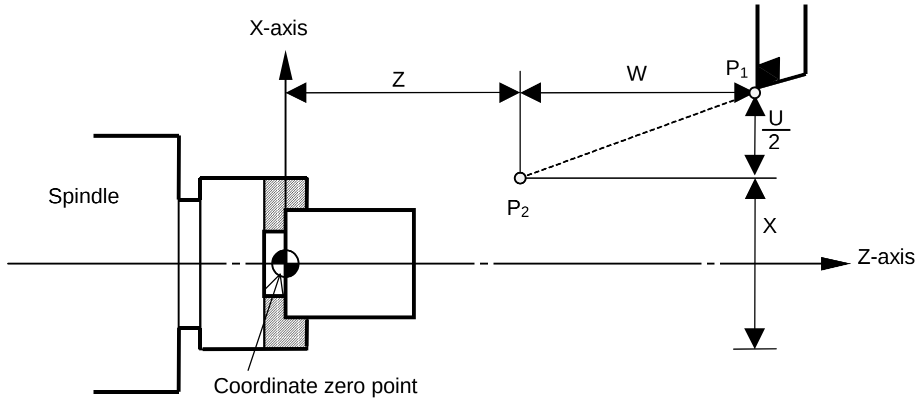
5-1 Absolute Data Command and Incremental Data Command: G90, G91

1. Function and purpose

Movement distance of a tool is designated by absolute data commands or incremental data commands. In the absolute data mode, each axis can be moved to the program-designated position within the workpiece coordinate system, irrespective of the current position. In the incremental data mode, each axis moves through the program-designated distance as relative data with respect to the current position which is taken to be the starting point, 0.

2. Detailed description

The following figure shows the tool path from point P_1 to point P_2.



Incremental and absolute data commands TEP011

For X-axis and Z-axis, incremental command and absolute command are distinguished by address or G-code (G90/G91). However, only when G-code system B or C is selected, absolute command/incremental command can be distinguished by G90/G91. Additional axis (C- or Y-axis) is also distinguished by address or G-code.

		Command system	Remarks
Absolute data	X-axis	Address X	- Address corresponding to the desired axis is set by machine parameter. - Absolute and incremental data can be used together in the same block. - Address of incremental command for A- and B-axes does not exist.
	Z-axis	Address Z	
	C/Y-aixs	Address C/Y	
Incremental data	X-axis	Address U	
	Z-axis	Address W	
	C/Y-aixs	Address H/V	

Example: X____W____;

Incremental data command for Z axis

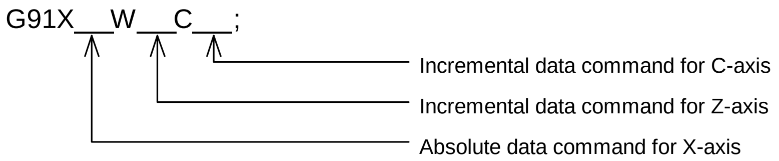
Absolute data command for X-axis

Return to Main Menu

3. Remarks

1. The coordinate value, if omitted, is taken as 0.
2. In G-code systems B or C, the absolute data and the incremental data can be distinguished by G90 and G91 for any axis.

Example: When C-axis is distinguished by G90/G91



G91X__W__C__;

Incremental data command for C-axis

Incremental data command for Z-axis

Absolute data command for X-axis

3. Description of related parameters

P11 bit 7
- 0 : Absolute/incremental is changed by the G-code and the parameter.
- 1 : Absolute/incremental is changed by the address.

P11 bit 2
- 0 : Initial G91 (Incremental command)
- 1 : Initial G90 (Absolute command)

	P11 bit 2 OFF (0)	**P11** bit 2 ON (1)
P11 bit 7 OFF	G91 (Incremental data)	G90 (Absolute data)
P11 bit 7 ON	X/Z : Absolute value U/W : Incremental value	

5-2 Inch/Metric Selection Commands: 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

G20 and G21 commands change only units of program data input but not units of data previously set.
Changeover between G20 and G21 is valid only for the linear axes; it is invalid for rotational axes.

Example: Input data processing according to G20/G21 (Only for decimal-point input type I)

Axis	Input data	[Initial inch] OFF		[Initial inch] ON	
		G21	G20	G21	G20
X	X100;	0.100 mm	0.254 mm	0.0039 inches	0.0100 inches
Z	Z100;	0.100 mm	0.254 mm	0.0039 inches	0.0100 inches
Y	Y100;	0.100 mm	0.254 mm	0.0039 inches	0.0100 inches
C	C100;	0.100 deg	0.100 deg	0.100 deg	0.100 deg

[Initial inch] refers to parameter **P19**; OFF or ON denotes its setting to 0 or 1.

5-3 Decimal Point Input

1. Function and purpose

The decimal point for setting a zero point in millimeters or inches can be included in machining program input information which defines the path, distance, speed, etc. of a tool. Also, whether the least significant digit of data not having the decimal point is to be set as the minimum input command unit (type I) or as a zero point (type II) can be selected using a parameter (**P9** bit 5).

2. Programming format

○○○○○. ○○○	Metric system
○○○○. ○○○○	Inch system

3. Detailed description

1. Decimal-point commands are valid only for scaling factors such as the distance, angle, time and speed that have been set in the machining program.
2. For valid addresses for decimal-point commands, refer to "G-code address list" described below.
3. The number of effective digits for each type of decimal-point command is as follows:

		mm	inch
Move command (Linear)	Integral part	0. - 99999.	0. - 9999.
	Decimal part	.000 - .999	.0000 - .9999
Move command (Rotational)	Integral part	0 - 99999.	0. - 99999. (359.)
	Decimal part	.000 - .999	.0 - .999
Feed rate	Integral part	0. - 60000. 0. - 999.	0. - 2362. 0. - 99.
	Decimal part	.000 - . 999 .0000 - .9999	.0000 - .9999 .000000 - .999999
Dwell (X)	Integral part	0. - 99999.	0. - 99.
	Decimal part	.000 - .999	.000 - .999

Note: For feed rate column, the top row gives the feed rate as a per-minute rate and the bottom row as a per-revolution rate.

4. Decimal-point commands are also valid for definition of variables data used in subprograms.
5. 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, P, S and T. All types of variables command data are handled as the data having the decimal point.

4. Sample programs

Sample programs for addresses accepting the decimal point

Command category / Program example	Type I For 1 = 1 µ	Type II 1=1 mm
G00 X123.45 (With the decimal point always given as the millimeter point)	X123.450 mm	X123.450 mm
G00 X12345	X12.345 mm*	X12345.000 mm

* The least significant digit is given in 1 micron.

G-code address list

	G-code system			Command address																						
	A	B	C	X	Z	Y	U	W	V	C	H	R	I	K	J	D	P	Q	L	A	,C	,R	,A	T	N	S
	G00	G00	G00	S	S	S	S	S	S	S	S	-	-	-	-	-	-	-	-	C	S	S	S	-	-	-
	G01	G01	G01	S	S	S	S	S	S	S	S	-	-	-	-	-	-	-	-	C	S	S	S	-	-	-
	G02	G02	G02	S	S	S	S	S	S	S	S	S	S	S	S	-	B	-	-	-	E	E	E	-	-	-
	G03	G03	G03	S	S	S	S	S	S	S	S	S	S	S	S	-	B	-	-	-	E	E	E	-	-	-
	G04	G04	G04	A	-	-	A	-	-	-	-	-	-	-	-	-	B	-	-	-	-	-	-	-	-	-
*1	G07.1	G07.1	G07.1	-	-	-	-	-	-	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G09	G09	G09	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G10	G10	G10	S	S	S	S	S	S	S	S	S	-	-	-	-	B	B	B	-	-	S	-	-	B	-
	G11	G11	G11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
*2	G12.1	G1.2.1	G12.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
*2	G13.1	G13.1	G13.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G16	G16	G16	-	-	-	-	-	-	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G17	G17	G17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G18	G18	G18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G19	G19	G19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G20	G20	G20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G21	G21	G21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G22	G22	G22	S	S	-	-	-	-	-	-	-	S	S	-	-	-	-	-	-	-	-	-	-	-	-
	G23	G23	G23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G27	G27	G27	S	S	S	S	S	S	S	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G28	G28	G28	S	S	S	S	S	S	S	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G29	G29	G29	S	S	S	S	S	S	S	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G30	G30	G30	S	S	S	S	S	S	S	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G30.1	G30.1	G30.1	S	S	S	S	S	S	S	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G31	G31	G31	S	S	S	S	S	S	S	S	-	-	-	-	-	-	-	-	-	E	E	E	-	-	-
	G32	G32	G32	S	S	S	S	S	S	-	-	-	-	-	-	-	-	-	-	-	E	E	S	-	-	-
	G34	G34	G34	S	S	S	S	S	S	-	-	-	-	S	-	-	-	-	-	-	E	E	S	-	-	-
*3	G36	G36	G36	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	E	-	-
*4	G36(G36.5)	G36(G36.5)	G36(G36.5)	S	S	-	E	E	-	S	E	S	S	A	-	B	B	B	B	B	-	-	-	B	-	-
*3	G37	G37	G37	-	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	E	-	-
*4	G37(G37.5)	G37(G37.5)	G37(G37.5)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G40	G40	G40	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G41	G41	G41	-	-	-	-	-	-	-	-	-	S	S	-	-	-	-	-	-	-	-	-	-	-	-
	G42	G42	G42	-	-	-	-	-	-	-	-	-	S	S	-	-	-	-	-	-	-	-	-	-	-	-
	G46	G46	G46	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G50	G50	G50	S	S	S	S	S	S	S	S	-	-	-	-	-	-	B	-	-	-	-	-	-	-	-
	G50.2	G50.2	G50.2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G51.2	G51.2	G51.2	-	-	-	-	-	-	-	-	-	-	-	-	-	B	B	-	-	-	-	-	-	-	-
*3	G52	G52	G52	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
*4	G52(G52.5)	G52(G52.5)	G52(G52.5)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
*3	G53	G53	G53	S	S	S	S	S	S	S	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
*4	G53(G53.5)	G53(G53.5)	G53(G53.5)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G54	G54	G54	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G55	G55	G55	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G56	G56	G56	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G57	G57	G57	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	G58	G58	G58	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Return to Main Menu

G-code system			Command address																						
A	B	C	X	Z	Y	U	W	V	C	H	R	I	K	J	D	P	Q	L	A	,C	,R	,A	T	N	S
G59	G59	G59	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G61	G61	G61	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G62	G62	G62	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G64	G64	G64	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G65	G65	G65	S	S	S	S	S	S	S	S	S	S	S	S	S	B	S	B	S	-	-	-	-	-	-
G66	G66	G66	S	S	S	S	S	S	S	S	S	S	S	S	S	B	S	B	S	-	-	-	-	-	-
G67	G67	G67	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G68	G68	G68	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G69	G69	G69	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G68.5	G68.5	G68.5	S	S	S	-	-	-	-	-	-	B	B	B	-	-	-	-	-	-	S	-	-	-	-
G69.5	G69.5	G69.5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G70	G70	G72	E	E	-	-	-	-	-	-	-	-	-	-	-	B	B	-	B	-	-	-	-	-	-
G71	G71	G73	E	E	-	S	S	-	-	-	S	-	-	-	B	B	B	-	B	-	-	-	-	-	-
G72	G72	G74	E	E	-	S	S	-	-	-	S	-	-	-	B	B	B	-	B	-	-	-	-	-	-
G73	G73	G75	E	E	-	S	S	-	-	-	S	-	-	-	B	B	B	-	B	-	-	-	-	-	-
G74	G74	G76	S	S	-	S	S	-	-	-	S	E	E	-	B	A	A	-	-	-	-	-	-	-	-
G75	G75	G77	S	S	-	S	S	-	-	-	S	E	E	-	B	A	A	-	-	-	-	-	-	-	-
G76	G76	G78	S	S	-	S	S	-	-	-	S	A	A	-	B	A	A	-	-	-	-	-	-	-	-
G80	G80	G80	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G83	G83	G83	S	S	S	S	S	S	S	S	S	-	A	-	-	-	-	A	-	-	-	-	-	-	-
G84	G84	G84	S	S	S	S	S	S	S	S	S	-	A	-	-	-	-	A	-	-	-	-	-	-	-
G84.2	G84.2	G84.2	S	S	S	S	S	S	S	S	S	-	A	-	-	B	-	A	-	-	-	-	-	-	-
G85	G85	G85	S	S	S	S	S	S	S	S	S	-	A	-	-	-	-	A	-	-	-	-	-	-	-
G87	G87	G87	S	S	S	S	S	S	S	S	S	-	A	-	-	-	-	A	-	-	-	-	-	-	-
G88	G88	G88	S	S	S	S	S	S	S	S	S	-	A	-	-	-	-	A	-	-	-	-	-	-	-
G88.2	G88.2	G88.2	S	S	S	S	S	S	S	S	S	-	A	-	-	B	-	A	-	-	-	-	-	-	-
G89	G89	G89	S	S	S	S	S	S	S	S	S	-	A	-	-	-	-	A	-	-	-	-	-	-	-
G90	G77	G20	S	S	S	S	S	S	S	S	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G92	G78	G21	S	S	S	S	S	S	S	S	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G94	G79	G24	S	S	S	S	S	S	S	S	S	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G96	G96	G96	-	-	-	-	-	-	-	-	-	-	-	-	-	B	-	-	-	-	-	-	-	-	B
G97	G97	G97	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	B
G98	G94	G94	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G99	G95	G95	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
—	G90	G90	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
—	G91	G91	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
—	G98	G98	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
—	G99	G99	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G109	G109	G109	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	B	-	-	-	-	-	-	-
G110	G110	G110	B	B	B	-	-	-	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G111	G111	G111	B	B	B	-	-	-	B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G112	G112	G112	-	-	-	-	-	-	-	-	-	-	-	-	-		-	-	-	-	-	-	-	-	-
G120	G120	G120	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G121	G121	G121	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G122	G122	G122	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G123	G123	G123	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

S : Used as data with sign
A : Used as absolute data
B : Decimal point input not allowed
E : Alarm caused
– : No address reference (ignored)
C : Usable dependent on occasions

*1 : Only for standard mode

*2 : T32 compatible mode and standard mode are different in G-code meaning, but are common in dealing with the address. (without address)

*3 : Only for standard mode

*4 : G○○ refers to T32 compatible mode, and G○○.5 refers to standard mode.

Return to Main Menu

5-4 Selection/Cancellation of X-axis Radial Command: G122.1/G123.1

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

		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.
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
G69.5	Program coordinate system rotation revoking
G123	Polar coordinate input OFF
G22	Stored stroke 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	Program coordinate system rotation
G122	Polar coordinate input ON

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

- NOTE -

Return to Main Menu

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 or nonlinearly to the ending point specified by a coordinate word.

2. Programming format

G00 Xx/Uu Zz/Ww αα ; (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

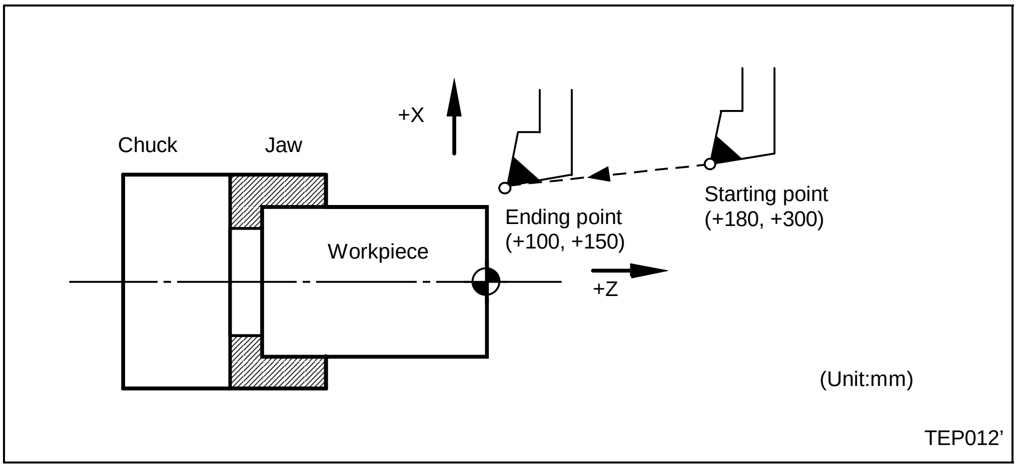
1. Once this command has been given, the G00 mode will be retained until any other G-code command that overrides this mode, that is, either G01, G02, G03, or G32 of command group 01 is given. Thus, a coordinate word will only need be given if the next command is also G00. This function is referred to 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 the machine parameter.
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 the parameter 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:



The diagram above is for:

G00 X100000 Z150000; Absolute data command
G00 U–80000 W–150000; Incremental data command

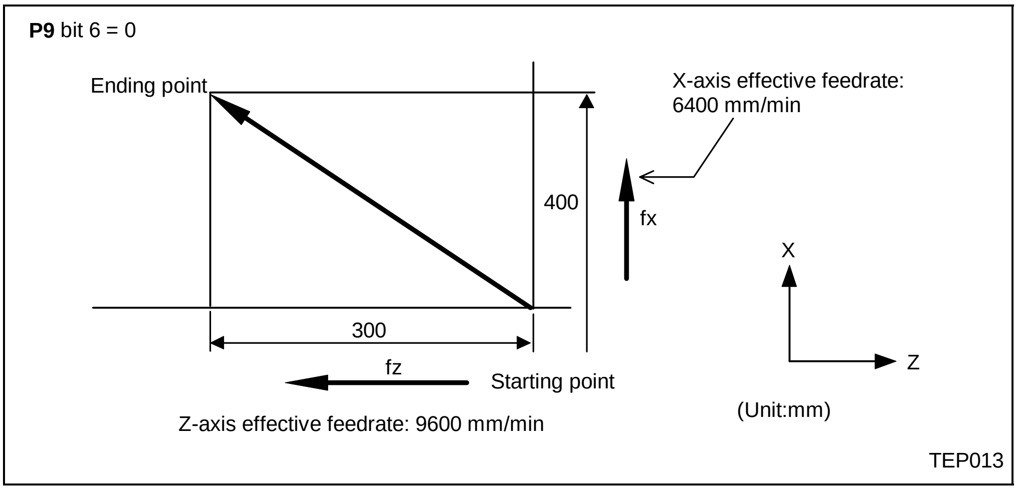
(where the unit of program data input is 0.001 mm)

5. Remarks

1. If bit 6 of user parameter **P9** 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–300000 X 400000 ; (the unit of program data input: 0.001 mm)

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

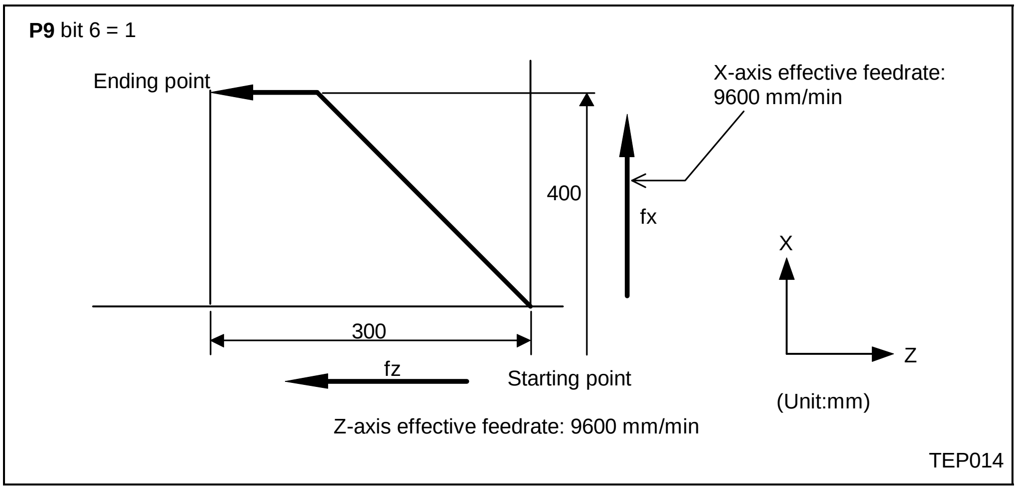


For inch-specification machines, the rapid feed rate of the C-axis is limited to 89 rpm (32000/360) even if item C of parameter **A1** is set to a value greater than 32000.

2. If bit 6 of user parameter **P9** 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-300000 X400000 ; (the unit of program data input: 0.001 mm)

   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 Linear Interpolation Command: G01

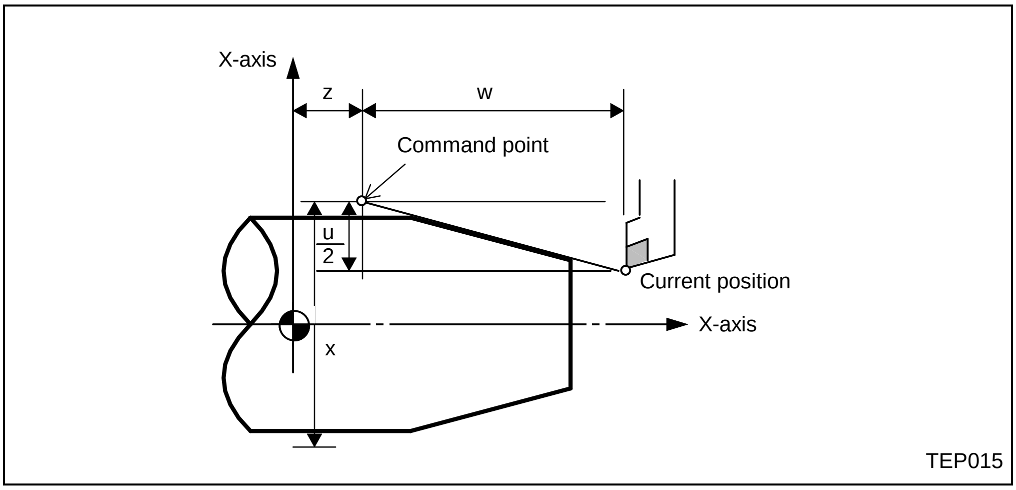
1. Function and purpose

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 αα Ff; (α: Additional axis)

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

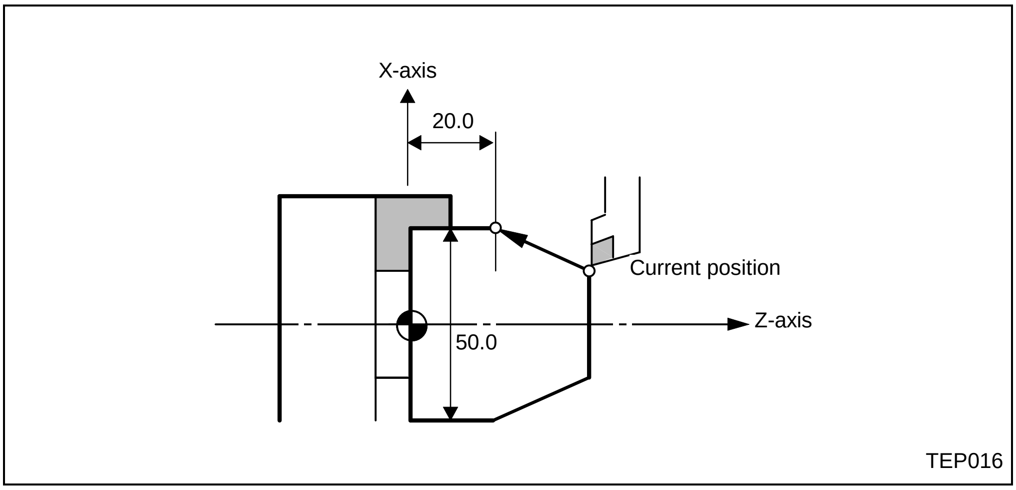


3. Detailed description

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).

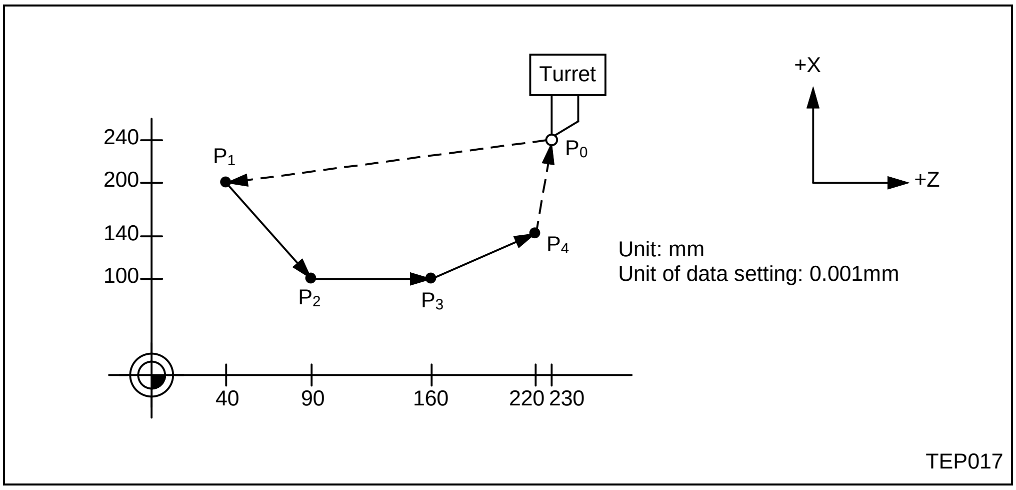
4. Sample program

Example 1: Taper turning



```
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):



```
G00  X200000 Z40000;        P0 → P1
G01  X100000 Z90000  F300;  P1 → P2
             Z160000;       P2 → P3
     X140000 Z220000;       P3 → P4
G00  X240000 Z230000;       P4 → P0
```

Return to Main Menu

6-3 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)

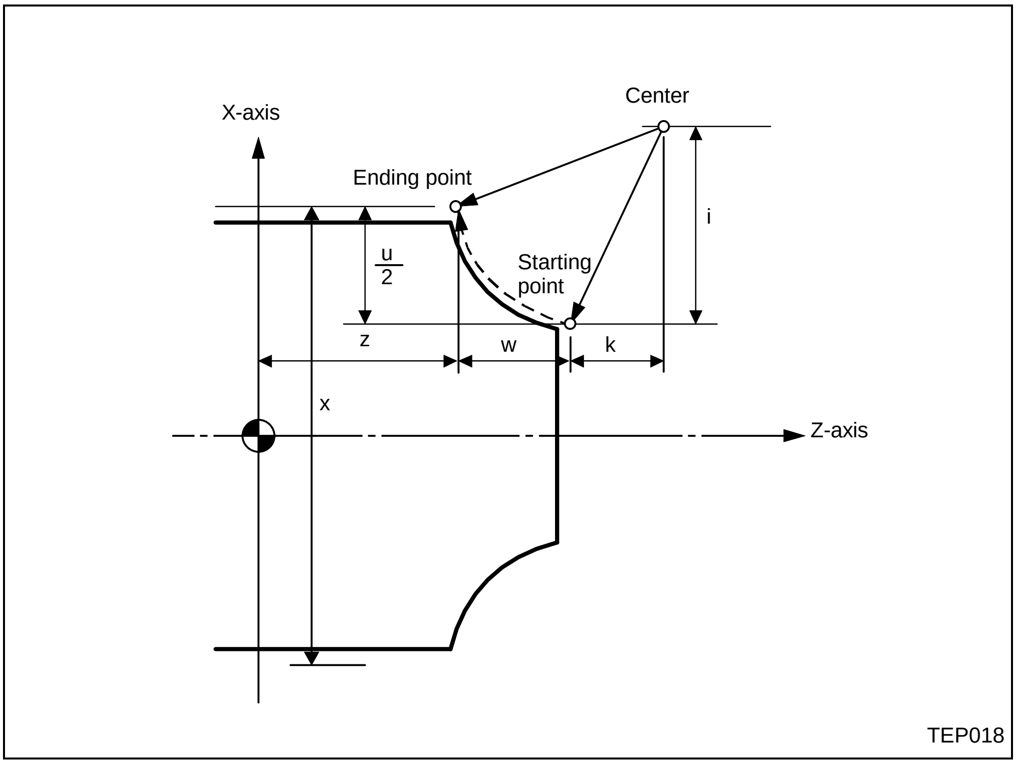
Y/V: Arc ending point coordinates, Y-axis (absolute value of workpiece coordinate system for Y, incremental value from present position for V)

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



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

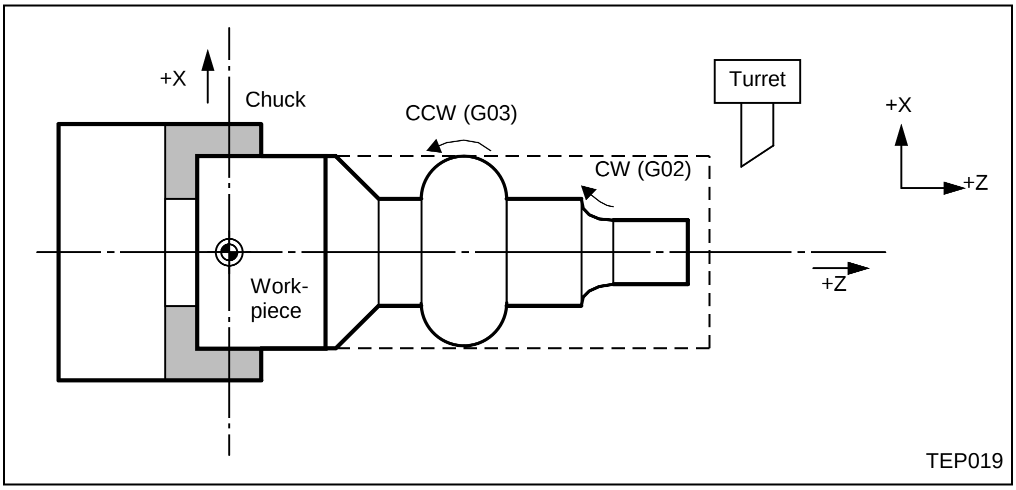
Y-Z plane G19;
G02 (G03) Y_Z_J_K_F_; For Y-axis milling on OD surface

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, G01 or G32 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:

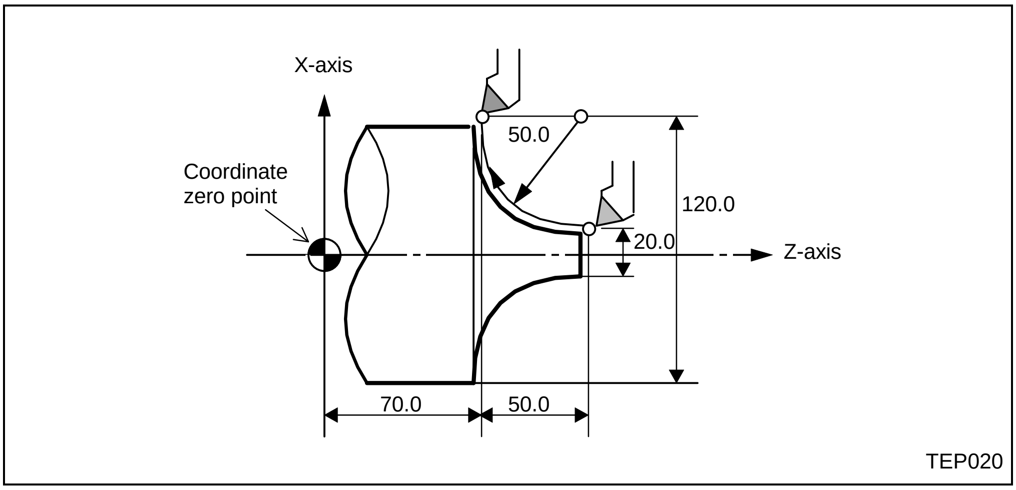
   - Rotational direction......................CW (G02) or CCW (G03)
   - Arc ending point coordinates........Given with address X, Z, Y, U, W, V.
   - Arc center coordinates ....Given with address I, K, J. (Incremental dimension)
   - Feed rate ....Given with address F.

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

Example:

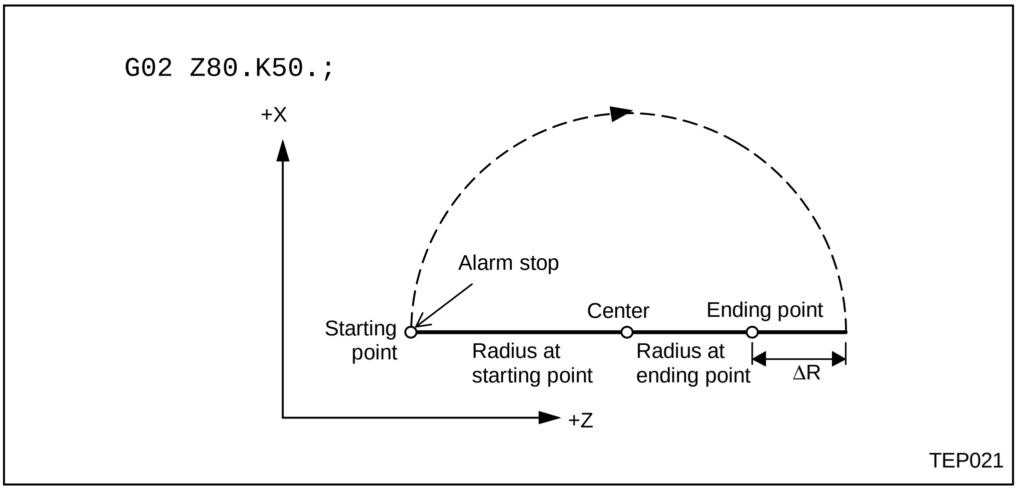


```
G02  X120.0  Z70.0  I50.0  F200;     Absolute data setting
G02  U100.0  W–50.0  I50.0  F200;    Incremental data setting
```

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.

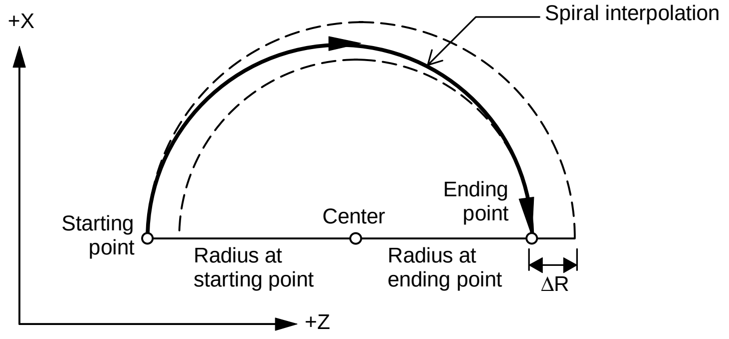
   - If error ΔR is larger than the parameter **U49** (tolerance for radial value difference at ending point), a program error will occur at the starting point of the arc.



Return to Main Menu

- 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.

```
G02 Z90.K50.;
```



TEP022

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

6-4 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).

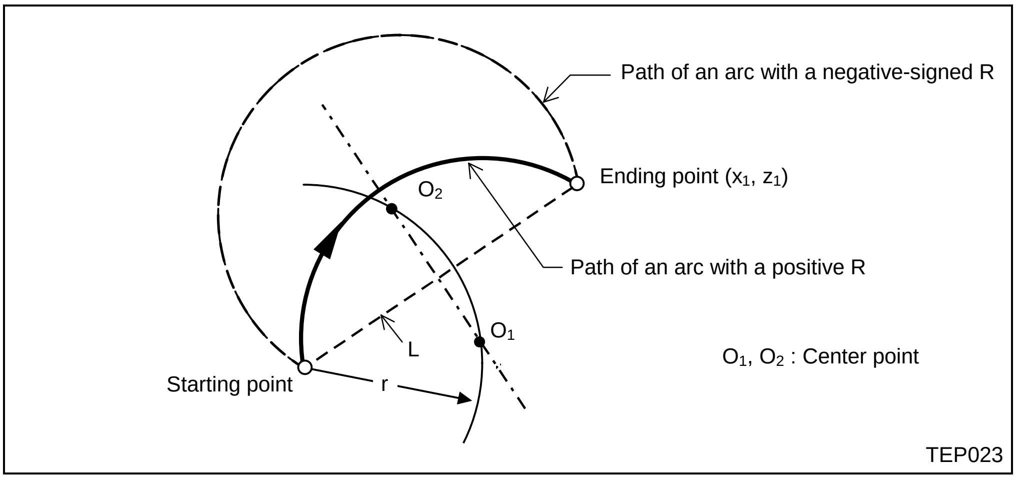
2. Programming format

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 bisector perpendicular to the segment which connects the starting point and the ending point. The crossing point of the bisector 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.


Path of an arc with a negative-signed R
Ending point (x_1, z_1)
O_2
Path of an arc with a positive R
L
O_1
O_1, O_2 : Center point
Starting point
r
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

Example 1: G02 Xx_1 Zz_1 Rr_1 Ff_1 ;

Example 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.

6-5 Plane Selection Commands: G16, G17, G18, G19

1. Function and purpose

Commands G16, 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.

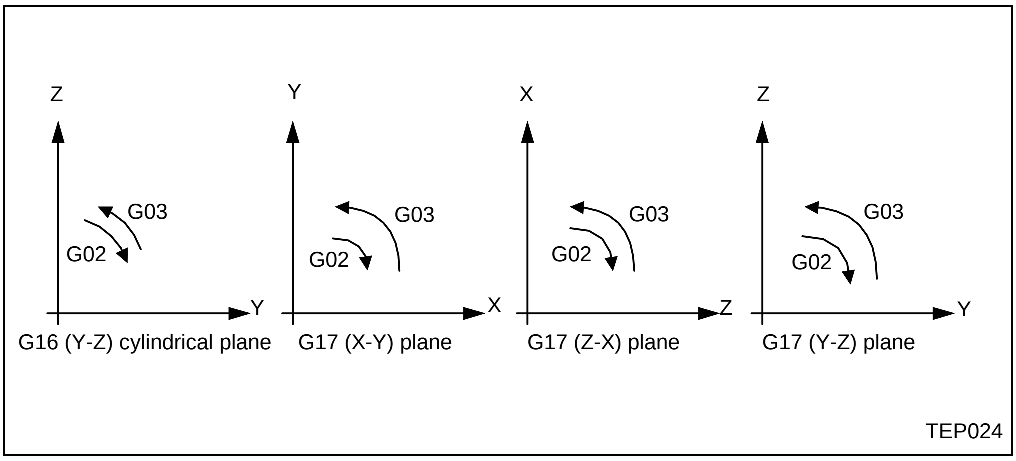
The available planes are the following three types:

- Plane for circular interpolation
- Plane for tool nose radius compensation
- Plane for milling interpolation

2. Programming format

G16; (Y-Z cylindrical plane selection)
G17; (X-Y plane selection)
G18; (Z-X plane selection)
G19; (Y-Z plane selection)

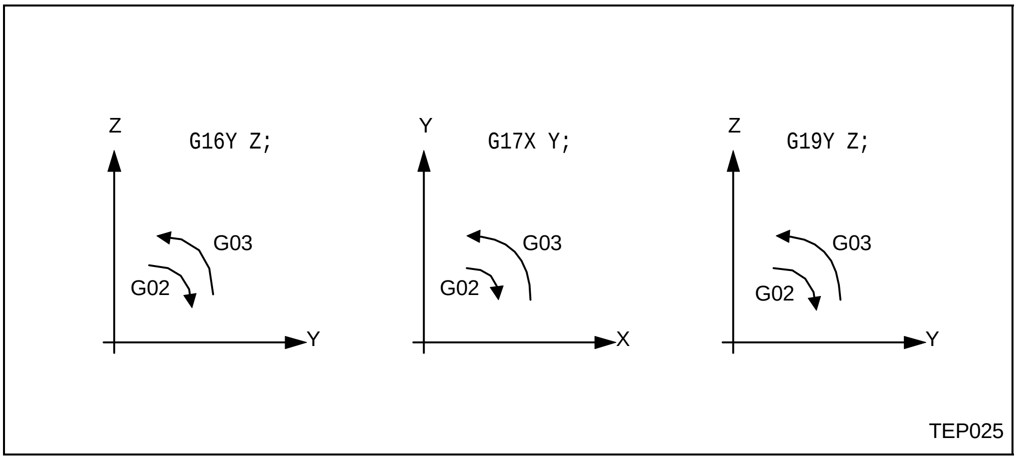
X, Y, and Z denote respective coordinate axes or their corresponding parellel axes.



6-5-1 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 (G16, G17 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, G16, 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 (G16, 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.

Return to Main Menu

6-5-2 Milling mode ON/OFF: G12.1/G13.1 (T32 compatible mode)

1. Function and purpose

When it is made possible to perform milling at the workpiece end or in the longitudinal direction using X-axis, Z-axis and C-axis (rotational axis), the milling shape can be programmed as a command of X-Y-Z rectangular coordinate system assuming Y-axis to be perpendicular to X- and Z-axes.

2. Programming format

G12.1: Milling mode ON
G13.1: Milling mode OFF

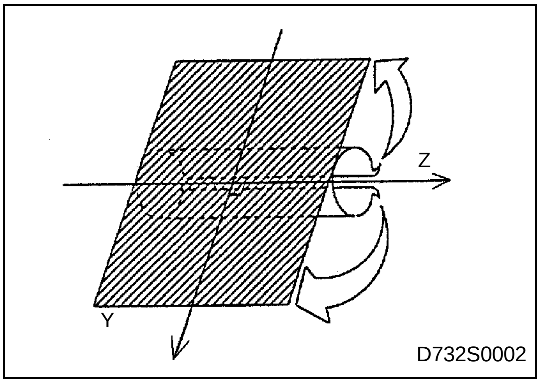
3. Detailed description

1. A plane where linear interpolation, circular interpolation or tool diameter offset is performed is selected by the G codes used for plane selection.

- Y-Z cylindrical plane (Cylindrical interpolation)

G16 Cx ; (x: cylindrical radius)

As shown right, it is a plane when a cylinder giving x to its bottom radius is spread. It is selected to machine the workpiece sides.



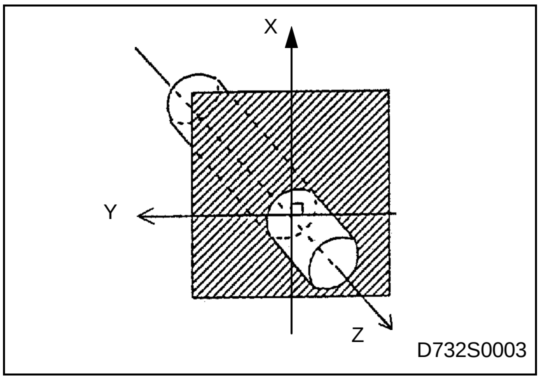
The example of programming shown on the right (F10) causes a C-axis feed of 10°/min to be performed for cylindrical interpolation, irrespective of mm or inch specification.

```
 :
G98;
G12.1 C_;
G01 Y_F10;
 :
```

- X-Y plane (Polar coordinate interpolation)

G17;

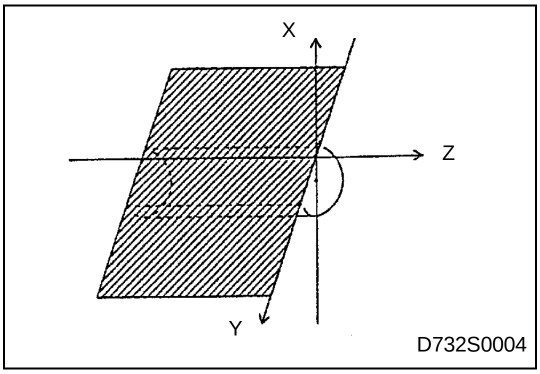
It is an X-Y plane for X-Y-Z rectangular coordinate system. It is selected to machine the workpiece end.



- Y-Z plane

  G19;

  It is a Y-Z plane for X-Y-Z rectangular coordinate system. It is selected to machine the workpiece sides



2. The movement on the virtual axis is commanded by address Y or V.
   Y: Absolute data command
   V: Incremental data command
3. Every axis is designated by radius at milling mode ON status.
4. A plane selected before giving a plane selection command (G16, G17, G19) is automatically selected by parameter (**P10** bit 0, **P10** bit 1) at milling mode ON status.

P10 bit 0	**P10** bit 1	Selected plane
1	1	G16
1	0	G16
0	1	G19
0	0	G17

5. G12.1 command turns the mode forcibly into the asynchronous feed mode (G98). And, by G13.1 command, the selected plane, feed mode and feed rate are returned to the previous ones (that is, those before G12.1 is commanded).
6. When turning the power on and resetting, G13.1 (milling mode cancel) is automatically selected.
7. If the following requirements are not filled, an alarm occurs.
   - Reference point return has been finished on X-, Z-and C-axes.
   - Tool nose radius compensation has been cancelled.
8. Proper machining requires that C-axis is positioned at 0. Because NC computes the end point assuming that C-axis is positioned at 0 when G12.1 is commanded.
9. Proper machining requires that the zero point of workpiece coordinate system is set on the center line of a workpiece.
10. When address C is not commanded with G16, a cylinder is defined using as a radius the current position of X-axis (treated as a radius value) when G16 is commanded.
    However, when C0 is commanded, an alarm occurs. (770 “WORKPIECE RADIUS NOT INPUT (G16)”)
11. G-codes capable of commanding Y-axis (virtual axis) among valid G-codes during milling mode are G00, G01, G02 and G03. These commands are referred to as milling interpolation command.
12. Every move command during milling mode is given by the coordinate system determined from the machining plane being selected. Therefore, the movement on the rotational axis cannot be commanded during milling mode.
    To perform milling from a particular position (C-axis), locate the C-axis before entering milling mode (before commanding G12.1).

13. Any other axis than X, Z and Y (virtual axis) cannot be commanded during milling mode.
14. G-codes available during milling mode are limited. Available G-codes are given below. If any G-code, which is not given in the list, is commanded during milling mode, an alarm (764 “ILLEGAL G-CODE (MILLING)”) is given.
15. During hole machining fixed cycle, the hole positioning data is commanded by Y-axis (virtual axis) and not by C-axis. Even when hole machining fixed cycle is commanded to perform machining on a plane different from that being selected, machining is possible, but executing a tool diameter offset does not provide correct compensating direction. For example, it is not provided when the fixed cycle designating hole positioning data on the X-Y plane is commanded during selecting Y-Z plane.
16. Commanding G16, G17 or G19 during milling mode OFF (G13.1) causes an alarm. (708 “ILLEGAL G-CODE”).
17. Commanding G18 during milling mode ON (G12.1) causes an alarm. (764 “ILLEGAL G-CODE (MILLING)”).

Related G-codes list

Classifi-cation	G-code	Function	Classifi-cation	G-code	Function
×	G00	Positioning		G80	Drilling cycle cancel
×	G01	Linear interpolation		G83	Front drill cycle (Z-axis)
×	G02	Circular interpolation (CW)		G84	Front tap cycle (Z-axis)
×	G03	Circular interpolation (CCW)		G84.2	Front synchronous tap cycle (Z-axis)
	G04	Dwell		G85	Front boring cycle (Z-axis)
	G09	Exact stop check		G87	Side drill cycle (X-axis)
	G13.1	Milling mode cancel		G88	Side tap cycle (X-axis)
○	G16	Y-Z cylindrical plane selection		G88.2	Side synchronous tap cycle (X-axis)
○	G17	X-Y plane selection		G89	Side boring cycle (X-axis)
○	G19	Y-Z plane selection		G61	Exact stop mode
	G40	Tool diameter offset cancel		G64	Cutting mode
	G41	Tool diameter offset left			
	G42	Tool diameter offset right			

×: Milling interpolation command
○: G-code available only during milling mode

6-5-3 Polar coordinate command ON/OFF: G122/G123

1. Function and purpose

The end point coordinate value can be entered by the polar coordinates (radius and angle). Whether it is polar coordinate command is determined by commanding G122 and G123.

2. Programming format

G122: Polar coordinate command ON
G123: Polar coordinate command OFF

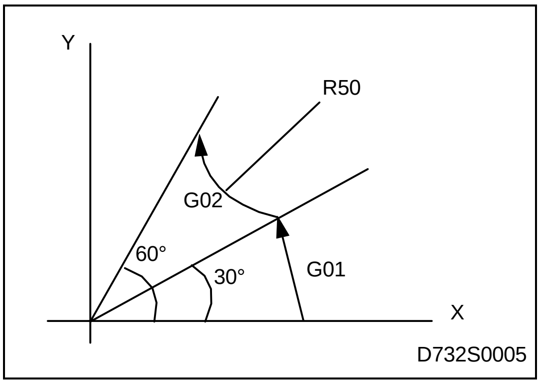
3. Detailed description

1. Used address

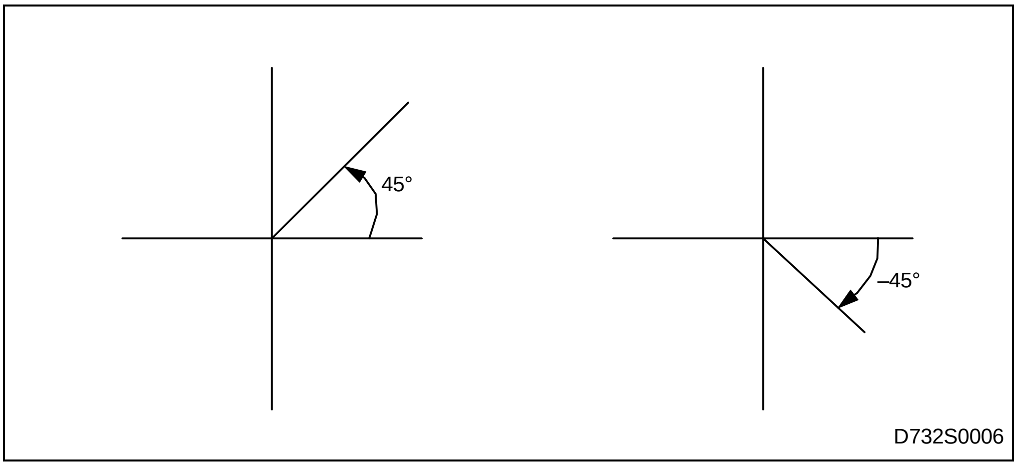
Selected plane	Radius command address	Angle command address	Remarks
G17	X	C	Milling

2. Sample programs (for both T32 compatible and Standard modes)

```
G00 G98;
G28 U0 W0;
M200;
G00 X100.Z0 C0;
G122 G12.1;
G01 X50.C30.F100;
G02 X50.C60.R50.;
G123 G13.1;
  ⋮
```



3. The angle command is given using counterclockwise direction as “+”.



4. Commanding the address Y during polar coordinate command mode is invalid.
5. Enter the length (X) by radius value.
6. Command the radius of circular interpolation command (G02, G03) by the address R.

Return to Main Menu

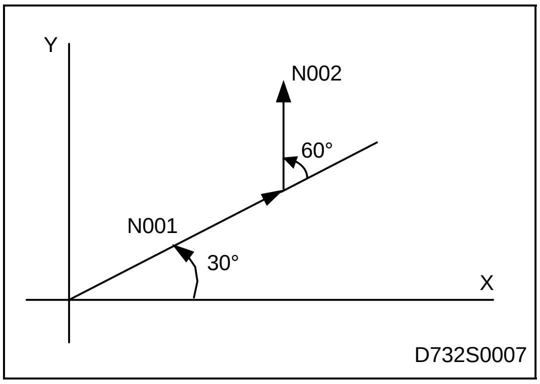
7. The incremental command is available.

Absolute data address	Incremental data address
X	U
Z	W
C	H

For the incremental command, the incremental data from the end point of the preceding block is given.

N001 G00 X100.C30.;
N002 G00 U50.H60.

- An angle given by the incremental data is processed as shown right.
- A length given by the incremental data is programmed as with the absolute data considering that the current position is at the center of the polar coordinates.



6-6 Polar Coordinate Interpolation ON/OFF: G12.1/G13.1 (Standard Mode)

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 19).

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

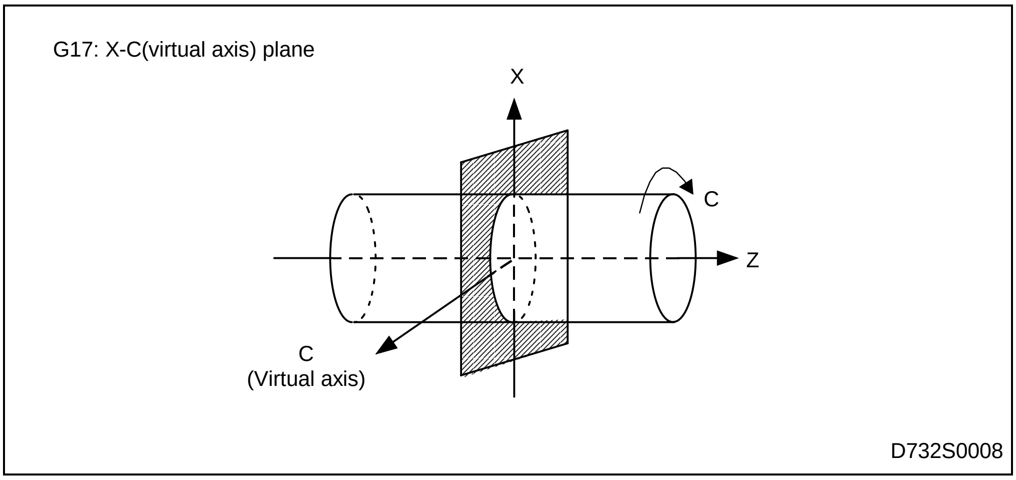
1. When turning on the power and resetting, the polar coordinate interpolation cancel mode (G13.1) is provided. Commanding G12.1 provides a plane selected by G17.
2. The polar coordinate interpolation uses the zero point of workpiece coordinate system as that of the coordinate system. A plane (hereinafter 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-axis) 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, G46). G12.1 and G13.1 must be commanded in G40 mode (Nose radius compensation cancel mode).

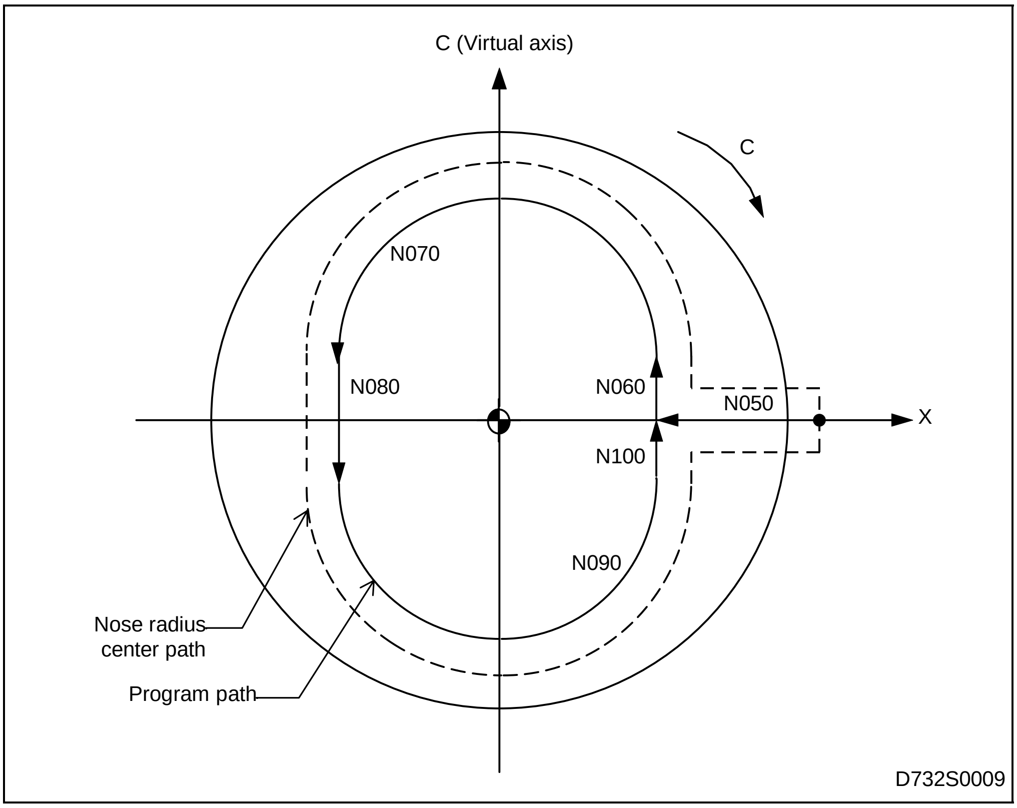
5. The feed rate is commanded using tangential speed (relative speed of the workpiece and a tool) on the polar coordinate interpolation plane (rectangular coordinate system) as F (mm/min or inch/min is used for a unit of F).

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.



Return to Main Menu

4. Sample programs



```
    ⋮
N001  G00 G97 G98;
N004  G28 U0 W0;
N008  M200;
N010  T0101;
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;
    ⋮
```

Return to Main Menu

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, G42 and G46.
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” 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.

6-7 Cylindrical Interpolation Command: G07.1 (G107) (Standard Mode)

1. Function and purpose

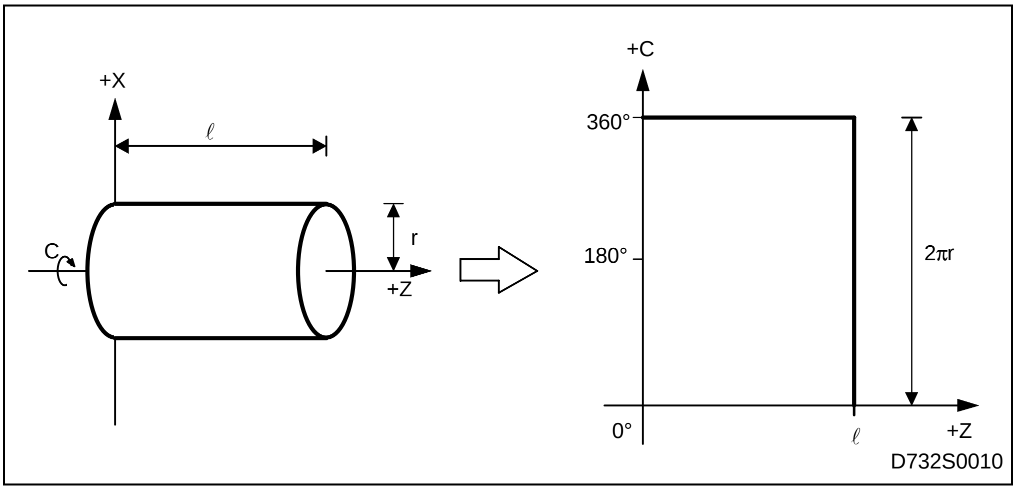
Cylindrical interpolation function refers to a function by which the sides of a cylindrical workpiece are machined. The cylindrical interpolation function capable of programming in the form in which the sides of a cylinder are spread can very easily prepare programs including cylindrical cam-grooving.

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

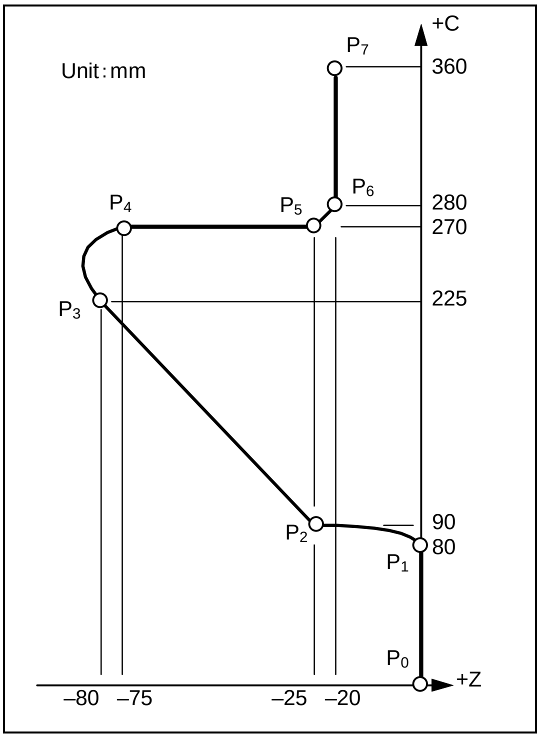


The moving distance of rotational axis commanded with an angle is converted to the linear distance on the circumference in CNC. After the conversion, linear interpolation or circular interpolation is given with the other axis. After the interpolation, the calculated movement is converted again to the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

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;
T0101;
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: Different from the case of the cylindrical interpolation in T32 compatible mode, the example of programming shown on the right (F10) realizes a C-axis feed of 143°/min [approximation of $10/(4 \times 2\pi) \times 360$].

```
 ⋮
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_ ;	or	G19 C_ Z_ ;
G02/G03 Z_ C_ R_ ;		G02/G03 Z_ C_ R_ ;
(when parameter **K65** = 5)		(when parameter **K65** = 6)

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 cylindircal 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.

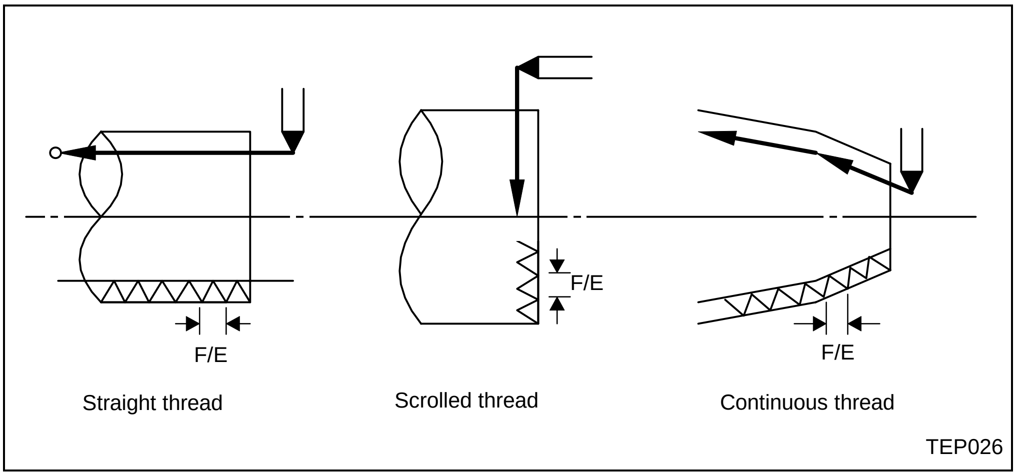
Return to Main Menu

6-8 Threading

6-8-1 Constant lead threading: G32

1. Function and purpose

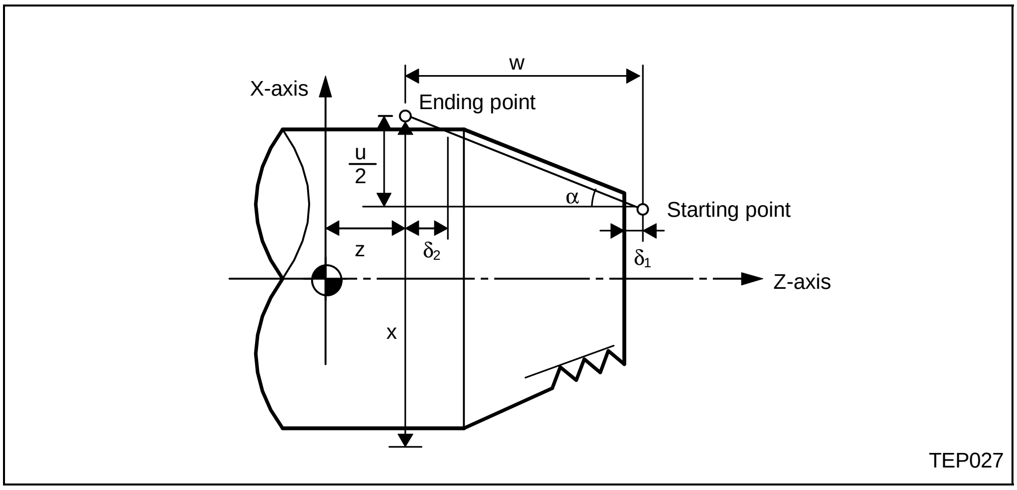
The G32 command controls the feedrate of the tool in synchronization with the spindle rotation and so this enables both the straight and taper thread cutting of constant leads and the continuous thread cutting.



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

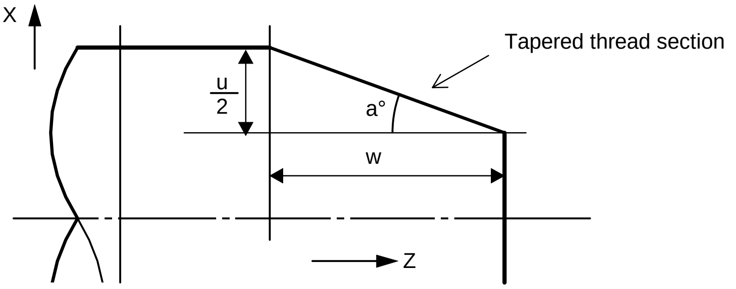


Return to Main Menu

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 **P9** is set to 1 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-asis 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

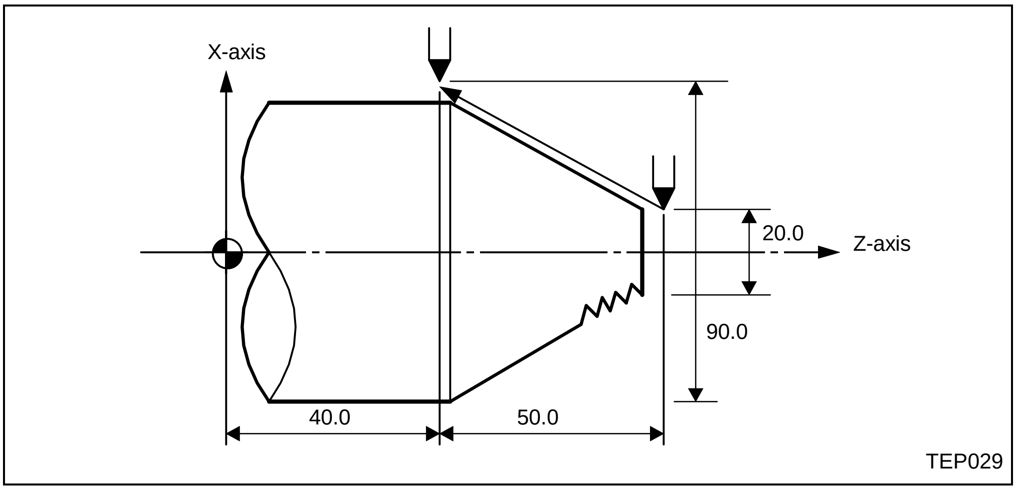
   where R : Spindle speed (rpm) ≤ Permissible speed of encoder (rpm)
   Thread lead = mm or inches
   Maximum feed rate = mm/min or inch/min (this is subject to the restrictions imposed by the machine specifications).

10. During threading, use or disuses of dry run can be specified by setting parameter **P13** bit 4.

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
```

Return to Main Menu

6-8-2 Inch threading: G32

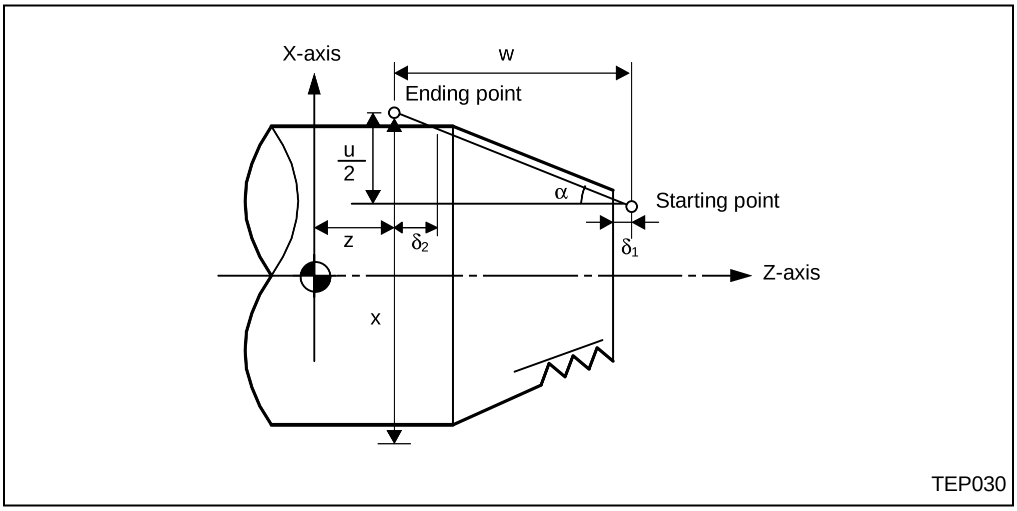
1. Function and purpose

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;

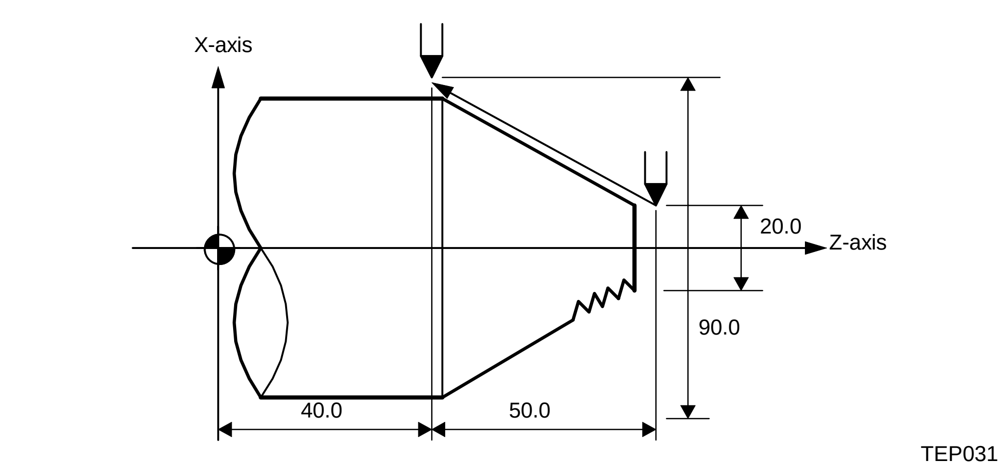
Where Zz, Ww, Xx, Uu: Thread ending point addresses and coordinates
Ee: Number of threads per inch in direction of long axis (axis of which the moving distance is the longest)
(Decimal point command can also be assigned.)



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 **P9** bit 7).
3. The E command value should be set within the lead value range when converted to the lead.
4. See Section 6-8-1 on “Constant lead threading” for further details.

4. **Sample programs**

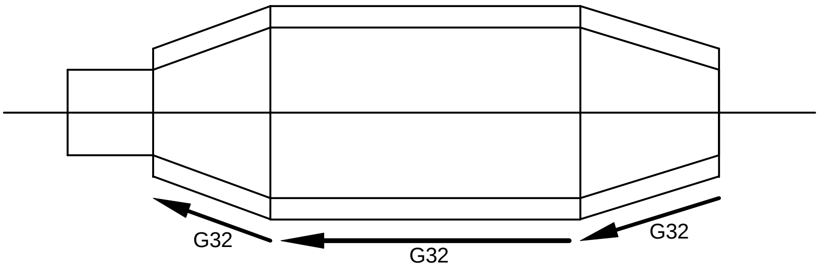


TEP031

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

6-8-3 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.

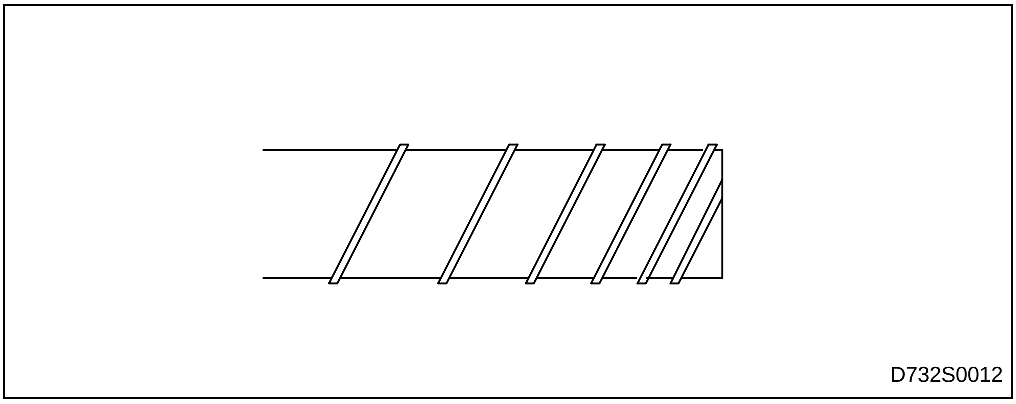


TEP032

6-8-4 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.



2. Programming

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 inch/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.

6-8-5 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.

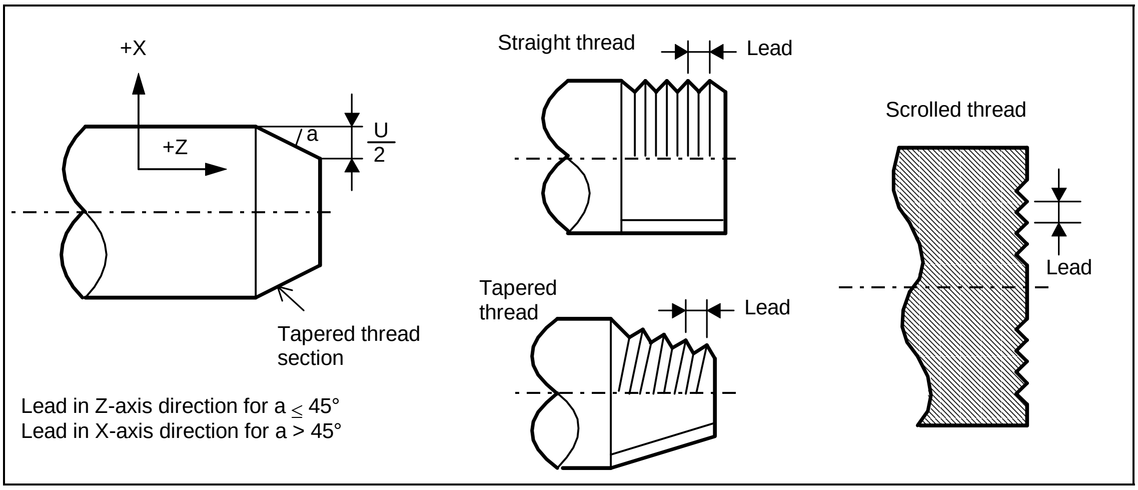
2. Programming format

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 **K50** to select the direction of C-axis rotation:
K50 = 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 **A4**) 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.

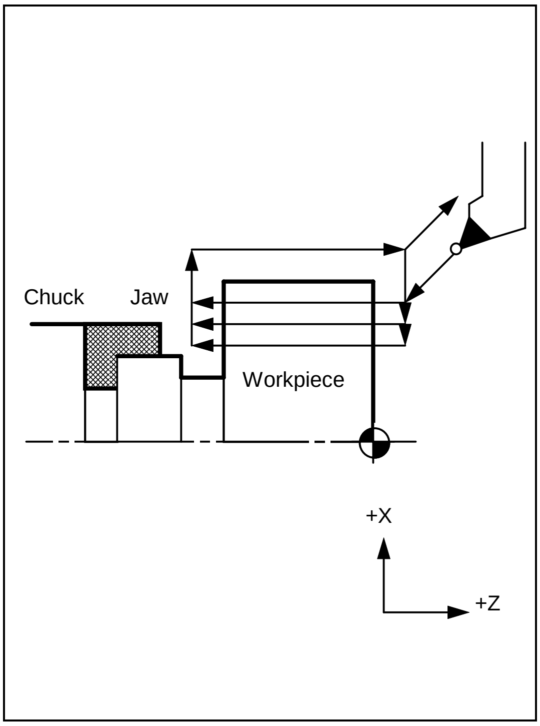
4. During execution of G01.1 command, it is possible, indeed, but not advisable at all to apply feed hold or to change the override value for fear of deformation of the thread.

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;
T0100;
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;
```

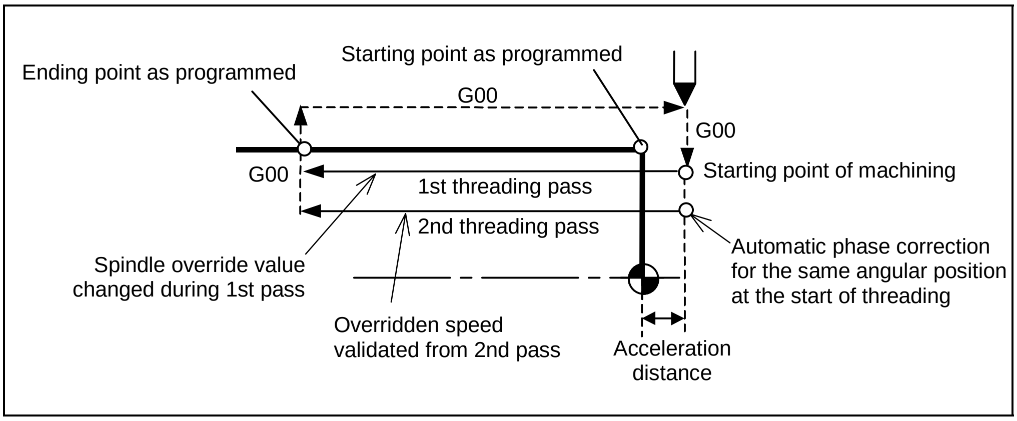


(*) Command for threading with C-axis control, 2 mm lead and 400 rpm

6-8-6 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.


Ending point as programmed
Starting point as programmed
G00
G00
Starting point of machining
G00
1st threading pass
2nd threading pass
Spindle override value changed during 1st pass
Automatic phase correction for the same angular position at the start of threading
Overridden speed validated from 2nd pass
Acceleration distance

2. Related G-codes

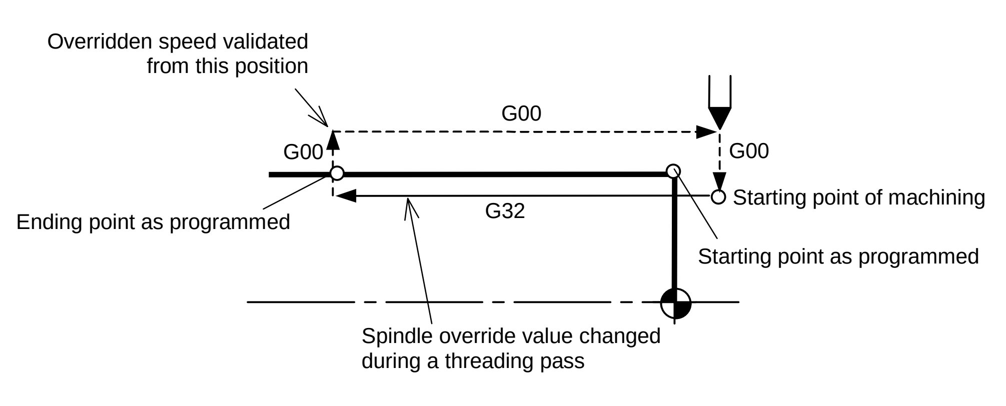
The automatic correction function is applicable to the following G-codes of threading:

Function	T32-compatible mode G-code series			Standard mode G-code series		
	A	B	C	A	B	C
Thread cutting (straight, taper)	G32	G33	G33	G32	G33	G33
Turning fixed cycle for threading	G92	G78	G21	G92	G78	G21
Compound fixed cycle for threading	G76	G76	G78	G76	G76	G78

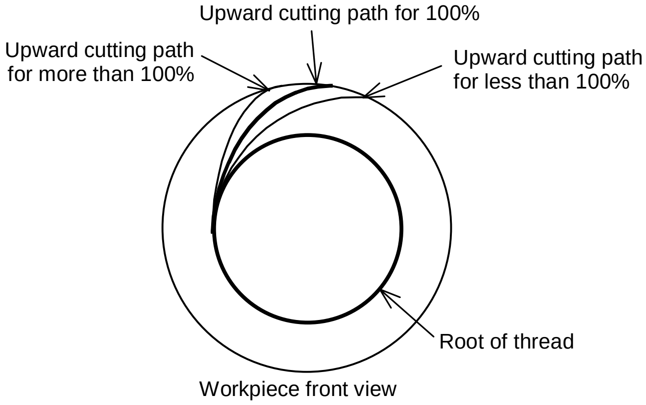
Note: Variable-lead threading (G34), or continuous threading for different-pitch sections, requires continuous or transitional acceleration between blocks, as well as different distances of acceleration. The automatic correction function cannot guarantee correct thread forming for a speed overriding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


Overridden speed validated from this position
G00
G00
G00
Starting point of machining
Ending point as programmed
G32
Starting point as programmed
Spindle override value changed during a threading pass

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.
   Refer to the calculation formulae and diagram of the approach distance (δ_1), given in APPENDIX 4, for details on the acceleration distance.
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%.


Upward cutting path for 100%
Upward cutting path for more than 100%
Upward cutting path for less than 100%
Root of thread
Workpiece front view

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

6-9 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)

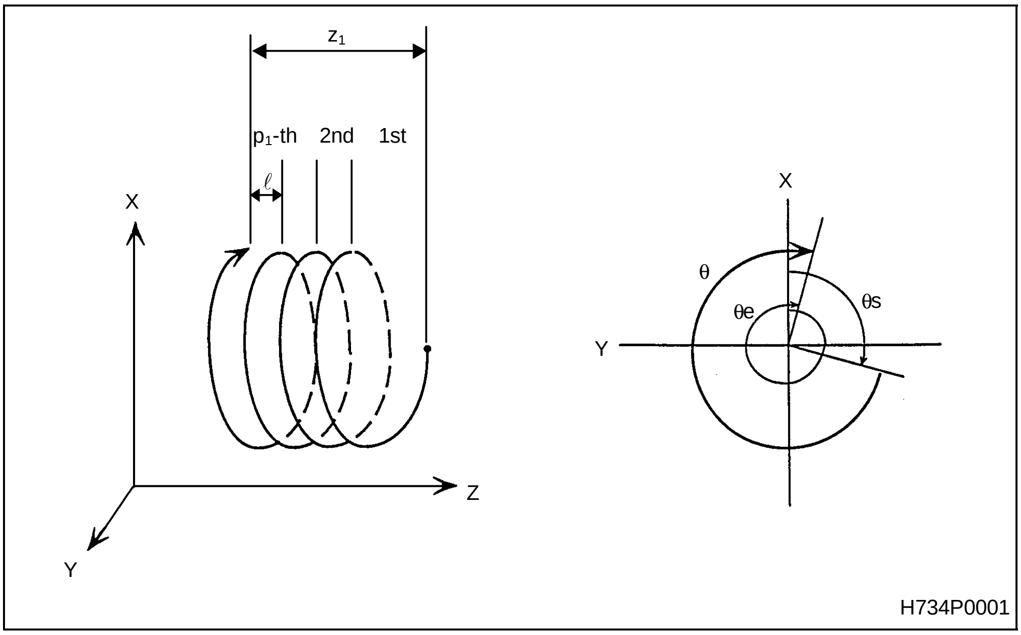
- Ff_1: Feed rate
- Pp_1: Number of pitches
- Ii_1 Jj_1: Arc center coordinates
- Zz_1: Linear axis ending point coordinate
- Xx_1 Yy_1: Arc ending point coordinates

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:

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

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

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

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.

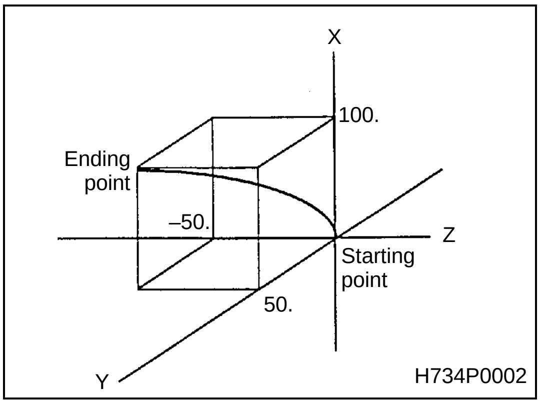
   - Y-Z plane circular, X-axis linear

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

4. Sample programs

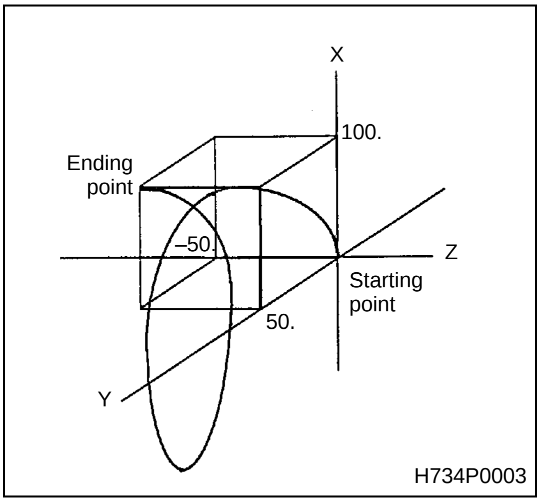
Example 1:

```
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;
```



Return to Main Menu

- NOTE -

Return to Main Menu

7 FEED FUNCTIONS

7-1 Rapid Feed Rates

A separate rapid feed rate can be set for each axis. The range of rapid feed rates which can be set is from 1 mm/min to 60000 mm/min for a program data input unit of 1 micron. The maximum feed rate, however, is limited according to the particular machine specifications.
Refer to the Operating manual for the machine for rapid feed 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 bit 6 of parameter **P9** to select the interpolation type or the non-interpolation type. The positioning time is the same for both types.

Note: Override can be executed for rapid feed in automatic operation and in manual operation by external input signal.

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 1: The alarm 713 “FEEDRATE ZERO” 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.

Note 2: If you set data in inches for machine operation on the metric system, the maximum available feed rate will become 2362 inches/min for both rapid feed and cutting feed.

7-3 Asynchronous/Synchronous Feed Commands: G98/G99

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.

   For T32 Compatible mode (**P16** bit 3 = 0)

	G98F_ (Feed per minute)	G99F_ (Feed per revolution)
Input in mm	1 to 99999 mm/min (F1 to F99999)	0.01 to 999.99 mm/rev (F1 to F99999)
Input in inches	0.01 to 9999.99 in./min (F1 to F999999)	0.0001 to 99.9999 in./rev (F1 to F999999)

   For Standard mode (**P16** bit 3 = 1)

	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 **P11**, 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

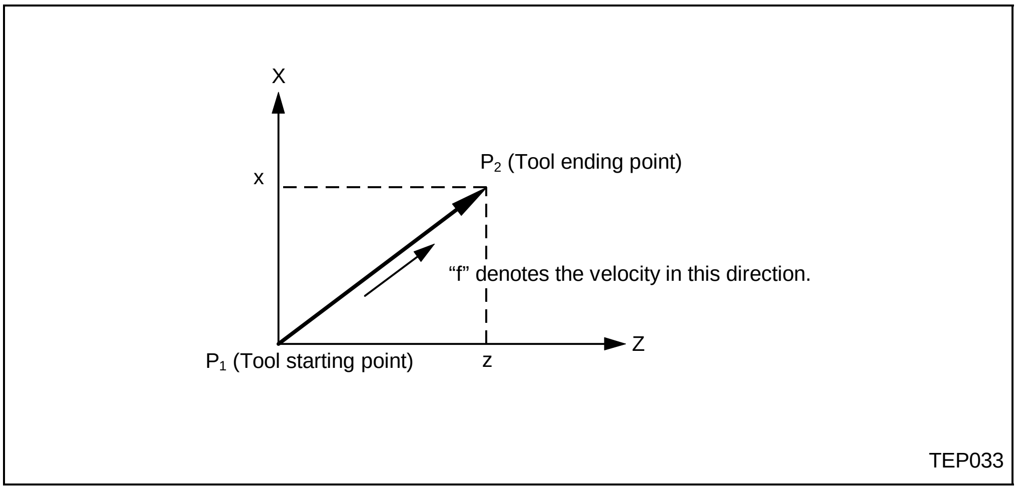
As mentioned earlier, the machine has various control axes. These control axes can be broadly divided into linear axes, which control linear motions, and rotational axes, which control rotational motions. Feed rates for control axes have different effects on the tool speed, which is of great importance 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:



TEP033

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:

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

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

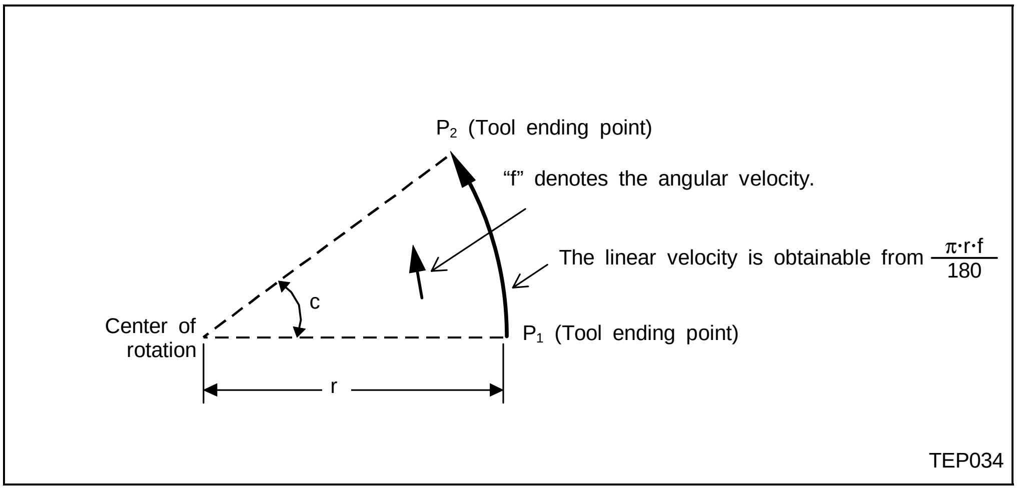
2. Controlling a rotational axis

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.

Return to Main Menu

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


P2 (Tool ending point)
"f" denotes the angular velocity.
The linear velocity is obtainable from $\frac{\pi \cdot r \cdot f}{180}$
c
Center of rotation
P1 (Tool ending point)
r
TEP034

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

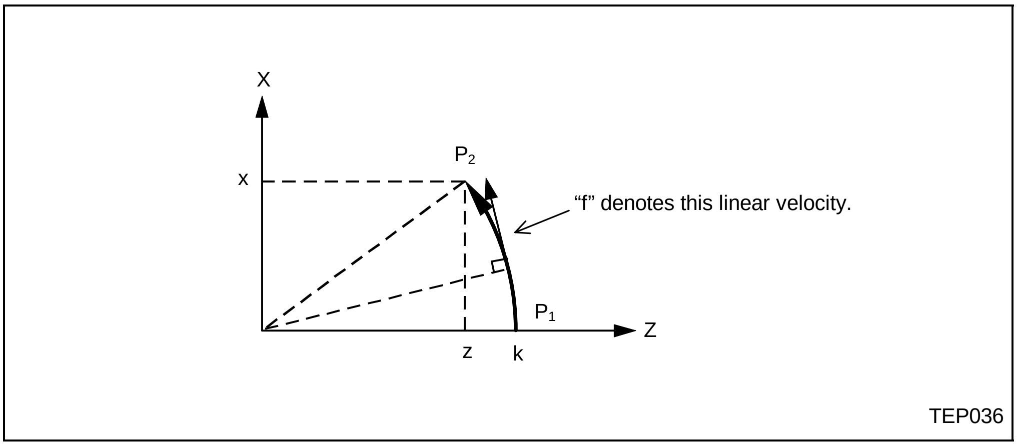
$$fc = f \times \frac{\pi \bullet r}{180}$$

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

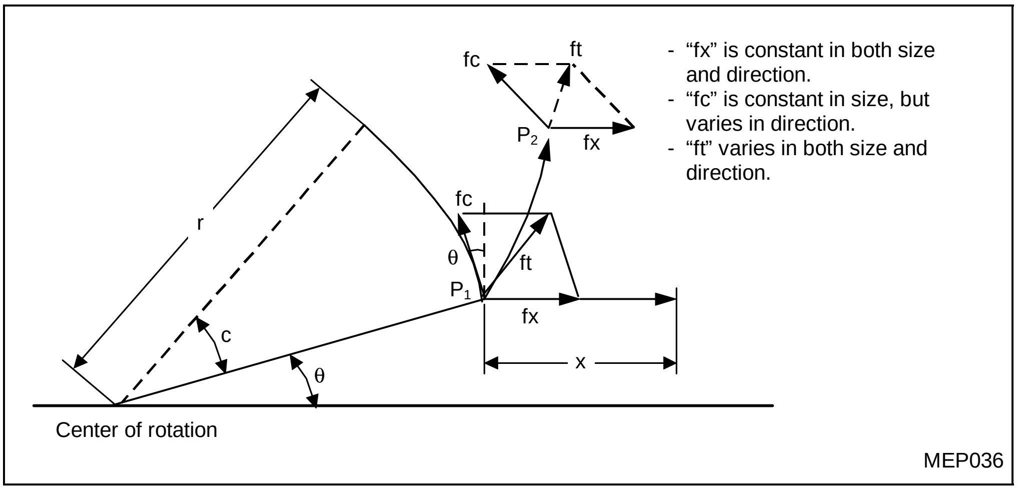


In this case, the X- and Z-axis feed rates will change with the movement of the tool. The resultant velocity, however, will be kept at the constant value, f.

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:



- "fx" is constant in both size and direction.
- "fc" is constant in size, but varies in direction.
- "ft" varies in both size and direction.

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:

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

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

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:

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

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

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

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

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)

Mode	Standard mode			T32 Compatible mode		
Unit of program data input	0.0001 mm			0.001 mm		
Command address	F (mm/rev)	E (mm/rev)	E (Number of threads/inch)	F (mm/rev)	E (mm/rev)	E (Number of threads/inch)
Unit of minimum data setting		1(=0.0001) (1.=1.0000)	1(=1) (1.=1.00)	1(=0.01) (1.=1.000)	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	0.001 to 999.999	0.0001 to 999.9999	0.01 to 9999999.9

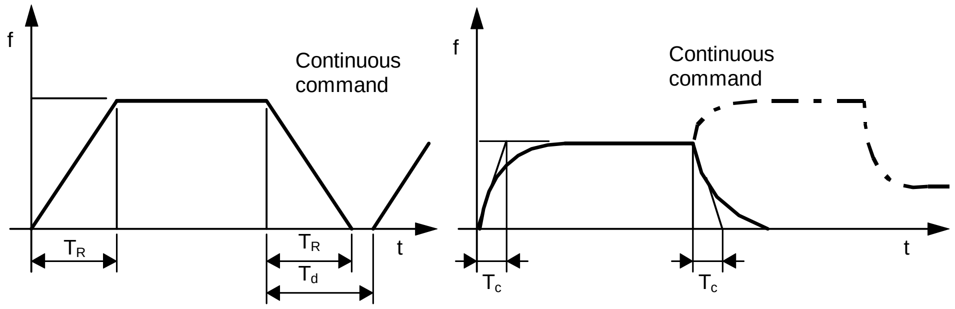
Thread cutting (inch input)

Mode	Standard mode			T32 Compatible mode		
Unit of program data input	0.000001 in.			0.0001 in.		
Command address	F (in./rev)	E (in./rev)	E (Number of threads/inch)	F (in./rev)	E (in./rev)	E (Number of threads/inch)
Unit of minimum data setting		1(=0.000001) (1.=1.000000)	1(=1) (1.=1.0000)	1(=0.0001) (1.=1.0000)	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	0.0001 to 99.9999	0.000001 to 99.999999	0.0001 to 9999.9999

Return to Main Menu

7-6 Automatic Acceleration/Deceleration

The rapid feed 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 feed 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 Main Menu

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

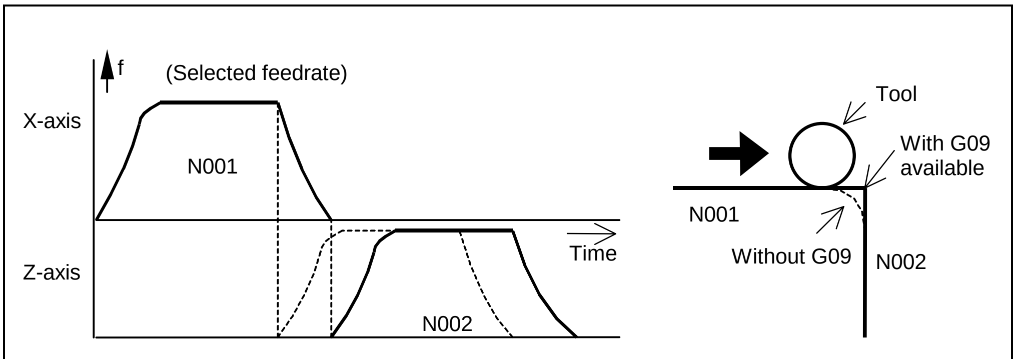
G09 G01 (G02, G03) ;

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 ;
```

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 Main Menu

4. Detailed description

A. Continuous cutting feed commands

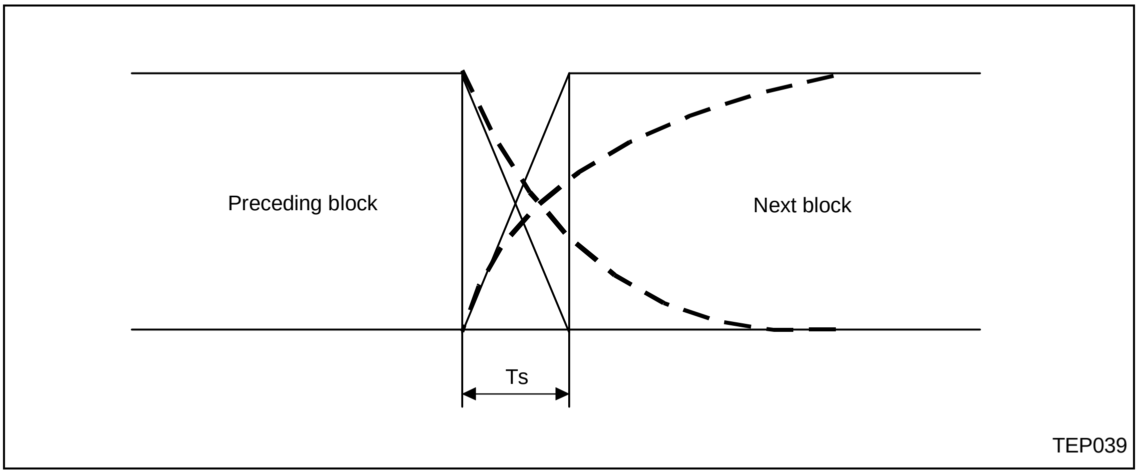


Fig. 7-2 Continuous cutting feed commands

B. Cutting feed commands with in-position status check

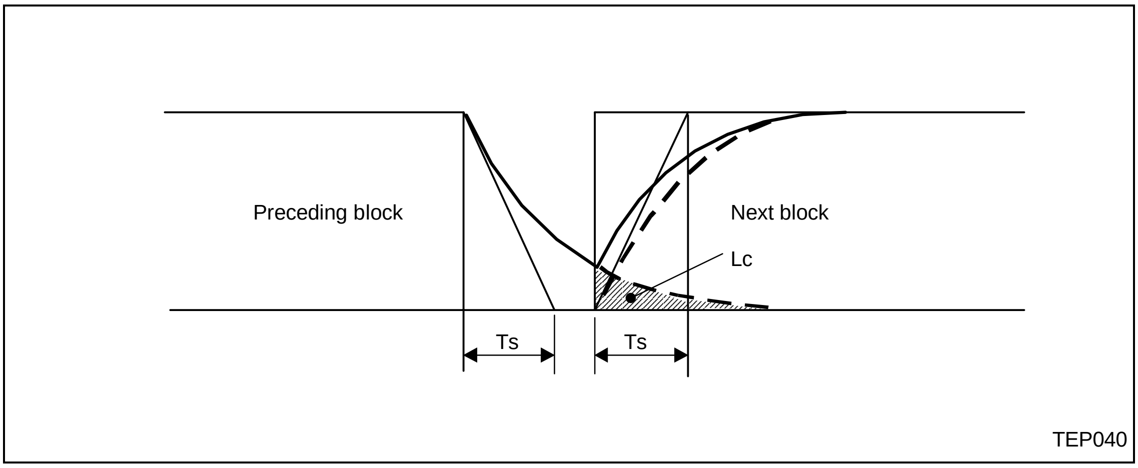


Fig. 7-3 Block-to-block connection in cutting feed in-position status check mode

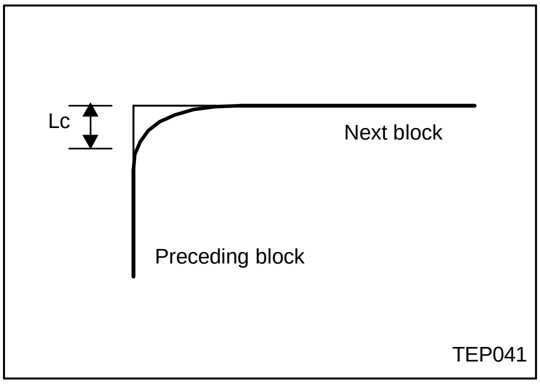
In Fig. 7-2 and 7-3 above,

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.

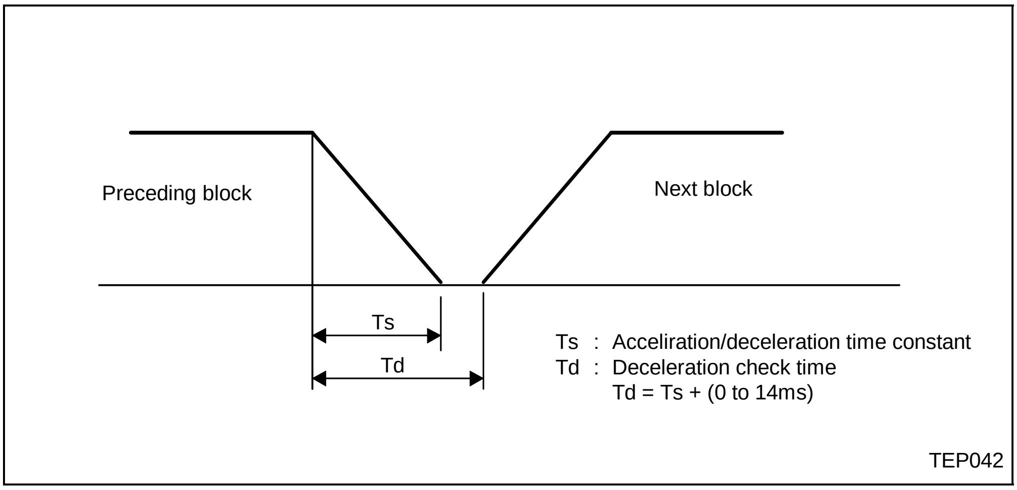
The in-position width helps keep any rounding of workpieces during corner cutting within a fixed level.

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

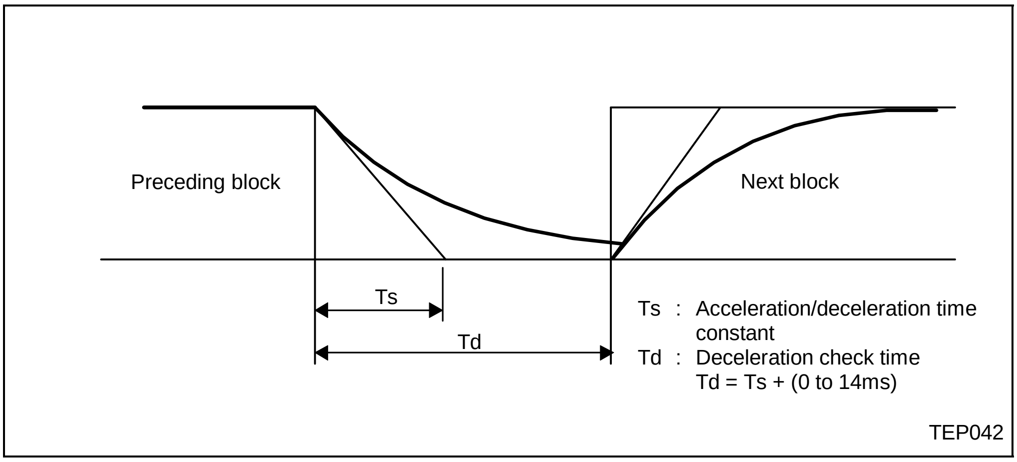


C. With deceleration check

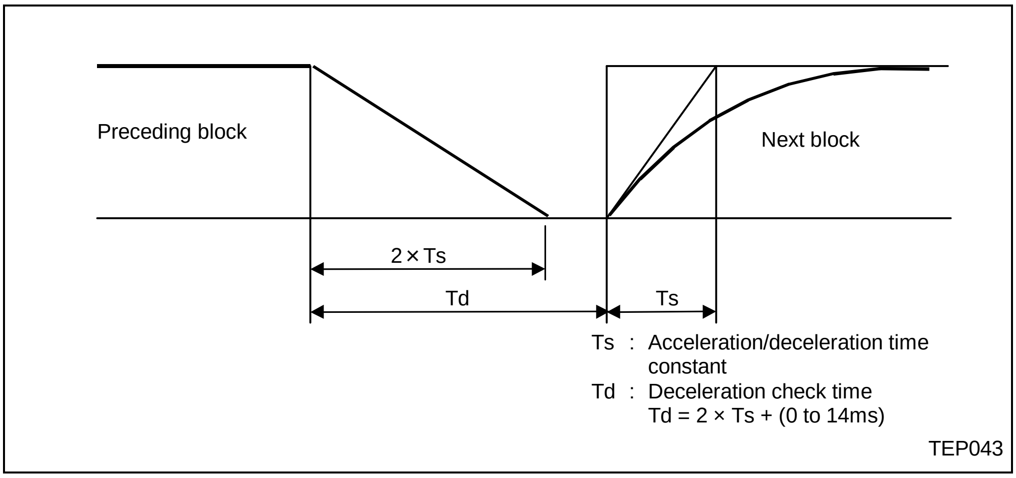
- With linear acceleration/deceleration


Preceding block
Next block
Ts
Td
Ts : Acceliration/deceleration time constant
Td : Deceleration check time
Td = Ts + (0 to 14ms)
TEP042

- With exponential acceleration/deceleration


Preceding block
Next block
Ts
Td
Ts : Acceleration/deceleration time constant
Td : Deceleration check time
Td = Ts + (0 to 14ms)
TEP042

- With exponential acceleration/linear deceleration


Preceding block
Next block
2 × Ts
Td
Ts
Ts : Acceleration/deceleration time constant
Td : Deceleration check time
Td = 2 × Ts + (0 to 14ms)
TEP043

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

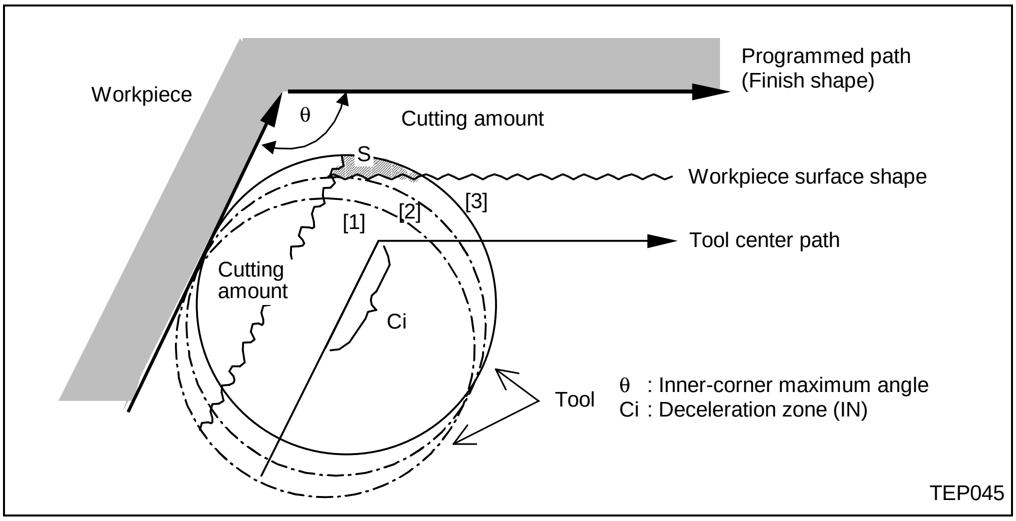
G62 ;

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.



<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.

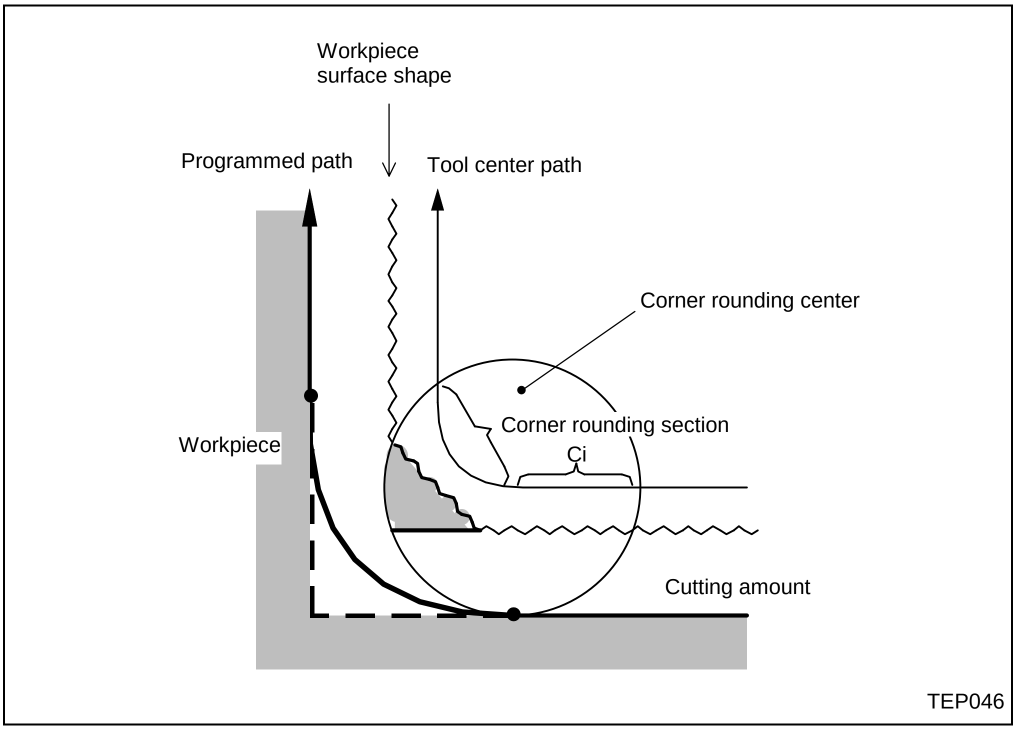
<Setting parameters>

Set the following parameters as user parameters:

- **K9**: Override	0 to 100 (%)
- **U29**: Inner-corner maximum angle θ	0 to 180 (deg)
- **U48**: 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

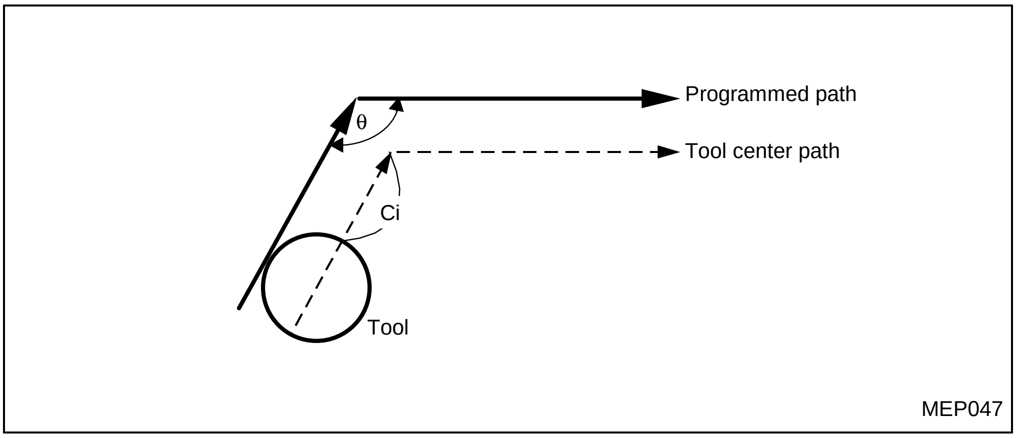


<Operation>

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).

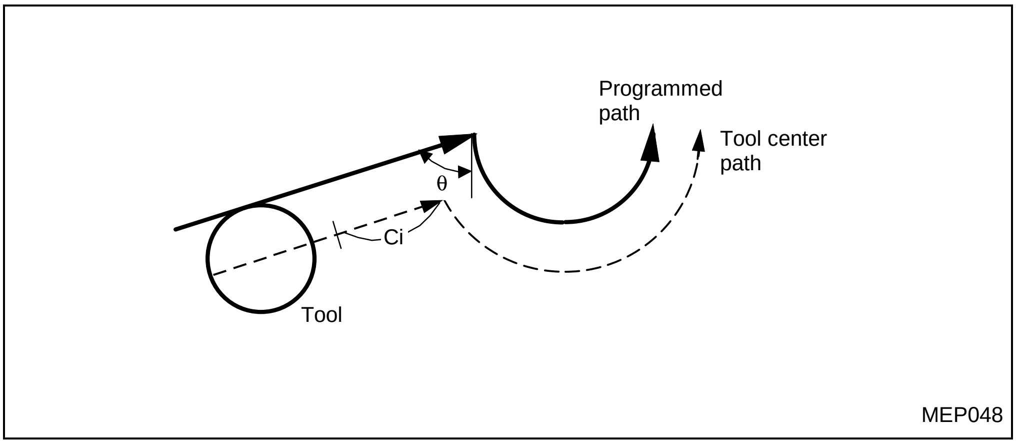
4. Operation examples

- Line-to-line corner



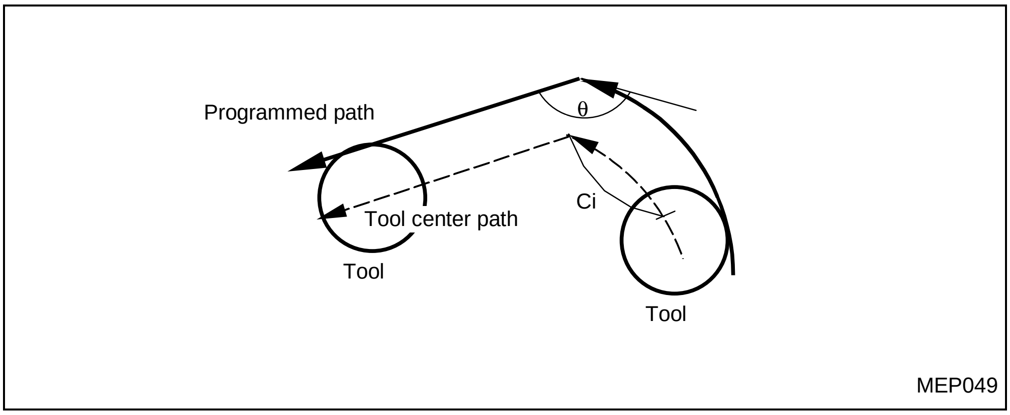
The feed rate is automatically overridden with the preset value by the parameter **K9** through deceleration zone Ci.

- Line-to-circular (outside offsetting) corner



The feed rate is automatically overridden with the preset value by the parameter **K9** through deceleration zone Ci.

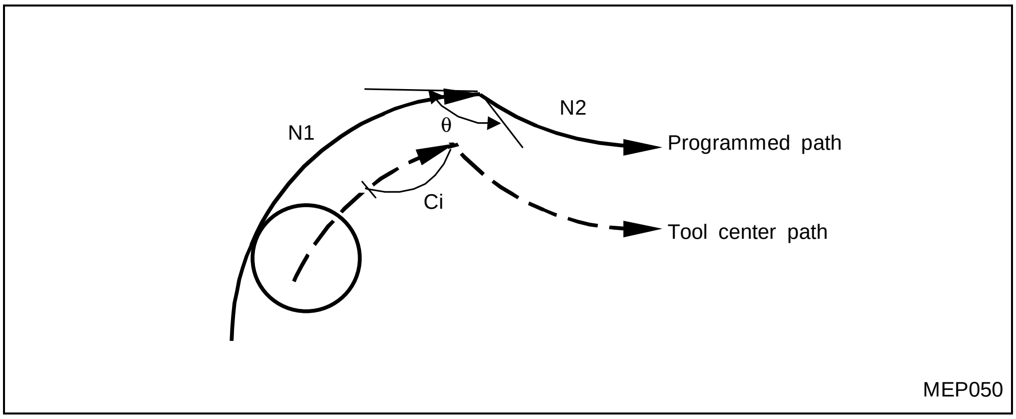
- Arc(internal compensation)-to-line corner



The feed rate is automatically overridden with the preset value by the parameter **K9** 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



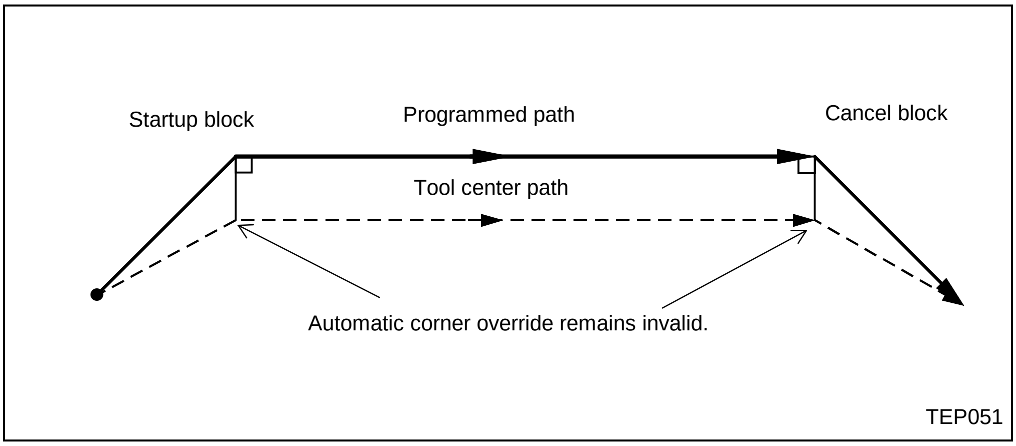
The feed rate is automatically overridden with the preset value by the parameter **K9** through deceleration zone Ci.

5. Correlationships to other command functions

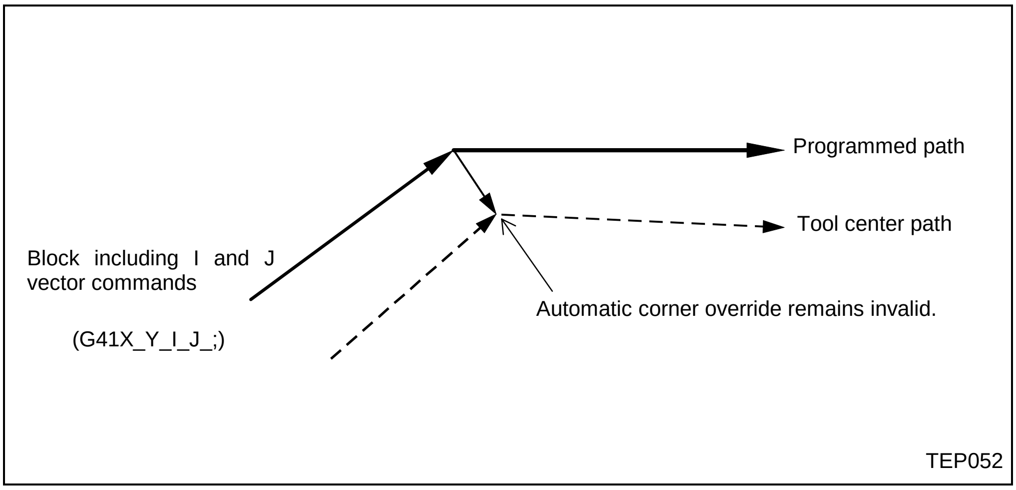
Function	Override at corners
Cutting feedrate override	Automatic corner override is applied after cutting feed override.
Override cancel	Automatic corner override is not cancelled by override cancel.
Feed rate clamp	Valid (for the feed rate after automatic corner override)
Dry run	Automatic corner override 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.


Programmed path
Tool center path
Block including I and J vector commands
(G41X_Y_I_J_;)
Automatic corner override remains invalid.
TEP052

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 Chapter 9 “GEOMETRY COMPENSATION FUNCTION” of the MAZATROL Programming 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
```

Return to Main Menu

4. Remarks

1. The geometry compensation function cannot be selected or canceled for EIA/ISO programs by the setting of the parameter **P76** (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 (**708 INCORRECT 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 **P9**), 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  ←
N300    U_W_
N400  ⋯
```

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.

<Example 2>

```
N001  G61.1
N200  G2I-10.,K30←
N300  G1U10.,K0  ←
N400  ⋯
```

Deceleration to 70% occurs for this block only.

The accuracy coefficient is canceled from this block onward.

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.

- 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

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_;

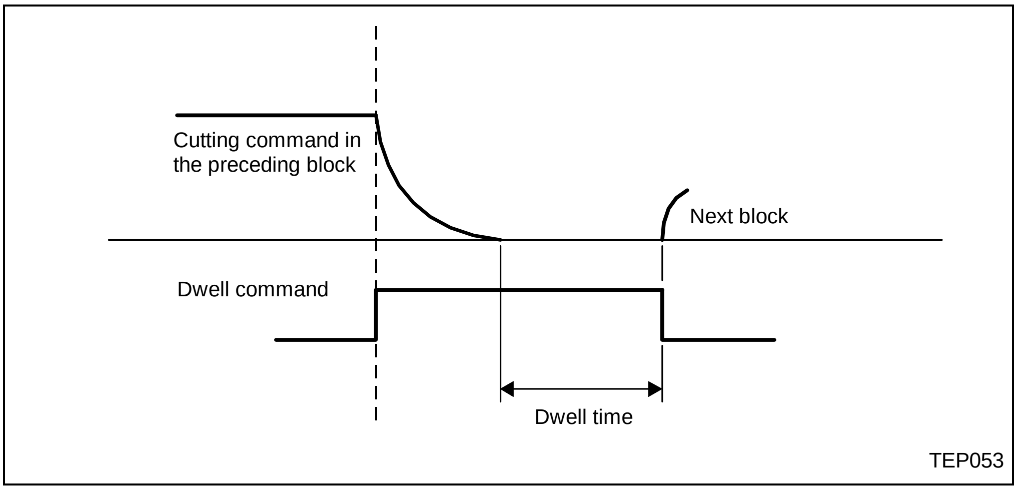
Data must be set in 0.001 seconds.
For address P, the decimal point is not available. Setting a decimal point will cause an alarm.

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. The dwell function is also valid during the machine lock mode. The dwell function, however, can be immediately terminated using bit 4 of parameter **P11**.

4. If the bit 2 of parameter **P10** 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

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_ ;

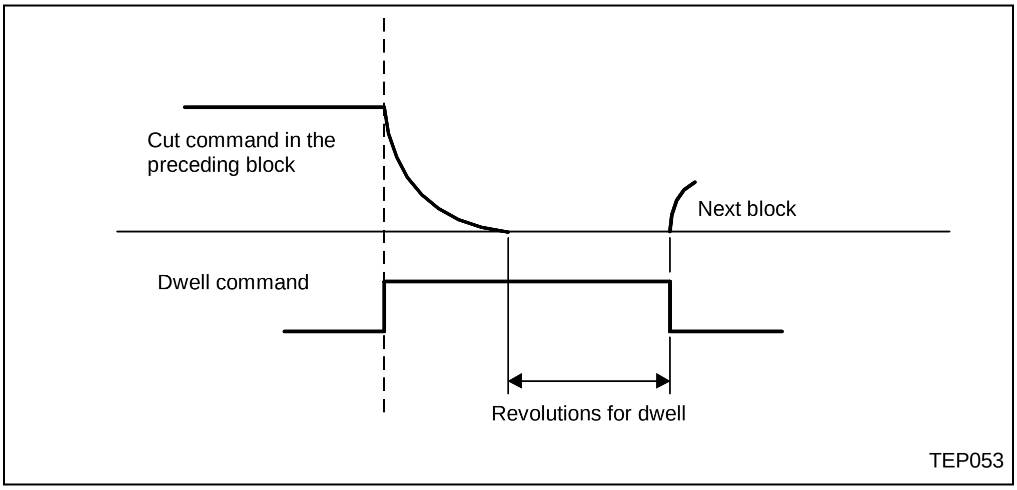
Data must be set in 0.001 revolutions.
For address P, the decimal point is not available. Setting a decimal point will cause an alarm.

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.

Return to Main Menu

3. The dwell function is also valid during the machine lock mode. The dwell function, however, can be immediately terminated using bit 4 of parameter **P11**
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 **P10** 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, M198 and M199, 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.

Return to Main Menu

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 ;       Constant peripheral speed control ON
       |    └──────── Axis for constant peripheral speed control
       └───────────── Peripheral speed
G97;                  Constant peripheral speed control OFF
```

3. Detailed description

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

   G96 Pp;

   p = 1: First axis
   p = 2: Second axis

   X-axis is provided if argument P is omitted.

Return to Main Menu

2. 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 status will be resumed.
```

4. Remarks

1. The initial modal status (G96 or G97) can be selected by parameter (**P11** bit 0).
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 spindle speed of x provides the spindle speed of G96 mode in the preceding block.
6. The peripheral speed constant control does not apply to the milling spindle.

10-3 Spindle Clamp Speed Setting: G50

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 ;
      |    └────────  Minimum spindle speed
      └─────────────  Maximum spindle speed
```

3. Detailed description

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.

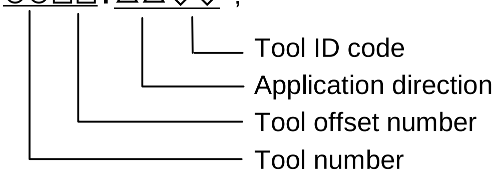
11 TOOL FUNCTIONS

11-1 Tool Function (4-Digit T-Code)

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.

The meaning of the decimal fractions depends upon the type of the NC unit.

For 640MT : T ○○□□.△△◇◇ ;



Use two digits after the decimal point to specify the application direction of the tool in accordance with the registration on the **TOOL DATA** display.

△△	0, 2, 4, 6, 8	1, 3, 5, 7, 9	11	13	14	15
DIRECTION	←	↓	→	←	↓	→

Note 1: The directions shown above may not all apply to some machine models.

Note 2: The reverse-display directions refer to a milling spindle orientation of 180 degrees, and can be used for the application of a turning tool with the tip reversely directed.

In addition to the application direction, the ID code (A to Z) of the tool can also be designated in accordance with the registration on the **TOOL DATA** display on condition that the bit 6 of the parameter **P111** is set to 1.

◇◇	1	2	3	4	5	6	7	8	9	11	12	13
ID code	A	B	C	D	E	F	G	H	J	K	L	M
◇◇	14	15	16	17	18	19	21	22	23	24	25	26
ID code	N	P	Q	R	S	T	U	V	W	X	Y	Z

Note 3: An alarm will result if no corresponding tool for the specified direction and ID code is registered on the **TOOL DATA** display.

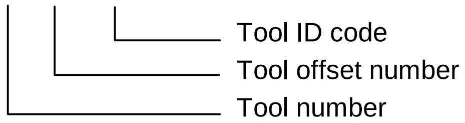
Note 4: Depending on the number of the decimal places, the T-code is processed as follows:

1) A designation of a single or two decimal places is always processed as that of the application direction. It is not possible, therefore, to merely designate the ID code.

2) The T-code with three or five and more decimal places causes an alarm.

3) The default attribute of the T-code is of "← direction without ID code".

For 640T : T ○○□□.◇◇ ;



The above ID code table also applies to the corresponding designation for the 640T.

Return to Main Menu

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:

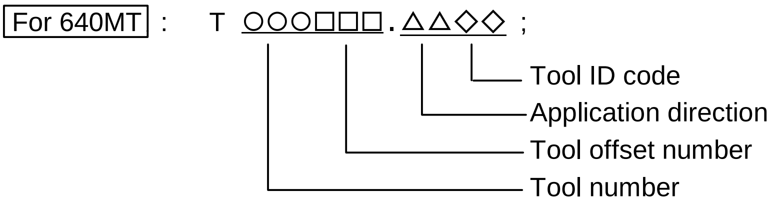
- The T-code is not executed till completion of the motion command, or
- The T-code is executed simultaneously with the motion command.

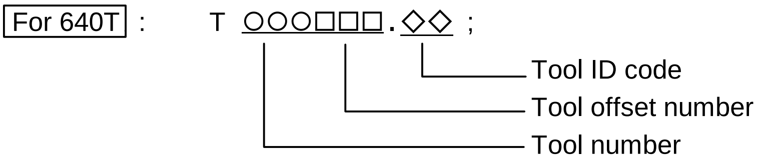
11-2 Tool Function (6-Digit T-Code)

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.

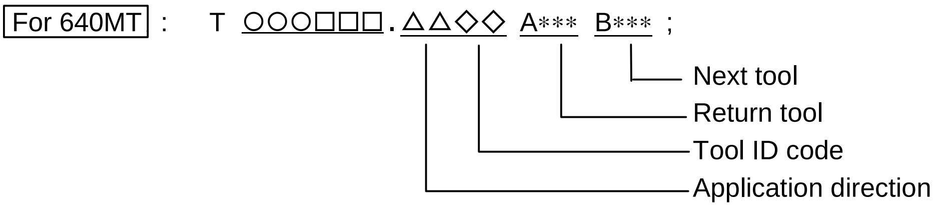

For 640MT : T ○○○□□□.△△◇◇ ;

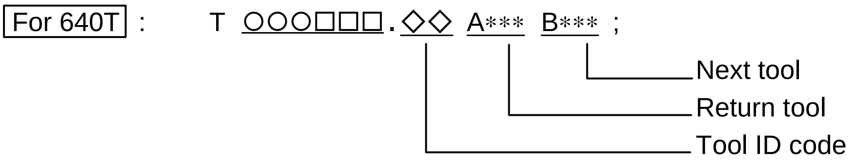

For 640T : T ○○○□□□.◇◇ ;

11-3 Next Tool Automatic Selection

You can designate return tool and a next tool for the machine provided with ATC function by commanding T-code in the format shown below. The return tool refers to an unnecessary tool returned to load another tool to be used. The assigned return tool is accomodated in the magazine. The next tool refers to a tool used for the next machining, which can be assigned when it is now 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.

See Section 11-1 for the details of the application direction and ID code.


For 640MT : T ○○○□□□.△△◇◇ A∗∗∗ B∗∗∗ ;


For 640T : T ○○○□□□.◇◇ A∗∗∗ B∗∗∗ ;

Return to Main Menu

11-4 Tool Life Management and Spare Tool Change

Tool life management is a function to have the life of a tool served when the frequency, service time or wear amount of the tool exceeds the preset value.

Spare tool change is a function to choose a spare tool to continue machining when the tool has served its life.

The method of data setting varies depending on the type of EIA/ISO program modes. When the program type of EIA/ISO is T32 compatible mode, the **TOOL DATA** display is used. When the program type of EIA/ISO is standard mode, the **TOOL LIFE** display is used.

11-4-1 Tool life management (T32 compatible mode)

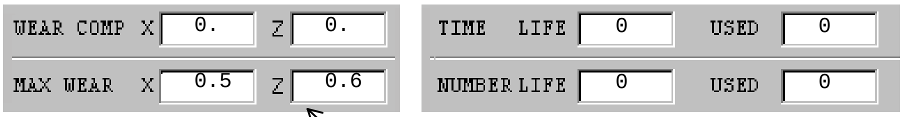
1. Tool life data setting

The life of each tool is entered in the **TOOL DATA** display beforehand. When tool life management is not required, avoid entering data.

- When tool life is managed from the wear amount, set data for MAX WEAR item.
- When tool life is managed from the machining time, set data for LIFE item of TIME.
- When tool life is managed from the number of machined workpieces, set data for LIFE item of NUMBER.

Example: For set up on the **TOOL DATA** display

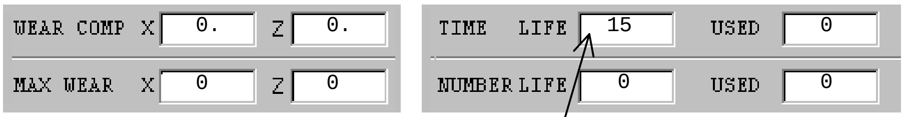
TNo.1



WEAR COMP	X	0.	Z	0.	TIME	LIFE	0	USED	0
MAX WEAR	X	0.5	Z	0.6	NUMBER	LIFE	0	USED	0

For TNo. 1, tool life is managed from wear amount.

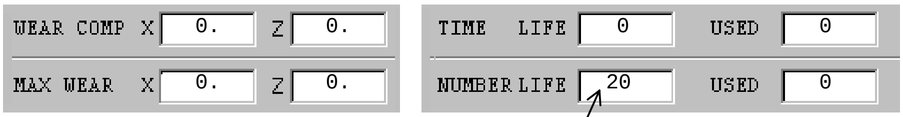
TNo.2



WEAR COMP	X	0.	Z	0.	TIME	LIFE	15	USED	0
MAX WEAR	X	0	Z	0	NUMBER	LIFE	0	USED	0

For TNo. 2, tool life is managed from machining time.

TNo.3



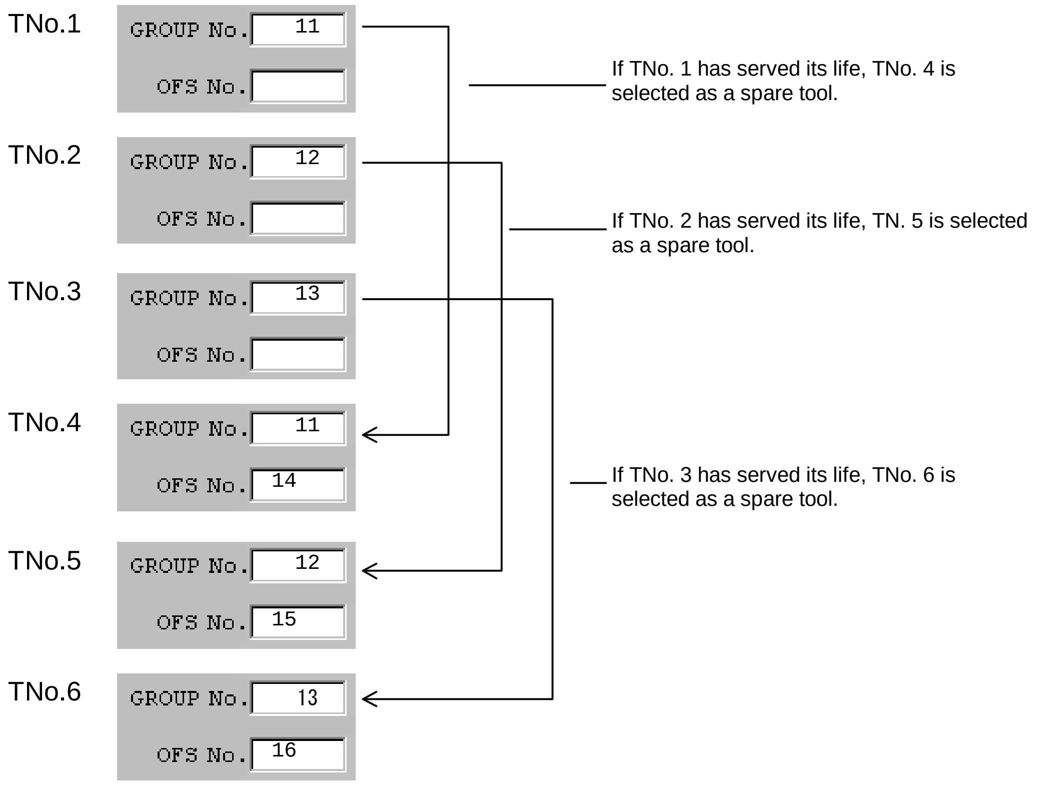
WEAR COMP	X	0.	Z	0.	TIME	LIFE	0	USED	0
MAX WEAR	X	0.	Z	0.	NUMBER	LIFE	20	USED	0

For TNo. 3, tool life is managed from number of machined workpieces.

2. Spare tool change

When a tool has served its life, spare tool change can be made by entering beforehand the spare tool in the **TOOL DATA** display. A tool capable of serving as spare tool is set with the same number of the original one in GROUP No. item of the **TOOL DATA** display. If offset number is required when the tool serves as a spare, it is also set in OFS No. item of the display.

Example: For set up on the **TOOL DATA** display



TNo.1 GROUP No. 11 OFS No.

TNo.2 GROUP No. 12 OFS No.

TNo.3 GROUP No. 13 OFS No.

TNo.4 GROUP No. 11 OFS No. 14

TNo.5 GROUP No. 12 OFS No. 15

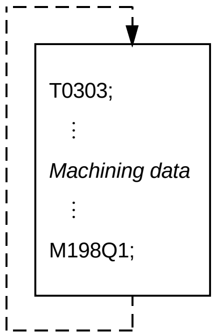
TNo.6 GROUP No. 13 OFS No. 16

If TNo. 1 has served its life, TNo. 4 is selected as a spare tool.

If TNo. 2 has served its life, TN. 5 is selected as a spare tool.

If TNo. 3 has served its life, TNo. 6 is selected as a spare tool.

Example of operation with the above data registered on the **TOOL DATA** display:



```
T0303;
  ⋮
Machining data
  ⋮
M198Q1;
```

Machining 20 workpieces with the program given left leads TNo. 3 to serve its life. Since TNo. 3 and TNo. 6 are connected by the same group number, TNo. 6 is selected as a spare tool when TNo. 3 has served its life. The offset number of TNo. 6 is 16, therefore T0303 will then act as T0616.

3. Remarks

The data which the tool life management should refer to is designated by parameter **P18**.

P18 = 0	Any data
P18 = 1	Number of machined workpieces
P18 = 2	Machining time
P18 = 4	Wear amount X
P18 = 8	Wear amount Z
P18 = 16	Wear amount Y

To manage the tool life by number of machined workpieces together with machining time, for example, add both parameter values and set this added value as follows:
1 (number of machined workpieces) + 2 (machining time) = 3,
therefore, set **P18** to 3.

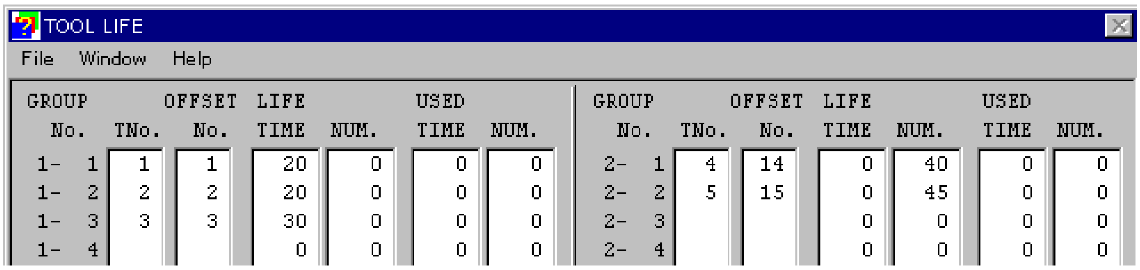
Return to Main Menu

11-4-2 Tool life management (Standard mode)

1. Tool life data setting

Tools of each group to be used in sequence and their life are entered in the **TOOL LIFE** display beforehand.

Example: TOOL LIFE display



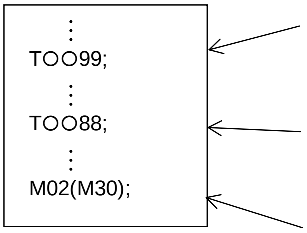
GROUP No.	TNo.	OFFSET No.	LIFE TIME	NUM.	USED TIME	NUM.	GROUP No.	TNo.	OFFSET No.	LIFE TIME	NUM.	USED TIME	NUM.
1- 1	1	1	20	0	0	0	2- 1	4	14	0	40	0	0
1- 2	2	2	20	0	0	0	2- 2	5	15	0	45	0	0
1- 3	3	3	30	0	0	0	2- 3			0	0	0	0
1- 4			0	0	0	0	2- 4			0	0	0	0

When the tool of TNo. 1 and OFFSET No. 1 has served its life, it is changed for the tool of TNo. 2 and OFFSET No. 2 as a spare tool. And also, when the tool of TNo. 2 and OFFSET No. 2 has served its life, it is changed for the tool of TNo. 3 and OFFSET No. 3.

2. Specifying tool groups in a machining program

In a machining program, tool group and others are specified using T-codes as follows.

Programming format



```
    :
TOO99;
    :
TOO88;
    :
M02(M30);
```

Use of the tool that has been used is finished, and use of a tool of OO group is started.
99 represents a differentiation from general command.

Offset cancel of a tool of OO group.
88 represents a differentiation from general command.

End of machining program.

Note: 88 and 99 cannot be used as the tool offset number.

3. Count of tool life

The count of the life counters is executed for each group, and the contents of the counters are not erased even after the power is turned off.

1. When the life is specified by time (minutes):
   In a machining program, the time for which a tool is actually used in cutting mode between the time TOO99 (OO: Tool group No.) was commanded and the time TOO88 is commanded is counted every second. The time taken for single block stop, feed hold, rapid feed, dwell, waiting for FIN, etc. is ignored. The maximum value of life capable of setting is 9999 (min.).
2. When the life is specified by frequency:
   The counter of the used tool group increases its number by 1 for every process between the time a machining program was started and the time M02 or M30 is commanded to put the NC into a reset state. No matter how many times the same group is commanded in a process, the counter only increases its number by 1. The maximum value of life capable of setting is 9999 (times).

Note: For the management of life by frequency, set the related parameter as required (**P106** bit 1 = 0).

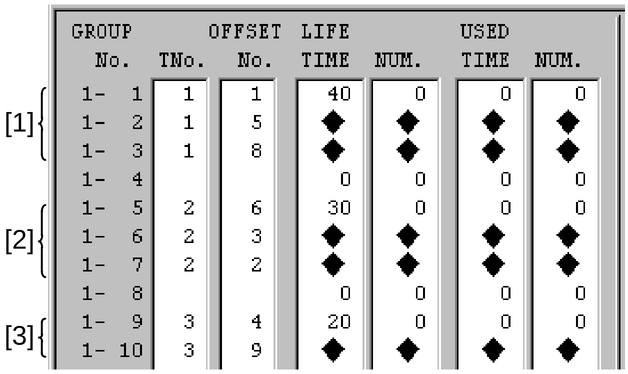
4. Remarks

1. By which data the tool life is to be managed is set by parameter **P18**.
2. The number of groups and the number of tools per group that can be registered are combined by selecting one of the following four ways of parameter setting.

Parameter **K72**	Number of groups	Number of tools per group
0	16	16
1	32	8
2	64	4
3	128	2

   In any combination, up to 256 tools can be registered. To change a combination, change the parameter before erasing the **TOOL LIFE** display totally, and reset the tool group. If the tool groups are not reset, former tool groups are selected.

3. The same TNo. can appear at any time and at any place.
4. Group number does not have to be set in sequence. It is also not necessary to set all the groups.
5. When two or more tool offset numbers are used for the same tool in a process, setting must be done as follows:
   (When TNos. are set in sequence, LIFE and USED items on the second and subsequent lines provide ◆.)



```
GROUP        OFFSET  LIFE          USED
  No.   TNo.   No.   TIME  NUM.    TIME  NUM.
 1-  1    1     1     40     0       0     0
 1-  2    1     5     ◆      ◆       ◆     ◆
 1-  3    1     8     ◆      ◆       ◆     ◆
 1-  4                 0     0       0     0
 1-  5    2     6     30     0       0     0
 1-  6    2     3     ◆      ◆       ◆     ◆
 1-  7    2     2     ◆      ◆       ◆     ◆
 1-  8                 0     0       0     0
 1-  9    3     4     20     0       0     0
 1- 10    3     9     ◆      ◆       ◆     ◆
```

- Tools of GROUP No. 1 are used in order of [1], [2] and [3] for 40, 30 and 20 minutes respectively.
  OFFSET Nos., when this group is specified three times in a process, are selected in the following order.

  For [1]   1 → 5 → 8
  For [2]   6 → 3 → 2
  For [3]   4 → 9 → 9

12 TOOL OFFSET FUNCTIONS

12-1 Tool Offset

1. Outline

Tool offset must be set using four-digits or six-digits numerical data preceded by address T. Whether the offset number is set by lower two or three digits is selected by parameter **P10** 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/G52 (MAZATROL coordinate system selection/cancel) and parameter **P10** bit 3 (MAZATROL tool wear offset data valid/invalid) as in the following table.

Program Parameter	G53 (MAZATROL coordinate system)		G52 (Cancellation of MAZATROL coord. sys.)	
	T 0 1 0 0 [1] [2]	T 0 1 0 1 [1] [2]’	T 0 1 0 0 [1] [2]	T 0 1 0 1 [1] [2]’
P10 bit 3 = 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
P10 bit 3 = 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.*

Note: Tool offset data are to be registered separately for each headstock (for two-line system) or each turret (for SQR series).

2. Tool offset number designation

The tool offset number is set using the lower two or three digits. Whether the offset number is set by lower two or three digits is selected by parameter **P10** bit 4.

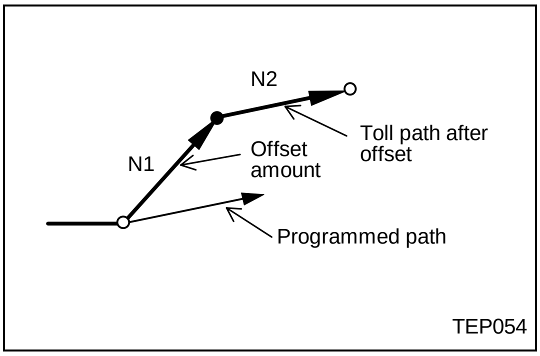
3. Tool offset start

There are two ways to execute tool offset and these can be selected by parameter **P13** bit 0: 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 T0101;
N2 X100.Z200.;
```

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



TEP054

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.

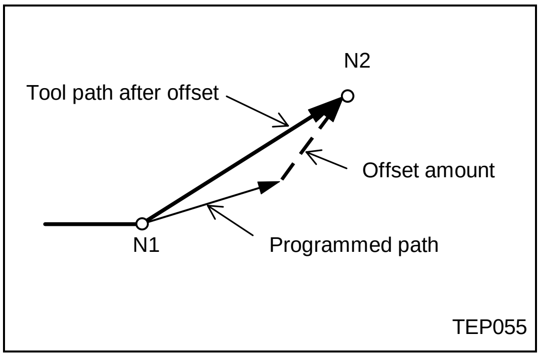
Note 3: When performing offset in T command execution, offset will not function until the arrival of any command G except those listed below when the T command is included in the same block as those commands G.

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

B. Offset with move command

```
N1 T0101;
N2 X100.Z200.;
```

Tool offset is executed simultaneously.



TEP055

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.)

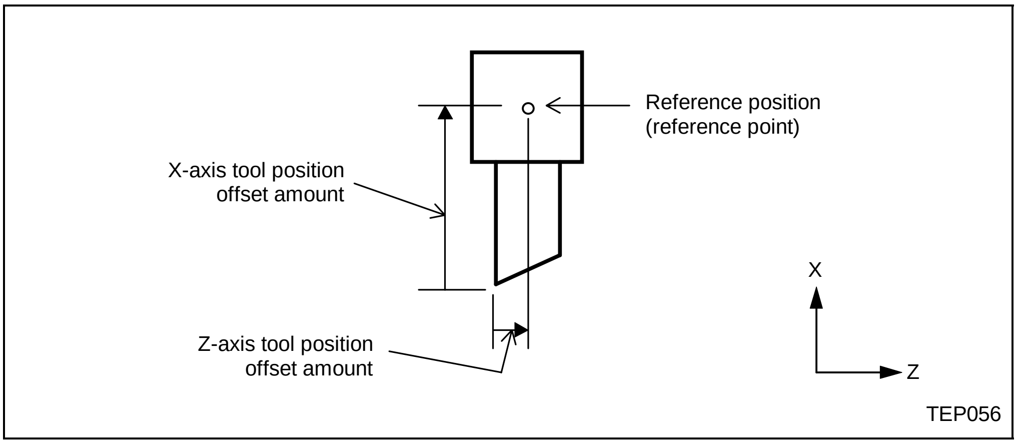
Return to Main Menu

12-2 Tool Position Offset

1. Tool position offset amount setting

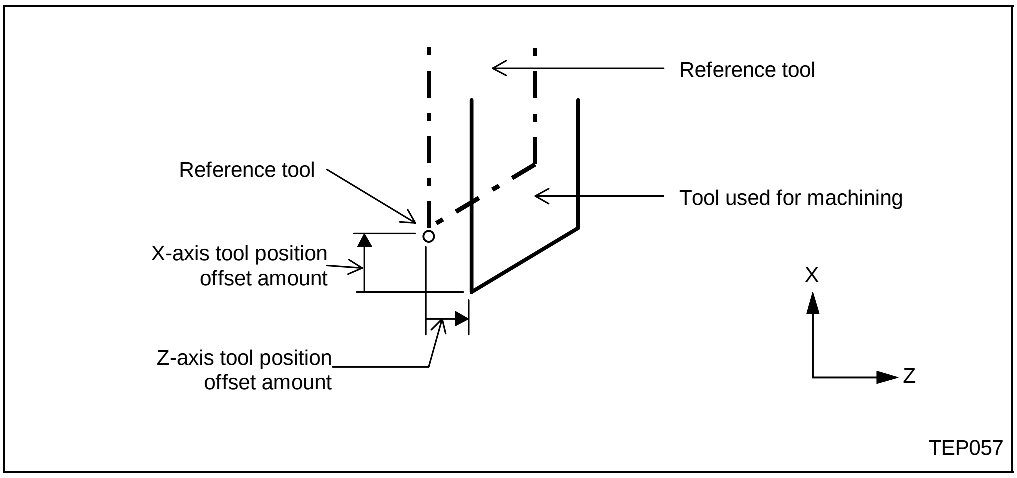
This function offsets tool position with respect to the program reference position. This position may generally be set to either the center position of the turret or the tool nose position of the reference tool.

A. Setting to the center position of turret



TEP056

B. Setting to the tool nose position of reference tool



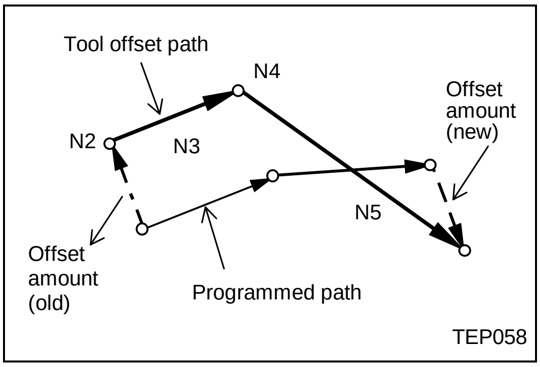
TEP057

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  T0100;
N2  G1 X10.0 Z10.0 F100;
N3  G1 X13.0 Z15.0 F20.0;
N4  T0200;
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

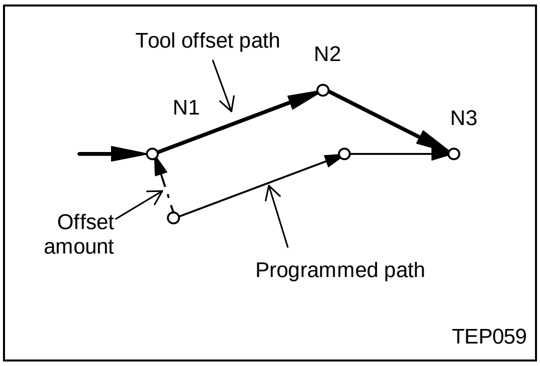
3. Tool position offset cancel

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 (only when parameter **P15** bit 5 = 1).

```
N1  X10.0 Z10.0 F10;
N2  T0000;
N3  G1 X10.0 Z20.0;
```

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

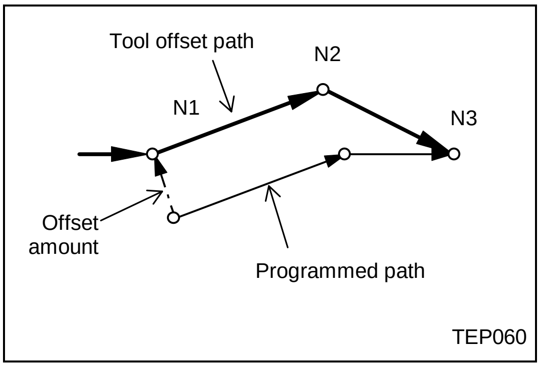


B. When 0 is set as the offset amount

Offset is cancelled when 0 is set as the offset amount of the tool position offset number (Only when parameter **P15** bit 5 = 1).

```
N1  G1 X10.0 Z10.0 F10;
N2  T0100;
N3  G1 X10.0 Z20.0;
```

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



4. Remarks

1. 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.
2. The tool position offset is cleared by resetting and by emergency stop.

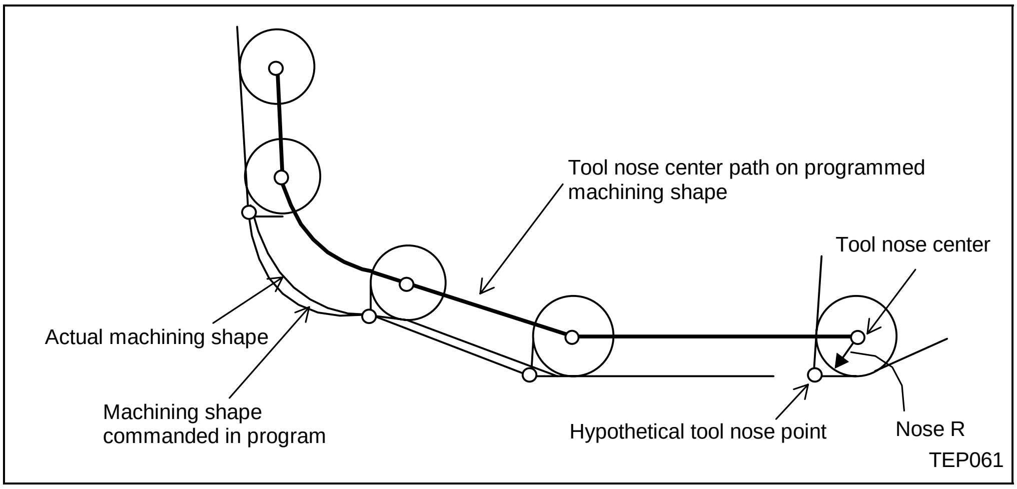
12-3 Tool Nose Radius Compensation: G40, G41, G42, G46

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. Tool nose radius compensation is a function for automatically calculating and offsetting this error by setting the tool nose radius value.

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

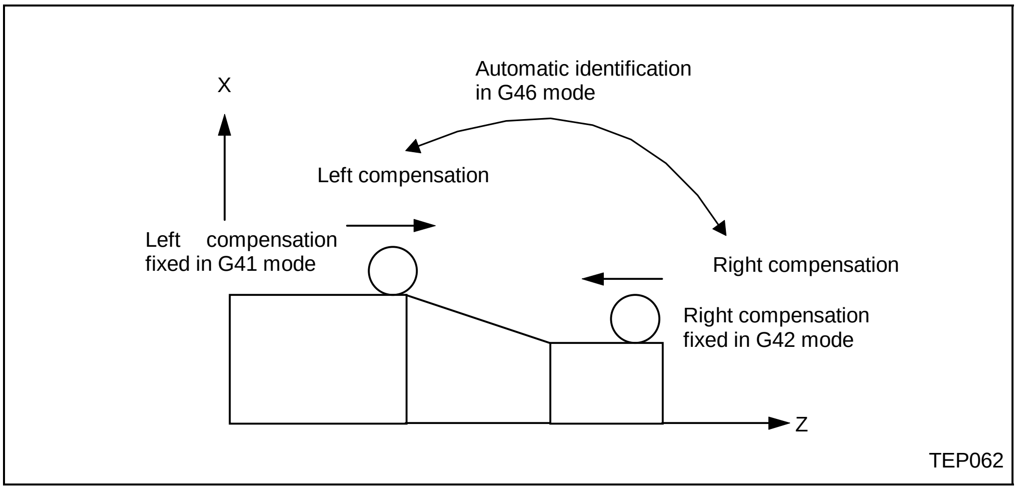

Tool nose center path on programmed machining shape
Tool nose center
Actual machining shape
Machining shape commanded in program
Hypothetical tool nose point
Nose R
TEP061

2. Programming format

Code	Function	Programming format
G40	Tool nose radius compensation mode cancel	G40 Xx/Uu Zz/Ww Ii Kk ;
G41	Tool nose radius compensation left mode ON	G41 Xx/Uu Zz/Ww ;
G42	Tool nose radius compensation right mode ON	G42 Xx/Uu Zz/Ww ;
G46	Tool nose radius compensation automatic direction identification mode ON	G46 Xx/Uu Zz/Ww ;

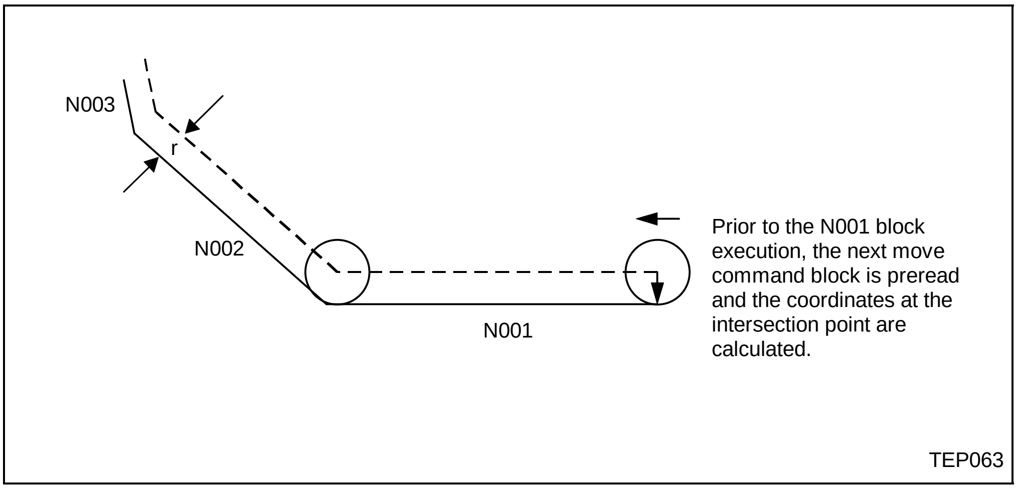
3. Detailed description

1. By means of the preset hypothetical tool nose point and move commands in the machining program, the G46 tool nose radius compensation function automatically identifies the compensation direction and executes tool nose radius compensation.



2. G40 serves to cancel the tool nose radius compensation mode.

3. 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).

4. The tool nose radius compensation amount corresponds to the tool length number and it should be preset with the tool nose point.

5. 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.

6. 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.

7. With threading commands, compensation is temporarily cancelled in one block before.

8. A tool nose radius compensation (G41 or G42) command can be set during tool nose radius compensation (G46). There is no need to cancel the compensation mode with G40.

9. The compensation plane, move axes and next advance direction vector follow the plane selection command designated by G17, G18 or G19.

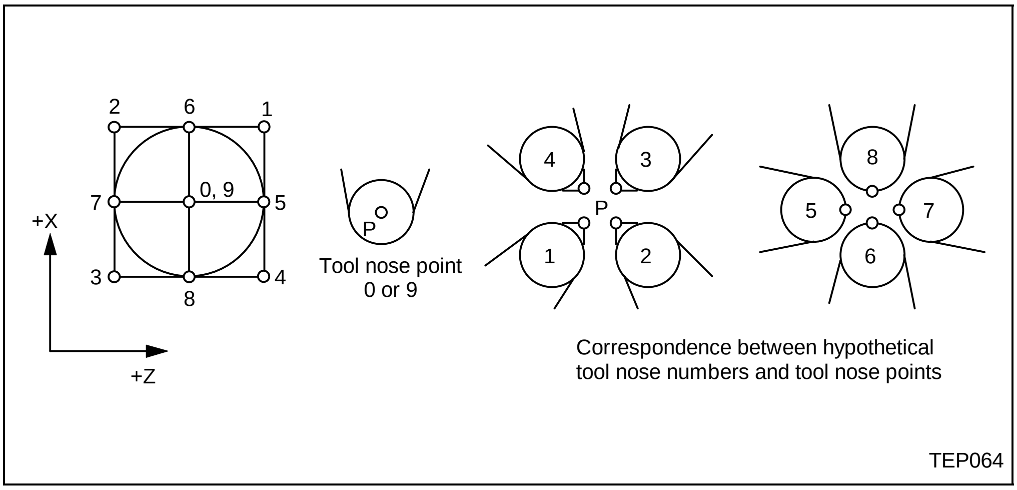
G17 .... XY plane	X, Y;	I, J
G18 .... ZX plane	Z, X;	K, I
G19 .... YZ plane	Y, Z;	J, K

12-3-2 Tool nose point and compensation directions

1. Tool nose point

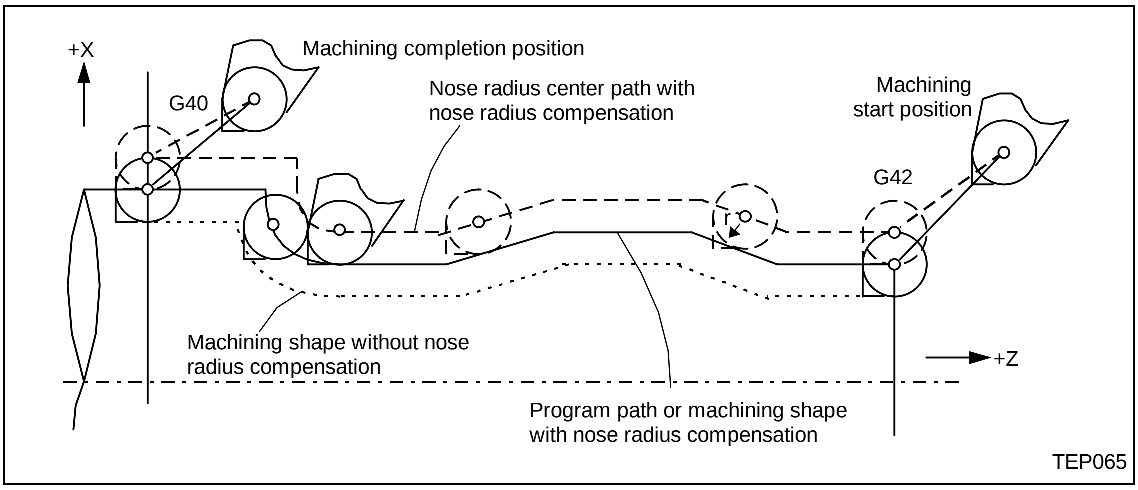
Since the tool nose is generally rounded, the programmed tool nose position is aligned with point P shown in the examples of the figures below.

For tool nose radius compensation, select one point among those in the figures below for each tool length number and preset. (Selection from points 1 through 8 in the G46 mode and 0 through 9 in the G41/G42 mode.)

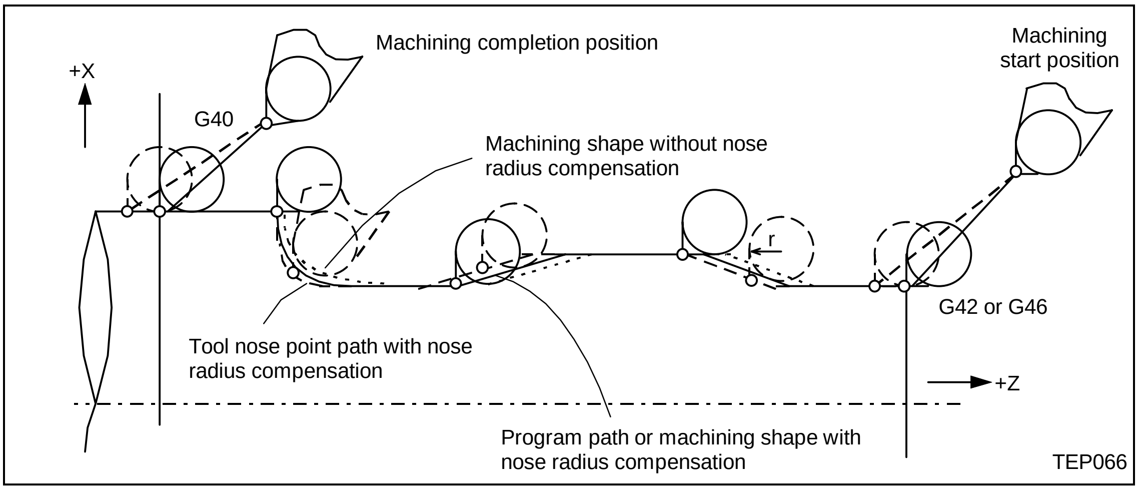

2
6
1
7
0, 9
5
3
8
4
+X
+Z
P
Tool nose point
0 or 9
4
3
P
1
2
8
5
7
6
Correspondence between hypothetical tool nose numbers and tool nose points
TEP064

2. Tool nose point and compensation operation

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


+X
G40
Machining completion position
Nose radius center path with nose radius compensation
Machining start position
G42
r
Machining shape without nose radius compensation
+Z
Program path or machining shape with nose radius compensation
TEP065

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


Machining completion position
Machining start position
+X
G40
Machining shape without nose radius compensation
r
G42 or G46
Tool nose point path with nose radius compensation
+Z
Program path or machining shape with nose radius compensation
TEP066

3. Compensation directions

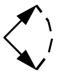
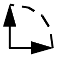
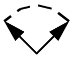
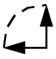
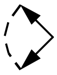
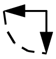
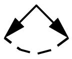
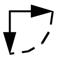
The compensation direction of the commands G41/G42 is determined by the G41/G42 codes. The compensation direction in a command G46 is automatically determined from the relationship between the tool nose points and the commanded movement vectors as shown in the table below.

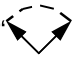
When nose radius compensation is started and the initial movement vector (including G0) corresponds to an “ × ” mark in the table, the compensation direction cannot be specified and so it is determined by the next movement vector. When the direction cannot be determined even after reading 5 blocks ahead, a program error wil occur.

When the offset direction is reversed during nose radius compensation, a program error will occur except when the reversal is done in the G00 block. Even if the compensation directions differ before and after the G28, G30, G30.1 or G53 block, the program error will not occur since compensation is temporarily cancelled. Using parameter **P13** bit 7, the tool can also be moved in the same compensation direction.

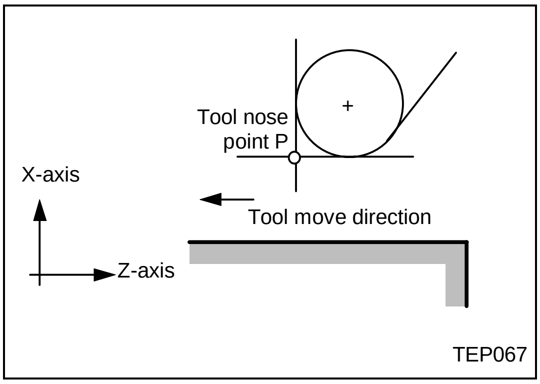
When the compensation direction during nose radius compensation coincides with an “ × ” mark in the table below, the direction becomes the previous compensation direction.

Determining the offset direction by the movement vectors and tool nose point in G46 command

Offset direction of tool nose / Direction of tool nose advance		Tool nose points								Offset direction of tool nose / Direction of tool nose advance	
		①	②	③	④	⑤	⑥	⑦	⑧		
Movement vector (tool nose points 1 to 4)		Right	Right	Left	Left	×	Right	×	Left	→	Movement vector (tool nose points 5 to 8)
	↗	×	Right	×	Left	Left	Right	Right	Left		
		Left	Right	Right	Left	Left	×	Right	×	↑	
	↖	Left	×	Right	×	Left	Left	Right	Right		
		Left	Left	Right	Right	×	Left	×	Right	←	
	↙	×	Left	×	Right	Right	Left	Left	Right		
		Right	Left	Left	Right	Right	×	Left	×	↓	
	↘	Right	×	Left	×	Right	Right	Left	Left		

- An “ } ” mark in the table indicates that the offset direction is not determined by the movement vector and the tool nose point.
- The “↗ ” mark denotes a movement vector in the 45° direction.
  (The other movement vectors are based on this.)
- The “” mark denotes a movement vector in a range larger than 45° and smaller than 135°.
  (The other movement vectors are based on this.)

Example: With tool nose point 3, move vector in the Z-axis (–) direction (with ← move vector)



As shown in the figure above, the workpiece is on the X-axis (–) side from the above tool nose position and tool move direction. Consequently, the compensation in the right side of the workpiece direction seen from the tool advance direction will be realized.

Return to Main Menu

12-3-3 Tool nose radius compensation operations

1. Tool nose radius compensation cancellation

Tool nose 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 tool nose radius compensation

Tool nose radius compensation will begin when all the following conditions are met:

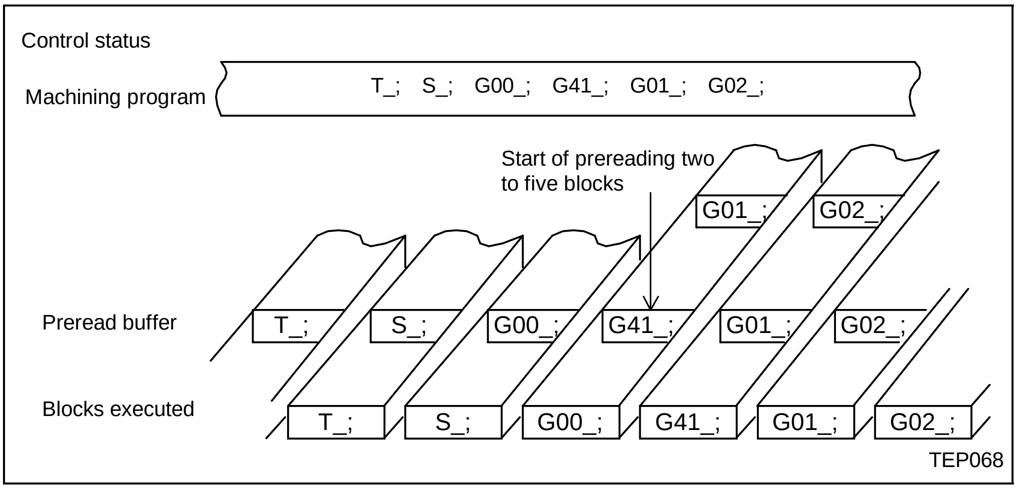
- Command G41, G42 or G46 has been executed.
- The command used with the offsetting command is a move command other than those used for arc interpolation.

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

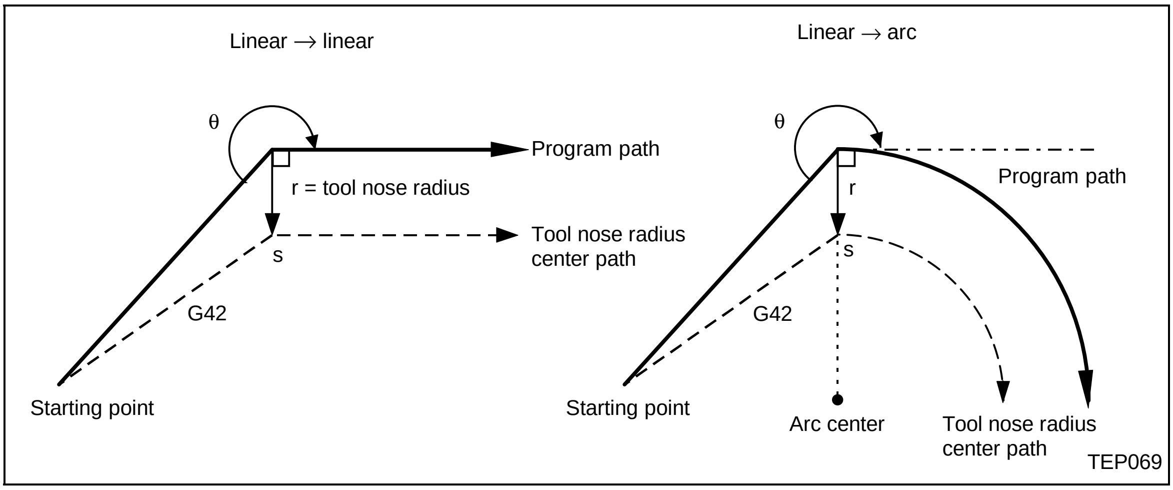
If error is caused because of the reason above, provide serveral blocks including move commands after G41, G42, G46, or T command.


Control status
Machining program
T_; S_; G00_; G41_; G01_; G02_;
Start of prereading two to five blocks
G01_; G02_;
Preread buffer
T_; S_; G00_; G41_; G01_; G02_;
Blocks executed
T_; S_; G00_; G41_; G01_; G02_;
TEP068

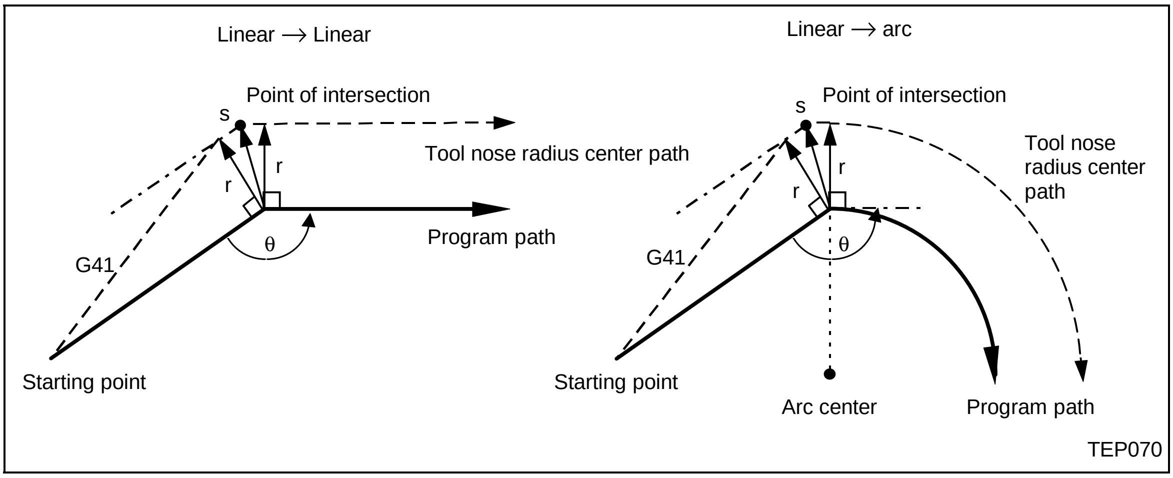
3. Start operation for tool nose radius compensation

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

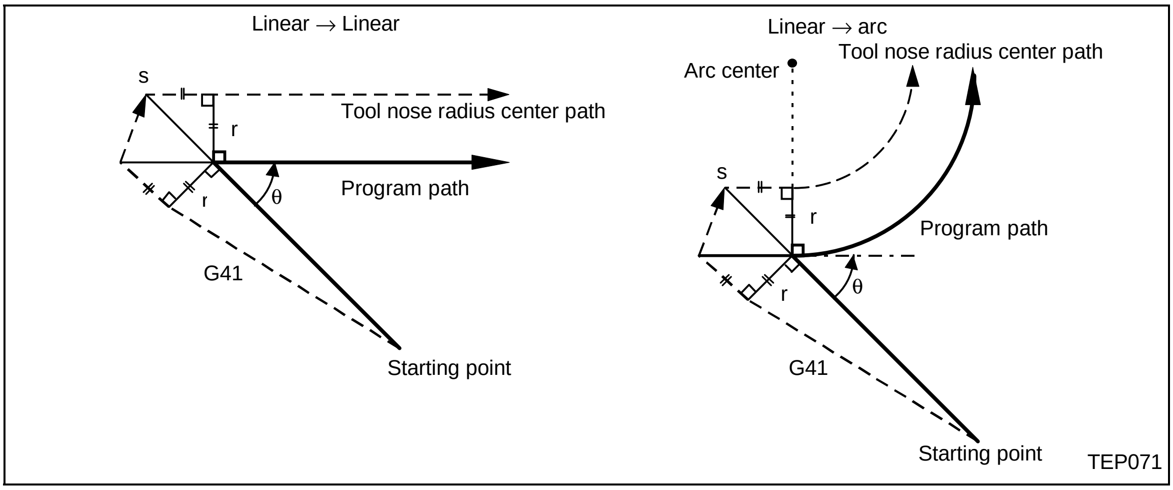
A. For the corner interior



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



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



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.

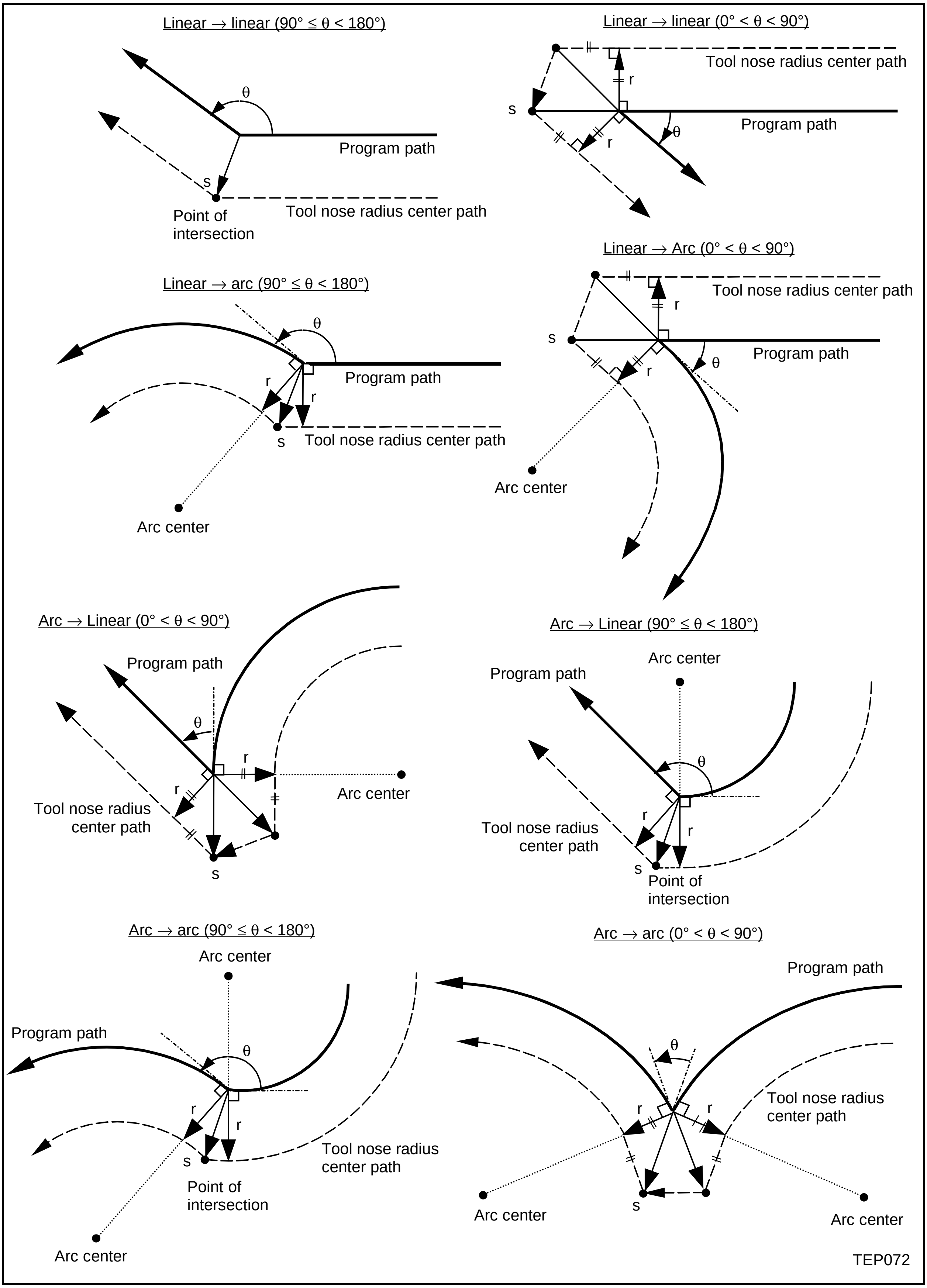
Even if the same compensation command G41/G42/G46 is set in a tool nose radius compensation G41/G42/G46 mode, the command will be ignored.

When four or more blocks not including move command are commanded in the compensation mode, overcutting or undercutting will result.

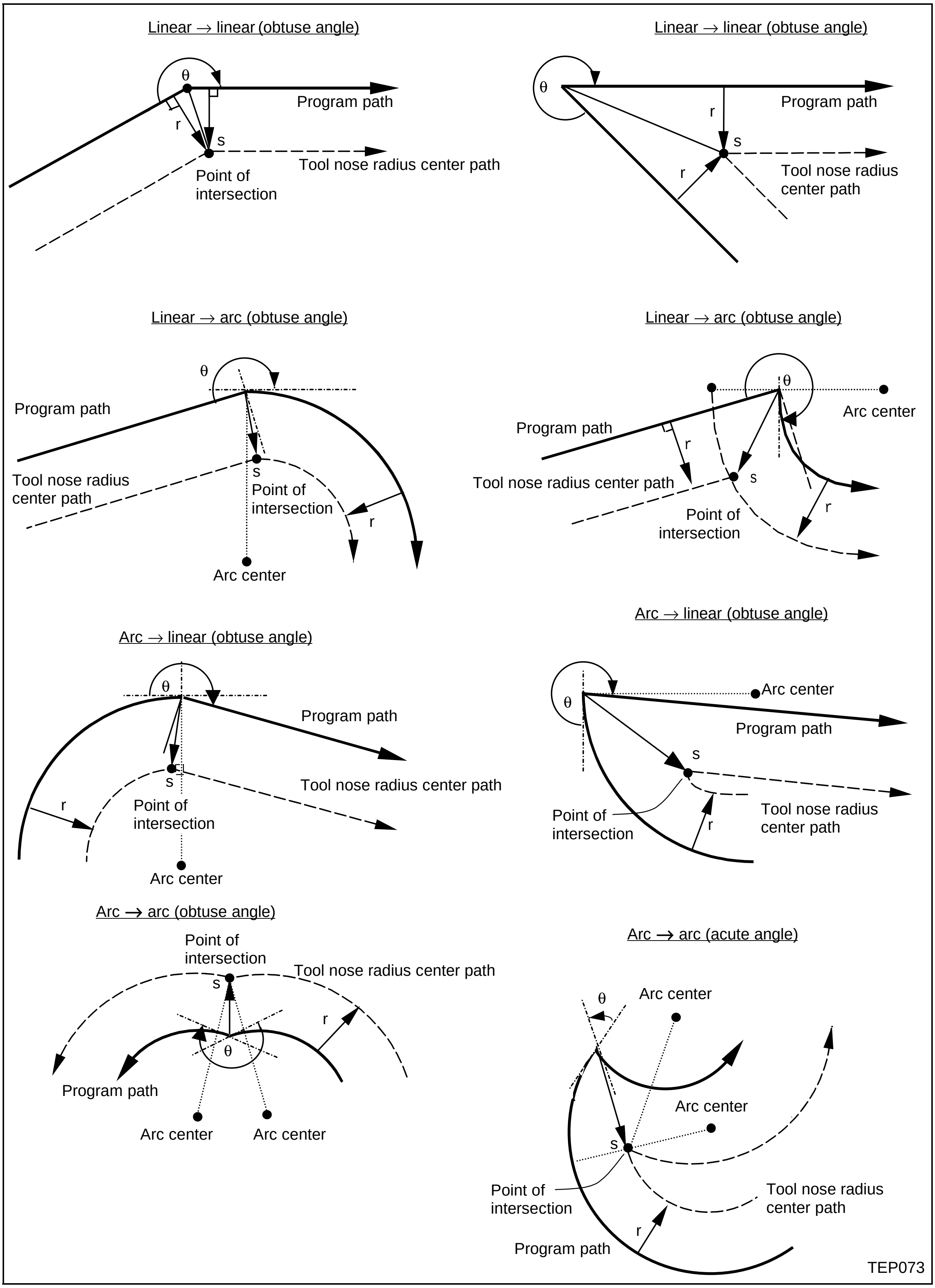
When the M00 command has been set during tool nose radius compensation, pre-reading is prohibited.

Return to Main Menu

A. For the corner exterior

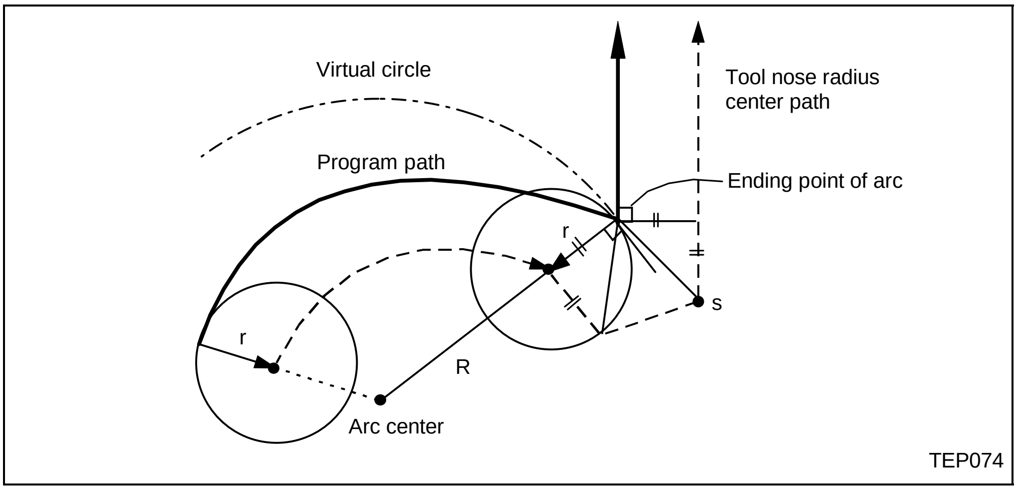

Linear → linear (90° ≤ θ < 180°)
θ
Program path
s
Point of intersection
Tool nose radius center path
Linear → linear (0° < θ < 90°)
Tool nose radius center path
r
s
Program path
θ
r
Linear → arc (90° ≤ θ < 180°)
θ
Program path
r
r
s
Tool nose radius center path
Arc center
Linear → Arc (0° < θ < 90°)
Tool nose radius center path
r
s
Program path
θ
r
Arc center
Arc → Linear (0° < θ < 90°)
Program path
θ
r
r
Tool nose radius center path
Arc center
s
Arc → Linear (90° ≤ θ < 180°)
Arc center
Program path
θ
r
r
Tool nose radius center path
s
Point of intersection
Arc → arc (90° ≤ θ < 180°)
Arc center
Program path
θ
r
r
s
Point of intersection
Tool nose radius center path
Arc center
Arc → arc (0° < θ < 90°)
Program path
θ
r
r
Tool nose radius center path
Arc center
s
Arc center
TEP072

B. For the corner interior



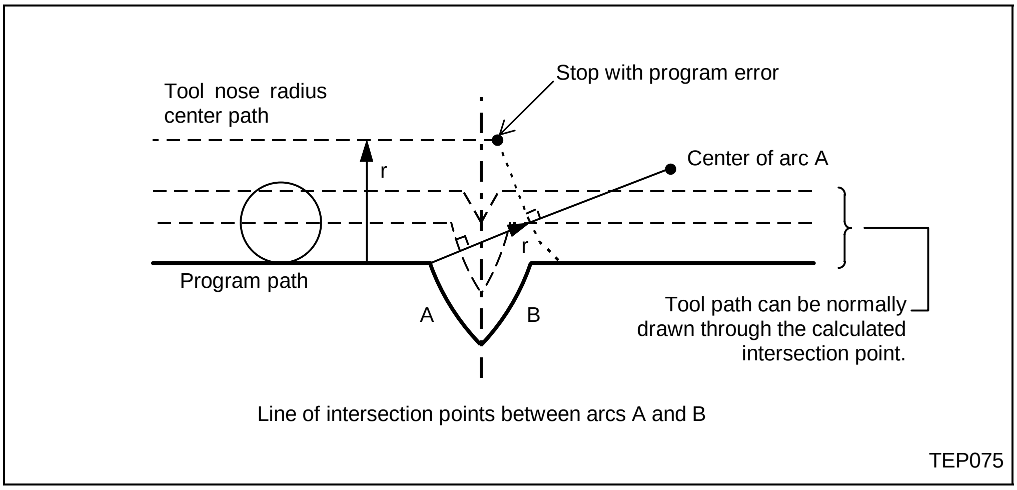
C. For the arc on which the ending point is not found

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.



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.


Stop with program error
Tool nose radius center path
Center of arc A
r
r
Program path
A
B
Tool path can be normally drawn through the calculated intersection point.
Line of intersection points between arcs A and B
TEP075

E. Tool nose radius compensation cancel

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

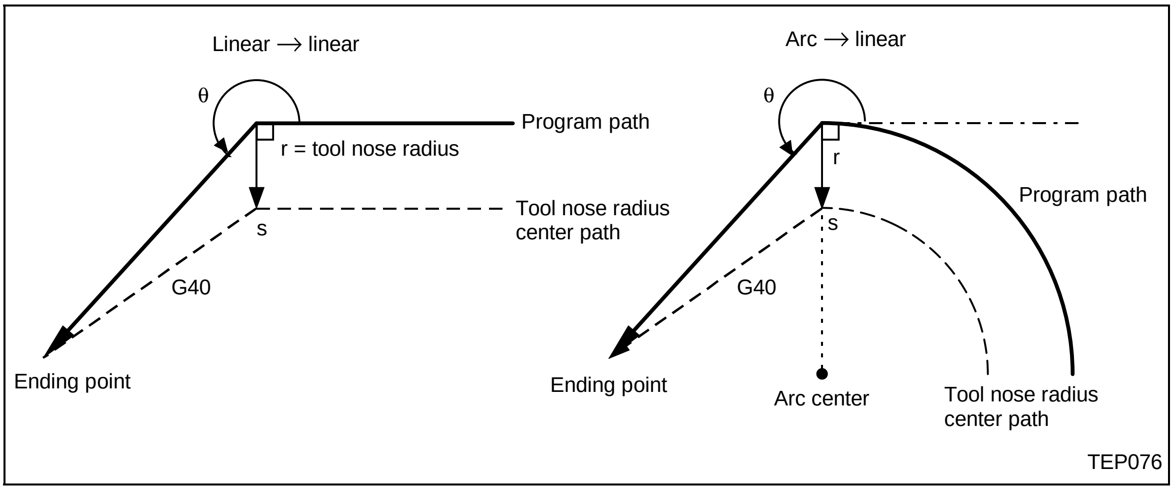
- Command G40 has been executed.
- Tool number T00 has been executed.

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.

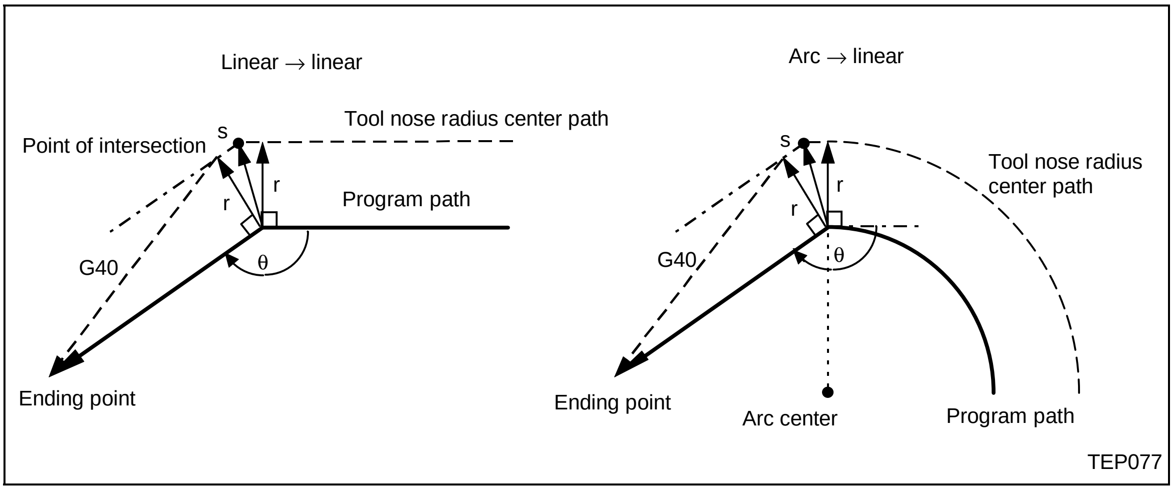
The cancel mode is established once the compensation cancel command has been read, five-blocks prereading is suspended and one-block pre-reading is made operational.

5. Tool nose radius compensation cancel operation

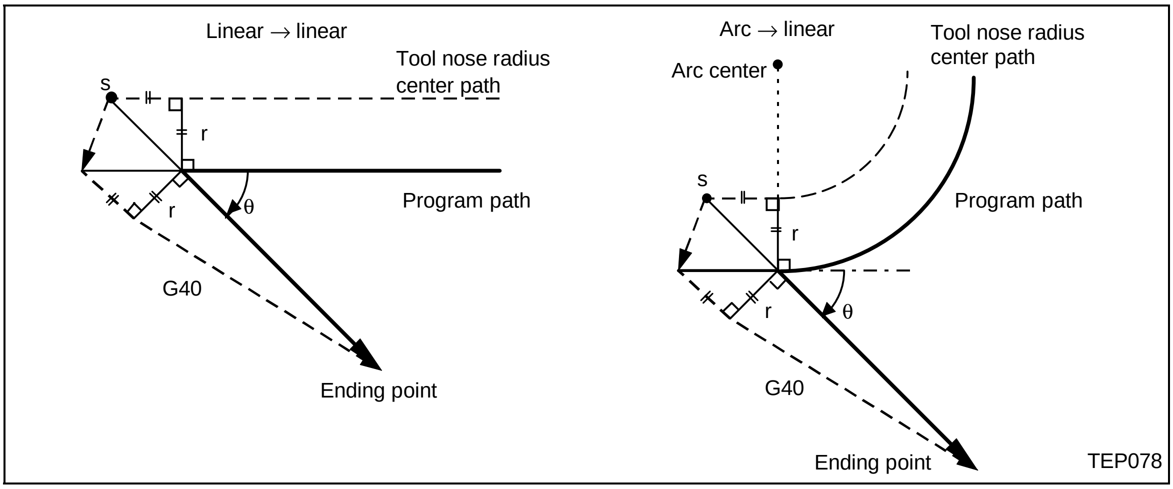
A. For the corner interior



B. For the corner exterior (obtuse angle)



C. For the corner exteiror (acute angle)



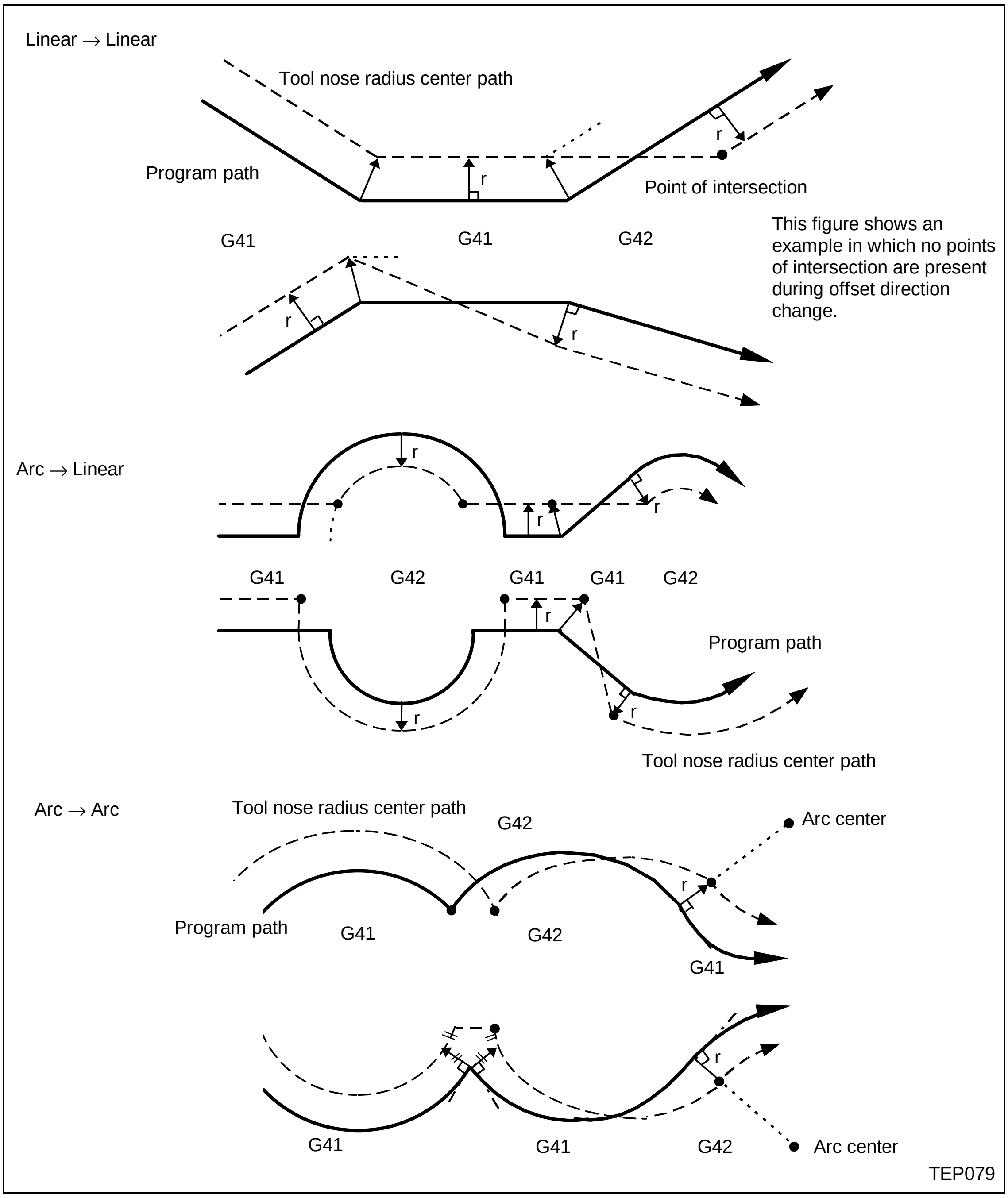
12-3-4 Other operations during tool nose radius compensation

1. Changing the compensation direction during tool nose radius compensation

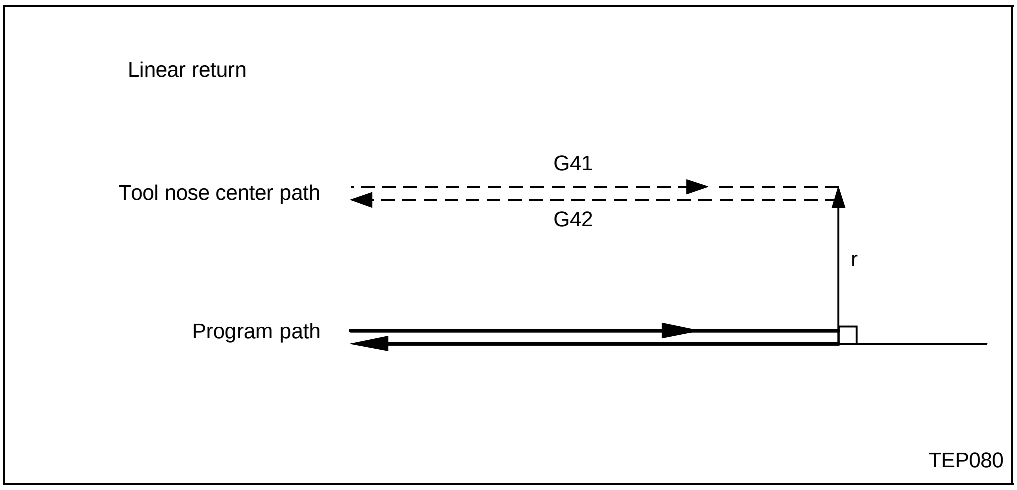
The compensation direction is determined by the tool nose radius compensation commands (G41, G42).

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.



Linear → Linear
Tool nose radius center path
Program path
r
r
Point of intersection
G41
G41
G42
This figure shows an example in which no points of intersection are present during offset direction change.
r
r
Arc → Linear
r
r
r
G41
G42
G41
G41
G42
r
Program path
r
r
Tool nose radius center path
Arc → Arc
Tool nose radius center path
G42
Arc center
r
Program path
G41
G42
G41
r
G41
G41
G42
Arc center
TEP079

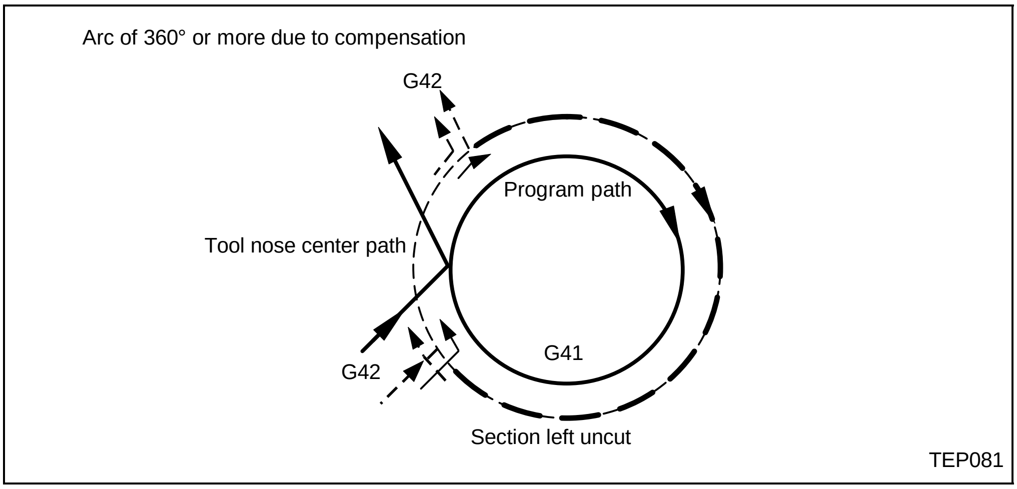


In the following cases, it is possible that the arc may exceed 360°.

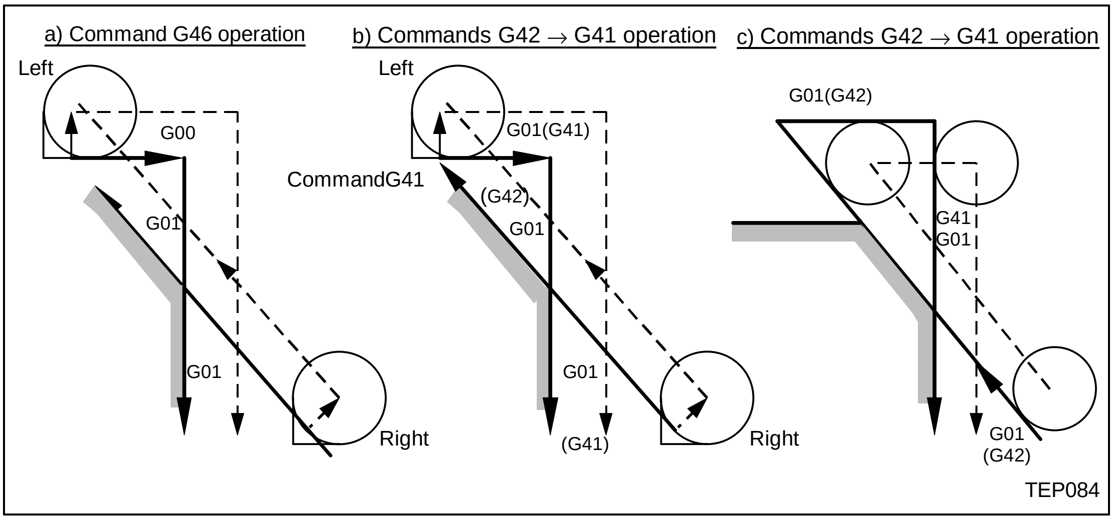
- Compensation direction is changed by the selection of G41 or G42.

- I, J, K are commanded with G40.

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



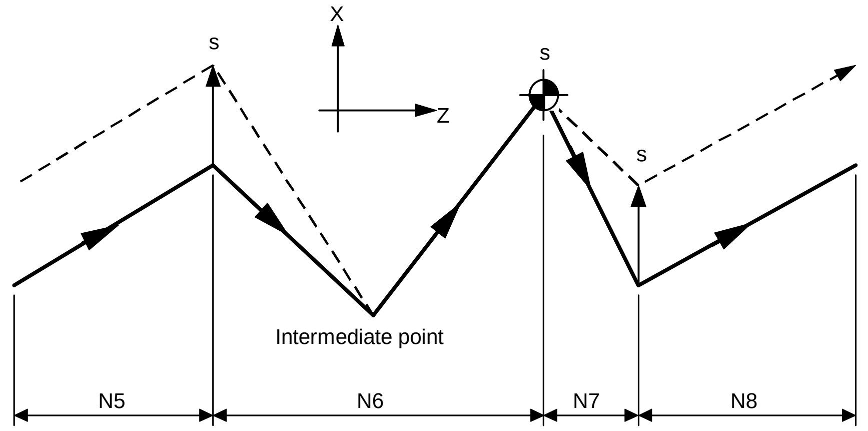
2. Tool nose radius compensation of path close by G46/G41/G42



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.; ←
N7     U-60. W+30.;
N8     U+40. W+70.;
           ⋮
```

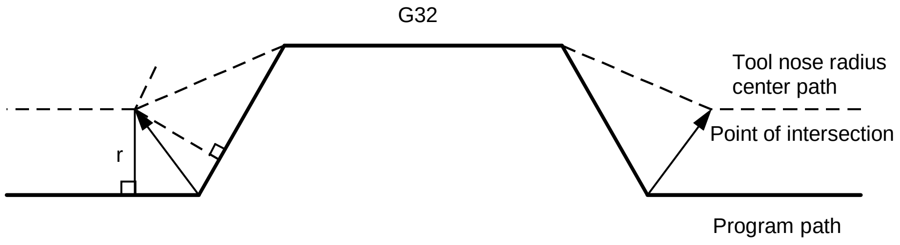
Temporarily vecto 0 as compensation at intermediate point (Reference point when the intermediate point is not available)

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

Return to Main Menu

C. Multiple repetitive fixed cycles

When a multiple repetitive 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

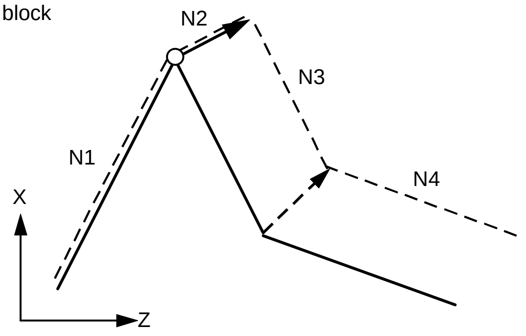
The following blocks are referred to as those which do not include movement.

`M03;`	M command	Move-free
`S12;`	S command	
`T0101;`	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

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.

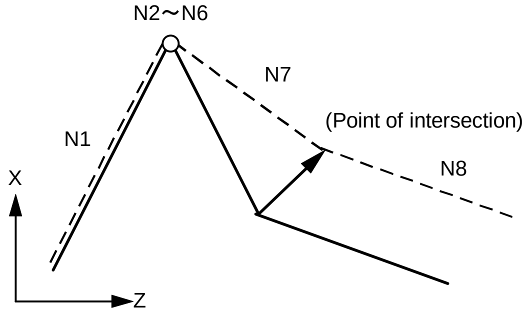
```
N1  U60.W30.T0101;
N2  G41;———————— Move-free block
N3  U-50.W20.;
N4  U-20.W50.;
```



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.T0101;
N2  G41;        ⎫
N3  G4 X1000;   ⎪
N4  F100;       ⎬ Move-free block
N5  S500;       ⎪
N6  M3;         ⎭
N7  U-50.W20.;
N8  U-20.W50.;
```

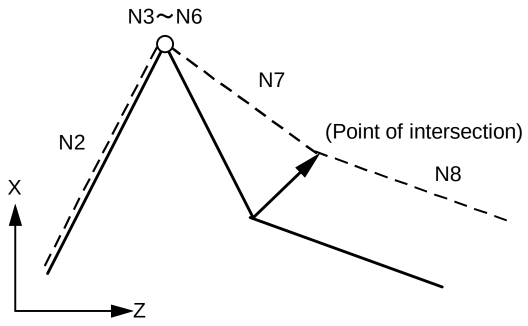


TEP086

Return to Main Menu

```
N1  T0101;
N2  G41 U60.W30.;
N3  G4 X1000;
N4  F100;
N5  S500;
N6  M3;
N7  U-50.W20.;
N8  U-20.W50.;
```

N3 to N6: Move-free block



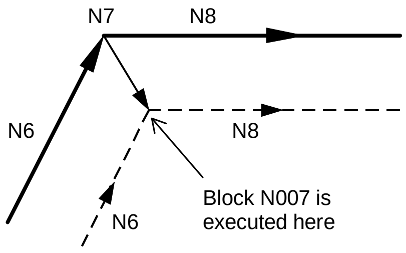
TEP086

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.

```
N6  U200.W100.;
N7  G04 X1000;
N8  W200.;
```

N7: Move-free block



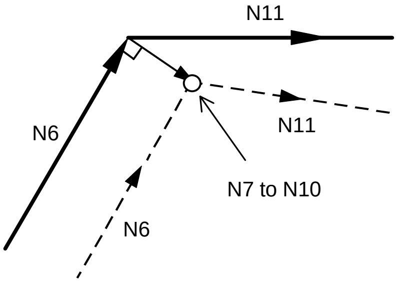
TEP087

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.

```
N6  U200.W100.;
N7  G4 X1000;
N8  F100;
N9  S500;
N10 M4;
N11 W100.;
```

N7 to N10: Move-free block

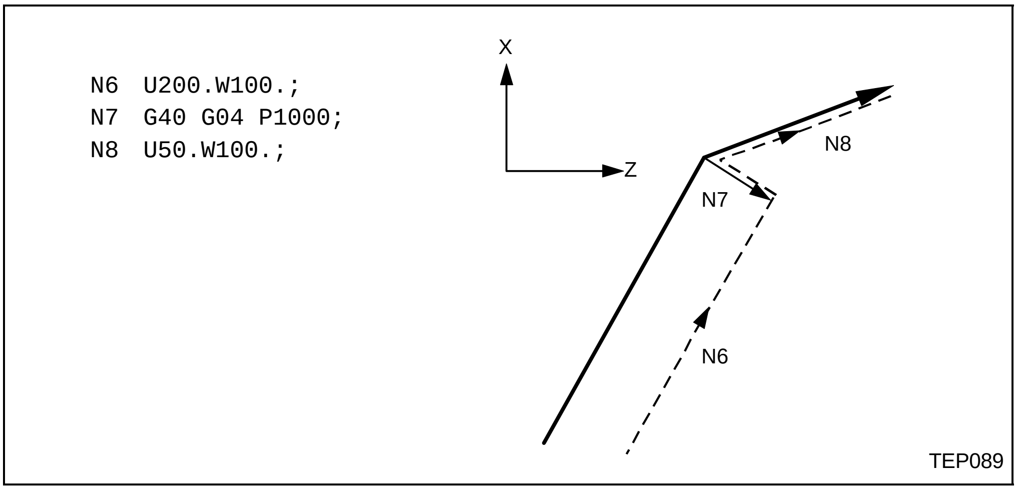
In this case, excessive cutting may occur.



TEP089

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.

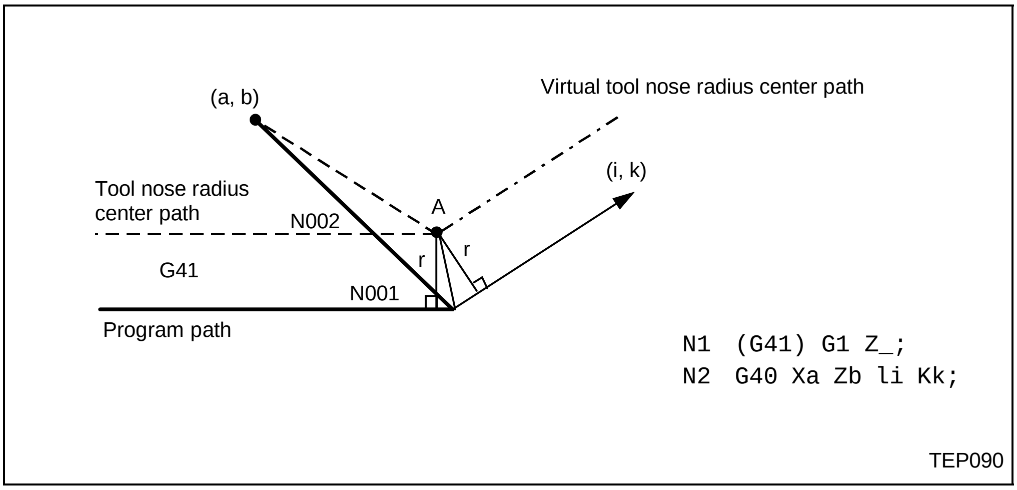


```
N6  U200.W100.;
N7  G40 G04 P1000;
N8  U50.W100.;
```

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.

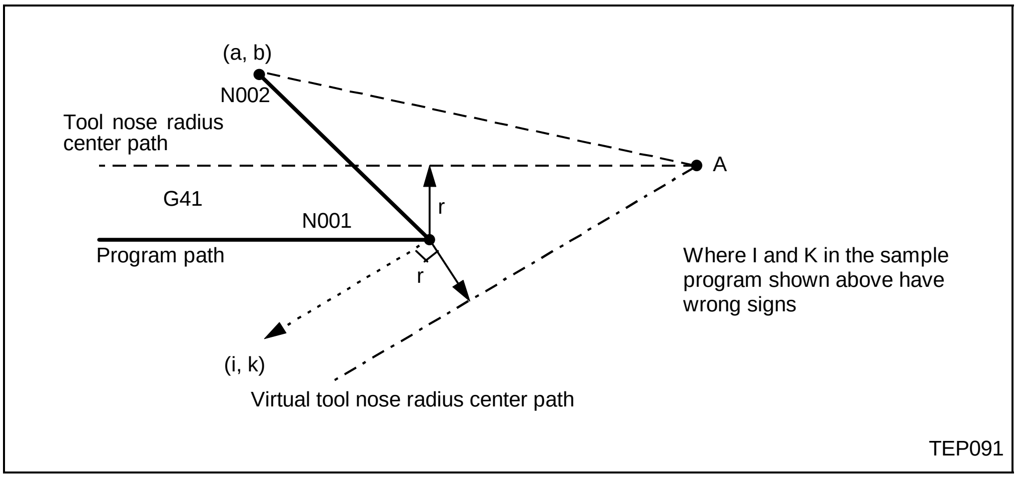


```
N1  (G41) G1 Z_;
N2  G40 Xa Zb li Kk;
```

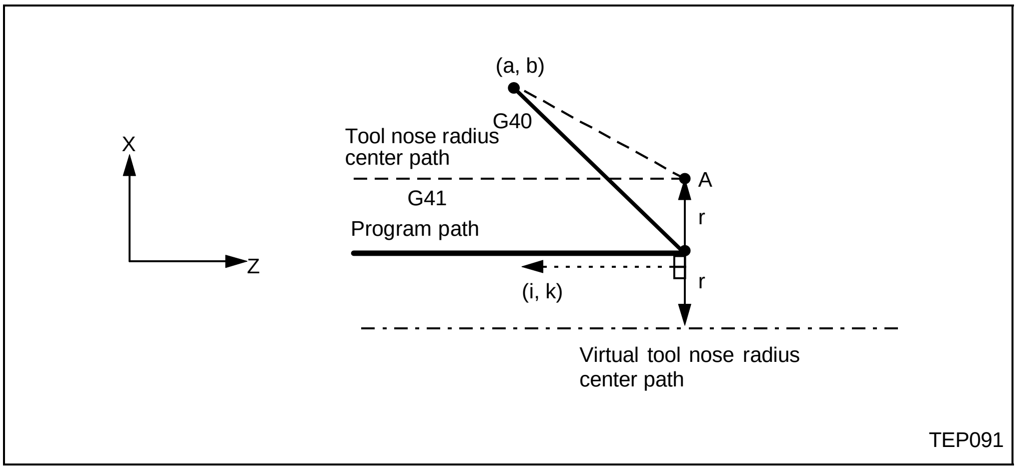
TEP090

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.

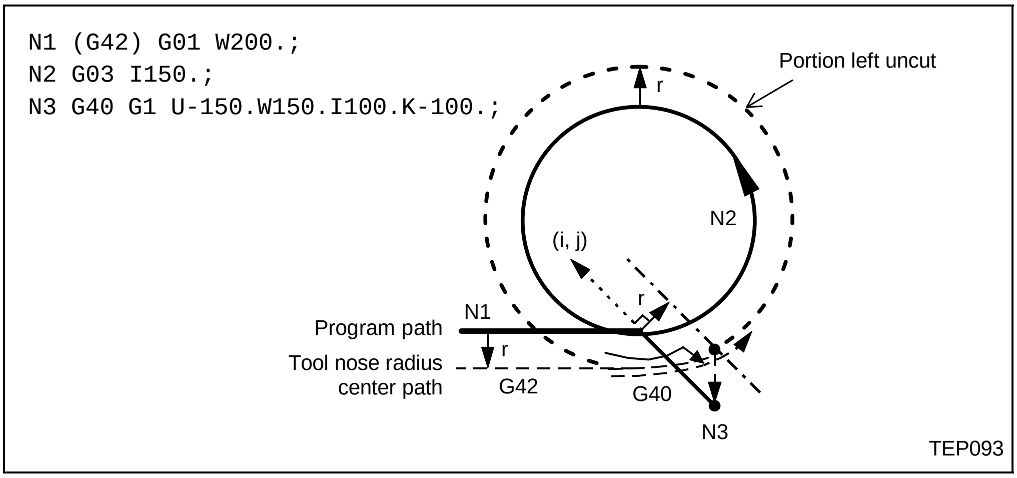
Return to Main Menu


(a, b)
N002
Tool nose radius center path
A
G41
r
N001
Program path
r
Where I and K in the sample program shown above have wrong signs
(i, k)
Virtual tool nose radius center path
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.


(a, b)
G40
Tool nose radius center path
X
A
G41
r
Program path
Z
(i, k)
r
Virtual tool nose radius center path
TEP091

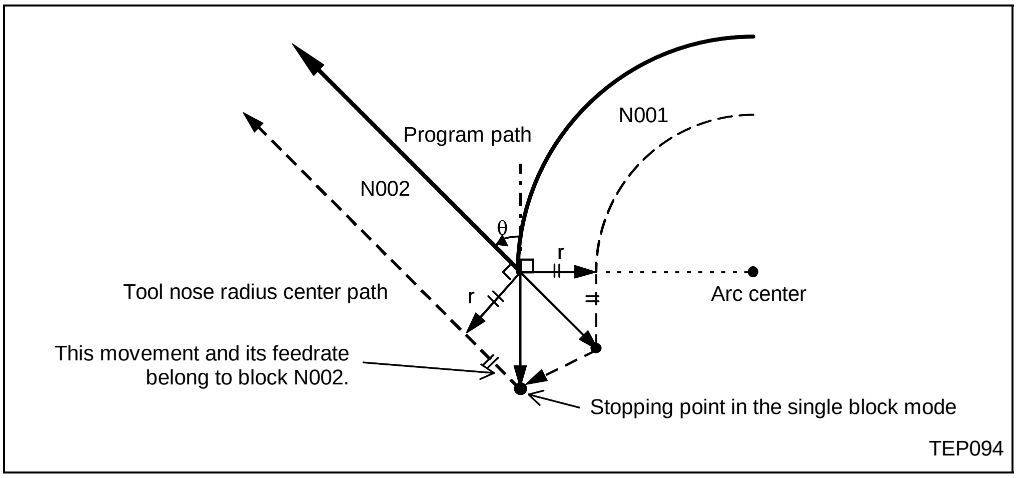
Note: Part of the workpiece will be left uncut if the I/J/K command data in G40 preceded by an arc command generates an arc of more than 360 degrees.



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

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.


Program path
N001
N002
θ
r
r
Tool nose radius center path
Arc center
This movement and its feedrate belong to block N002.
Stopping point in the single block mode
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

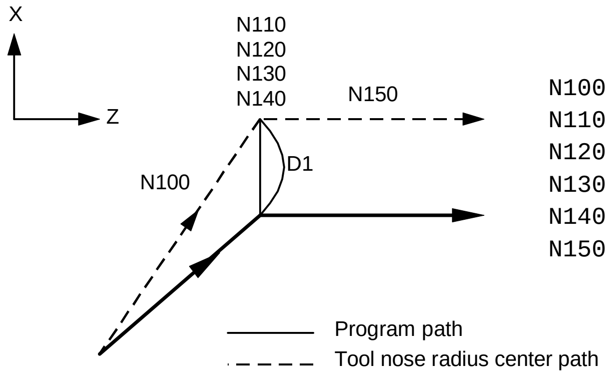
G18 (Z-X plane) G41/G42 X_ Z_ I_ K_ ;

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).

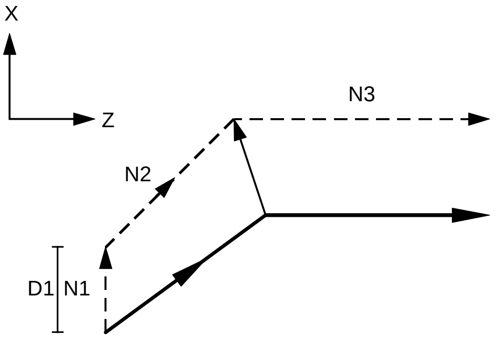
A. When I, K is commanded at compensation start:



```
      (G40)
        ⋮
N100 G41 U100.W100.K150.T0101;
N110 G04 X1000;
N120  G01 F1000;
N130 S500;
N140 M03;
N150 Z150.;
        ⋮
```

TEP095

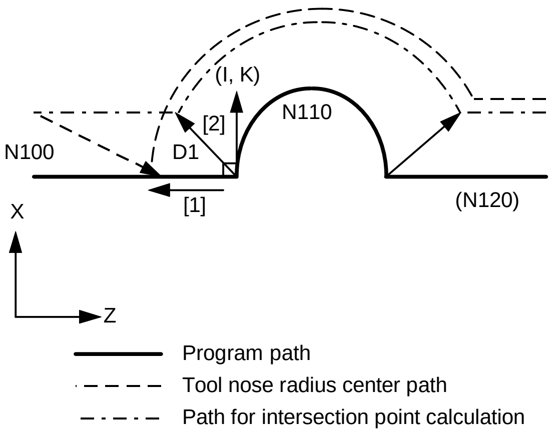
When there are no move commands at the compensation start



```
   (G40)
     ⋮
N1 G41 K150.T0101;
N2 U100.W100.;
N3 W150.;
     ⋮
```

TEP096

B. When I, K has been commanded in the tool nose radius compensation mode (G18 plane)

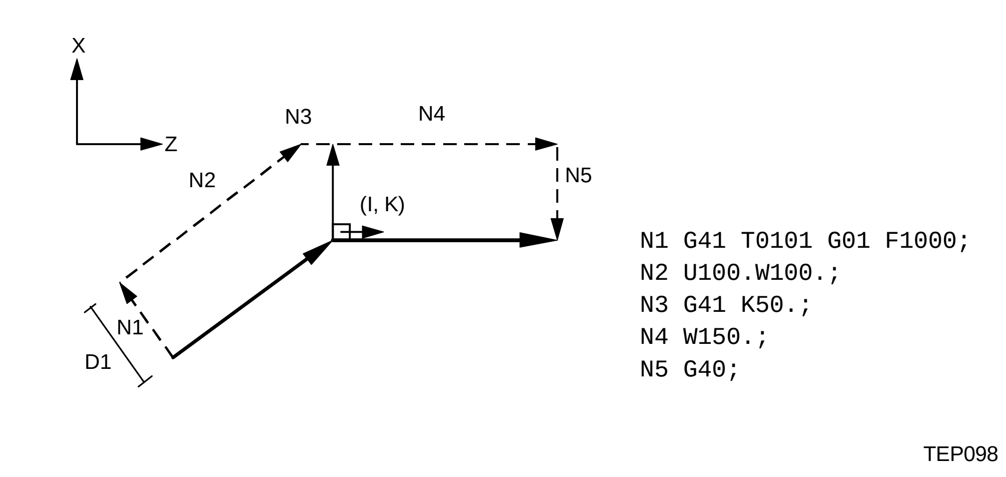

(I, K)
[2]
N110
N100
D1
[1]
(N120)
X
Z
Program path
Tool nose radius center path
Path for intersection point calculation

```
          ⋮
     (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

C. When I, K has been commanded in a block without move command



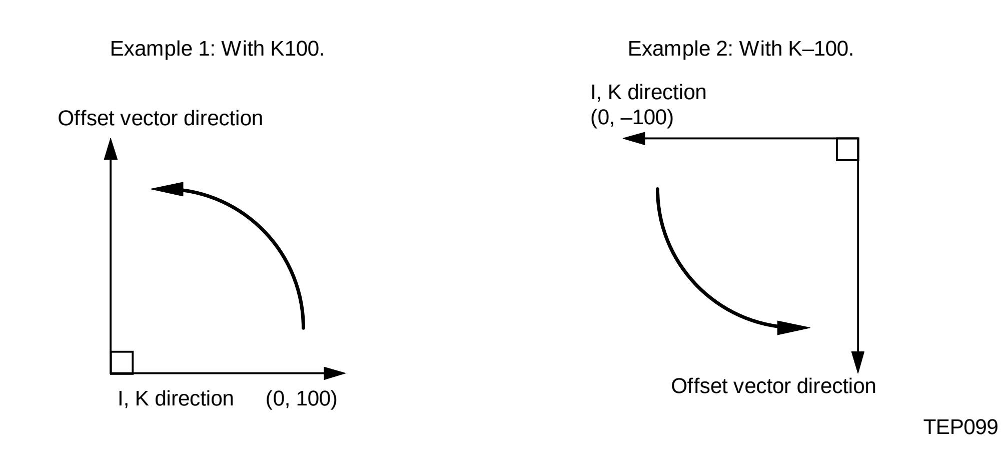
```
N1 G41 T0101 G01 F1000;
N2 U100.W100.;
N3 G41 K50.;
N4 W150.;
N5 G40;
```

TEP098

3. Direction of offset vectors

A. In G41 mode

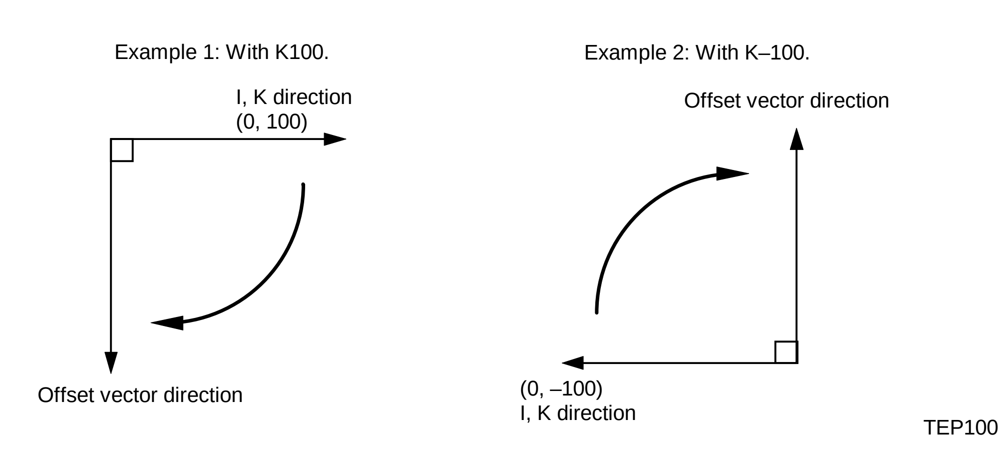
Direction produced by rotating the direction commanded by I, K vector through 90° to the left as seen from the forward direction of the Y-axis (third axis) to the zero point.



TEP099

B. In G42 mode

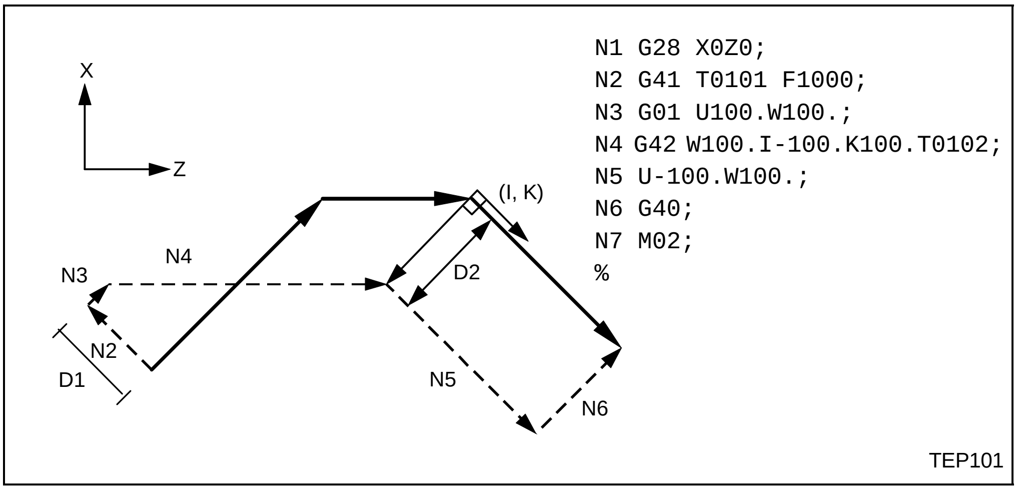
Direction produced by rotating the direction commanded by I, K vector through 90° to the right as seen from the forward direction of the Y-axis (third axis) to the zero point



TEP100

4. Selection of offset modal

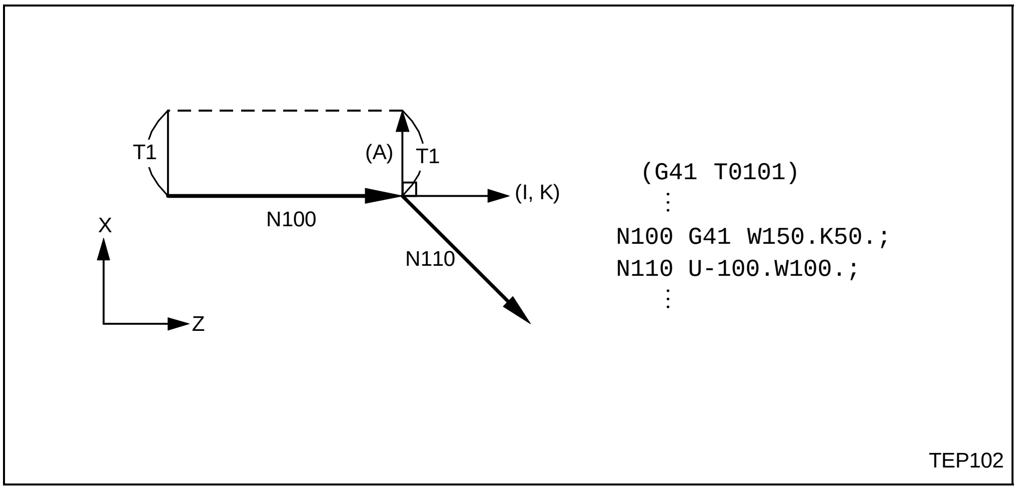
The G41 or G42 modal can be selected at any time.



```
N1 G28 X0Z0;
N2 G41 T0101 F1000;
N3 G01 U100.W100.;
N4 G42 W100.I-100.K100.T0102;
N5 U-100.W100.;
N6 G40;
N7 M02;
%
```

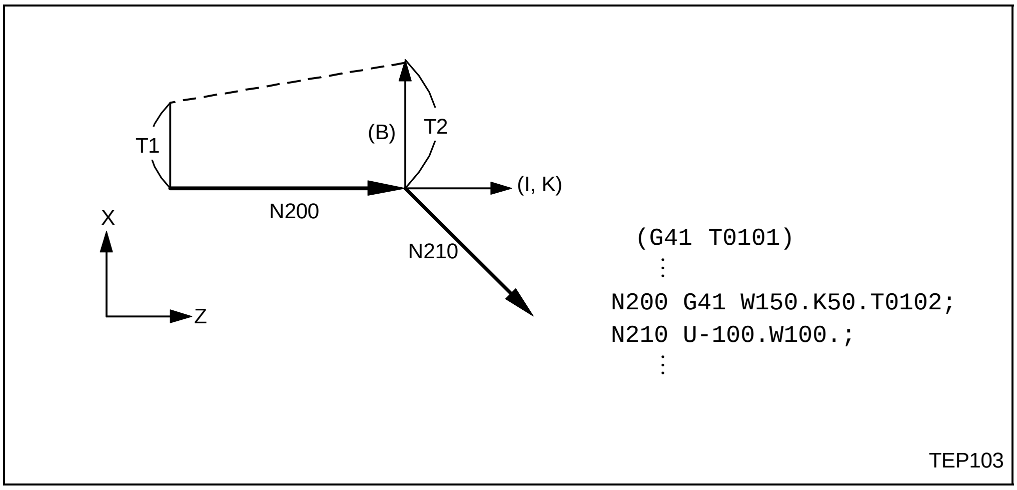
5. Offset stroke of offset vectors

The offset stroke is determined by the offset number (modal) in the block including the I, K designation.



```
 (G41 T0101)
   ⋮
N100 G41 W150.K50.;
N110 U-100.W100.;
   ⋮
```

Vector (A) is the offset stroke entered in tool offset number modal 1 in the N100 block.

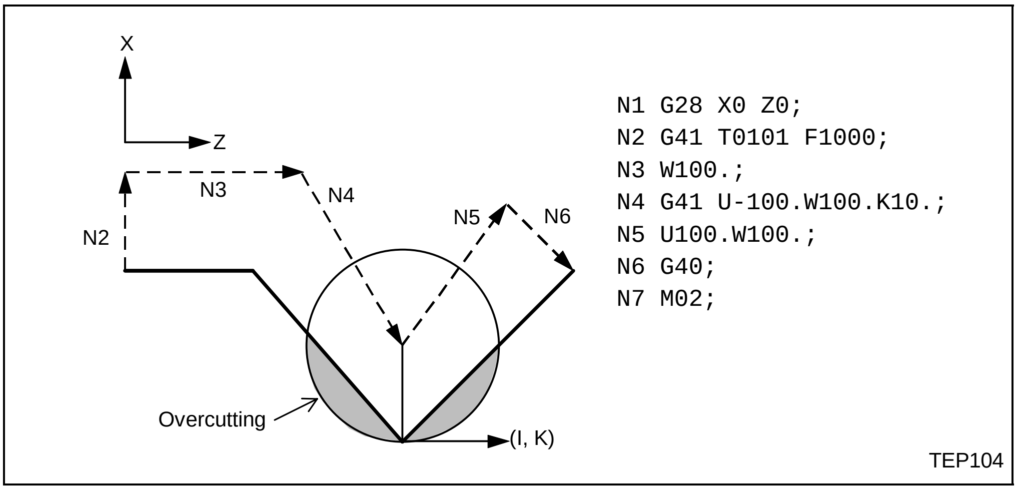


```
 (G41 T0101)
   ⋮
N200 G41 W150.K50.T0102;
N210 U-100.W100.;
   ⋮
```

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 (G01). 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.



```
N1 G28 X0 Z0;
N2 G41 T0101 F1000;
N3 W100.;
N4 G41 U-100.W100.K10.;
N5 U100.W100.;
N6 G40;
N7 M02;
```

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 tool nose radius compensation

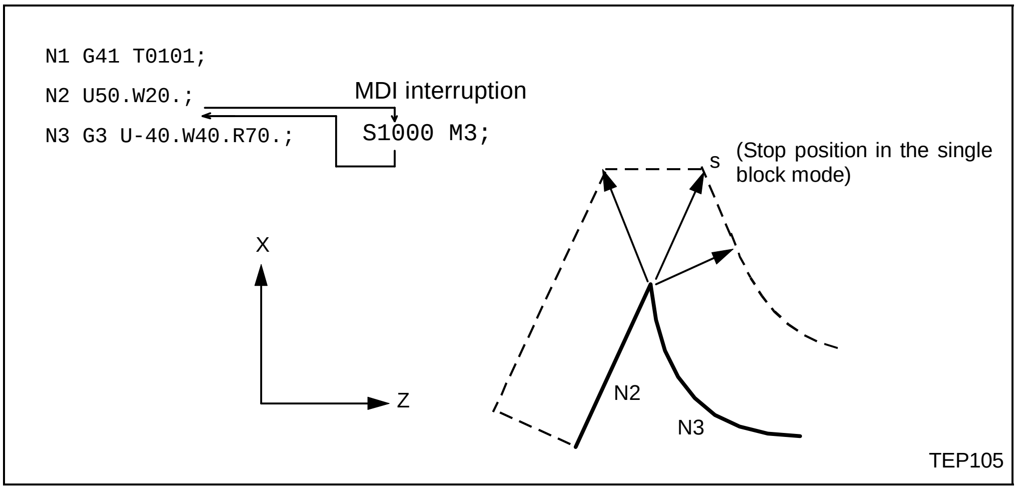
1. Interruption by MDI

Tool nose 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.



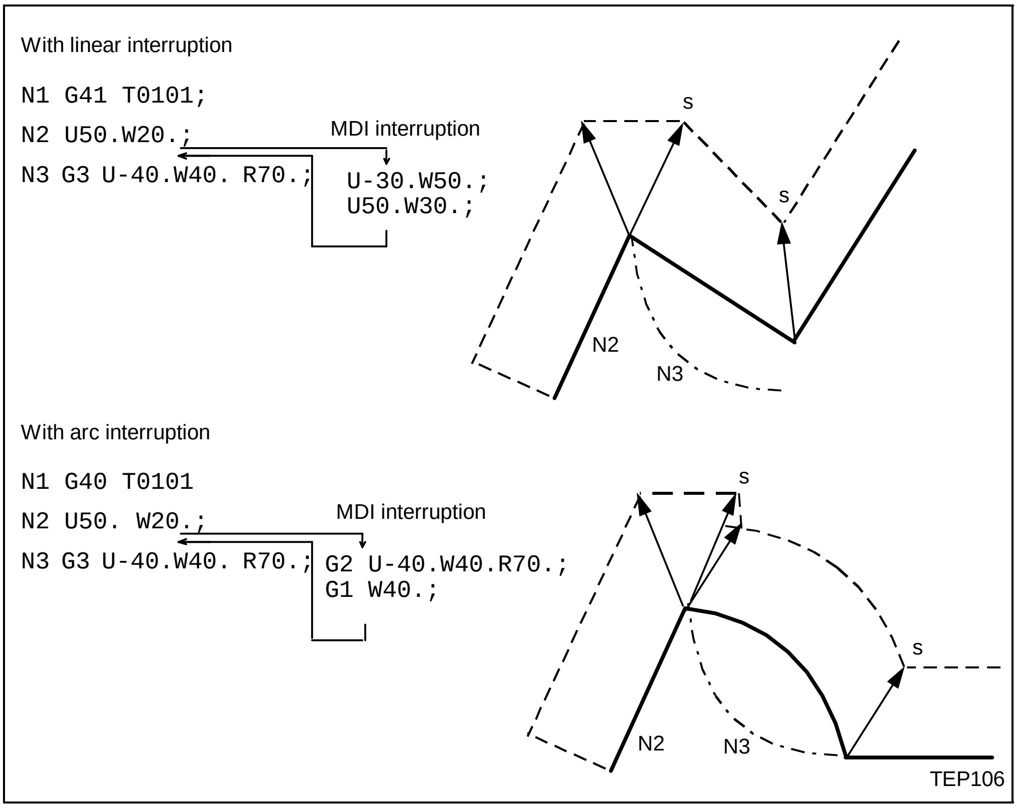
```
N1 G41 T0101;
N2 U50.W20.;
N3 G3 U-40.W40.R70.;
```

MDI interruption

```
S1000 M3;
```

B. Interruption with movement

The offset vectors are recalculated automatically at the move command block after interruption.



With linear interruption

```
N1 G41 T0101;
N2 U50.W20.;
N3 G3 U-40.W40. R70.;
```

MDI interruption

```
U-30.W50.;
U50.W30.;
```

With arc interruption

```
N1 G40 T0101
N2 U50. W20.;
N3 G3 U-40.W40. R70.;
```

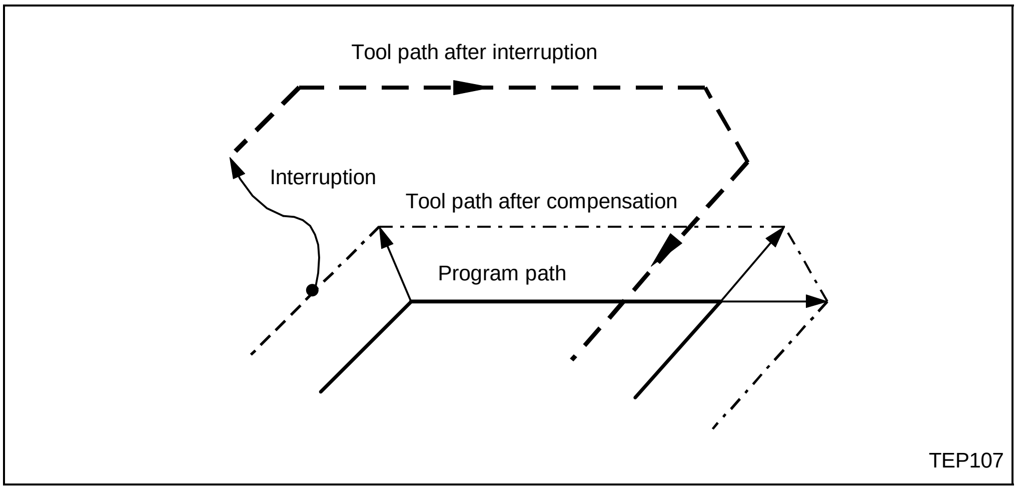
MDI interruption

```
G2 U-40.W40.R70.;
G1 W40.;
```

2. Manual interruption

A. Interruption with manual absolute OFF

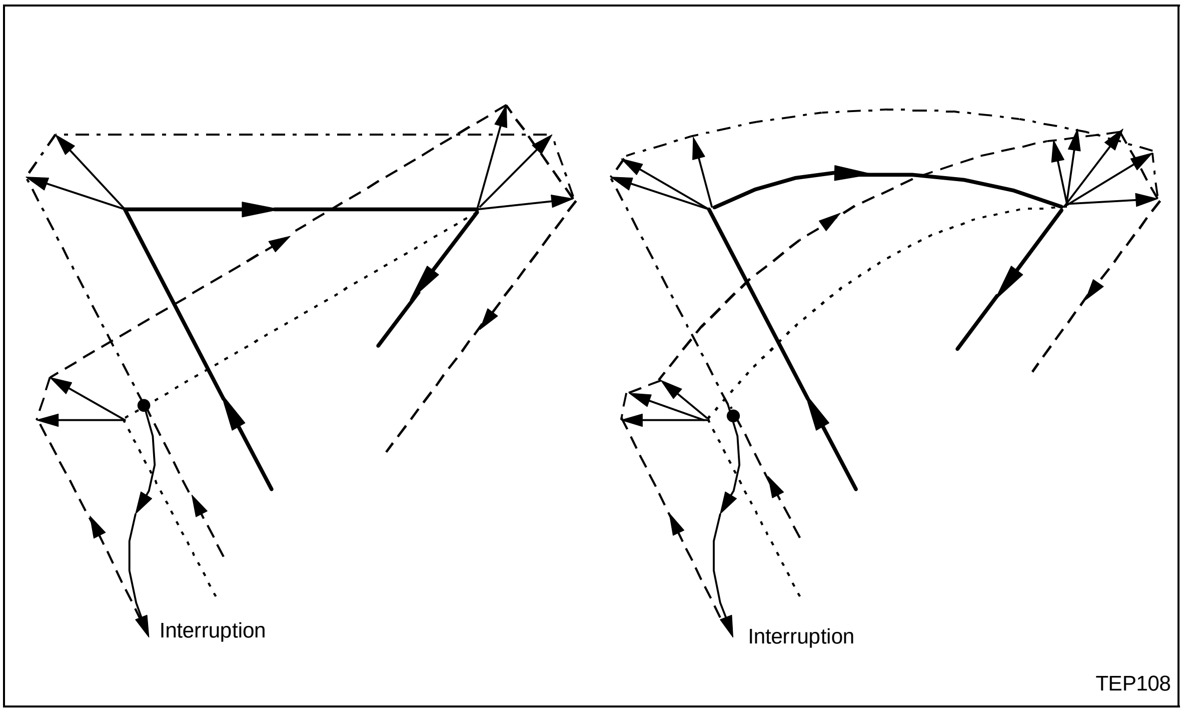
The tool path is shifted by an interruption amount.



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.



Return to Main Menu

12-3-7 General precautions on tool nose 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 a tool nose point other than 1 through 8 has been designated in the G46 mode.

3. An error results when the compensation direction is not determined by the movement vector of the initial cutting command even when the tool nose radius compensation operation has started in the G46 mode and 5 blocks have been preread.

4. An error results when an arc command is set in the first or last block of the tool nose radius compensation.

5. A programming error occurs when the compensation direction is reversed in the G46 mode. A parameter can be set to move the tool in the same compensation direction. (Parameter **P13** bit 7 “reversal error prevention in G46 mode”)

6. 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.

7. A programming error results when an error occurs in one of the preread blocks during tool nose radius compensation.

8. A programming error results when interference can arise without no interference avoidance function during tool nose radius compensation.

Return to Main Menu

12-3-8 Interference check

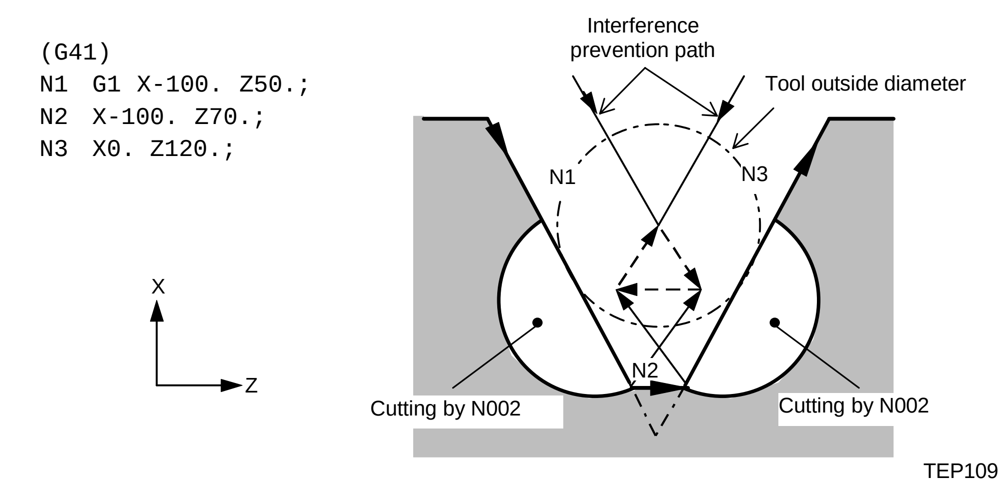
1. Overview

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 (**P10** bit 5)	Operation
Interference check/alarm	Interference check/prevention off (**P10** bit 5 = 0)	The system will stop, with a program error resulting before executing the cutting block.
Interference check/prevention	Interference check/prevention on (**P10** bit 5 = 1)	The path is changed to prevent cutting from taking place.

Example:



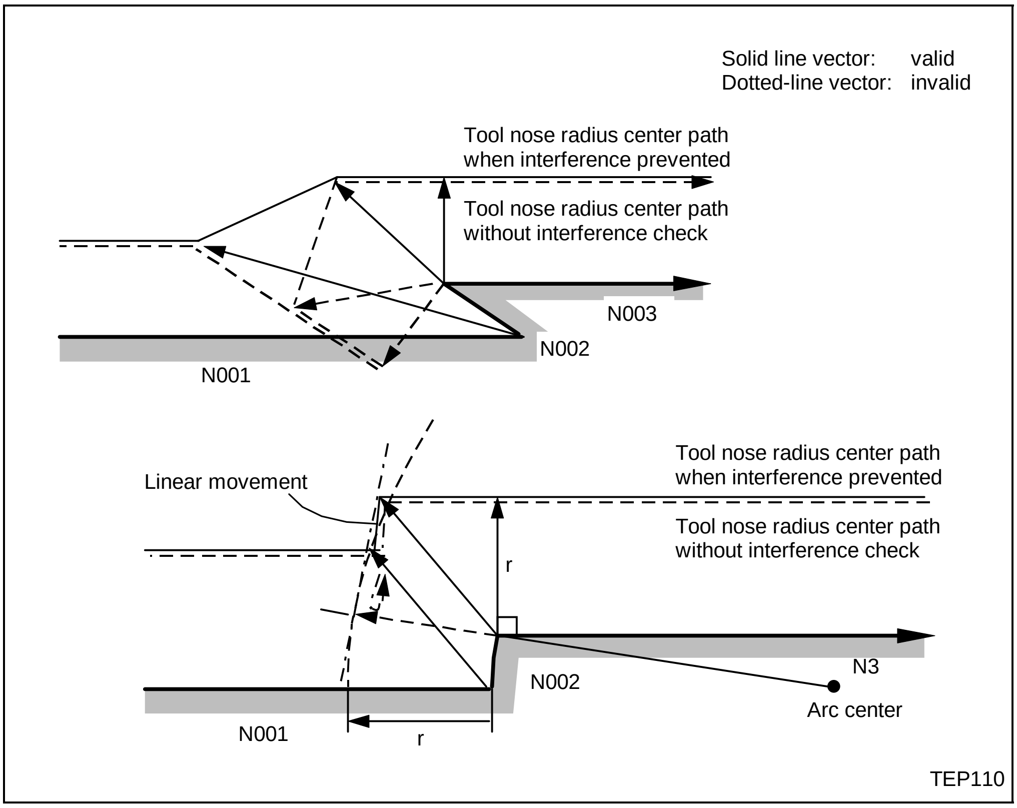
```
(G41)
N1  G1 X-100. Z50.;
N2  X-100. Z70.;
N3  X0. Z120.;
```

TEP109

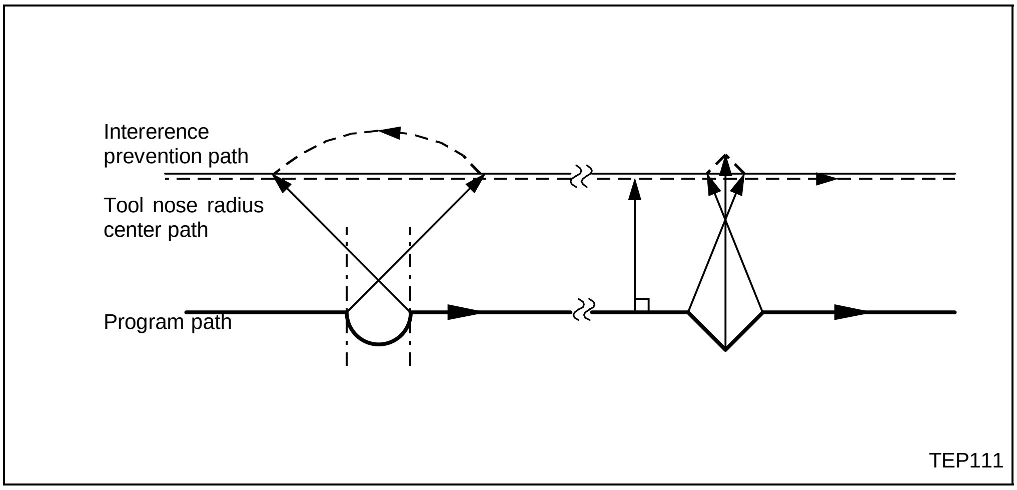
- For the alarm function
  An alarm occurs before N001 is executed. Machining can be continued by updating the program into, for example,
  N001 G1 X–100. Z–20.;

- Using the buffer correction function.
  For the interference check/prevention function
  Interference prevention vectors are generated by N001 and N003 intersection point calculation.

2. Operation during interference prevention


Solid line vector: valid
Dotted-line vector: invalid
Tool nose radius center path when interference prevented
Tool nose radius center path without interference check
N003
N002
N001
Linear movement
Tool nose radius center path when interference prevented
Tool nose radius center path without interference check
r
N3
N002
Arc center
N001
r
TEP110

In the case of the figure below, the groove will be left uncut.

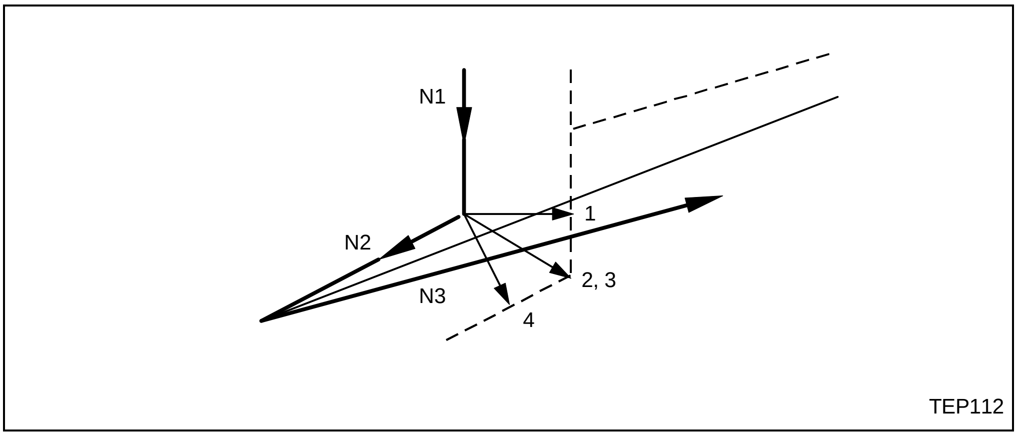


3. Interference check/alarm

Cases that an interference check/alarm occurs are listed below.

A. When interference check/alarm is selected

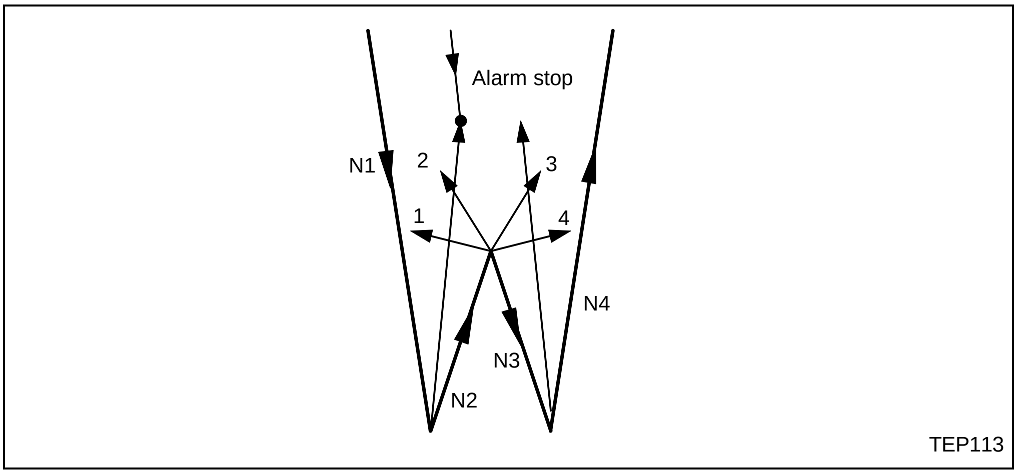
1. If all vectors at the ending point of the current block are erased:
   Prior to execution of N001, a program error will result if vectors 1 through 4 at the ending point of the N001 block are all erased as shown in the diagram below.



B. When interference check/prevention is selected

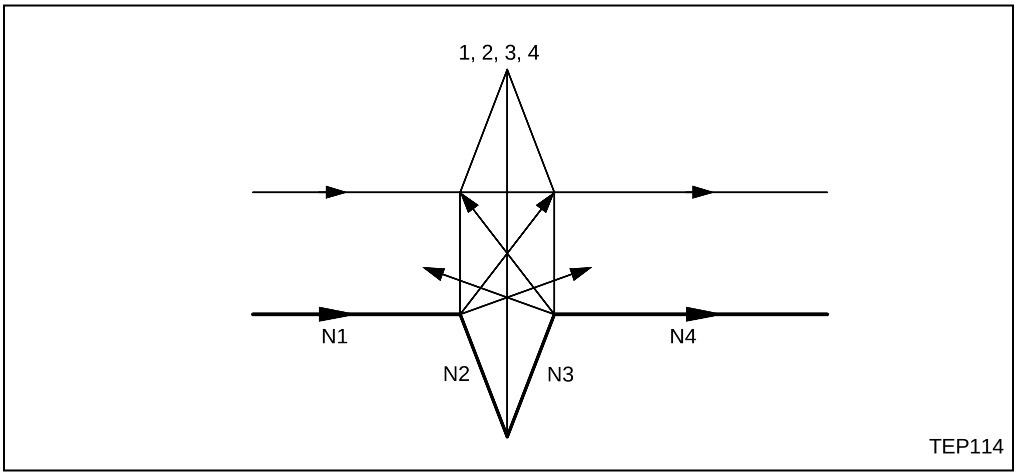
1. If all vectors at the ending point of the current block are erased but an effective vector(s) remains at the ending point of the next block:

   - 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.



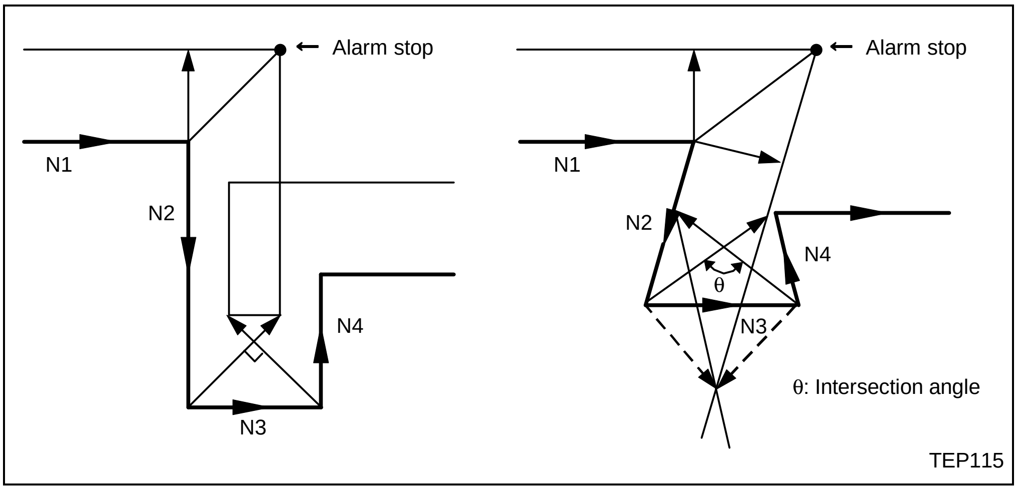
At this time, a program error will occur at the ending point of N001.

- 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.

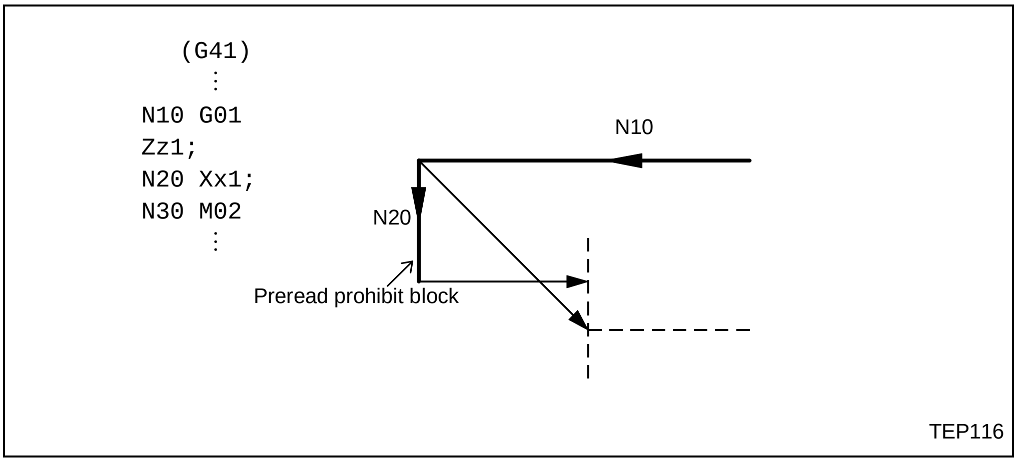


2. When prevention vectors cannot be generated:

- 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.



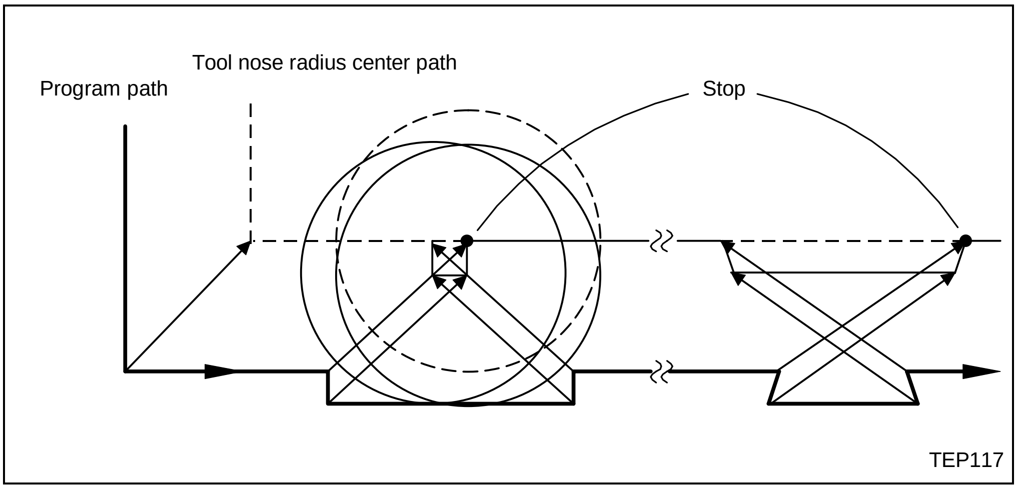
- Prevention vectors may not be generated when preread prohibit blocks are interfered with and so program error occurs.



```
  (G41)
    ⋮
N10 G01
Zz1;
N20 Xx1;
N30 M02
    ⋮
```

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.



Return to Main Menu

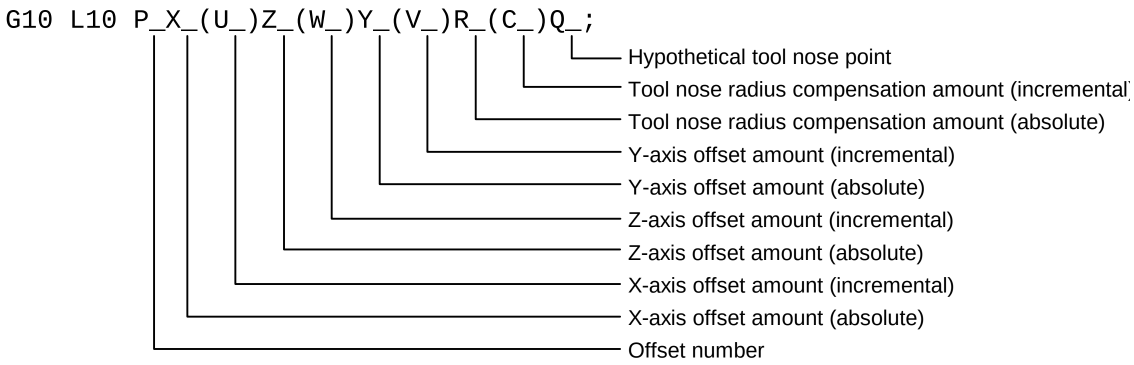
12-4 Programmed Tool Offset Input: G10 L10

1. Outline

The amount of tool offset can be set or changed by the G10 command. When commanded with absolute values (X, Z, Y, R), the commanded offset amounts serve as the new amounts; when commanded with incremental values (U, W, V, C), the new offset amounts are equivalent to the commanded amounts plus the current offset amount settings.

2. Programming format

Tool offset input (L10)



```
G10 L10 P_X_(U_)Z_(W_)Y_(V_)R_(C_)Q_;
```

Hypothetical tool nose point

Tool nose radius compensation amount (incremental

Tool nose radius compensation amount (absolute)

Y-axis offset amount (incremental)

Y-axis offset amount (absolute)

Z-axis offset amount (incremental)

Z-axis offset amount (absolute)

X-axis offset amount (incremental)

X-axis offset amount (absolute)

Offset number

The address L can be omitted.

3. Detailed description

1. The following table shows the setting ranges of the offset numbers and the hypothetical tool nose points.

Address	Signification	Setting range
		L10
P	Offset number	When L command is present : 1 to max. number of offset sets
Q	Hypothetical tool nose point	0 to 9

For type B, distinguish between the two types of data as follows:
- Set the offset data No. as it is at address P for GEOMETRIC OFFSET data, or
- Set the offset data No. plus 10000 at address P for WEAR COMP. data.

2. The setting range for the offset amount is given below.
Program error occurs for any value not listed in the table after command unit conversion.
With an incremental value command, the offset amount is the sum of the present setting and command value.

	Tool offset amount	
	Metric system	Inch system
Setting range	±99999.999 (mm)	±9999.9999 (inch)

3. G10 is an unmodal command and is valid only in the commanded block.

4. Offset input can be performed similarly for the third axis but even when the C-axis has been designated as the third axis, the value for address C is treated as an incremental command value for tool nose radius in the L10 command.

5. If an illegal number for addresses L or P is commanded, program error will result, respectively.
6. Program error occurs when the P command is omitted.
7. Program error occurs when the offset amount exceeds the setting range.
8. X, Z and U, W can be inputted together in a same block but when the addresses which specify the same offset input (X/U or Z/W) are commanded, the address which is inputted last is valid.
9. Offset will be input if even one address following G10L10P_ is commanded. Program error results when not even a single command has been set.

   Example: G10 L10 P3 Z50.;

   ⇩ Input

   ```
   [Tool position data]
   #   Z
   3   50.000
   ```

10. Decimal points are valid for offset amounts.
11. The commands G40 to G42 are ignored when they have been commanded in the same block as G10.
12. Do not command G10 in the same block as fixed cycles and subprogram call commands. This will cause malfunctioning and program errors.

12-5 Programmed Parameter Input: G10, G11

1. Function and purpose

Parameters can be input from a tape or others with this command.
Executing a program which contains parameter numbers and data in blocks subsequent to G10 L50 allows parameter setting automatically.

Parameters including maximum cutting feed can be changed in the flow of program execution according to the respective machining conditions.

2. Programming format

```
G10   L50 ; ..........................Parameter input mode
N_ P_ R_ ;
N_ R_ ;
G11 ;....................................Parameter input mode cancel
```

N _: Parameter number (See next page.)
R _: Parameter setting
P _: Axis number (for axis type parameter)
Device number (for parameter by device)

3. Detailed description

1. Other NC sentences cannot be commanded in parameter input mode.
2. As a general rule, avoid using a decimal point for address R. Values actually set varies with input unit. It should be noted that the use of a user macro variable is as with that of a decimal point.
3. Pocket calculator type decimal point programming is invalid for address R in parameter input mode.
4. Always cancel fixed cycle mode beforehand. Failure to do it performs drilling operation.
5. Commading an improper parameter number causes an alarm.
6. Describe all the data with decimal numbers. (Change HEX type and bit type data to decimal numbers.)
7. Specify data at addresses N and P as follows:

Parameter	Parameter number (N)	Parameter data (R)	Axis number (P)
P1 to **P32**	1001 to 1032		—
P33 to **P40**	1033 to 1040		—
P41 to **P64**	1041 to 1064		—
P65 to **P96**	1065 to 1096		—
P97 to **P104**	1097 to 1104		—
P105 to **P112**	1105 to 1112		—
U1 to **U96**	2001 to 2096		—
K1 to **K48**	3001 to 3048		—
K81 to **K96**	4001 to 4016		1, 2
A1 to **A16**	7001 to 7016		1 to 10
B1 to **B32**	8001 to 8032		—
B33 to **B80**	8033 to 8080		—
C1 to **C32**	9001 to 9032		—
C33 to **C80**	9033 to 9080		—
External workpiece origin	20000		1 to 10
G54 to G59 workpiece origin	20001 to 20006		1 to 10
K49 to **K80**	30001 to 30032		—
A17 to **A24**	35001 to 35008		1 to 10
A25 to **A32**	35009 to 35016		1 to 10
B81 to **B144**	38001 to 38064		—
B145 to **B208**	39001 to 39064		—
B209 to **B272**	40001 to 40064		—
C81 to **C144**	41001 to 41064		—
C145 to **C208**	42001 to 42064		—
C209 to **C272**	43001 to 43064		—

Return to Main Menu

4. Sample programs

```
G10  L50 ; ............. Parameter input mode
N1001  R6 ; ............ Set 00000110 to P1. ←   “6” in decimal notation corresponds
N2001  R10 ; ........... Set 10 to U1.            to “00000110” in binary notation.
N3003  R20 ; ........... Set 20 to K3.
N4001  P1  R100 ; ...... Set 100 as calling code to K81.
N4001  P2  R9000 ; ..... Set 9000 as the number of calling program to K81.
N7001  P2  R12000 ; .... Set 12000 as the data of the 2nd axis (Z-axis) for A1.
N8001  R10 ; ........... Set 10 to B1.
N9002  R20 ; ........... Set 20 to C2.
G11 ; .................. Parameter input mode cancel.
```

Return to Main Menu

13 PROGRAM SUPPORT FUNCTIONS

13-1 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	Edge turning cycle

1. The program format is as follows:

   G90 X/U_ Z/W_ I _ F_ ; (T32 compatible mode)
   G90 X/U_ Z/W_ R_ F_ ; (Standard mode)
   (Same for G92, G94)

   The taper values of fixed cycles G90, G92 and G94 are specified by argument I in T32 compatible mode, while specified by argument R in standard mode.

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.

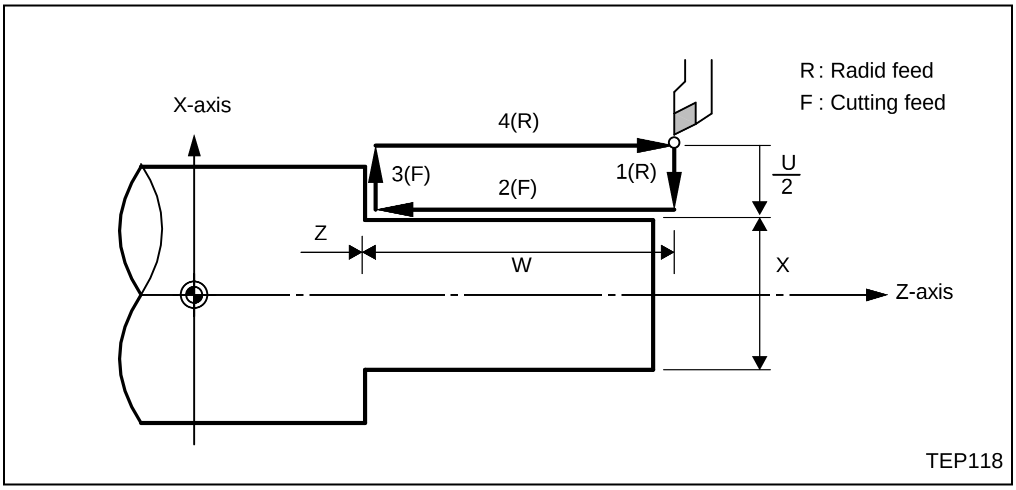
4. A manual interruption can be applied while a fixed cycle for turning (G90, G92 and G94) is being executed. Upon completion of the interruption, however, the tool must be returned to the position where the manual interruption was applied and then the fixed cycle for turning should be restarted. If it is restarted without the tool having been returned, all subsequent operation movements will deviate by an amount equivalent to the manual interruption value.

13-1-1 Longitudinal turning cycle: G90

1. Straight turning

This function enables continuous straight turning in the longitudinal direction using the following command.

G90 X/U_Z/W_F_;

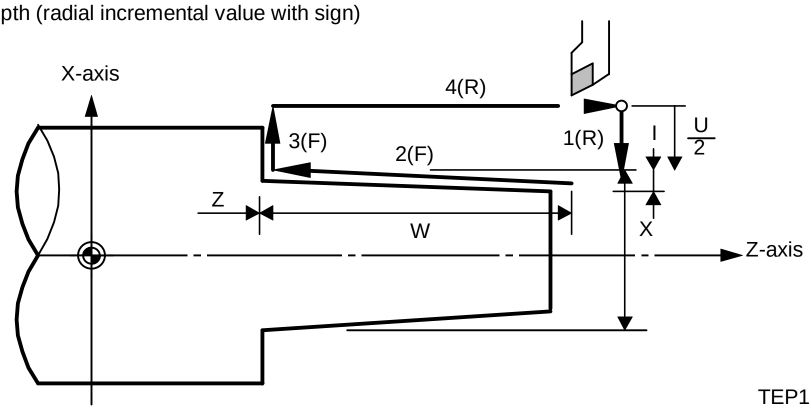


2. Taper turning

This function enables continuous taper turning in the longitudinal direction using the following command.

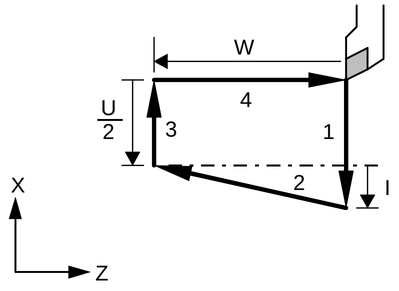
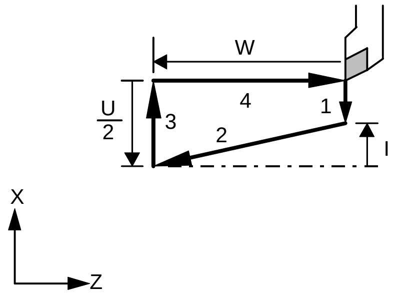
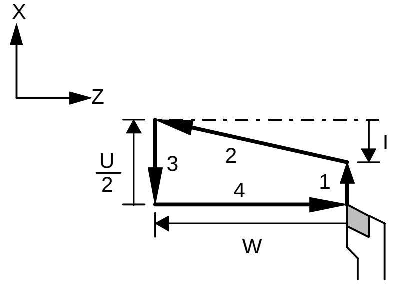
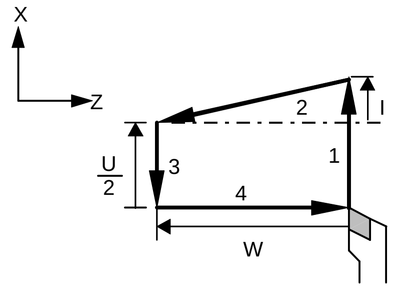
G90 X/U_Z/W_I_F_;

R : Radid feed
F : Cutting feed
I : Taper depth (radial incremental value with sign)



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 2 of parameter **P108**)
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 736 "ILLEGAL TAPER LENGTH" occurs in shapes [2] and [3] unless the following condition is satisfied.

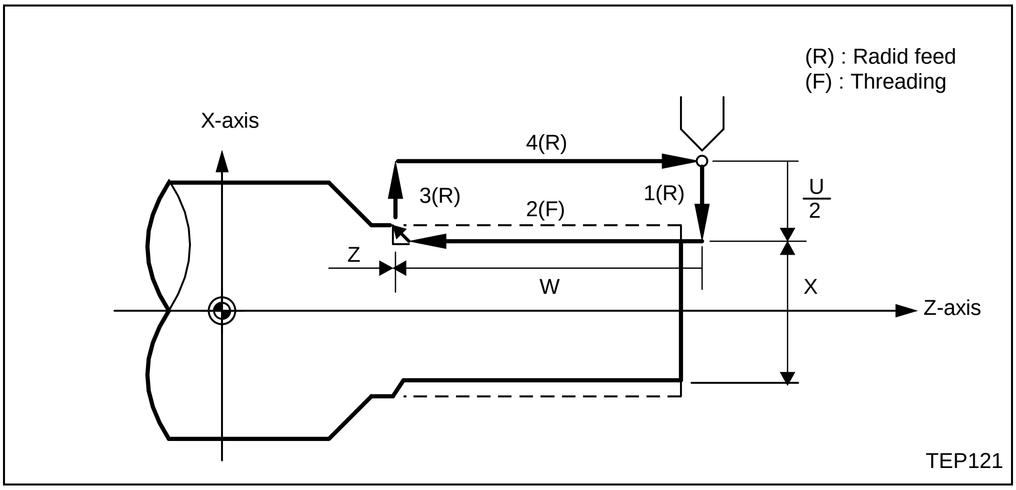
$$\left| \frac{U}{2} \right| \geq \left| \quad I \quad \right|$$

13-1-2 Threading cycle: G92

1. Straight threading

This function enables straight threading using the following command.

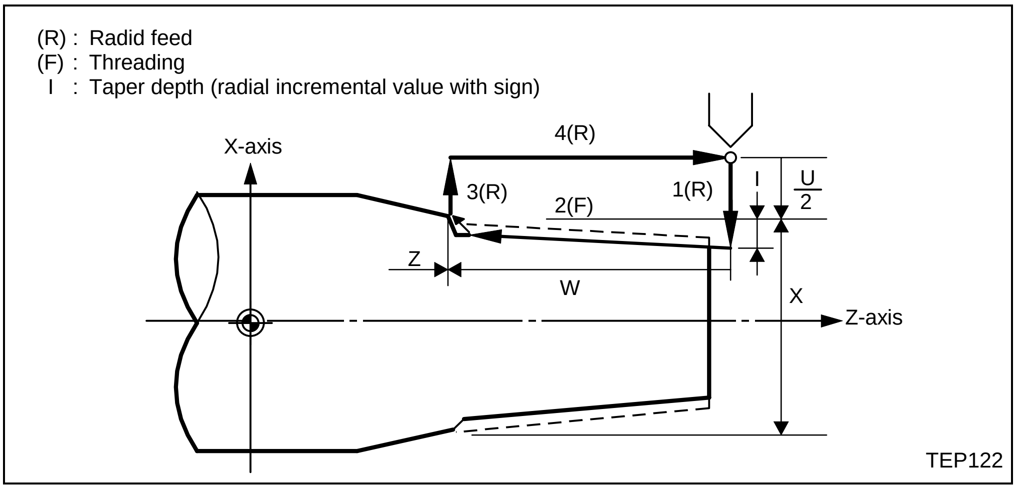
G92 X/U_ Z/W_ F/E_ ;



2. Taper threading

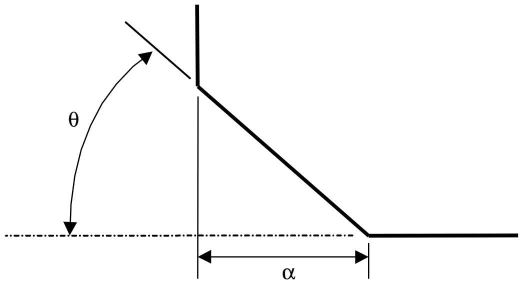
This function enables taper threading using the following command.

G92 X/U_ Z/W_ I_ F/E_ ;



3. Remarks

- Details for chamfering

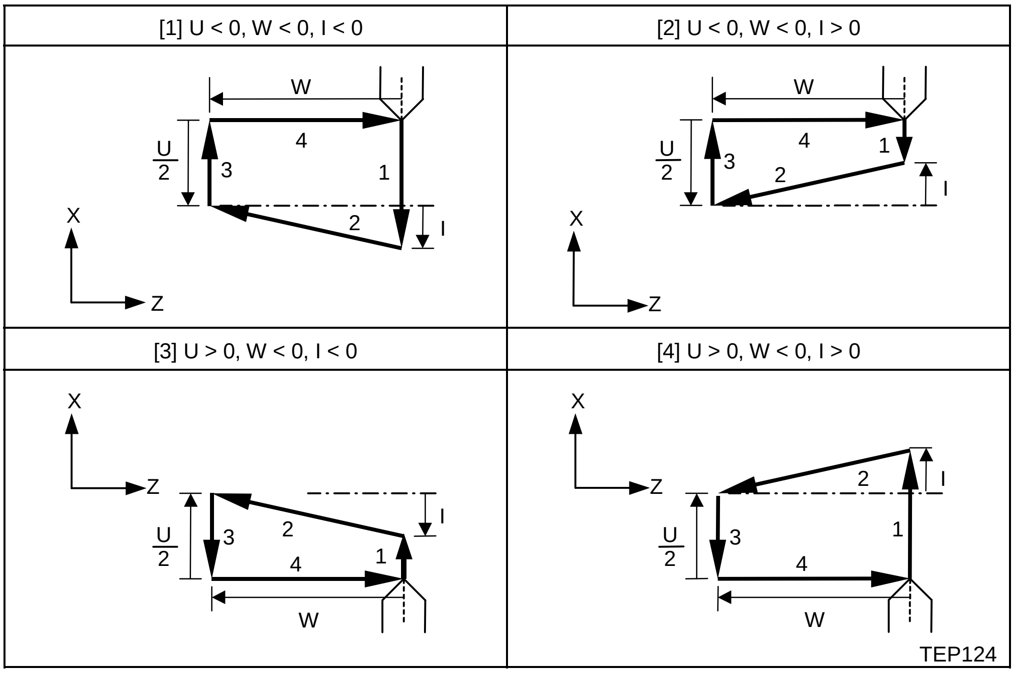


α : Chamfering amount of threading
If the thread lead is assumed to be L, then the parameter can be set by 0.1 L in the range 0 to 4.0.

θ : Chamfering angle of threding
The parameter (**P30**) 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 2 of parameter **P108**).
- 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 **P13** bit 5, 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.



TEP124

Program error 736 "ILLEGAL TAPER LENGTH" occurs in shapes [2] and [3] unless the following condition is satisfied.

$$\left|\frac{U}{2}\right| \geq \left|\ I\ \right|$$

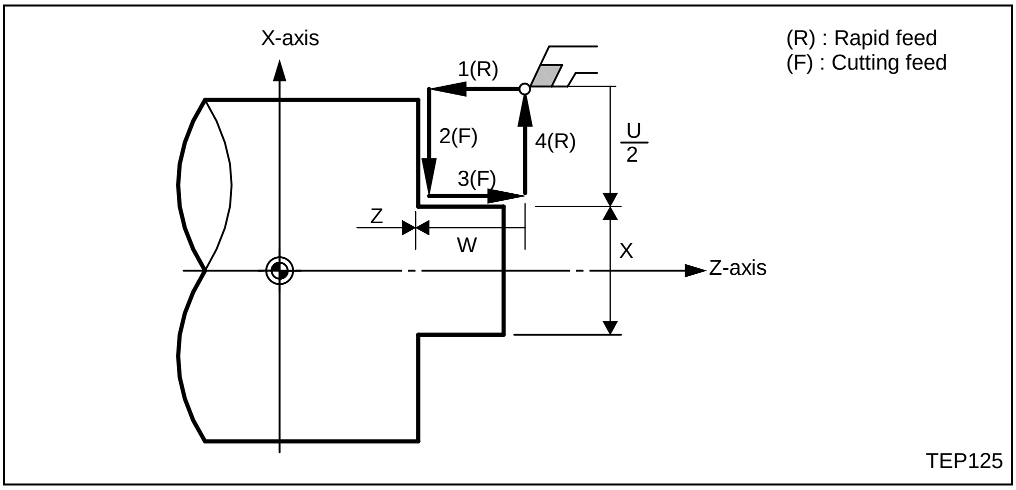
- For machines with the optional function for automatic correction of threading start position, the thread cutting conditions can be changed by "overriding" the spindle speed.
See Subsection 6-8-6 for more information.

13-1-3 Edge turning cycle: G94

1. Straight turning

This function enables continuous straight turning in the face direction using the following command.

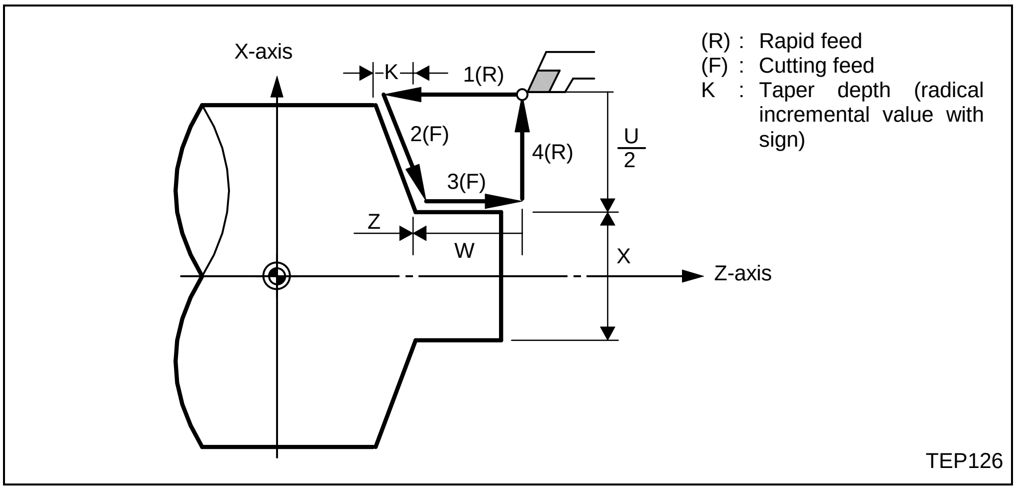
G94 X/U_Z/W_F_;



2. Taper turning

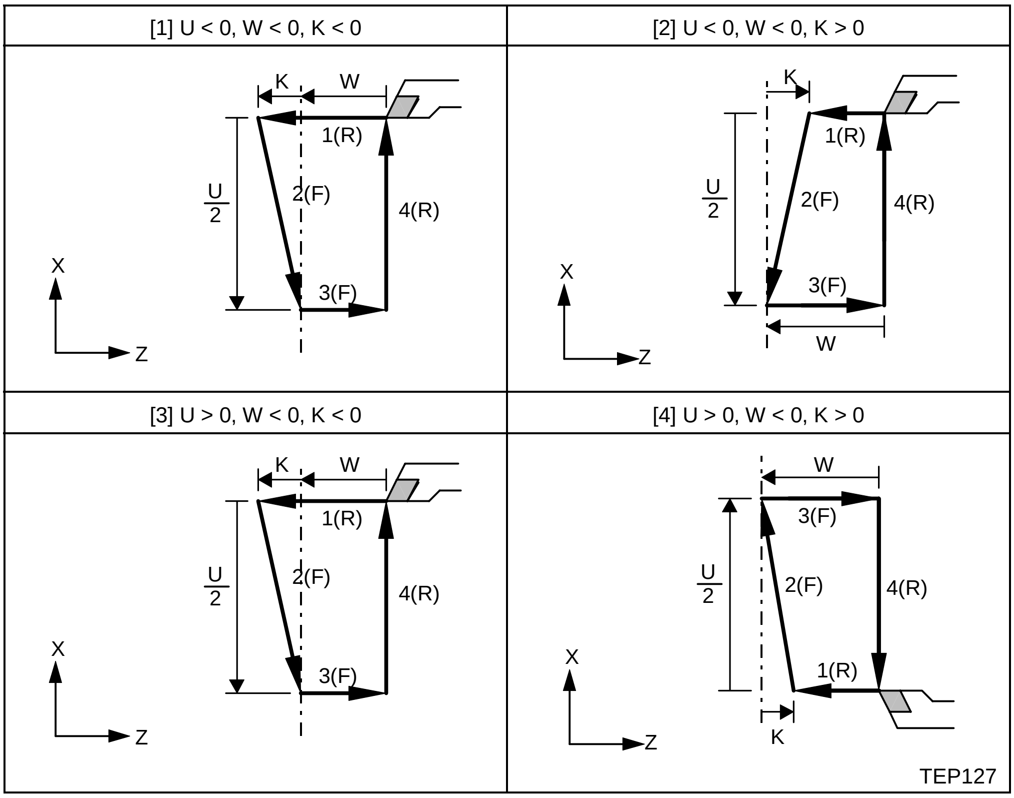
This function enables continuous taper turning in the face direction using the following command.

G94 X/U_Z/W_K_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 2 of parameter **P108**).
- Depending on the U, W and K signs, the following shapes are created.



TEP127

Program error 736 “ILLEGAL TAPER LENGTH” occurs in shapes [2] and [3] unless the following condition is satisfied.

$$| W | \geq | K |$$

13-2 Multiple Repetitive Fixed Cycles

These functions permit to execute the fixed cycle by designating a program in a block with corresponding G-code.
The types of multiple repetitive fixed cycles are listed below.

G-code	Function	
G70	Finishing cycle	Multiple repetitive fixed cycles I
G71	Longitudinal roughing cycle (roughing along finish shape)	
G72	Edge roughing cycle (roughing along finish shape)	
G73	Cast or forged workpiece roughing	
G74	Edge cut-off cycle	Multiple repetitive fixed cycles II
G75	Longitudinal cut-off cycle	
G76	Multiple repetitive threading cycle	

- If the finish shape program has not been entered in the memory, any of the above functions for the multiple repetitive fixed cycles I (G70 to G73) cannot be used.
- The program formats are as follows.

G-code	T32 compatible mode	Standard mode
G70	G70 A_P_Q_;	G70 A_P_Q_ ;
G71	G71 A_P_Q_U_W_D_F_S_T_;	G71 U_R_ ; G71 A P_Q_U_W_F_S_T_ ;
G72	G72 A_P_Q_U_W_D_F_S_T_;	G72 W_R_ ; G72 A_P_Q_U_W_F_S_T_ ;
G73	G73 A_P_Q_I_K_U_W_D_F_S_T_;	G73 U_W_R_; G73 P_Q_U_W_F_S_T_ ;
G74	G74 X(U)_Z(W)_I_K_D_F_S_T_;	G74 R_ ; G74 X(U)_Z(W)_P_Q_R_F_S_T_ ;
G75	G75 X(V)_Z(W)_I_K_D_F_S_T_;	G75 R_ ; G75 X(U)_Z(W)_P_Q_R_F_S_T_ ;
G76	G76 X(V)_Z(W)_I_K_D_A_ F_ S_T_; E_	G76 P_Q_R_ ; G76 X(U)_Z(W)_R_P_Q_F_ ;

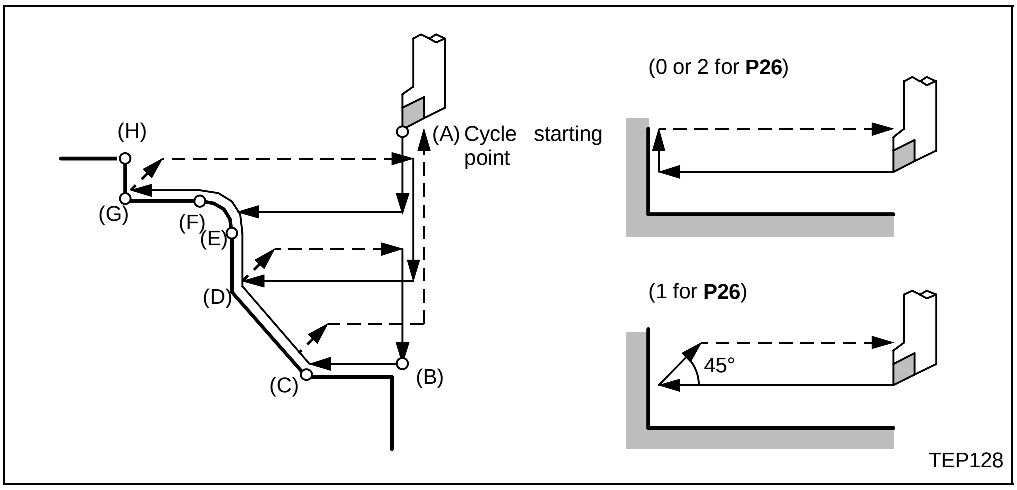
Return to Main Menu

13-2-1 Longitudinal roughing cycle : G71

1. T32 compatible mode

A. 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 **P26** 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 **P26**, chamfering speed can be changed. (Refer to parameter **K3**.)

B. Programming format

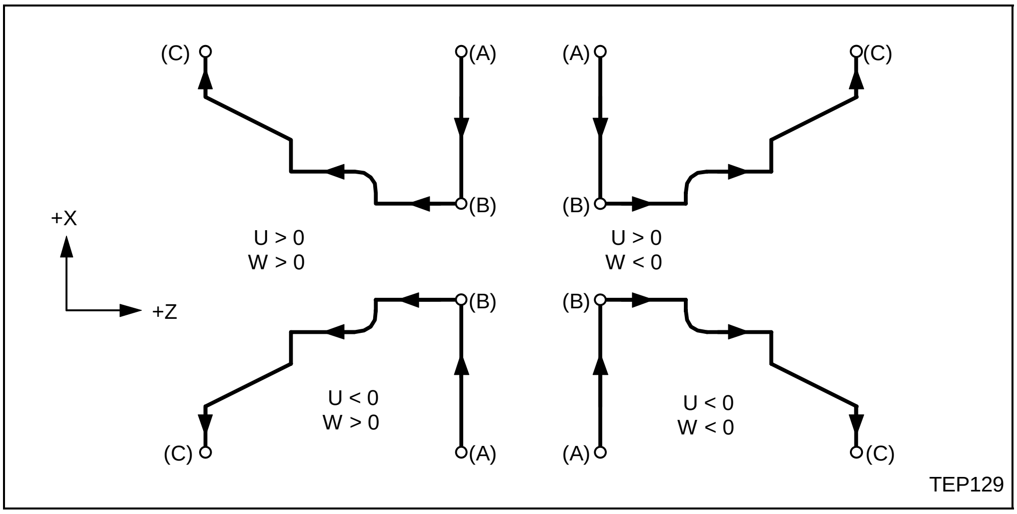
```
G71 A_P_Q_U_W_D_F_S_T_;
N○○○○
                 S_  ⎫
                 F_  ⎬ Finish shape between A to H in the figure above
                 F_  ⎪
                 S_  ⎭
N ＊＊＊＊
```

A : Finish shape program No. (if omitted, program number under execution)
P : Head sequence No. for finishing shape (○○○○)
Q : End sequence No. for finishing shape (＊＊＊＊)
U : Finishing allowance in X-axis direction (diametral value)
W : Finishing allowance in Z-axis direction
D : Cutting depth (radial value) . . . Absolute value, decimal point input not allowed
F : Roughing cycle feed rate
S : Peripheral speed or rpm for spindle in roughing cycle
T : T command

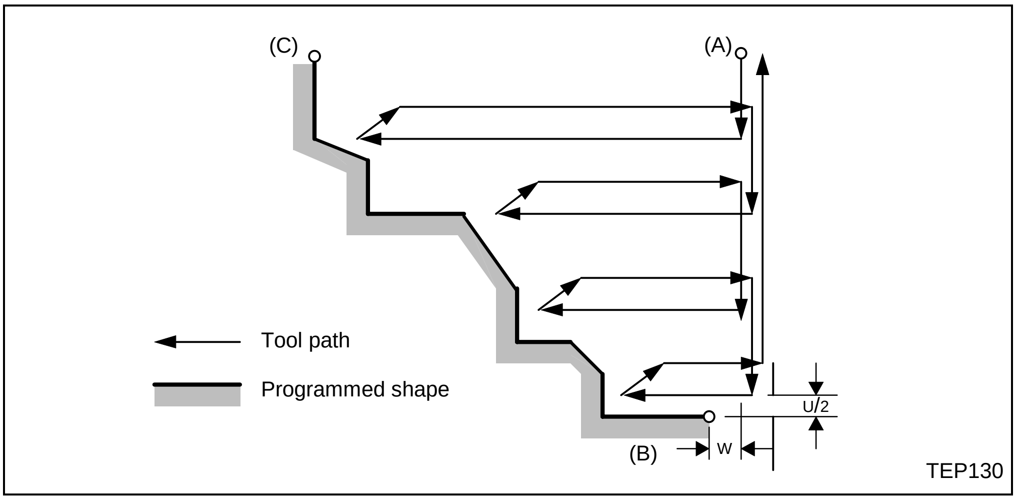
Note: Even if F and S commands exist in blocks defined by P and Q, they will be ignored during roughing cycle because they are considered for finishing cycle.

C. 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.

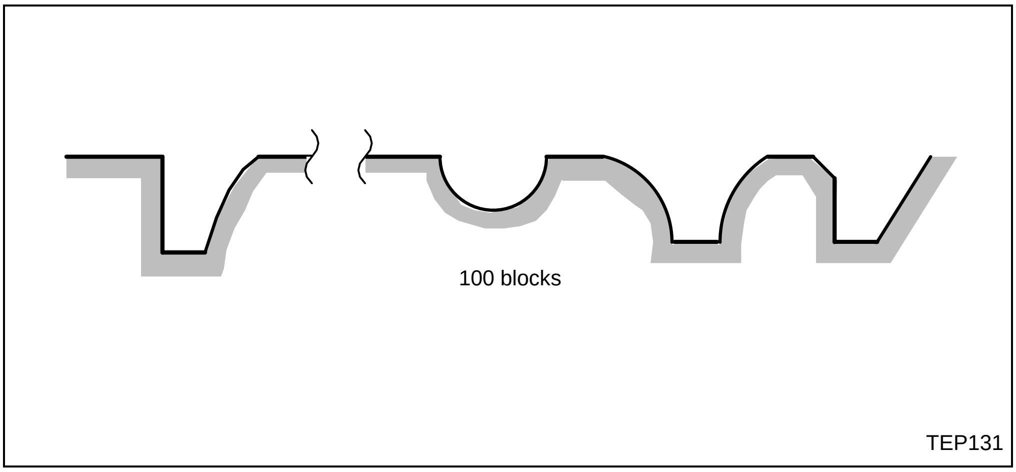


For section between A and B, command should be executed by sequence block of P data, with move command in X axis only.
For section between B and C, a maximum of 32 recesses are allowed.
When G00 is commanded for section between A and B, cutting during the cycle will be made by rapid feed. With G01, cutting feed rate will be used.

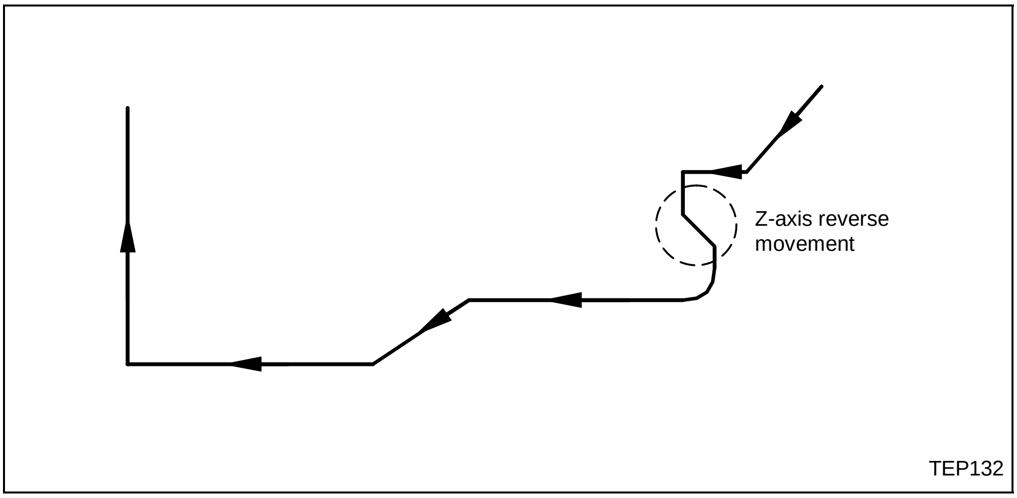


D. Remarks

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.

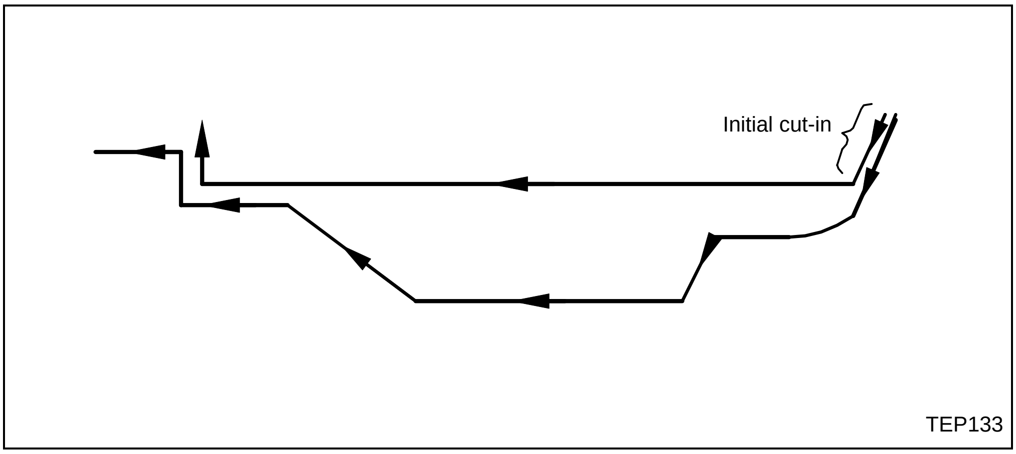


Coordinates, however, must be increased monotonously in Z-axis direction. For shapes as shown below, alarm 740 “ILLEGAL SHAPE DESIGNATED” will occur and machining stops.



Return to Main Menu

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.

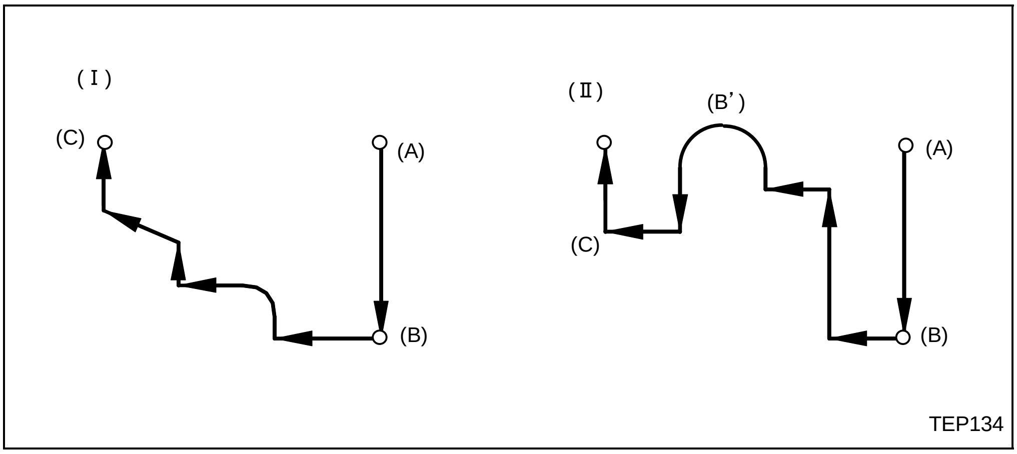


5. If this cycle is commanded during nose radius compensation mode, nose radius compensation will be effected on the finishing shape program, and this cycle will be accordingly executed for the compensated shape.

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.

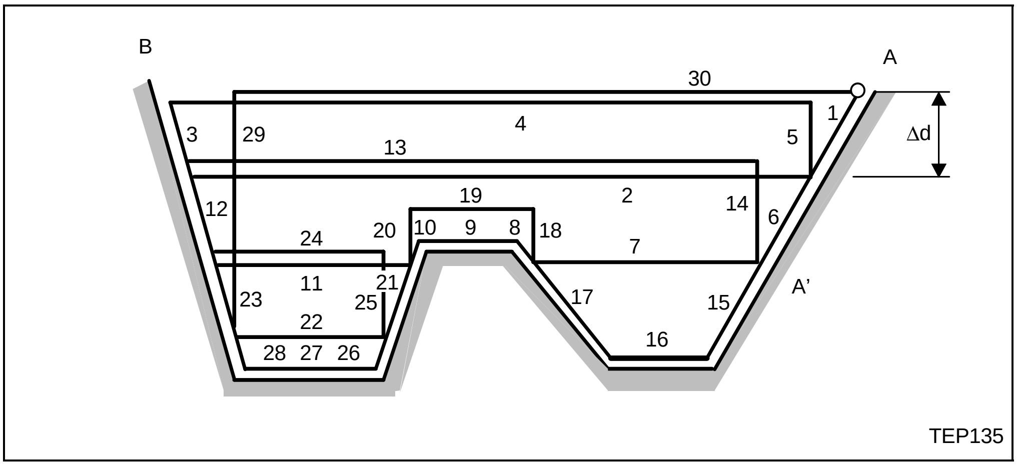
7. In the first block of repeating section, give a command for each of the pair X(U) and Z(W). If no Z-axis movement is made, W0 must be commanded.

8. In machining shape defined for A-B-C, if the command which results in A ≤ B is made as with B', alarm 740 "ILLEGAL SHAPE DESIGNATED" will be caused. Case (I) is correct while case (II) will cause alarm.



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.



2. Standard mode

A. Programming format

G71 U<u>Δd</u> R_;
G71 A_ P_ Q_ U<u>Δu</u> W_ F_ S_ T_;

U<u>Δd</u> : 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

U<u>Δu</u> : 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.

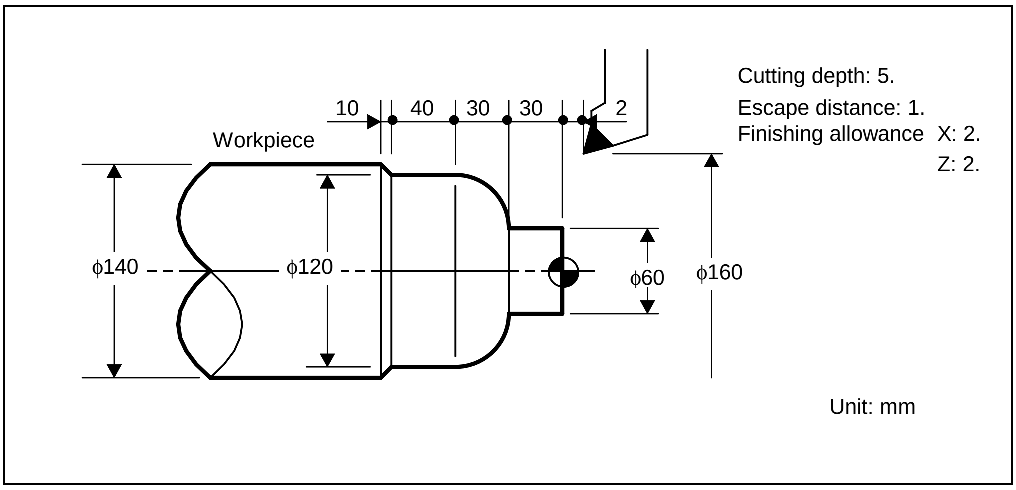
- Δd and Δu are both specified by address U. The differentiation depends on whether P and Q are specified in the same block.
- Other notes are as with T32 compatible mode.

B. Parameter

- Cutting depth can be set by parameter **P97**, whose value will be overridden with the program command.
- Escape distance can be set by parameter **U34**. The parameter setting value will be overridden with the program command.

Return to Main Menu

C. Sample programs



For T32 compatible mode

```
N001 G00 G96 G98;
N002 G28 U0 W0;
N003 X160.Z2.;
N010 G71 P012 Q016 U4.W2.D5.
     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;
```

For Standard mode

```
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;
```

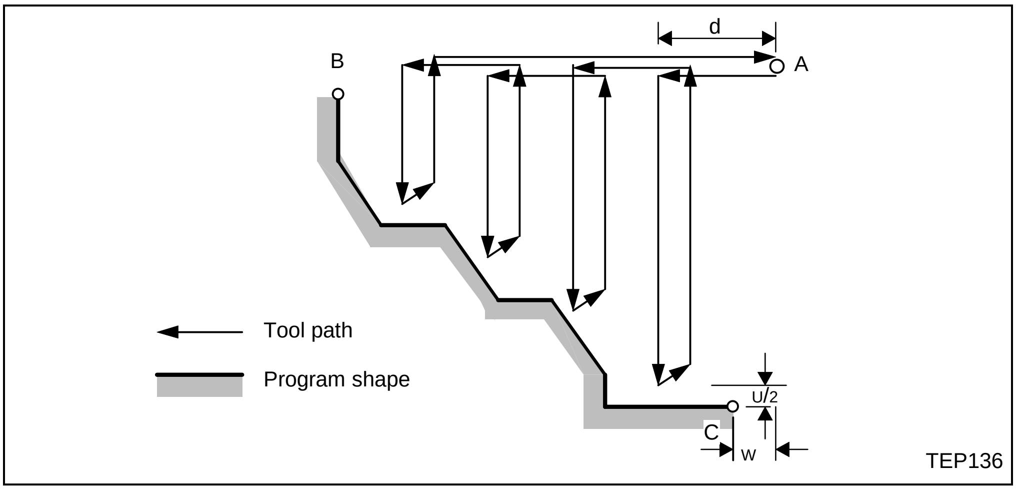
Return to Main Menu

13-2-2 Edge roughing cycle: G72

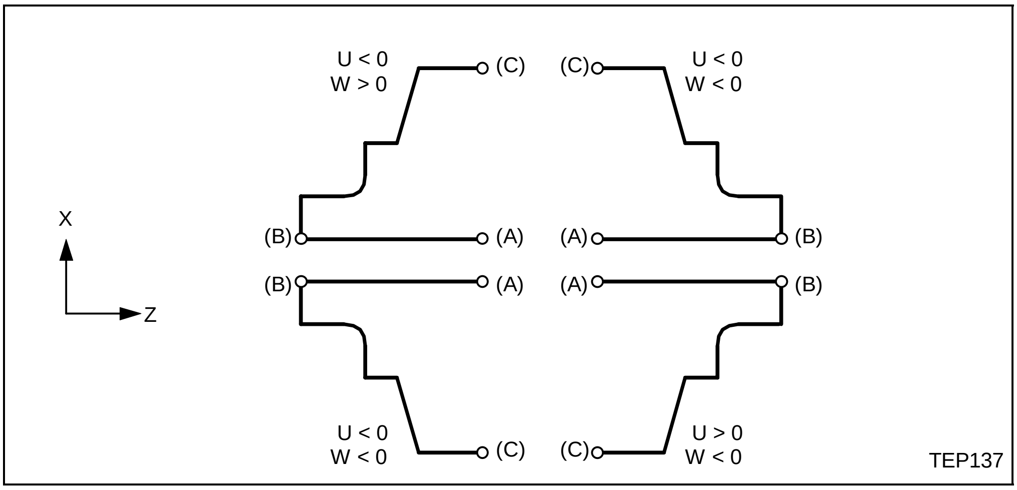
1. T32 compatible mode

G72 A_ P_ Q_ U_ W_ D_ F_ S_ T_ ;

Machining will be executed in the manner similar to G71. Movement, however, is parallel to the X-axis.



In G72, parameter **P26** will determine whether escape at 45° from wall should be selected or not in the cycle as in G71. By setting 2 for **P26**, feed can be accelerated in up-going slope. (Refer to parameter **K3**.)
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.

Return to Main Menu

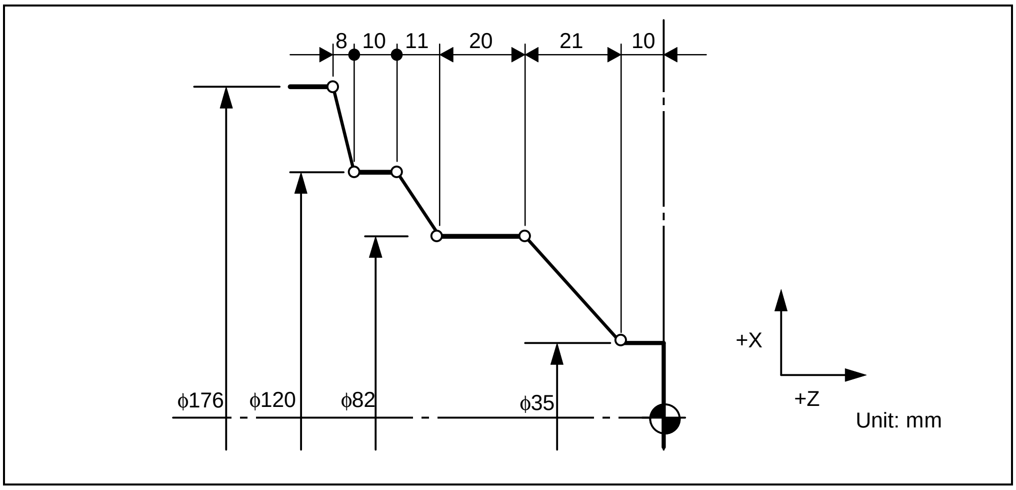
2. Standard mode

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 (standard mode).

3. Sample programs



For T32 compatible mode

```
N001  G00  G96  G98;
N002  G28  U0  W0;
N003  T0101;
N004  X176.Z2.;
N010  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;
```

For Standard mode

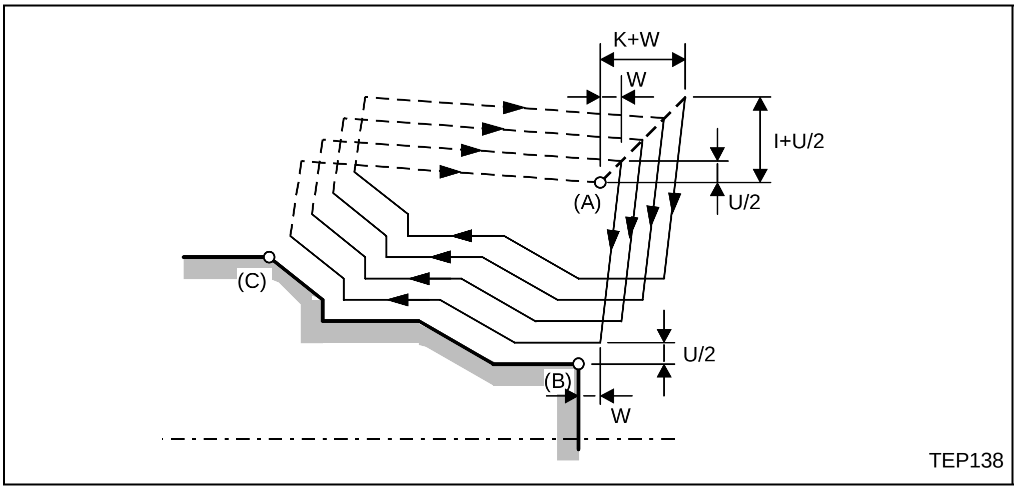
```
N001  G00  G96  G98;
N002  G28  U0  W0;
N003  T0101;
N003  X176.Z2.;
N010  G72  W7.R1.;
N011  G72  P012  Q018  U4.W2.D7.F100
      S100  M3;
N012  G00  Z-80.S150;
N013  G01  X120.W8.F100;
N014  W10.;
N015  X82.W11.;
N016  W20.;
N017  X35.W21;
N018  W12.;
N019  G70  P012  Q018;
N020  G28  U0  W0  M5;
N021  M30;
```

13-2-3 Cast or forged workpiece roughing cycle: G73

1. T32 compatible mode

A. Overview

This function will allow efficient execution in roughing when cast or forged parts are to be cut along finish shape.



B. Programming format

G73 A_ P_ Q_ I_ K_ U_ W_ D_ F_ S_ T_;

```
NOOOO
  ⎰          F_ ⎫
  ⎱          S_ ⎬ Finish shape between B and C
N＊＊＊＊         ⎭
```

A : Finsih shape program No. (if omitted, program under execution)

P : Head sequence No. for finish shape (OOOO)

Q : End sequence No. for finish shape (＊＊＊＊)

I : Escape distance in X-axis direction (radial value)

K : Escape distance in Z-axis direction

U : Finishing allowance in X-axis direction (diametral value)

W : Finishing allowance in Z-axis direction

D : Number of infeed for roughing

T : T command

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.

C. Detailed description

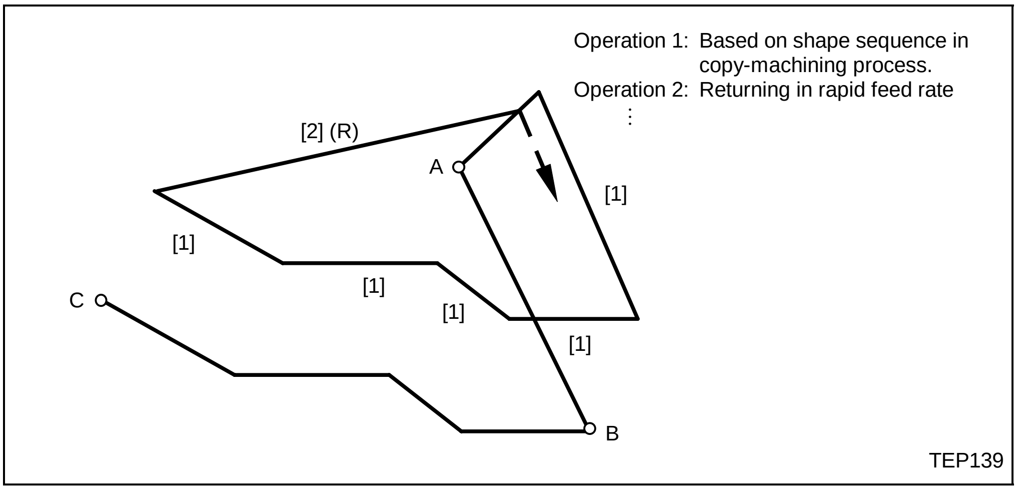
- Finish shape

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.


Operation 1: Based on shape sequence in copy-machining process.
Operation 2: Returning in rapid feed rate
⋮
[2] (R)
A
[1]
[1]
C
[1]
[1]
[1]
B

TEP139

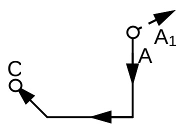
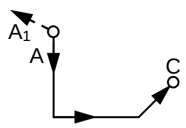
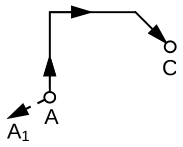
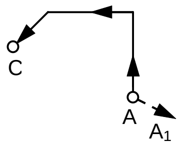
- Tool nose radius compensation

  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	–	–	+

Return to Main Menu

2. Standard mode

A. 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.

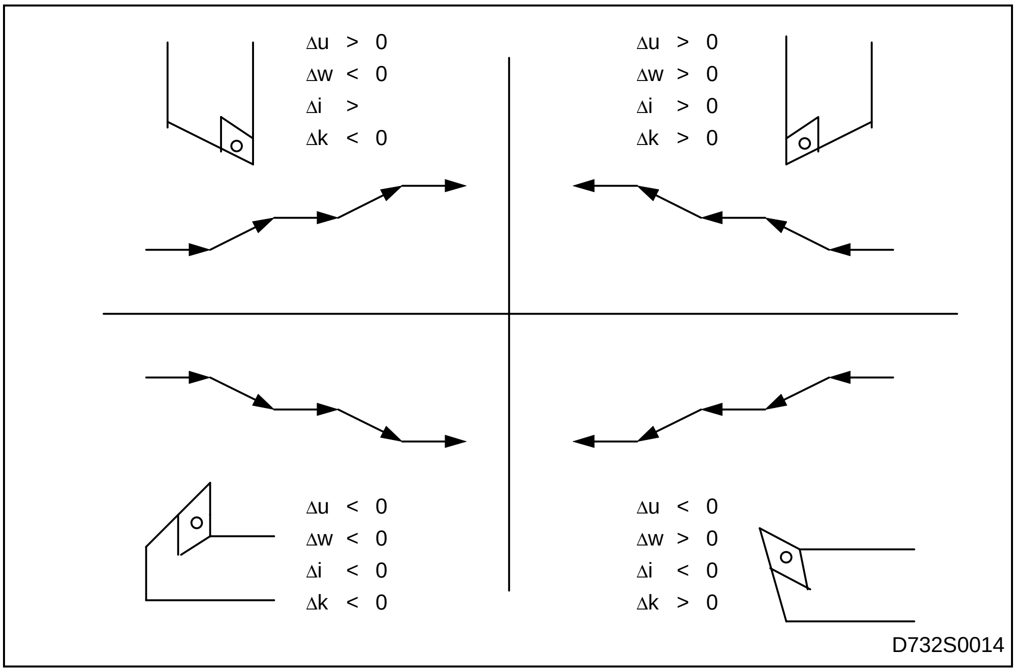
* Other addresses are as with G71 (standard mode).

B. Parameter

- Escape distance and direction in the X-axis direction can be set by parameter **P98**. Parameter setting values will be overridden with the program command.
- Escape distance and direction in the Z-axis direction can be set by parameter **P99**. Parameter setting values vary according to the program command.
- Times of divisions can be set by parameter **P33**. Parameter setting values vary according to the program command.

C. Detailed description

- 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.

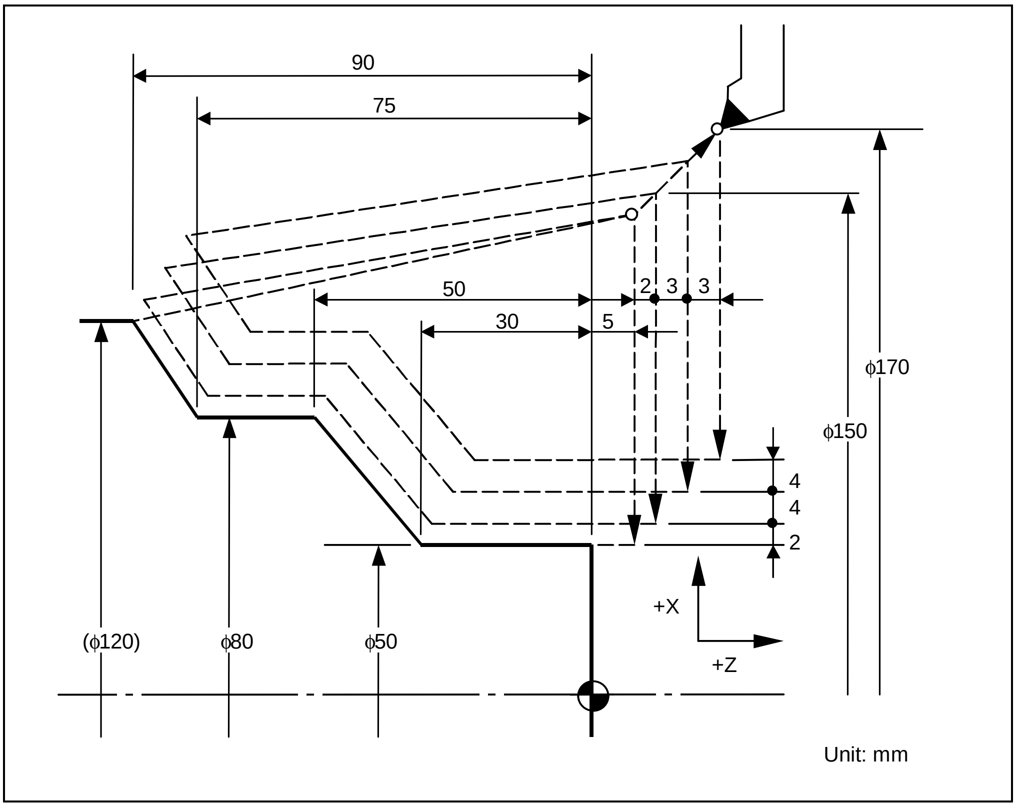


Δi, Δk and Δu, Δw are both specified by addresses U and W. The differentiation is given by whether P and Q are commanded in the same block. That is, addresses U and W when P and

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.

3. Sample programs



For T32 compatible mode

```
N010  G00 G96  G98;
N011  G28 U0 W0;
N012  T0101;
N013  X150.Z5.;
N014  G73 P016 Q020 I8.K6.U4.
      W2.D3.F150 S100 M3;

N016  G00 X50.;
N017  G01Z-30;
N018  X80.Z-50.;
N019  Z-75.;
N020  X120.Z-90.;
N021  G70 P016 Q020;
N022  G28 U0 W0 M5;
N023  M30.;
```

For Standard mode

```
N010  G00 G96 G98;
N011  G28 U0 W0;
N012  T0101;
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.;
```

13-2-4 Finishing cycle: G70

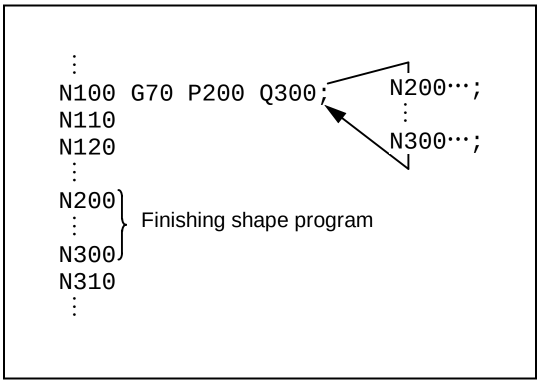
After roughing have been carried out by the G71 to G73 commands, finishing can be performed by the following programming format.

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

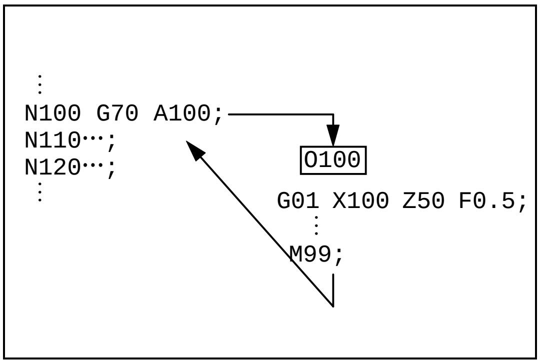
- The F, S and T commands in the finishing shape program are valid during the finishing cycle.
- When the G70 cycle is completed, the tool returns to the starting point by rapid feed and the next block is read.

Example 1: When designating a sequence number



```
 ⋮
N100 G70 P200 Q300;     N200···;
N110                     ⋮
N120                    N300···;
 ⋮
N200
 ⋮
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.

Note: For G70, T32 compatible mode and standard mode are the same.

13-2-5 Edge cut-off cycle: G74

1. T32 compatible mode

A. Overview

This function is used for smooth disposal of machining chips in edge cut-off machining. For SS materials which produce hard-to-cut machining chips this function can be managed for easy machining chip disposal.

B. Programming format

G74 X(U)_ Z(W)_ I_ K_ D_ F_ S_ T_;

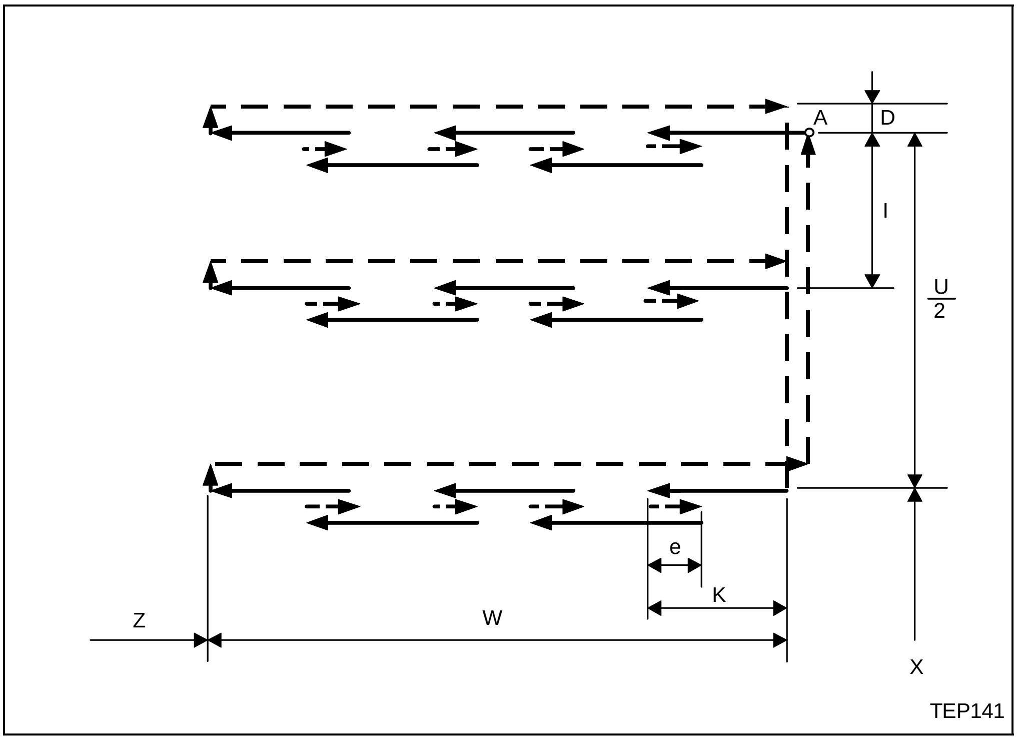
X :
U :
Z :
W : } Refer to the illustration below.

I : X-axis movement distance
K : Z-axis cut depth } Absolute data

D : X-axis escape distance (decimal point input not allowed)

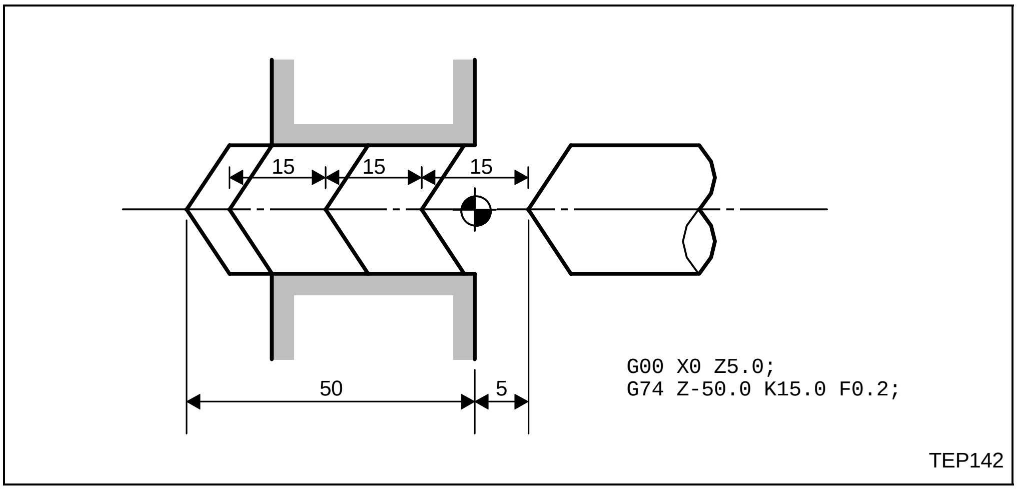
F, S, T: These are as with G71.

The distance “e” is set by parameter **U41** (pecking return distance in grooving process).



C. Remarks

1. For drilling X(U), I and D are not required. Omit these data.

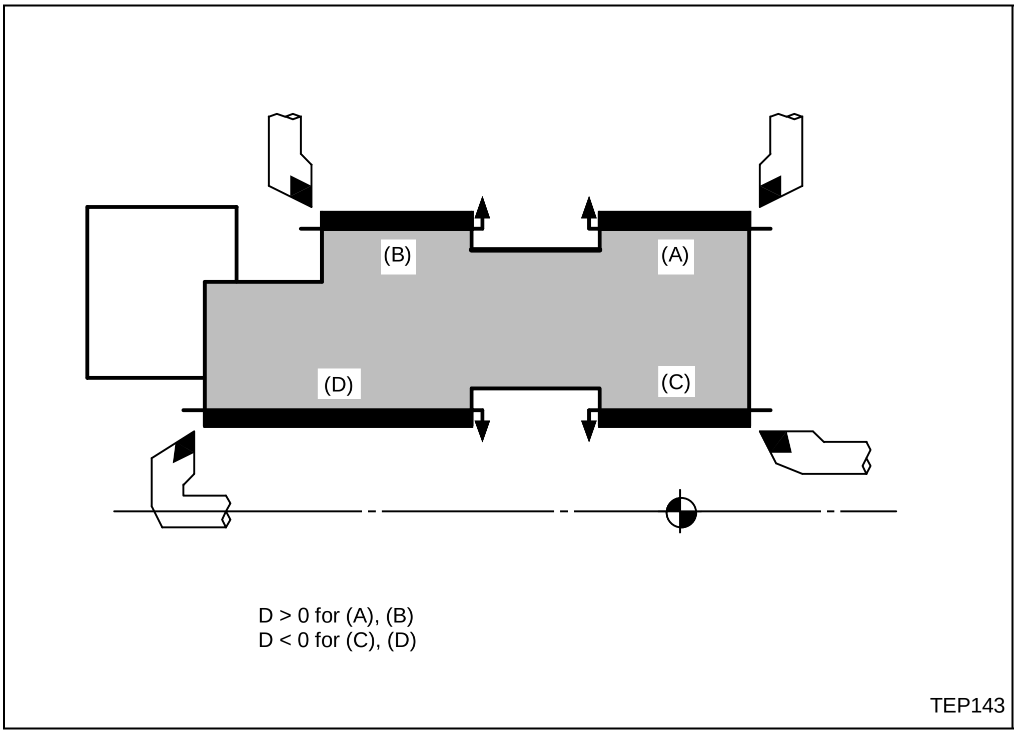


```
G00 X0 Z5.0;
G74 Z-50.0 K15.0 F0.2;
```

TEP142

2. Without D, escape will be considered as 0. Normally D is specified with plus data. When X (U) and I are omitted in outside or inside diameter machining, however, D requires a sign.

   Four combinations of G74



D > 0 for (A), (B)
D < 0 for (C), (D)

TEP143

3. During single block operation, all the blocks are executed step by step.
4. Omission of address I enables the operation of Z-axis alone, resulting in peck drilling cycle.

2. Standard mode

A. Programming format

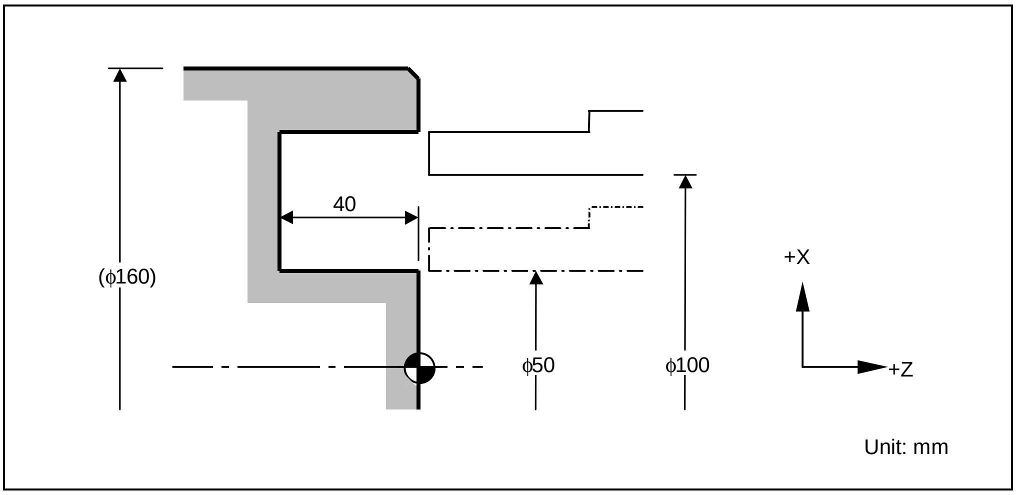
G74 Re;
G74 Xx/Uu Zz/Ww PΔi QΔk RΔd Ff Ss Tt;

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

B. Remarks

1. During single block operation, all the blocks are executed step by step.
2. Omission of address X (U) and P 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.

3. Sample programs



For T32 compatible mode

```
G00 G96 G98;
G28 U0 W0;
T0101;
X100.Z2.;
G74 U-50.Z-40.I5.K7.F150 S100 M3;

G28 U0 W0;
M30;
```

For Standard mode

```
G00 G96 G98;
G28 U0 W0;
T0101;
X100.Z2.;
G74 R2.;
G74 U-50.Z-40.P5.Q7.F150 S100 M3;
G28 U0 W0;
M30;
```

Return to Main Menu

13-2-6 Longitudinal cut-off cycle : G75

1. T32 compatible mode

A. Overview

This function is used for smooth disposal of machining chips in longitudinal cut-off machining. This allows easy disposal of machining chips in edge machining as well.

B. Programming format

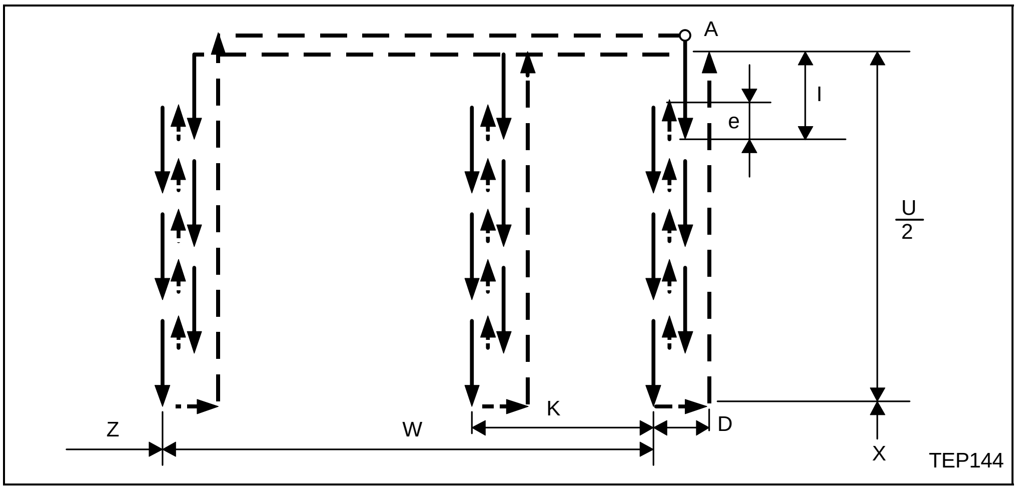
G75 X(U)_ Z(W)_ I_ K_ D_ F_ S_ T_ ;

X :
U :
Z :
W :
} Refer to the illustration below.

I : X-axis cut depth
K : Z-axis movement distance
} Absolute data
D : Z-axis escape distance (decimal point input not allowed)
F, S, T: These are as with G71.

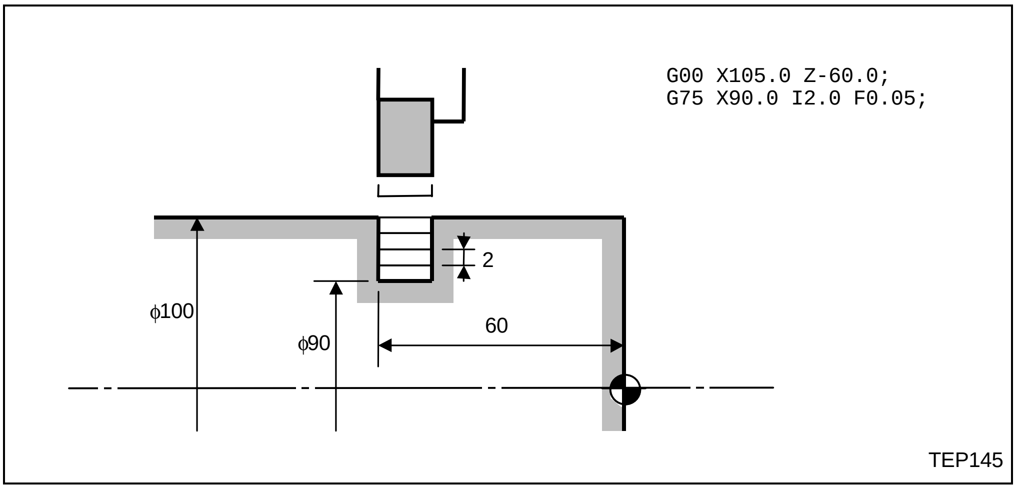
The distance “e” is set by parameter **U41** (pecking return distance in grooving process).

G75 executes cycle as shown below.

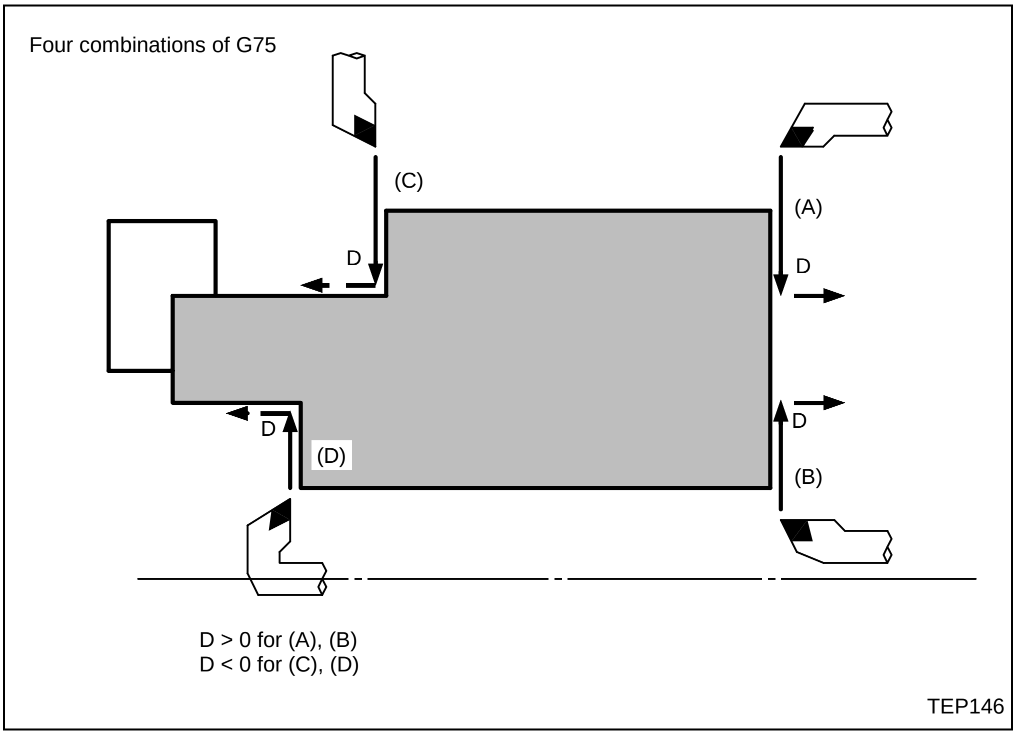


C. Remarks

1. For outside and inside diameter groove machining, Z(W), K and D are not required. Omit these data.



2. Without D, escape distance in Z-axis direction will be considered as 0. Normally D is specified with plus data. When Z(W) and K are omitted in edge machining, however, D requires a sign.



3. During single block operation, all the blocks are executed step by step.

2. Standard mode

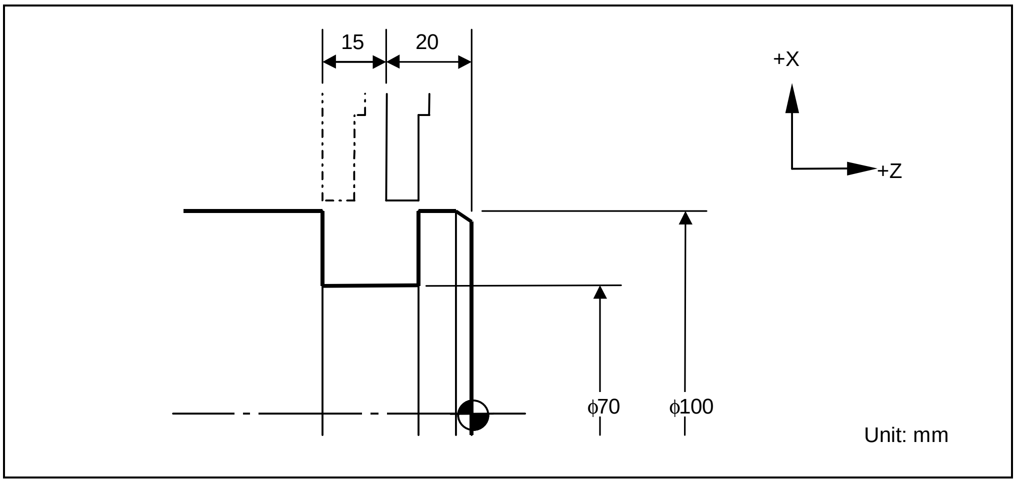
A. Programming format

G75 Re ;
G75 X(U)_ Z(W)_ PΔi QΔk RΔd F_ S_ T_ ;

B. 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 **U41**. The parameter setting value will be overridden with program command.
3. During single block operation, all the blocks are executed step by step.

C. Sample programs



For T32 compatible mode

```
G00 G96 G98;
G28 U0 W0;
T0101;
X102.Z-20.;
G75 W-15.X70.I6.K5.F150 S100 M3;

G28 U0 W0;
M30;
```

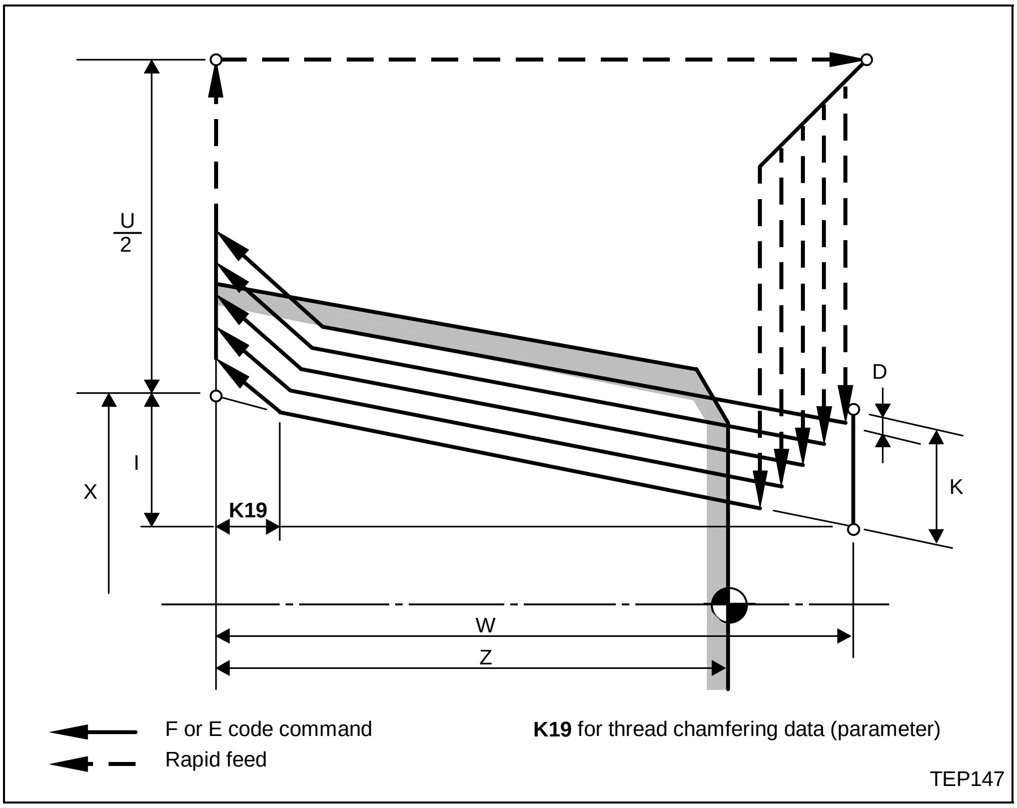
For Standard mode

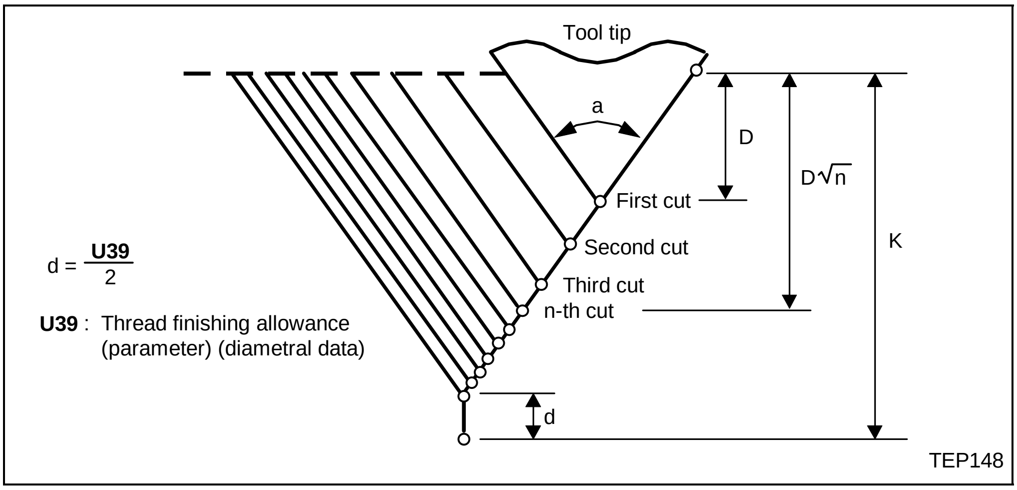
```
G00 G96 G98;
G28 U0 W0;
T0101;
X102.Z-20.;
G75 R2.;
G75 W-15.X70.P6.Q5.F150 S100 M3;
G28 U0 W0;
M30;
```

13-2-7 Multiple repetitive threading cycle: G76

1. T32 compatible mode

A. Cycle configuration





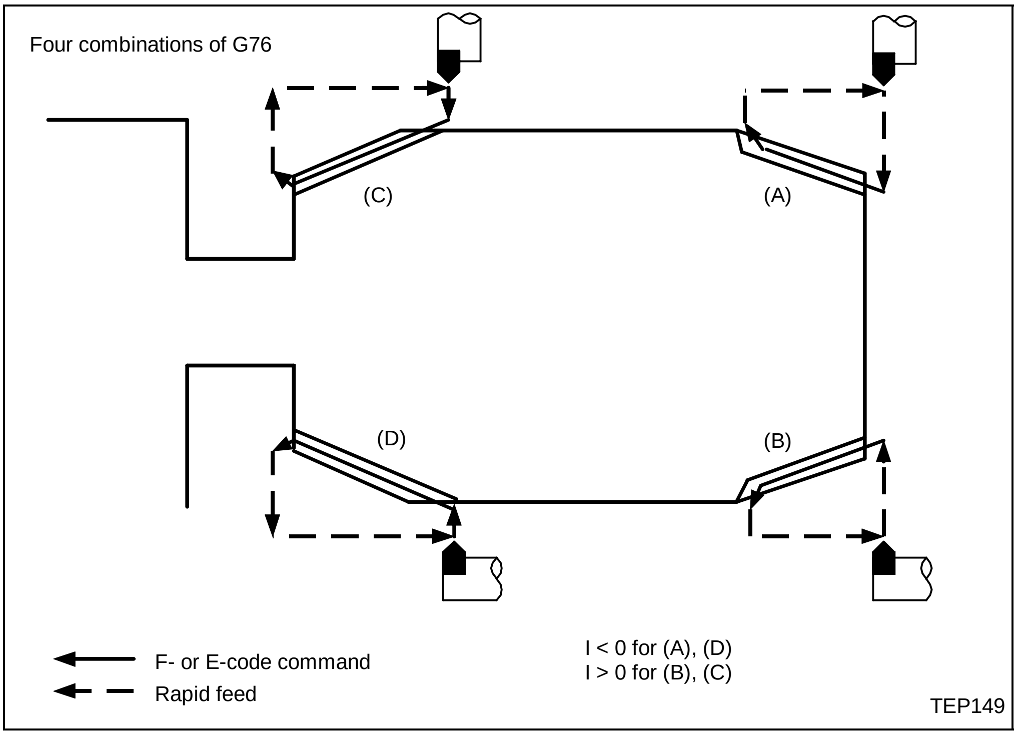
B. Programming format

G76 X(U)_ Z(W)_ I_ K_ D_ A_ $\left\{\begin{array}{l}F\_\\E\_\end{array}\right.$ S_ T_;

X : Thread bottom diameter at ending point of threading
U : X-axis movement distance from cycle starting point to ending point (thread bottom)
Z : Z-axis coordinate at ending point of threading
W : Z-axis movement distance from starting point to ending point of threading
I : Radial difference at starting and ending of threading portion
(generally I < 0 for outside diameter, and I > 0 for inside diameter)
K : Thread height (radial data)
F, E : Lead of thread
A : Tool tip angle (single edge cutting data which can be commanded with a resolution of 1 degree in a range 0° to 120°)
D : First cut depth (radial data) (decimal point input not allowed)
S, T : These are as with G71.

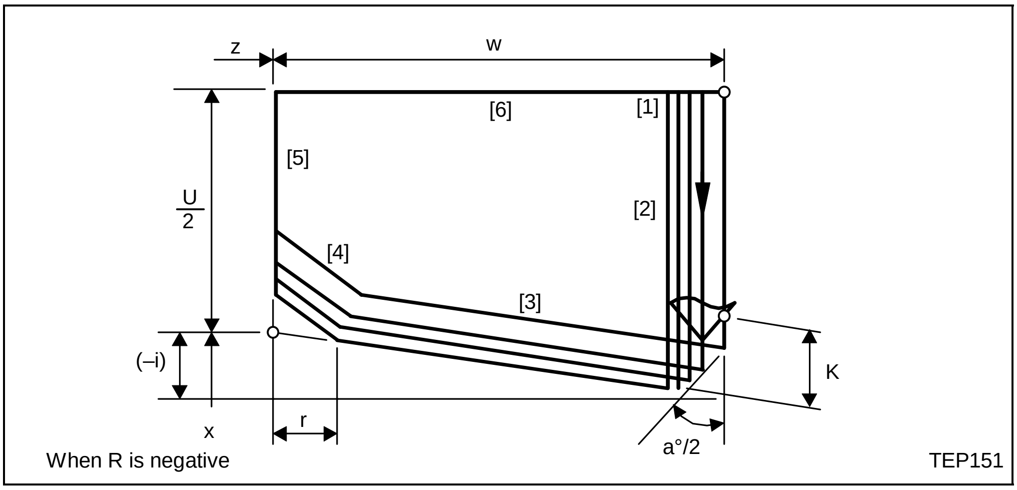
C. Detailed description

- Chamfering data can be set by parameter **K19** by 0.1 × L units in a range 0.1 × L to 4.0 × L (L as lead length).
- Cut depth is determined with D for initial cut, and $D\sqrt{n}$ for n-th cut to have a constant depth for each cut.



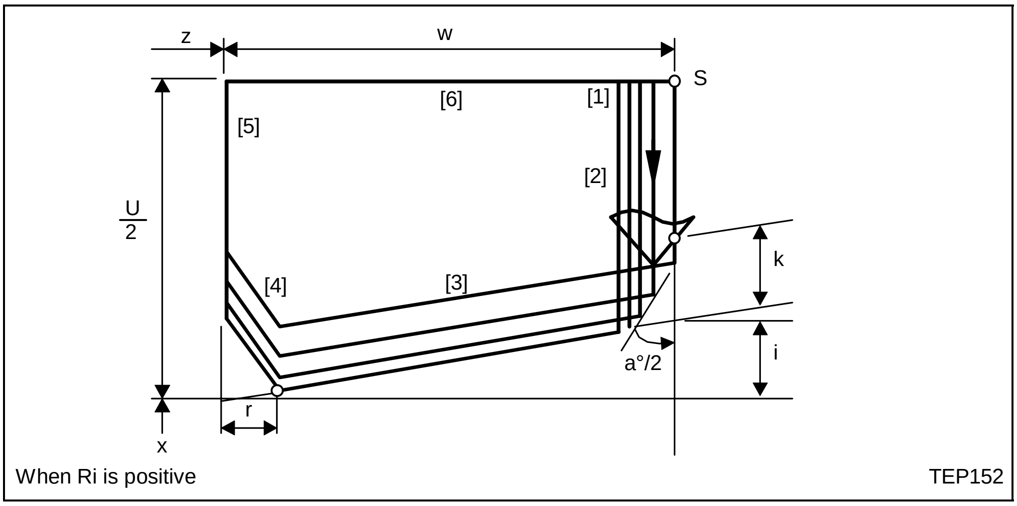
- One cycle configuration

  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].



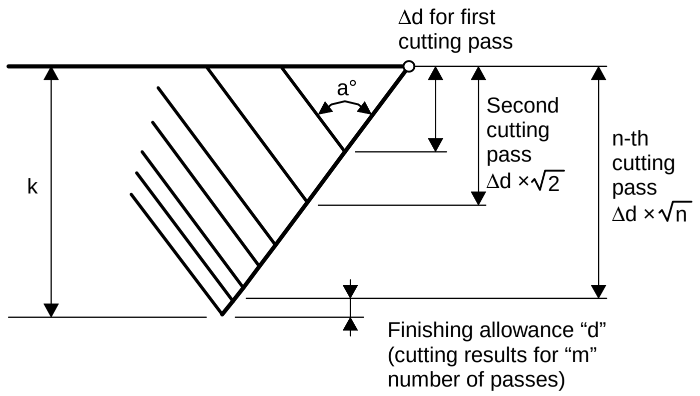
When R is negative

TEP151



When Ri is positive

TEP152


Δd for first cutting pass
a°
Second cutting pass Δd ×√2
n-th cutting pass Δd ×√n
k
Finishing allowance “d” (cutting results for “m” number of passes)

TEP153

Return to Main Menu

D. 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 **P13** bit 5 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 6-8-6 for more information.

2. Standard mode

A. Programming format

G76 Pmra Rd; (omission allowed)
G76 Xx/Uu Zz/Ww Ri Pk Q Δd Fℓ S_ T_;

m : Repeat times of final finishing (1 to 99)
This command is modal and valid until a new value is commanded.

r : Chamferring amount of threading
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)

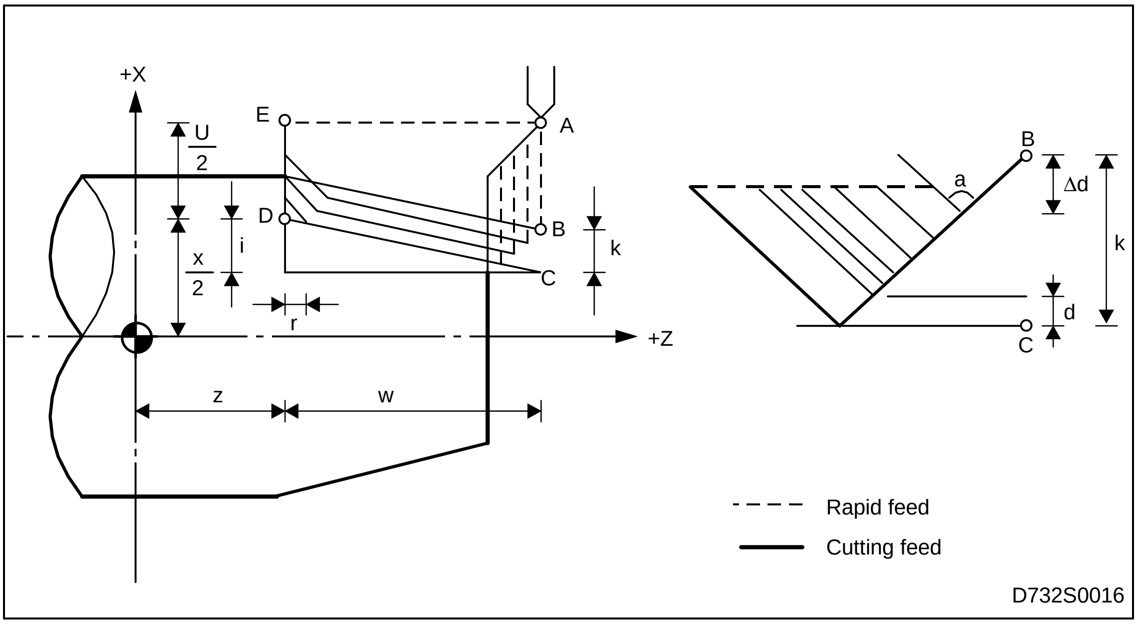
Δd : First cut depth (radial data)

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 02 12 60
a r m

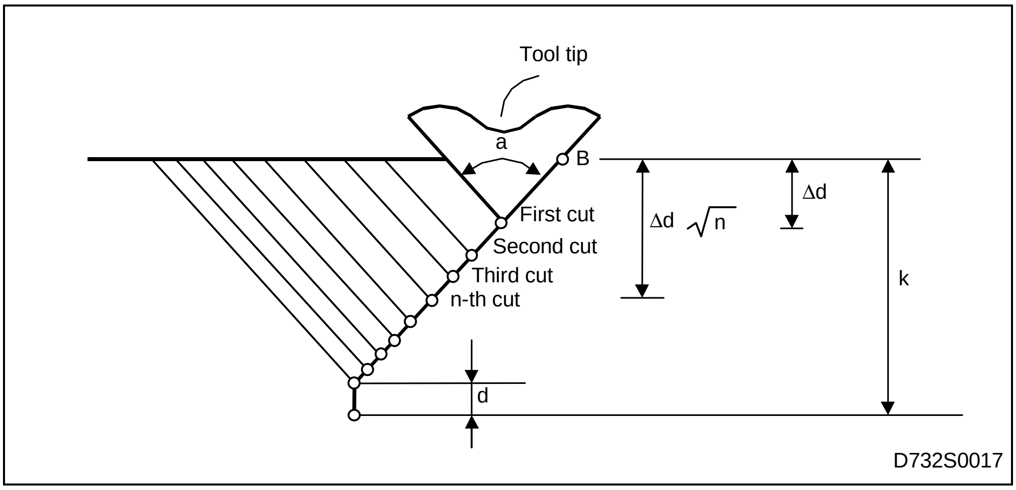


B. Parameter

- Repeat times of final finishing can be set by parameter **P34**. Parameter setting values will be overridden with program command.
- Chamferring amount can be set by parameter **K19**. The parameter setting value will be overridden with program command.
- Tool tip angle can be set by parameter **K67**. Parameter setting values will be overridden with program command.
- Finishing allowance can be set by a parameter **U39**. The parameter setting will be overridden with program command.

C. Detailed description

- Setting the tool tip angle provides the machining of a single tip, permitt-ing 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.

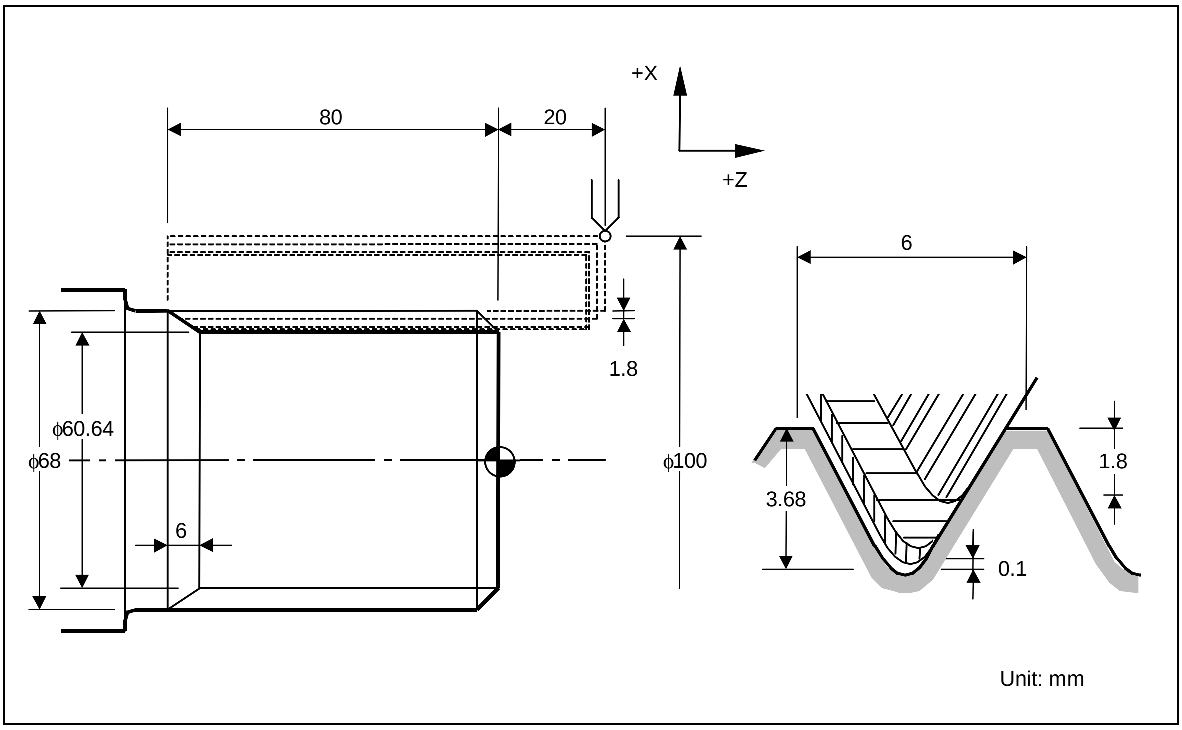
- Threading cycle provides a feed commanded by F code or E code only between C and D, and rapid feed for others.
- For the cycle shown above, the signs of increment are as follows:

  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; diameter value) can be set by parameter (U39) within the range as follows:

  0 to 65.535 mm (6.5535 inches)

3. Sample programes



For T32 compatible mode

```
G00 G97 G99;
G28 U0 W0;
T0101;
S500 M3;
X100.Z20.;
G76  X60.64  Z-80.K3.68  D1.8  A60
F6.0;
G28 U0 W0 M5;
M30;
```

For Standard mode

```
G00 G97 G99;
G28 U0 W0;
T0101;
S500 M3;
X100.Z20.;
G76 P011060 R0.2;
G76 X60.64 Z-80.P3.68 Q1.8 F6.0;
G28 U0 W0 M5;
M30;
```

Return to Main Menu

4. 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 (**P13** bit 5 = 1), the tool stops at the chamferring position at that moment (see item 3 below).
   Refer to G92 threading cycle for details.

2. Chamferring angle can be set in parameter **P30** 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 (**P13** bit 5).
   - 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, the next block is also executed in sequence.

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 6-8-6 for more information.

13-2-8 Checkpoints for multiple repetitive fixed cycles: G70 to G76

1. Except for the parameters which have been preset, set all the required parameters in the blocks for the multiple repetitive fixed cycle commands.

2. Provided that the finishing shape sequence has been registered in the memory, multiple repetitive 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.

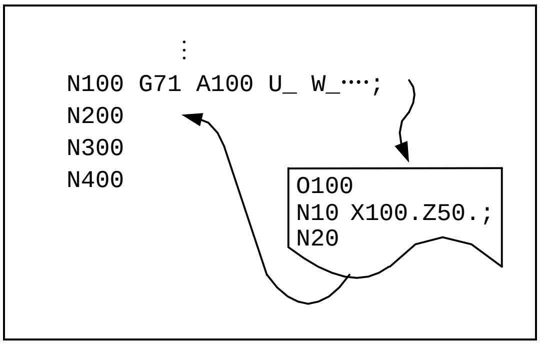
4. The finishing shape sequence specified to addresses P and Q in the blocks G71 to G73 should be prepared so that the maximum number of blocks is 100 for all the commands for corner chamfering, corner rounding and other commands including the automatic insertion blocks based on tool nose radius compensation. 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 number is designated The next block 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 number is designate The next block 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 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 multiple repetitive 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 multiple repetitive 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. Multiple repetitive fixed cycle commands are unmodal commands and so they must be set every time they are required.

15. Programming error “ILLEGAL SHAPE DESIGNATED” 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.

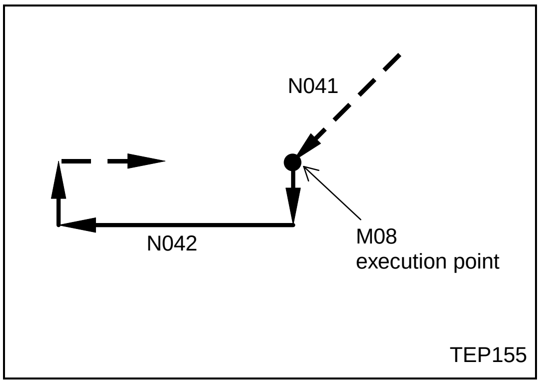
18. In blocks for finishing shape defined by P and Q for G70 to G73, if command for final shape is chamfering (G01 X_ I_) (G01 Z_ K_) or corner rounding (G01 Z_ R_) (G01 X_ R_), alarm “NO DIRECTIVE FOR NEXT MOVE R/C” occurs.

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.;
```



TEP155

Return to Main Menu

13-3 Hole Machining Fixed Cycles: G80 to G89

13-3-1 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.

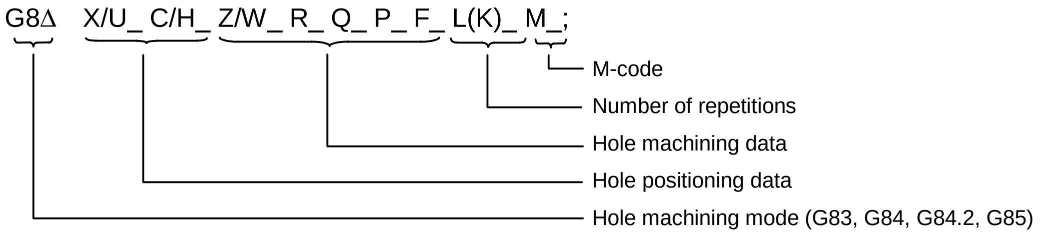
The following types of fixed cycles for hole machining are available.

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 command in the 01 group is set. The various data will also be cleared simultaneously to zero.

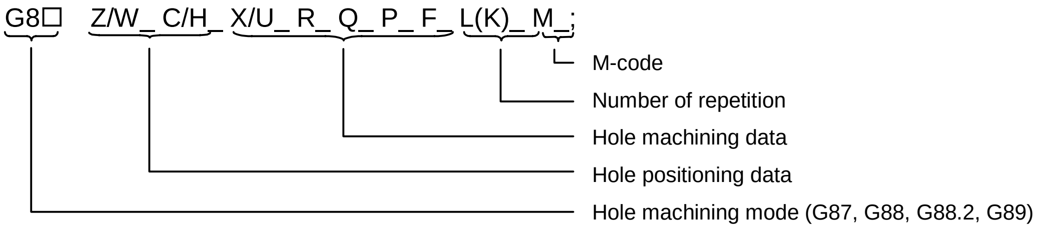
2. Programming format

A. Edge hole machining



G8Δ X/U_ C/H_ Z/W_ R_ Q_ P_ F_ L(K)_ M_;

B. Longitudinal hole machining



G8□ Z/W_ C/H_ X/U_ R_ Q_ P_ F_ L(K)_ M_;

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

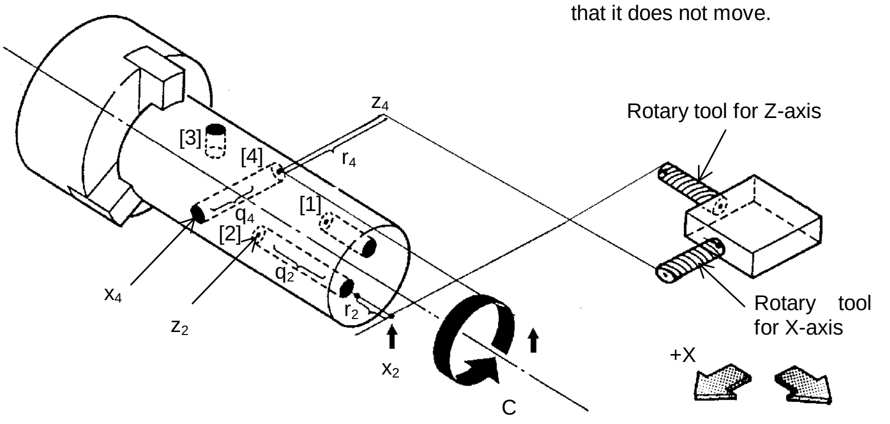
* Addresses in parentheses apply for commands G87, G88, G88.2 and G89.

E. Use of the hole machining fixed cycles on the 2nd spindle side

For a machine whose secondary spindle is equipped with the function for the C-axis control or 0.001-degree index (orientation), the hole machining fixed cycles can also be used on the 2nd spindle side with the aid of the related G-code (G110).
See Section 20-2 for details on the programming method.

3. Outline drawing

The hole machining axes for the hole machining fixed cycle and the positioning are shown in the outline drawing below.

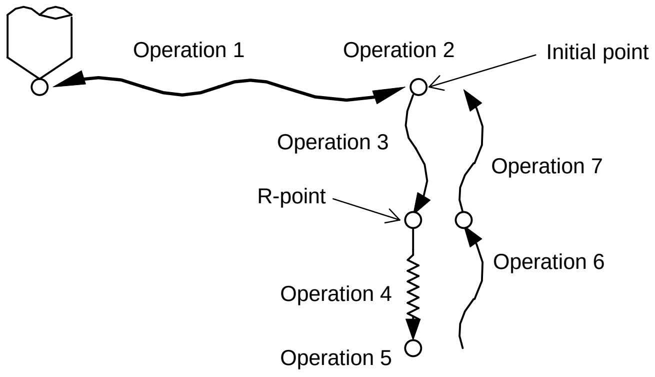


During the hole machining cycle, the C-axis (spindle) is clamped so that it does not move.

[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 (M04), rotary tools forward rotation (M03) and dwell.

Operation 6 : Return to the R-point

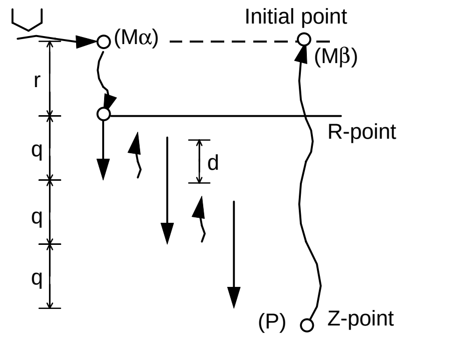
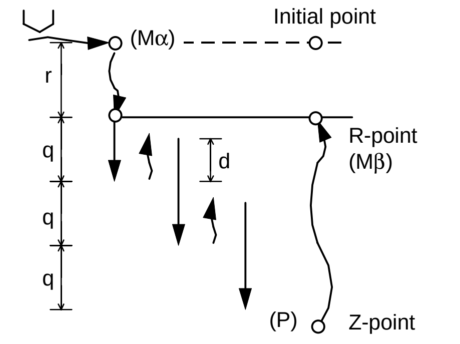
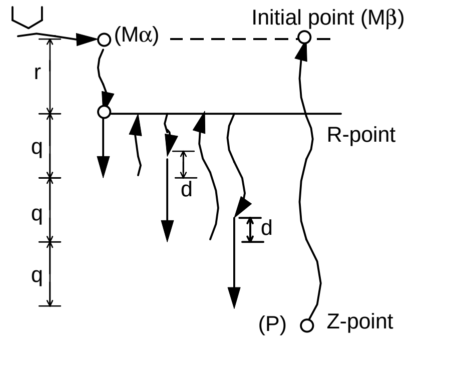
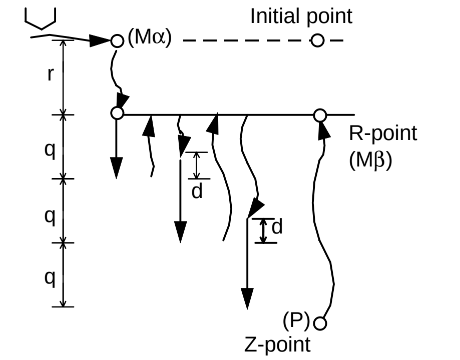
Operation 7 : Return to the initial point at rapid feed

(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 **P10** bit 7.
Parameter **P10** bit 7 = 0: Initial level return
Parameter **P10** bit 7 = 1: R-point level return

13-3-2 Face/Longitudinal deep hole drilling cycle: G83/G87

1. When the Q command is present (deep hole drilling)

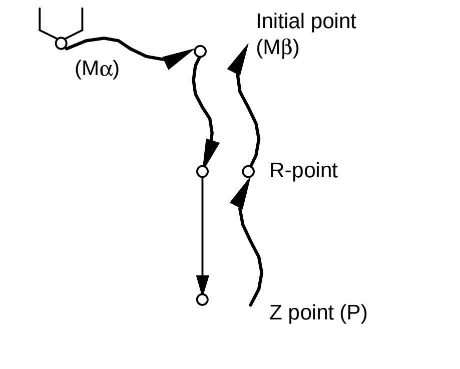
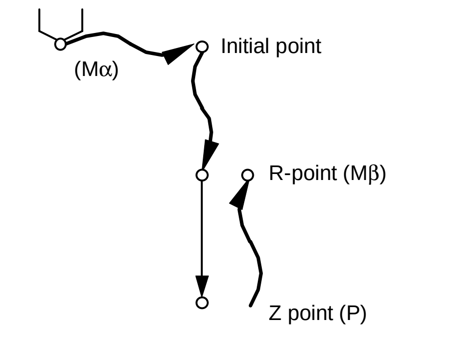
G83(G87)X(Z)_ C_ Z(X)_ Rr Qq Pp Ff Ll Mm ;

Type	Parameter **P10** bit 7 : 0	Parameter **P10** bit 7 : 1
A (high speed)	 Initial point (Mα) (Mβ) r q q q d R-point (P) Z-point	 Initial point (Mα) r q q q d R-point (Mβ) (P) Z-point
B (normal speed)	 Initial point (Mβ) (Mα) r q q q d d R-point (P) Z-point	 Initial point (Mα) r q q q d d R-point (Mβ) (P) Z-point TEP157

- Types A and B can be selected by the parameter (**P10** bit 6 : Selection between return to R-point/by escape distance in deep hole drilling cycle G83/G87)
- Return distance "d" is set by the parameter (**U45** : 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_ ;

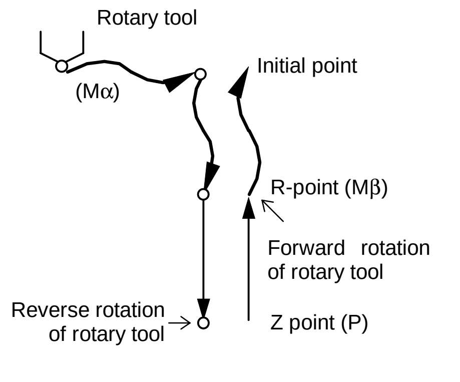
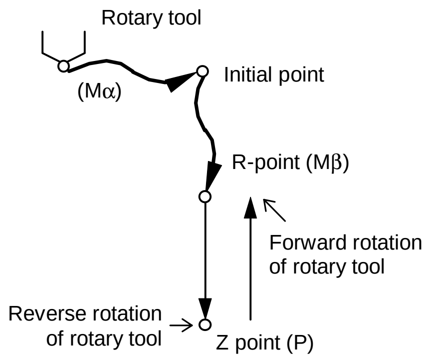
Parameter **P10** bit 7 : 0	Parameter **P10** bit 7 : 1
	

TEP158

See 1 for details on (Mα), (Mβ) and (P).

13-3-3 Face/Longitudinal tapping cycle: G84/G88

G84 (G88) X(Z)_ C_ Z(X)_ R_ P_ F_ L_ M_ ;

Parameter **P10** bit 7 : 0	Parameter **P10** bit 7 : 1
	

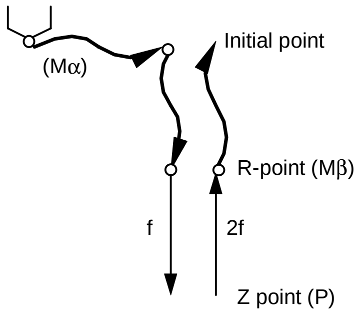
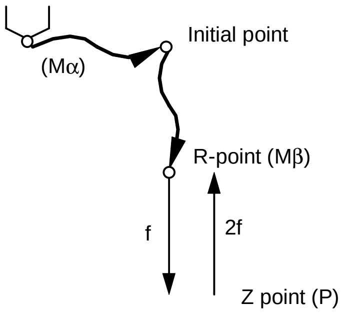
TEP159

- See 1 Section 13-3-1 for details on (Mα), (Mβ) and (P).
- 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 M-code for specifying rotation direction which is to be output at hole bottom can be selected by setting parameter **P12** bit 5. At R-point, M-code for direction reverse to the selected direction will be output. For M-code command specifying rotation direction of milling tool, the parameter **B82** (**C82**) or **B83** (**C83**) must be set.
- The fixed cycle subprograms should be edited if the rotary tool stop (M05) command is required before the rotary tool reverse rotation (M04) or forward rotation (M03) signal is output.

Note: Tapping cycle in the turning mode is not available on the side of secondary spindle.

13-3-4 Face/Longitudinal boring cycle: G85/G89

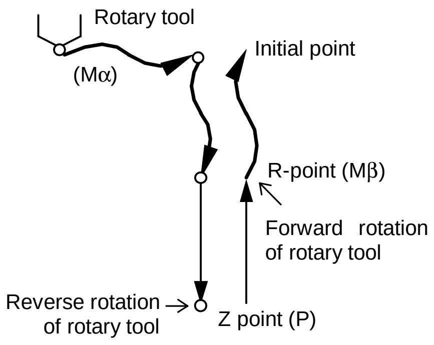
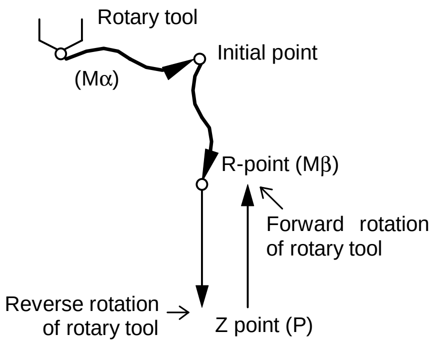
G85 (G89) X(Z)_ C_ Z(X)_ R_ P_ F_ L_ M_ ;

Parameter **P10** bit 7 : 0	Parameter **P10** bit 7 : 1
 (Mα) Initial point R-point (Mβ) f 2f Z point (P)	 (Mα) Initial point R-point (Mβ) f 2f Z point (P) TEP160

- See 1 in Section 13-3-1 for details on (Mα), (Mβ) and (P).
- 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.

13-3-5 Face/Longitudinal synchronous tapping cycle: G84.2/G88.2

G84.2 (G88.2) X(Z)_ C_ Z(X)_ R_ P_ F_ L_ M_ ;

Parameter **P10** bit 7 : 0	Parameter **P10** bit 7 : 1
 Rotary tool (Mα) Initial point R-point (Mβ) Forward rotation of rotary tool Reverse rotation of rotary tool Z point (P)	 Rotary tool (Mα) Initial point R-point (Mβ) Forward rotation of rotary tool Reverse rotation of rotary tool Z point (P) TEP159

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 forward/reverse rotation with M-codes (M03, M04, M203, M204).

Output to the machine side is as follows:

Program command	Z point	R point
M03	M04	M03
M04	M03	M04
M203	M204	M203
M204	M203	M204

- If none of the above-mentioned M-codes is specified in the program, one of the following four types of tapping cycle will automatically be selected according to the settings of the related parameters (bit 5 of **P12**, **B82** and **B83**):

		B82 = 3, **B83** = 4	**B82** = 203, **B83** = 204
P12 bit 5	0	Normal tapping cycle in turning mode	Normal tapping cycle in milling mode
	1	Reverse tapping cycle in turning mode	Reverse tapping cycle in milling mode

- Set M-code for spindle forward rotation to parameter **B82**. When the value of **B82** is 0, it results in M03.
- Set M-code for spindle reverse rotation to parameter **B83**. When the value of **B83** is 0, it results in M04.
- Use the parameters **C82** (forward) and **C83** (reverse) for the side of HD2.
- 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.

13-3-6 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.

13-3-7 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

$$-\ \underbrace{\mathrm{Gm}}_{\text{executed}}\ \underbrace{\mathrm{Gn}}_{\text{ignored}}\ \underbrace{\mathrm{X(Z)\_\ C\_\ Z(X)\_}}_{\text{executed}}\ \underbrace{\mathrm{R\_\ Q\_\ P\_\ L(K)\_}}_{\text{ignored}}\ \underbrace{\mathrm{F\_;}}_{\text{q}}$$

$$-\ \underbrace{\mathrm{Gn}}_{\text{executed}}\ \underbrace{\mathrm{Gm\ X(Z)\_\ C\_\ Z(X)\_}}_{\text{executed}}\ \underbrace{\mathrm{R\_\ Q\_\ P\_\ L(K)\_}}_{\text{ignored}}\ \underbrace{\mathrm{F\_;}}_{\text{memorized}}$$

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.

10. T32 compatible mode and standard mode of hole machining fixed cycle are the same except the address for repeat times.

Return to Main Menu

13-3-8 Sample programs with fixed cycles for hole machining

1. Face deep hole drilling cycle (G83)

```
G00 G97 G98;
G28 UW;
T0101;
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;
T0101;
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;
T0101;
M200;
M203 S600;
X100.Z2.C0;
G85 X50.H30.Z-20.R5.P.2 F150 L3 M210;
G80;
G28 UW;
M30;
```

4. Longitudinal deep hole drilling cycle (G87)

```
G00 G97 G98;

G28 UW;
T0101;
M200;
M203 S800;
X102. Z-50. C0;
G87 Z-50.H30.X70.R5.Q5000 P.2 F200 L3 M210;
G80;
G28 UW;
M30;
```

5. Longitudinal tapping cycle (G88)

```
G00 G97 G98;
G28 UW;
T0101;
M200;
M203 S600;
X102.Z-50.C0;
G88 Z-50.H30.X70.R5.P.2 F300 L3 M203 M210;
G80;
G28 UW;
M30;
```

6. Longitudinal boring cycle (G89)

```
G00 G97 G98;
G28 UW;
T0101;
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;
T0101;
M200;
M203 S600;
X100.Z2.C0;
G84.2 X50.H30.Z-20.R5.F2.00 L3 M203 M210;
G80;
G28 UW;
M30;
```

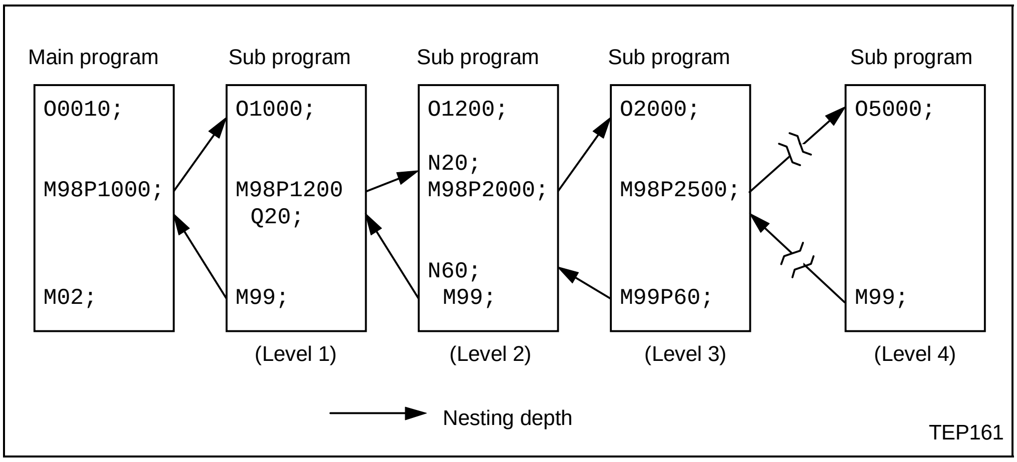
8. Longitudinal synchronous tapping cycle (G88.2)

```
G00 G97 G98;
G28 UW;
T0101;
M200;
M203 S600
X102. Z-50. C0;
G88.2 Z-50.H30.X70.R5.F2.L3 M203 M210;
G80;
G28 UW;
M30;
```

13-4 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 sub-program. Furthermore, it is possible to call other subprograms from particular subprograms and the nesting depth can include as many as 15 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 15 levels.

Return to Main Menu

2. Programming format

Subprogram call

M98 P_ Q_ L_;

Number of subprogram repetitions (L1 if omitted)

Sequence number in subprogram to be called (head block if omitted)

Program number of subprogram to be called (own program if omitted). P can only be omitted during memory operation and MDI operation.

Return to main program from subprogram

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 15 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.)

Example:

```
      ;
O○○○○  ;
................. ;     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 15th nesting level.

- Fixed cycles
- Pattern cycles

4. Subprogram execution

M98: Subprogram call command
M99: Subprogram return command

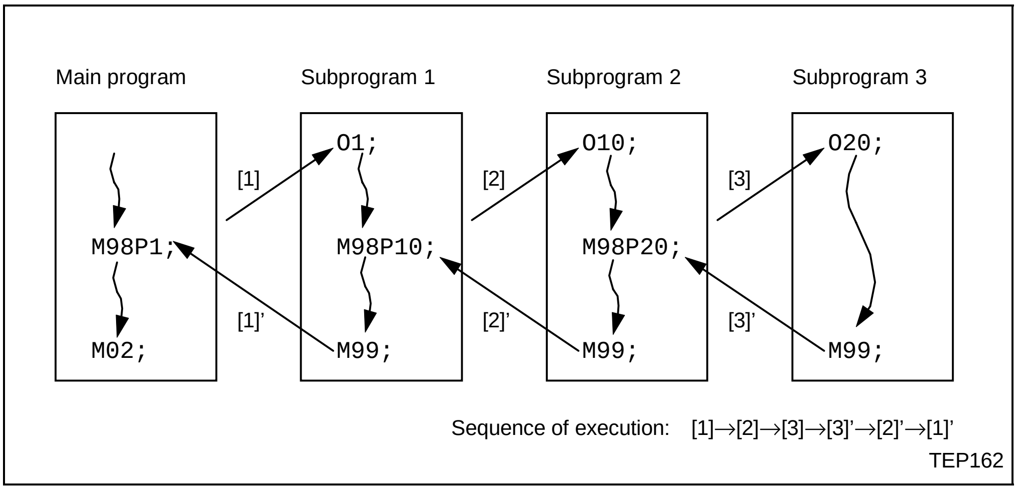
Programming format

M98 P_ Q_ L_;

Where P : Subprogram number to be called by the numerical value of maximum eight figures
Q : Any sequence number within the subprogram to be called by the numerical value of maximum five figures
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;

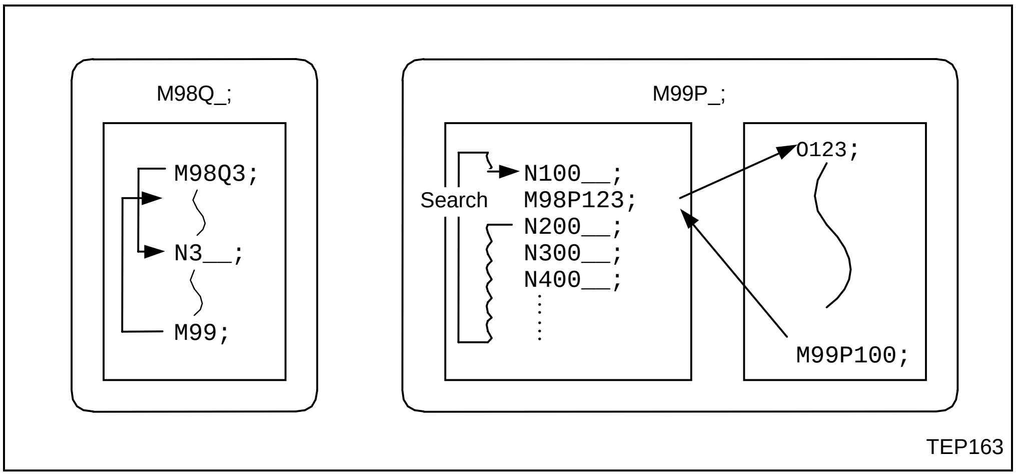
Example 1: When there are 3 subprogram calls (known as 3 nesting levels)



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.



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 l_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_ ;

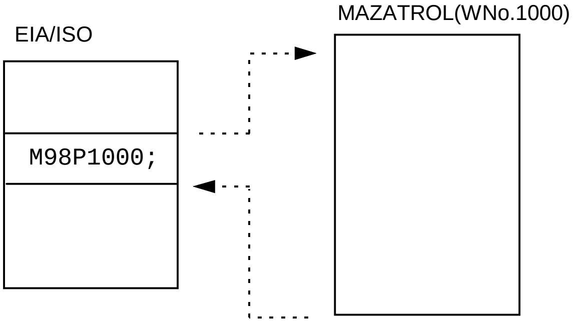
Return to Main Menu

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: MAZATROL machining program is executed as is the case with normal start, and not affected by EIA/ISO program at all. 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.

B. Programming format

M98 P□□□□ L ○○○○; or M98 P□□□□ ○○○○;

P□□□□: 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

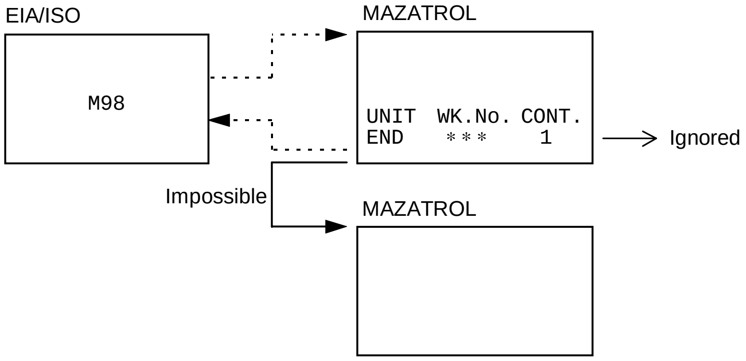
1. END unit of the MAZATROL program

End unit does not have to be specified at the end of MAZATROL machining program.
When end unit is specified:

UNo.	UNIT	COUNTER	RETURN	WK.No.	CONT.	NUM.	SHIFT
□	END			□	□		
(a)		(b)	(c)	(d)	(e)	(f)	(g)

(a) Unit No.
(b) Parts count
 0 : Yes 1 : No
(c) 0: No return to zero point
 1: Return to zero point
 2: Return to fixed point
(d) Work No. or program No. to make program chain
(e) Continuous operation performed or not
(f) Number of repetitions
(g) Z offset shift amount

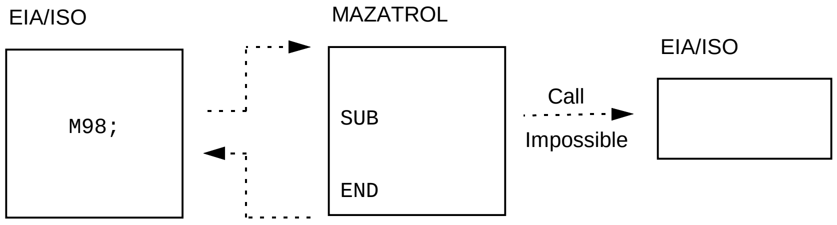
Even if WK.No. and CONT. are specified, they are ignored. This means that program chain cannot be made with MAZATROL program called from EIA/ISO program.



Also, NUM. 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) cannot be used.



Refer to the MAZATROL Programming Manual for SUB unit.

Note: As is the case with a SUB 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 error (alarm 752 “CANNED CYCLE PROGRAM ERROR”).
4. MAZATROL program call in the mode of nose radius compensation results in an error (alarm 726 “CANNED CYCLE IN NOSE-R COMP.”).
5. MAZATROL program call is not available in the MDI operation mode (alarm 601 “NO DESIGNATED PROGRAM”).
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 Main Menu

13-5 End Processing: M02, M30, M198, M199

If the program contains M02, M30, M198, M199 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. M198, M199
   Tool life processing, parts count, and work No. search will be executed.

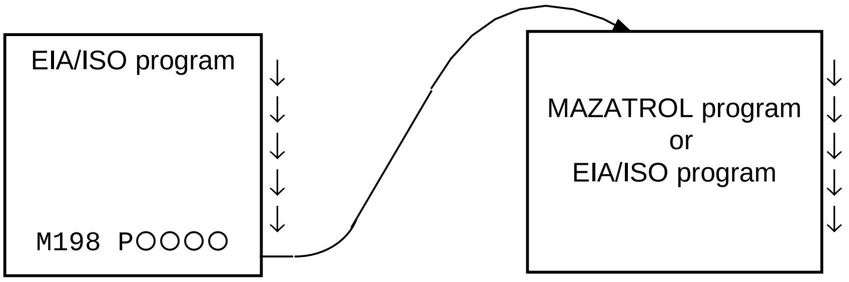
M198(199) P111 Q1;

Q1: Specification of execution or nonexecution of parts count
(counting updated on POSITION display)
0: Parts count nonexecution
1: Parts count execution

P111: Specification of next work No.

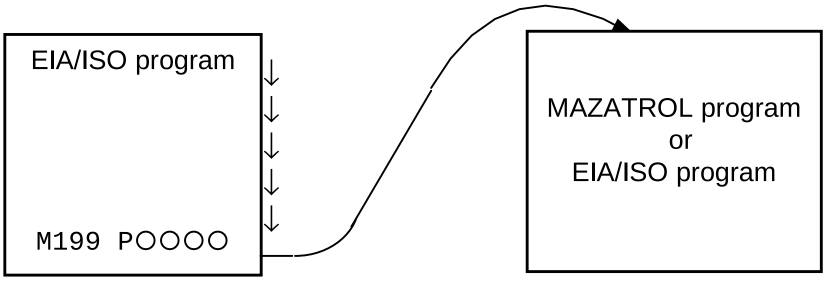
M198(199): M-code for program chain
M198: Continuous execution after parts count and work NO. search
M199: Ending after parts count and work No. search

- M198P○○○○



MAZATROL or EIA/ISO program is called from EIA/ISO program and executed as the next program.

- M199P○○○○



MAZATROL or EIA/ISO program is only called from EIA/ISO program and the operation is terminated.

13-6 Opposite Turret Mirror Image: G68, G69

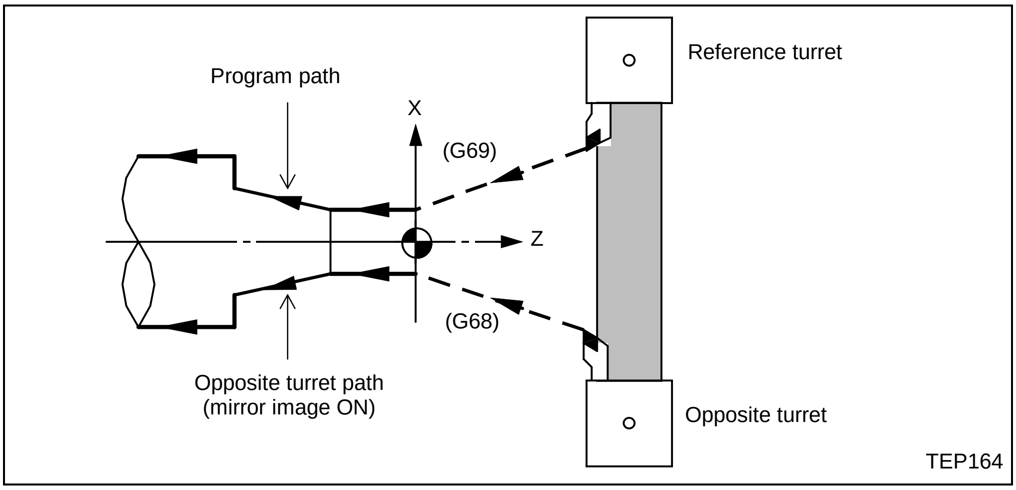
1. Overview

With machines in which the reference turret and opposite turret are integrated, this function enables the workpiece to be cut by the tools on the opposite turret using the programs prepared for the reference turret side (See figure below). The distance between the two turrets is set by G50U_ ; before G68 command. However, if the distance is set with parameter **B229** beforehand, setting by G50U; is not required.

2. Programming format

G68: Opposite turret mirror image ON
G69: Opposite turret mirror image OFF

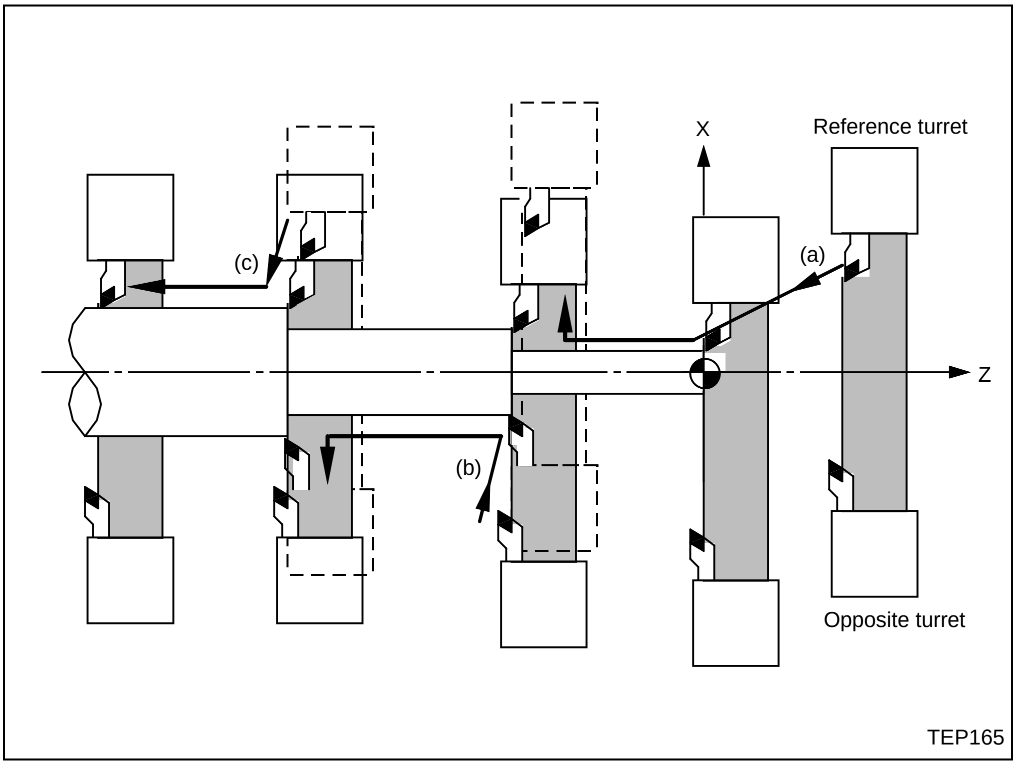
When the G68 command is set, the subsequent programmed coordinate systems are shifted to the opposite turret side and the movement direction of the X-axis is made the opposite of that commanded by the program.
When the G69 command is set, the subsequent programmed coordinate systems are returned to the reference turret side.



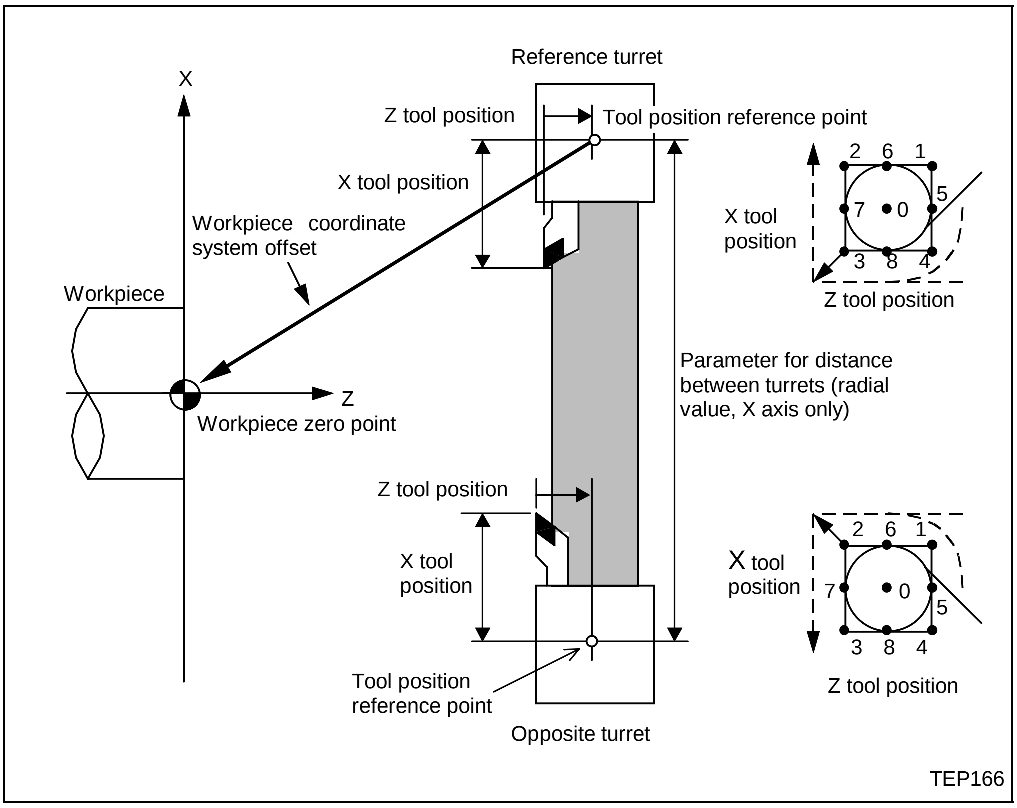
Return to Main Menu

3. Program and example of operation

```
T0101;            Selection on reference turret
G00 X10.Z0;                                             Machining by reference turret···(a)
G01 Z-40.F400;
X20.;
G68;              Opposite turret mirror image ON
T0202;            Selection on opposite turret
G00 X20.Z–40;                                           Machining by opposite turret···(b)
G01 Z-80.F200;
X30;
G69;              Opposite turret mirror image cancel
T0101;            Selection on reference turret         Machining by reference turret···(c)
G00 X30.Z-80.;
G01 Z-120.F400;
```

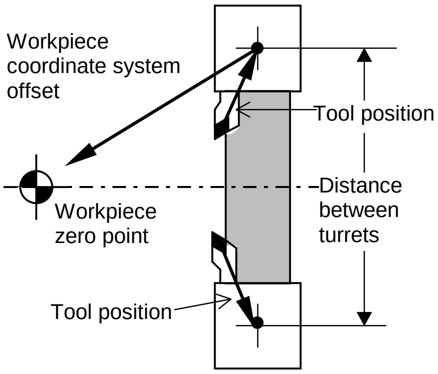


4. Opposite turret offset amounts



A. Tool position offset amount

The tool position is the distance from the tool nose to the tool position reference point. This definition is the same for the opposite turret.

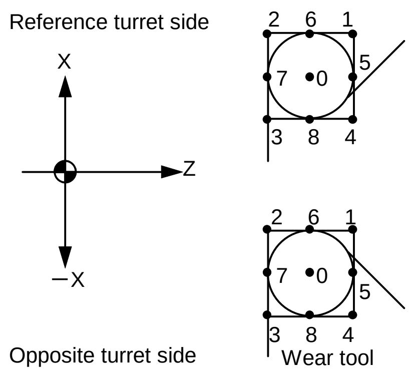
Workpiece zero point	Workpiece face center
Tool position reference point	Reference points of each turret
Distance between turrets	Distance between reference points of each turret (radial value)
Workpiece coordinate system offset	Workpiece zero point – tool position reference point of reference turret
Tool position	Tool position reference point – tool nose position
Outline diagrams	 Workpiece coordinate system offset Tool position Distance between turrets Workpiece zero point Tool position TEP167

Return to Main Menu

B. Tool nose point for tool nose radius compensation

The tool nose point for tool nose radius compensation is as follows. It is the same for both the reference and opposite turrets.

Tool nose radius tool nose point



TEP168

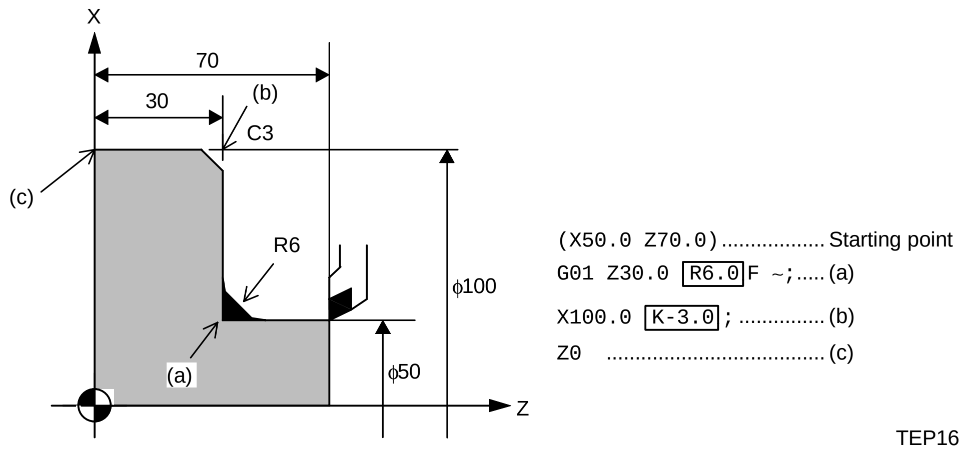
C. Distance between turrets

The distance between the turrets is the distance from the tool position reference point of the opposite turret to the tool position reference point of the reference turret. It is set by G50U ; only for the X-axis. “0” is set when the tool position reference point is common.

13-7 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.



```
(X50.0 Z70.0).................. Starting point
G01 Z30.0 R6.0 F ~;..... (a)
X100.0 K-3.0; ............... (b)
Z0 ...................................... (c)
```

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.

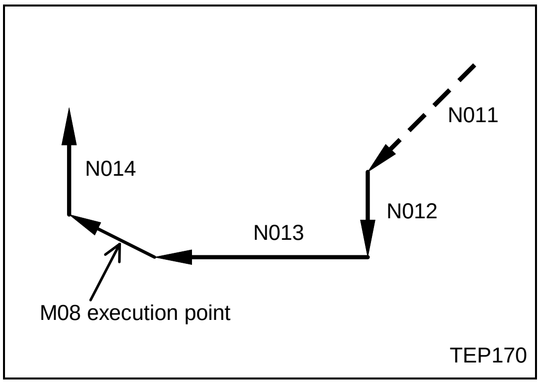
   - I, J, K or R is commanded while two axes of X and Z are commanded in G01.
   - Two of I, J, K or R are commanded in G01.
   - X and I, Y and J or Z and K are commanded at the same time in G01.
   - In a block commanding chamfering or corner rounding, movement distance in X- or Z-axis is smaller than 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.

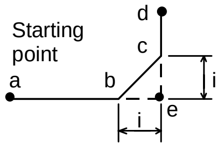
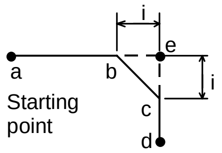
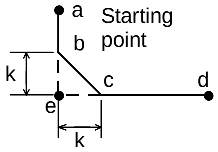
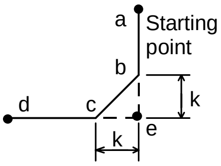
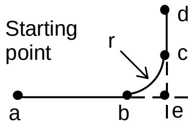
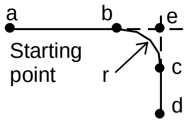
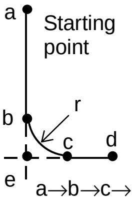
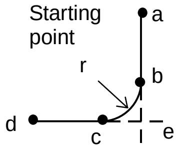
5. Execution by single step mode will require two steps to complete the operation.

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.;
```



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

13-8 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.

13-8-1 Chamfering at arbitrary angle corner: , C_

1. **Function**

The arbitrary corner is chamferred 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**

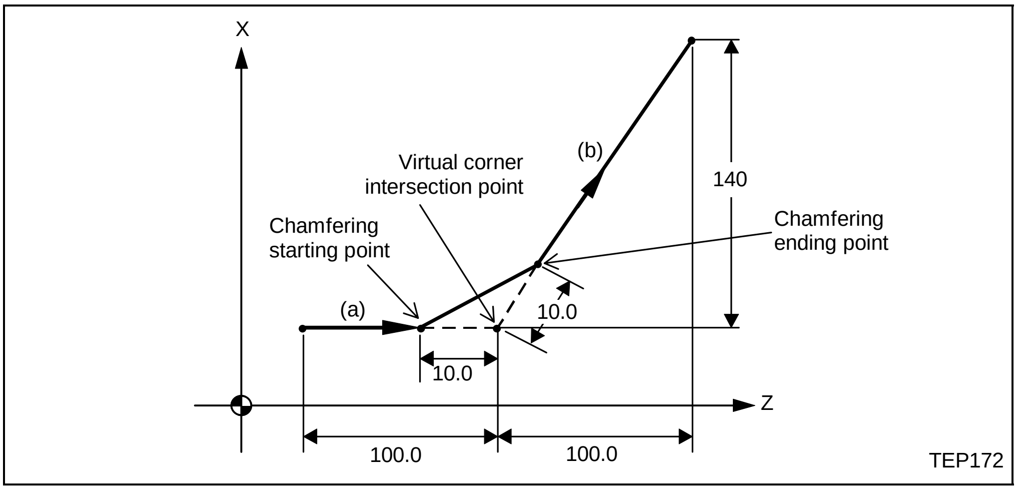
```
N100  G01 X_ Z_ ,C_;
N200  G01 X_ Z_;
```

} 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.

13-8-2 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.

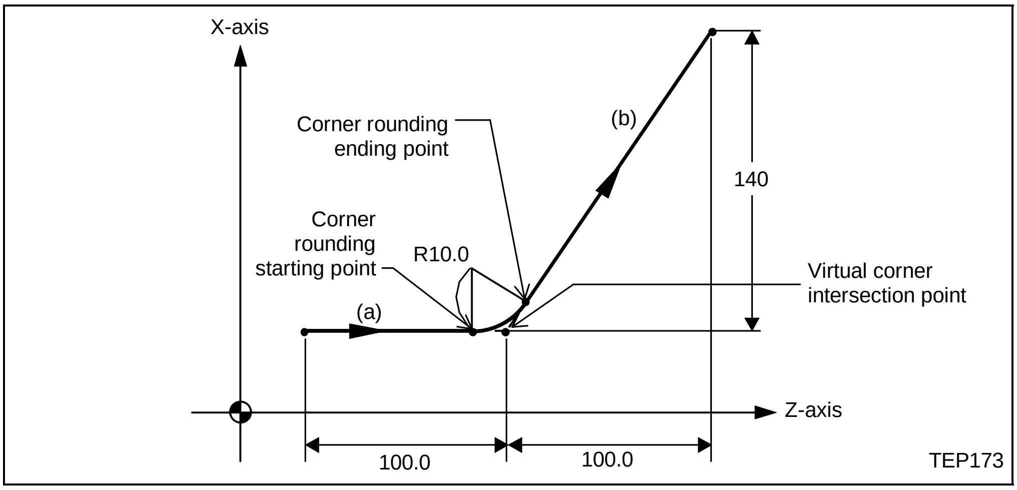
2. Programming format

```
N100  G01 X_ Z_ ,R_;  } Rounding is performed at the point where N100 and N200
N200  G01 X_ Z_;      }   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.

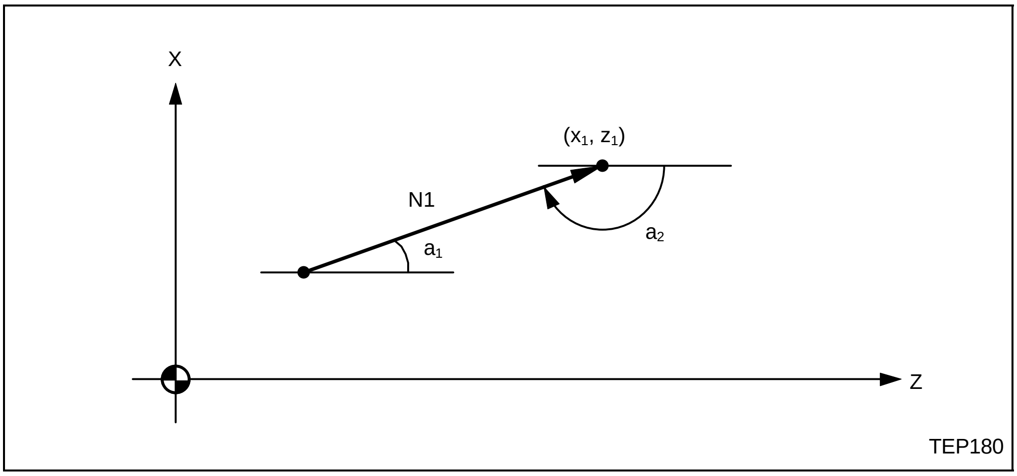
13-9 Linear Angle Command

1. Function

The ending point coordinates are calculated automatically by commanding the linear angle and one of the ending point coordinate axes.

2. Programming format

N1 G01 Aa_1 Zz_1 (Xx_1) ;   Designate the angle and the X- or Z-axis.
N2 G01 A–a_2 Zz_1 (Xx_1) ;



3. Detailed description

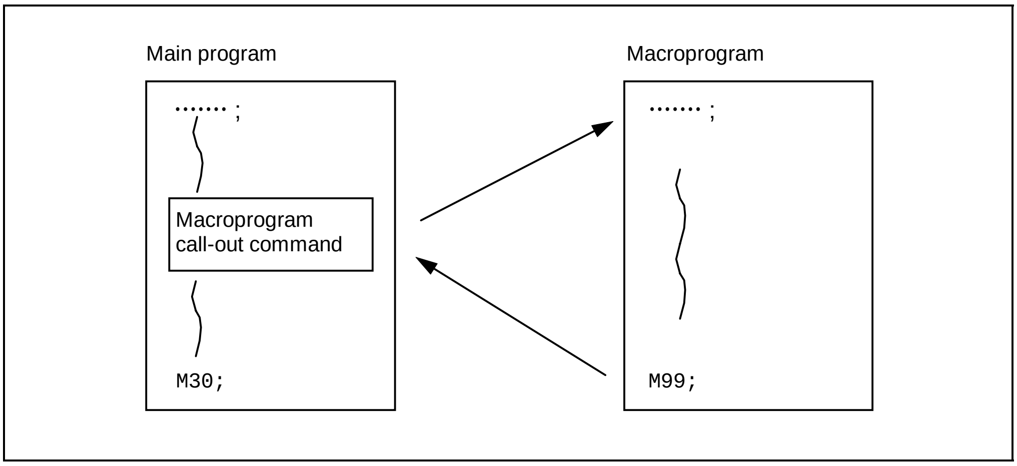
1. The angle is from the “+” direction of the horizontal axis on the selected plane. The counterclockwise (CCW) direction is considered to be “+” and the clockwise direction (CW) to be “–”.
2. Either of the axes on the selected plane is commanded for the ending point.
3. The angle is ignored when the angle and the coordinates of both axes are commanded.
4. When only the angle has been commanded, this is treated as a geometric command.
5. The angle of either the starting point (a_1) or ending point (a_2) may be used.
6. The function cannot be used when address A is used for the axis name or as the second miscellaneous function.
7. This function is valid only for the G01 command; it is not valid for other interpolation or positioning commands.

14 MACRO CALL FUNCTION

14-1 User Macroprogram

1. Function and purpose

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.

2. Detailed description

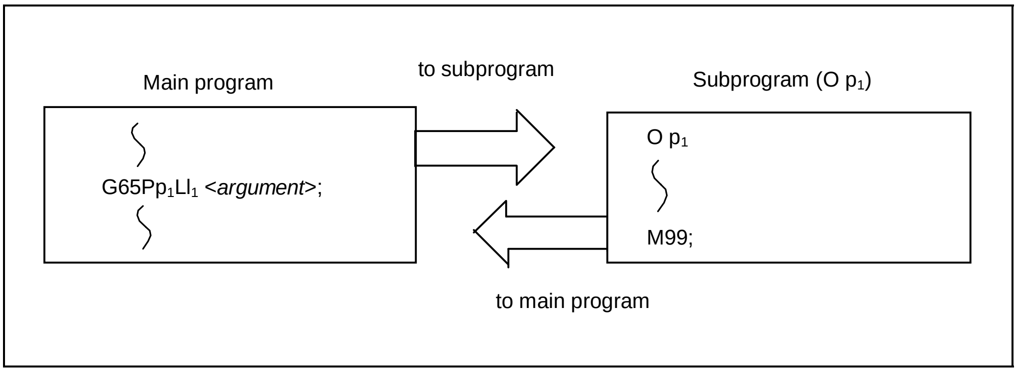
- When command G66 is entered, the designated user macro subprogram will be called every time after execution of the move commands within a block until G67 (cancellation) is entered.
- Command codes G66 and G67 must reside in the same program in pairs.

Return to Main Menu

14-2 Macro Call Instructions: G65, G66 (Cancellation: G67)

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 the call modal area.

1. Single call



The designated user macro subprogram ends with M99.

```
Format:
G65 P__ L__ <argument> ;
       |    |
       |    └──────────── Repeat times
       |
       └───────────────── Program No.
```

<Argument>

When <argument> is to be delivered to the user macro subprogram as a local variable, the address must be followed by actual data.

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 Main Menu

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 do 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
Argument specification Ⅰ addresses	Variables in macroprograms	G65, G66
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

Return to Main Menu

B. Argument specification II

Format: A_B_C_I_J_K_I_J_K_ ···

[Detailed description]

- 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:

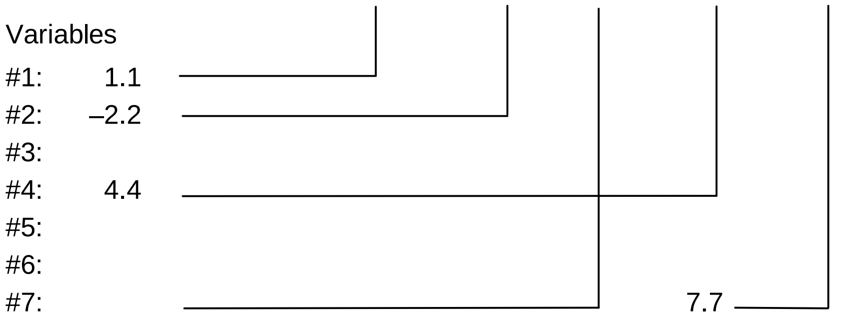
Argument specification II addresses	Variables in macro-programs	Argument specification II addresses	Variables in macro-programs
A	#1	K5	#18
B	#2	I6	#19
C	#3	J6	#20
I1	#4	K6	#21
J1	#5	I7	#22
K1	#6	J7	#23
I2	#7	K7	#24
J2	#8	I8	#25
K2	#9	J8	#26
I3	#10	K8	#27
J3	#11	I9	#28
K3	#12	J9	#29
I4	#13	K9	#30
J4	#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 K 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:

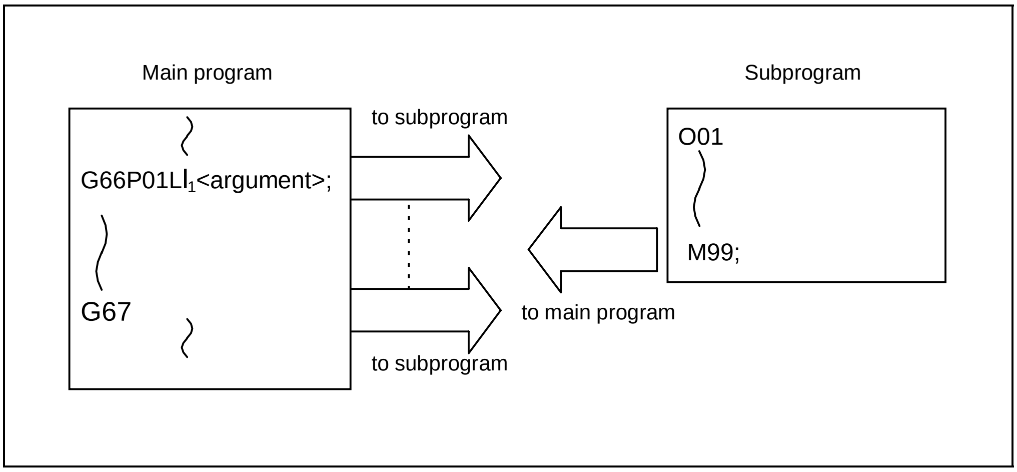
#6:

#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 Main Menu

2. Modal 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 I_1 number of times for each call.
The methods of specifying <argument> are the same as used for single call.

Format:
G66 P__ L__ *<argument>* ;

Repeat times

Program number

[Detailed description]

- 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.

3. G-code macro call

The user macro subprograms of the preset 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 that of the instruction below.

  G65P○○○○ *<argument>* ;

- Use parameters to set the relationship between G××(macro call G-code) and P○○○○ (program number of the macro to be called). (Parameter **K81** to **K88**, **P41**, **P42**, **P49**, **P50**)
- Of G00 through G255, up to a maximum of 10 command codes can be used with this instruction. (However, G65, G66 and G67 cannot be used.)

- The command code cannot be included in user macro subprograms that have been called using G-codes.
- The command code cannot be included also in user macro subprograms that have been called using M-codes or T-codes.

4. M-code macro call

The user macro subprograms of the preset program number can be called just by setting M-code. (Only the registered M-codes can be used.)

```
Format:
Mm;
 |
 └────────── M-code which calls macro-subprogram
```

[Detailed description]

- The instruction shown above performs the same function as that of the instruction below.

  G65P○○○○Mm;   (Mm is not outputted.)

- Use parameter to set the relationship between Mm (macro call M-code) and P○○○○ (program number of the macro to be called).
  Up to a maximum of 10 M-codes, ranging from M06 to M255, can be registered. (Parameter **K89** to **K93**, **P43** to **P47** and **P51** to **P55**)
  Do not register the M-codes that are fundamentally required for your machine, nor M00, M01, M02, M30, and M96 through M99.
- M-codes and MF are not outputted.
- If registered auxiliary command codes are set in the user macro subprograms that have been called using M-, S-, T- or B-codes, macro calls will not occur since those special auxiliary command codes will be handled as usual ones (M-, S-, T-, or B-codes).

5. M-code subprogram call

The user macro subprograms of the preset program number can be called just by setting M-codes. (Only the registered M-codes can be used.)

```
Format:
Mm;
 |
 └────────── M-code which calls macro-subprogram
```

[Detailed description]

- The instruction shown above performs the same function as that of the instruction below.

  M98P○○○○;   (M98 is not outputted.)

- Use parameter to set the relationship between Mm (macro call M-code) and P○○○○ (program number of the macro to be called).
  Up to a maximum of 3 M-codes, ranging from M03 to M255, can be registered. (Parameter **P57** to **P62**)
  Do not register the M-codes that are fundamentally required for your machine, nor M00, M01, M02, M30, and M96 through M99.
- M-codes and MF are not outputted as well as M98.

- If registered auxiliary command codes are set in the user macro subprograms that have been called using M-, S-, T- or B-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. S-, T-, B-code subprogram call

The user macro subprograms of the preset program number can be called just by setting S-, T-, or B-codes. (All the S-, T- and B-codes can be used.)

Format:
Ss; (or Tt;, Bb;)

S-code which calls macro-subprogram

[Detailed description]

- The instruction shown above performs the same function as that of the instruction below.

  M98P○○○○;

- Use parameter to set P○○○○ (program number of the macro to be called). (Parameter **K94**, **K95** and **K96**)
- T- and B-codes and SF, TF and BF are not outputted.
- If registered auxiliary command codes are set in the user macro subprograms that have been called using M-, S-, T- or B-codes, macro calls will not occur since those special auxiliary command codes will be handled as usual ones (M-, S-, T-, or B-codes).
- Any of S-, T- and B-codes can be used for this function.

7. Differences in usage between commands M98 and G65

- 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 and G66.
- 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 15 levels of call multiplexity when combined with G65 or G66 whereas the maximum available number of levels for command G65 is four when it is combined with G66.

8. Multiplexity of macro call commands

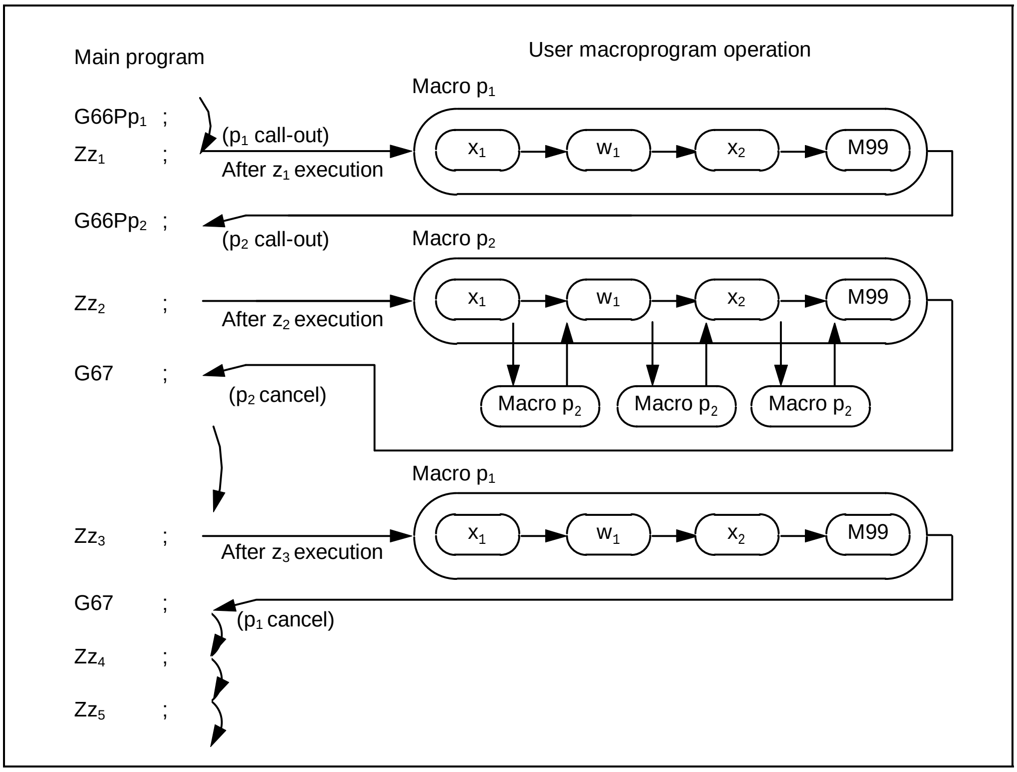
The maximum available number of levels of macro subprogram call is four, whether it is single or modal. Arguments in macro call instructions become valid only within the level of the called macro. Since the multiplexity of macro call is of up to a maximum of four levels, arguments can be included in a program as local variables each time a macro call is made.

Note 1: When a G65 or G66 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, 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 Main Menu

14-3 Variable

Variables used in user macro require both variable specification and user macro specification.
Of all types of variables available for the NC unit, only local variables, common variables, and system variables are retained even after power-off. (Local variables and common variables can be cleared by resetting according to the setting of the parameter **P16** bit 6 and bit 7.)

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;
  #5 = # [# [#1]];
        From #1=10, #[#[#1]] = #[#10] will result.
        From #10=20, #[#10] = #20 will result.
        Therefore #5 = #20, i.e. #5 = 30 will result.
```

```
- #1 = 10 #10 = 20 #20 = 30 #5 = 1000;
  # [# [#1]] = #5;
        From #1 = 10, #[#[#1]] = #[#10] will result.
        From #10 = 20, #[#10] = #20 will result.
        Therefore #20 = #5, i.e. #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;...........................#3 = –100 will result.
                                           (5/2 = 2.5 is rounded up to 3.)
```

2. Undefined variables

Under user macro specifications, variables remaining unused after power-on or local variables that are not argument-specified by G65 or G66 can be used as <empty>. Also, variables can be forcibly made into <empty>.
Variable #0 is always used as an <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
```

Return to Main Menu

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>
G0 X#1 Z1000; ········is equivalent to G0 Z1000;
G0 X[#1+10] Z1000; ···is equivalent to G0 X10 Z1000;

C. Conditional expression

Only for EQ and NE, does <empty> differ from 0 in meaning. (#0 means <empty>.)

If #101 = <empty>		If #101 = 0	
#101EQ#0	<empty> = <empty> holds.	#101EQ#0	0 = <empty> does not hold.
#101NE 0	<empty> ≠ 0 holds.	#101NE 0	0 ≠ 0 does not hold.
#101GE#0	<empty> ≥ <empty> holds.	#101GE#0	0 ≥ <empty> holds.
#101GT 0	<empty> > 0 does not hold.	#101GT 0	0 > 0 does not hold.
#101LE#0	<empty> ≤ <empty> holds.	#101LE#0	0 ≤ <empty> does not hold.
#101LT 0	<empty> < 0 holds.	#101LT 0	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.

Note: Use only integers for the comparison of EQ and NE. Use GE, GT, LE and LT for the comparison of values with decimal fractions.

Return to Main Menu

14-4 Types of Variables

1. Common variables (Types A and B)

Common variables refer to the variables to be used in common at any position, and two types are provided; type A (#100 to #149, #500 to #531) and type B (#100 to #199, #500 to #599). For two-line system, common variables are prepared for each turret.

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.

G65Pp_1Ll_1 <*argument*>;

where p_1 : Program number

l_1 : Number of repeat times

<Argument> must be: Aa_1 Bb_1 Cc_1 .... Zz_1

The following represents the relationship between the address specified by <argument> and the local variables number used in the user macro unit:

Argument specification I

Call commands	Argument address	Local variable	Call commands	Argument address	Local variable
G65, G66			G65, G66		
○	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			

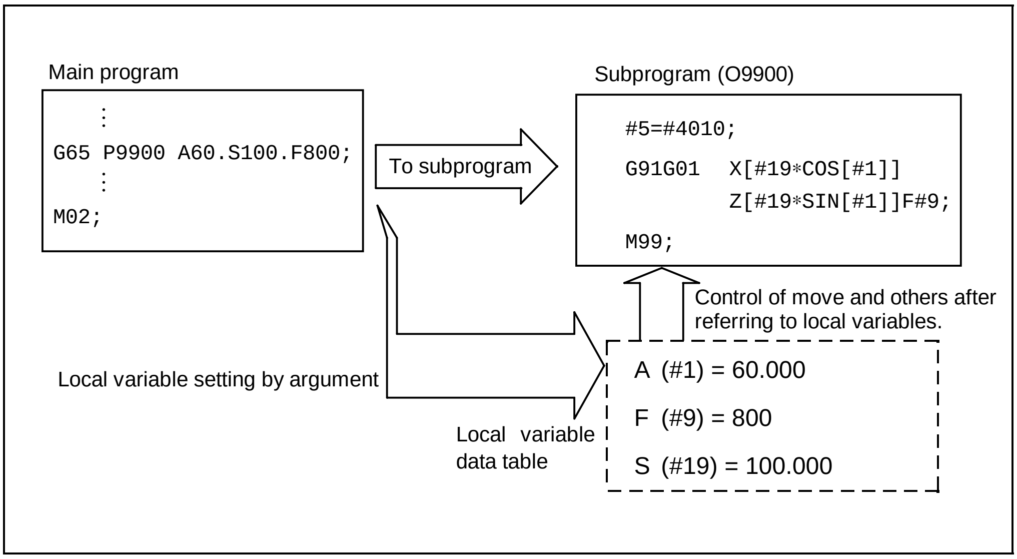
Argument addresses marked as × in the table above cannot be used. Also, the dash sign (–) indicates that no address is crosskeyed to the local variables number.

Argument specification II

Argument address	Local variable	Argument address	Local variable
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 K just to denote the order of arrangement of the designated sets of arguments: these numerals are not included in actual instructions.

1. Local variables for a subprogram can be defined by specifying <argument> when calling a macro.



Main program

```
    :
G65 P9900 A60.S100.F800;
    :
M02;
```

Subprogram (O9900)

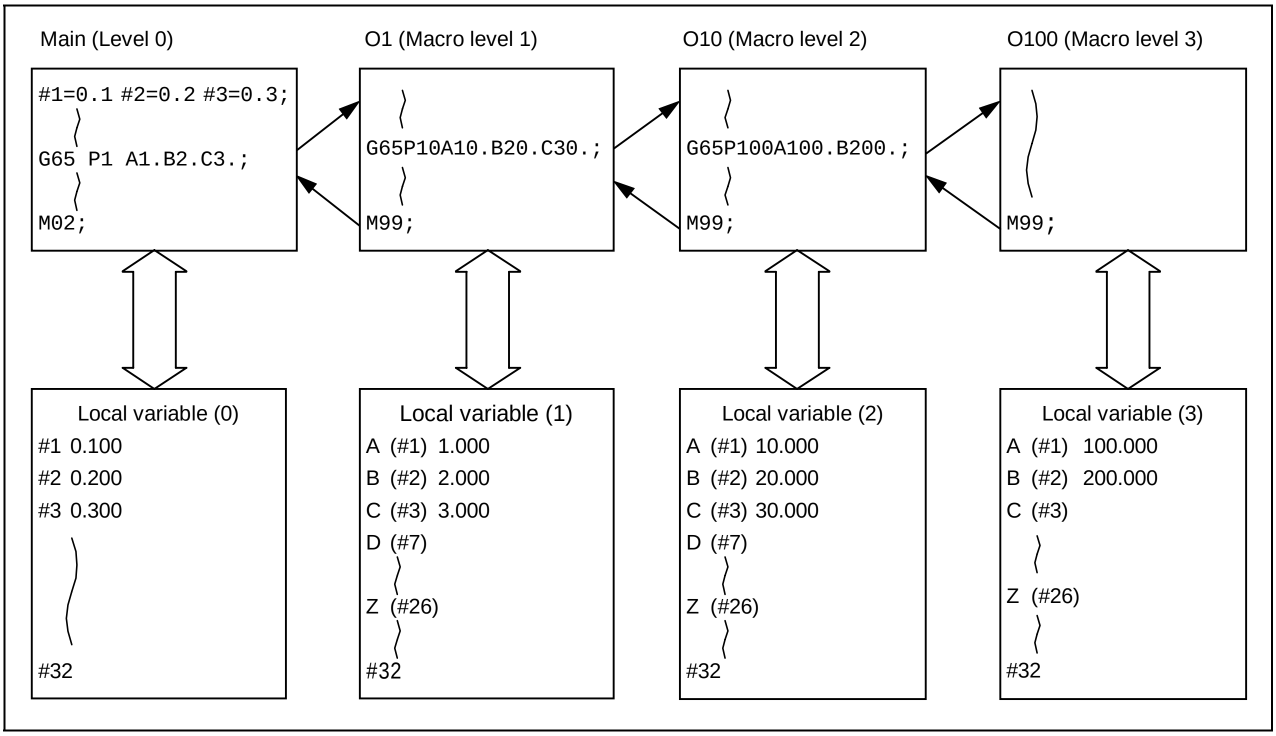
```
#5=#4010;
G91G01   X[#19*COS[#1]]
         Z[#19*SIN[#1]]F#9;
M99;
```

Local variable data table

```
A  (#1) = 60.000
F  (#9) = 800
S  (#19) = 100.000
```

To subprogram
Control of move and others after referring to local variables.
Local variable setting by argument
Local variable data table

2. 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.



Main (Level 0)

```
#1=0.1 #2=0.2 #3=0.3;
G65 P1 A1.B2.C3.;
M02;
```

O1 (Macro level 1)

```
G65P10A10.B20.C30.;
M99;
```

O10 (Macro level 2)

```
G65P100A100.B200.;
M99;
```

O100 (Macro level 3)

```
M99;
```

Local variable (0)

```
#1  0.100
#2  0.200
#3  0.300
#32
```

Local variable (1)

```
A  (#1)  1.000
B  (#2)  2.000
C  (#3)  3.000
D  (#7)
Z  (#26)
#32
```

Local variable (2)

```
A  (#1) 10.000
B  (#2) 20.000
C  (#3) 30.000
D  (#7)
Z  (#26)
#32
```

Local variable (3)

```
A  (#1)   100.000
B  (#2)   200.000
C  (#3)
Z  (#26)
#32
```

How the local variables are currently being used is displayed on the screen. A local variable not designated with an argument is <empty> in the initial state.

3. Macro interface input system variables (#1000 to #1015, #1032)

You can check the status of an interface input signal by reading the value of the appropriate variables number (#1000 to #1015, #1032).
The read value of the variables number is either 1 (contacts closed) or 0 (contacts open). You can also check the status of all input signals of the variables from #1000 to #1015 by reading the value of variables number 1032. Variables from #1000 to #1015 and #1032 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	
		HD1	HD2			HD1	HD2
#1000	1	Y580	Y680	#1009	1	Y589	Y689
#1001	1	Y581	Y681	#1010	1	Y58A	Y68A
#1002	1	Y582	Y682	#1011	1	Y58B	Y68B
#1003	1	Y583	Y683	#1012	1	Y58C	Y68C
#1004	1	Y584	Y684	#1013	1	Y58D	Y68D
#1005	1	Y585	Y685	#1014	1	Y58E	Y68E
#1006	1	Y586	Y686	#1015	1	Y58F	Y68F
#1007	1	Y587	Y687	#1032	16	Y580 to Y58F	Y680 to Y68F
#1008	1	Y588	Y688				

4. Macro interface output system variables (#1100 to #1115, #1132, #1133)

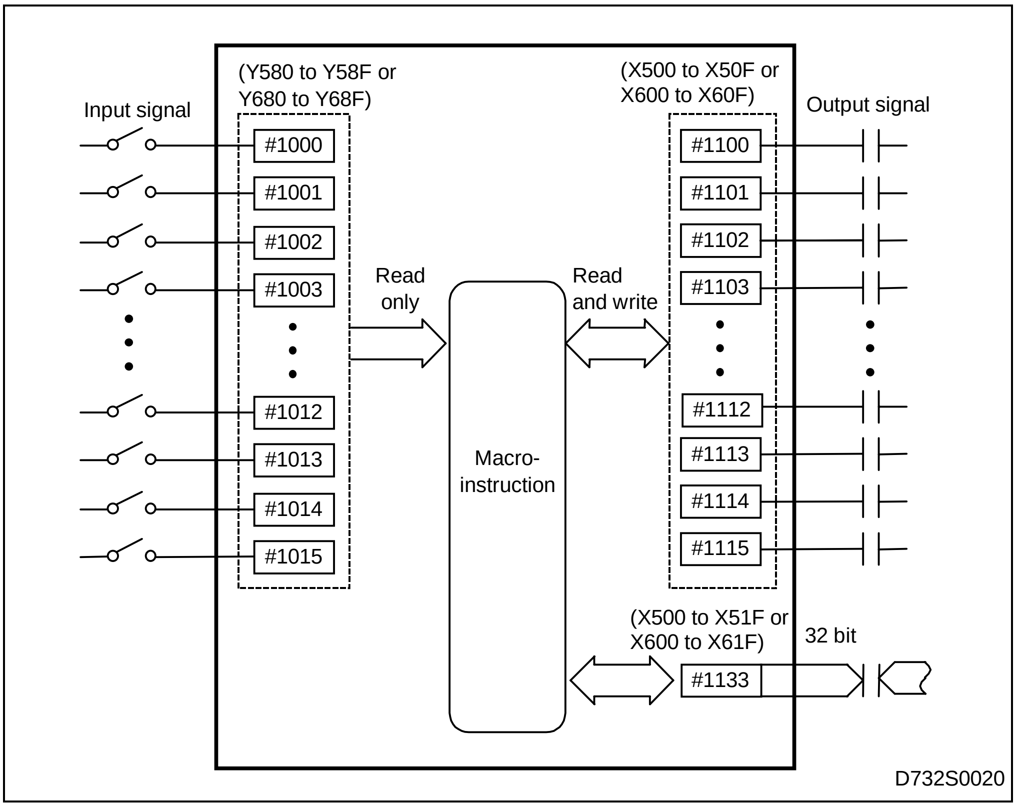
You can send an interface output signal by assigning a value to the appropriate variables number (#1100 to #1115, #1132, #1133).
All output signals can take either 0 or 1.
You can also send all output signals of the variables from #1100 to #1115 at the same time by assigning a value to variables number 1132 (2^0 to 2^{15}). In addition to the data writing for offsetting the #1100 to #1115, #1132 and #1133 output signals, the reading of the output signal status can be done.

System variable	Points	Interface output signal		System variable	Points	Interface output signal	
		HD1	HD2			HD1	HD2
#1100	1	X500	X600	#1109	1	X509	X609
#1101	1	X501	X601	#1110	1	X50A	X60A
#1102	1	X502	X602	#1111	1	X50B	X60B
#1103	1	X503	X603	#1112	1	X50C	X60C
#1104	1	X504	X604	#1113	1	X50D	X60D
#1105	1	X505	X605	#1114	1	X50E	X60E
#1106	1	X506	X606	#1115	1	X50F	X60F
#1107	1	X507	X607	#1132	16	X500 to X50F	X600 to X60F
#1108	1	X508	X608	#1133	32	X500 to X51F	X600 to X61F

Note 1: Data of the system variables from #1100 to #1115, #1132 and #1133 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 #1115:

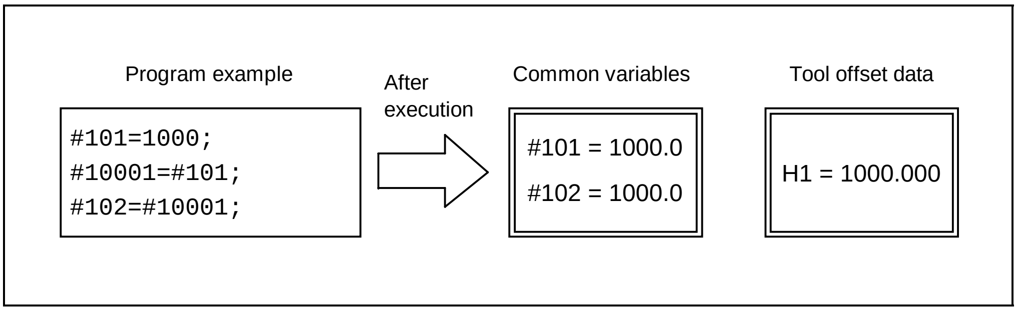
- <empty> is regarded as equal to 0.
- Data other than 0 and <empty> is regarded as equal to 1.
- Data less than 0.00000001, however, is regarded as undefined.



5. Tool offset

Range of variable Nos.		Description
#10001 to #10000+n	#2701 to #2700+n	X shape offset amount
#11001 to #11000+n	#2001 to #2000+n	X wear offset amount
#12001 to #12000+n	#2401 to #2400+n	Y shape offset amount
#13001 to #13000+n	#2401 to #2400+n	Y wear offset amount
#14001 to #14000+n	#2801 to #2800+n	Z shape offset amount
#15001 to #15000+n	#2101 to #2100+n	Z wear offset amount
#16001 to #16000+n	#2901 to #2900+n	R shape offset amount
#17001 to #17000+n	#2201 to #2200+n	R wear offset amount
#18001 to #18000+n	#2301 to #2300+n	Virtual tool nose point

Using variable numbers, you can read tool data or assign data.
Usable variable numbers are of the order of either #10000 or #2000.
The last two or 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.

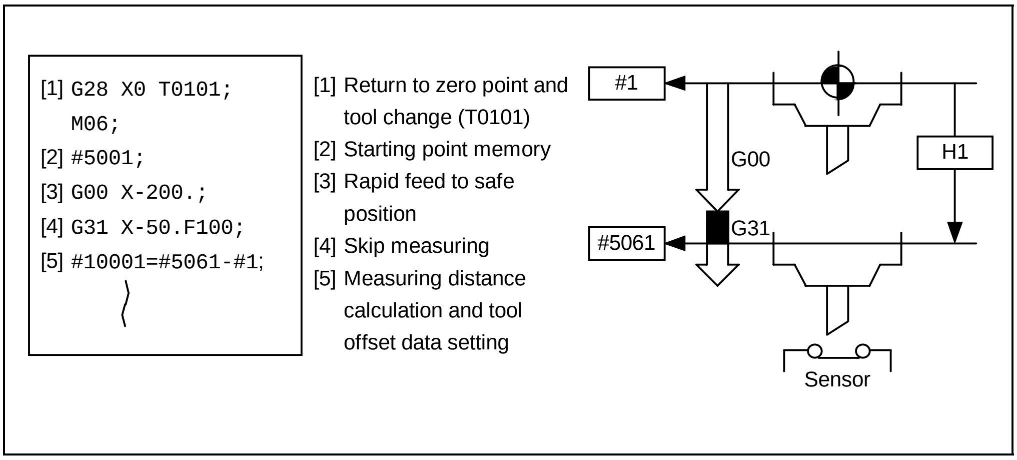


```
#101=1000;
#10001=#101;
#102=#10001;
```

#101 = 1000.0
#102 = 1000.0

H1 = 1000.000

Example: Tool offset data measuring



```
[1] G28 X0 T0101;
    M06;
[2] #5001;
[3] G00 X-200.;
[4] G31 X-50.F100;
[5] #10001=#5061-#1;
```

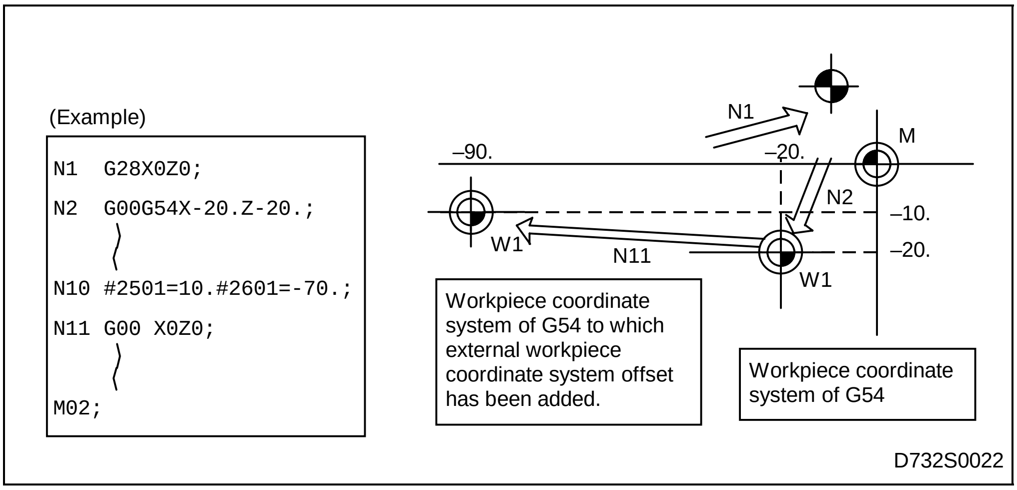
[1] Return to zero point and tool change (T0101)
[2] Starting point memory
[3] Rapid feed to safe position
[4] Skip measuring
[5] Measuring distance calculation and tool offset data setting

- The example shown above does not allow for any skip sensor signal delay.
- Also, #5001 denotes the position of the starting point of the X-axis, and #5061 denotes the skip coordinate of the X-axis, that is, the position at which a skip signal was input during execution of G31.

6. Workpiece coordinates system offset

Using variable numbers 2501 and 2601, you can read workpiece coordinate system offset data or assign data.

Axis name Coordinate name	X-axis	Z-axis
External workpiece coordinate system offset	#2501	#2601



(Example)

```
N1  G28X0Z0;
N2  G00G54X-20.Z-20.;

N10 #2501=10.#2601=-70.;
N11 G00 X0Z0;

M02;
```

N1
M
–90.
–20.
N2
–10.
W1
N11
–20.
W1
Workpiece coordinate system of G54 to which external workpiece coordinate system offset has been added.
Workpiece coordinate system of G54
D732S0022

Return to Main Menu

7. NC alarm (#3000)

The NC unit can be forced into an alarm status using variables number 3000.

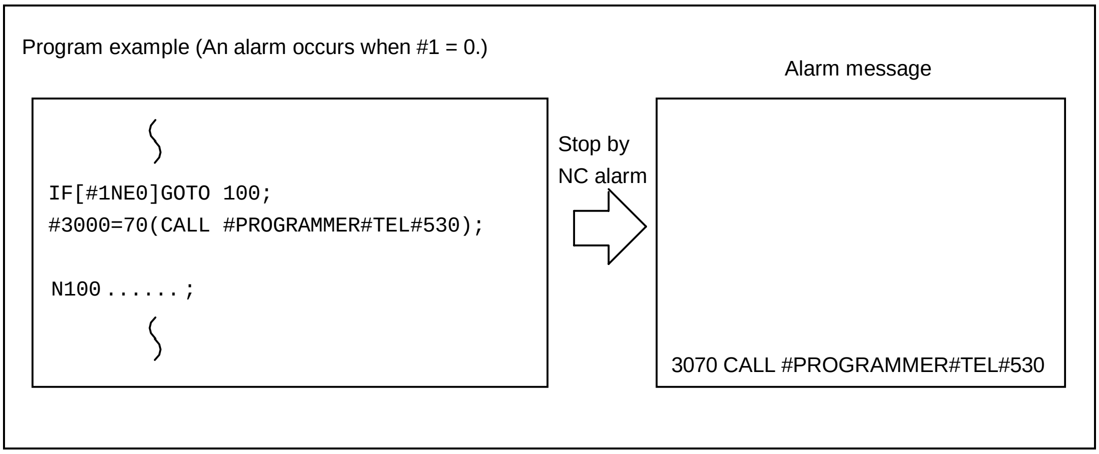
Format:

```
#3000 = 70 (CALL #PROGRAMMER#TEL#530) ;
```

Alarm No. — 70

Alarm message — CALL #PROGRAMMER#TEL#530

The alarm number must be between 0 and 200. The alarm message can be made with 30 characters or less.



Program example (An alarm occurs when #1 = 0.)

```
IF[#1NE0]GOTO 100;
#3000=70(CALL #PROGRAMMER#TEL#530);

N100......;
```

- As to the expression in the figure above, an alphabet on the right side and the subsequent characters can be regarded as an alarm message. That is, a numeral cannot be specified as the initial character of the alarm message. It is recommended to put an alarm message in parentheses.
- An alarm number displayed is a value adding 3000 to the programmed alarm number.

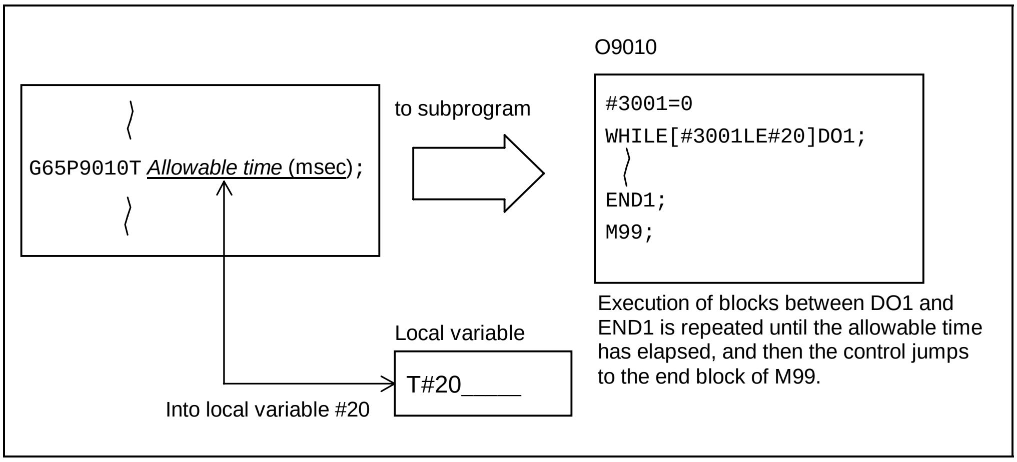
Return to Main Menu

8. Integrated time (#3001, #3002)

Using variable numbers #3001 and #3002, you can read the integrated time within automatic start operation or assign data.

Type	Variable No.	Unit	Data upon power-on	Initialization	Counting
Integrated time 1	3001	1 msec	Same as at power-off	Data is assigned in variables	Always during power-on
Integrated time 2	3002	1 hour			During automatic operation

- The integrated time 1 is cleared to 0 after having reached 65535 msec.
- The integrated time 2 is cleared to 0 after having reached 65535 hours.



```
G65P9010T Allowable time (msec);
```

```
#3001=0
WHILE[#3001LE#20]DO1;
END1;
M99;
```

Execution of blocks between DO1 and END1 is repeated until the allowable time has elapsed, and then the control jumps to the end block of M99.

9. Suppression of single-block stop or auxiliary-function finish signal wait

Assigning one of the values listed in the table below to variables number 3003 allows single-block stop to be suppressed 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-fucntion completion signal
0	No suppression	Wait
1	Suppression	Wait
2	No suppression	No wait
3	Suppression	No wait

- Variable #3003 is cleared to 0 by resetting.

10. Validation/invalidation of feed hold, feed rate override, or G09

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	Valid	Valid	Valid
1	Invalid	Valid	Valid
2	Valid	Invalid	Valid
3	Invalid	Invalid	Valid
4	Valid	Valid	Invalid
5	Invalid	Valid	Invalid
6	Valid	Invalid	Invalid
7	Invalid	Invalid	Invalid

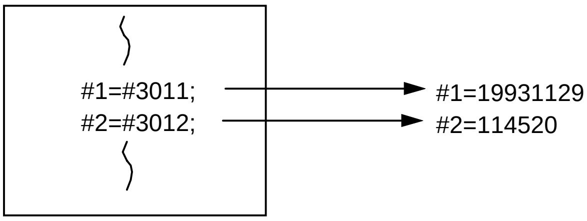
- Variable #3004 is cleared to 0 by resetting.
- Each of the listed bits makes the function valid if 0, or invalid if 1.

11. Clock information

Use of variable numbers 3011 and 3012 allows the date and time to be read.

Value type	Variable number
Year/month/day	3011
Hour/minute/second	3012

Example: In the case of 11:45':20" a.m. of November 29, 1993:



```
#1=#3011;  → #1=19931129
#2=#3012;  → #2=114520
```

12. Parts count (#3901, #3902)

Use of variable numbers 3901 and 3902 allows the actual value and the desired value of parts count to be read or substituted.

Value type	Variable number
Actual value	3901
Desired value	3902

- These variables must be integers from 0 to 9999.
- Data reading and writing by these variables is surely suppressed during tool path checking.

Return to Main Menu

13. G-command modal status

The G-command modal status in the preceding block can be checked using variable numbers from 4001 to 4023.

Variable No.	Function	
#4001	Interpolation mode	G00:0, G01:1, G02:2, G03:3, G32:32, G34:34
#4002	Plane selection	G17:17, G18:18, G19:19, G16:16
#4003	Absolute/incremental	G90:90, G91:91
#4004	Barrier check	G22:22, G23:23
#4005	Feed specification	G98:98, G99:99
#4006	Inch/metric	G20:20, G21:21
#4007	Tool diameter offset	G40:40, G41:41, G42:42, G46:46
#4008	No variable number	
#4009	Fixed cycle	G80:80, G70 ~ 76:70 ~ 76, G83 ~ G85:83 ~ 85 G84.2:84.2, G87 ~ G89:87 ~ 89, G88.2:88.2 G90:90, G92:92, G94:94
#4010	Return level	G98:98, G99:99
#4011		
#4012	Workpiece coordinate syste	G54 ~ G59:54 ~ 59
#4013	Acceleration/Deceleration	G61:61, G62:62, G64:64
#4014	Macro modal call	G66:66, G66.1:66.1, G67:67
#4015	Opposite turret mirror image	G68:68, G69:69
#4016	Program coordinate system rotation	G68.5:68.5, G69.5:69.5
#4017	Peripheral speed constant	G96:96, G97:97
#4018	Balance cut	G14:14, G15:15
#4019	Milling interpolation	G12.1:12.1, G13.1:13.1
#4020	Swiss type machining	G120:120, G121:121
#4021	Polar coordinate input	G122:122, G123:123
#4022		
#4023	Polygonal machining	G50.2:50.2, G51.2:51.2

Sample program

```
G28 X0 Z0;
G00 X150.Z200;
G65 P300 G02 W-30.K-15.F1000;
M02;
O300
#1=#4001; ....... #1=2.0 (Currently valid modal G-code of group 01)
G#1 W#24;
M99;
%
```

Return to Main Menu

14. Other modal information

Modal information about factors other than the G-command modal status in the preceding block can be checked using variable numbers from 4101 to 4120.

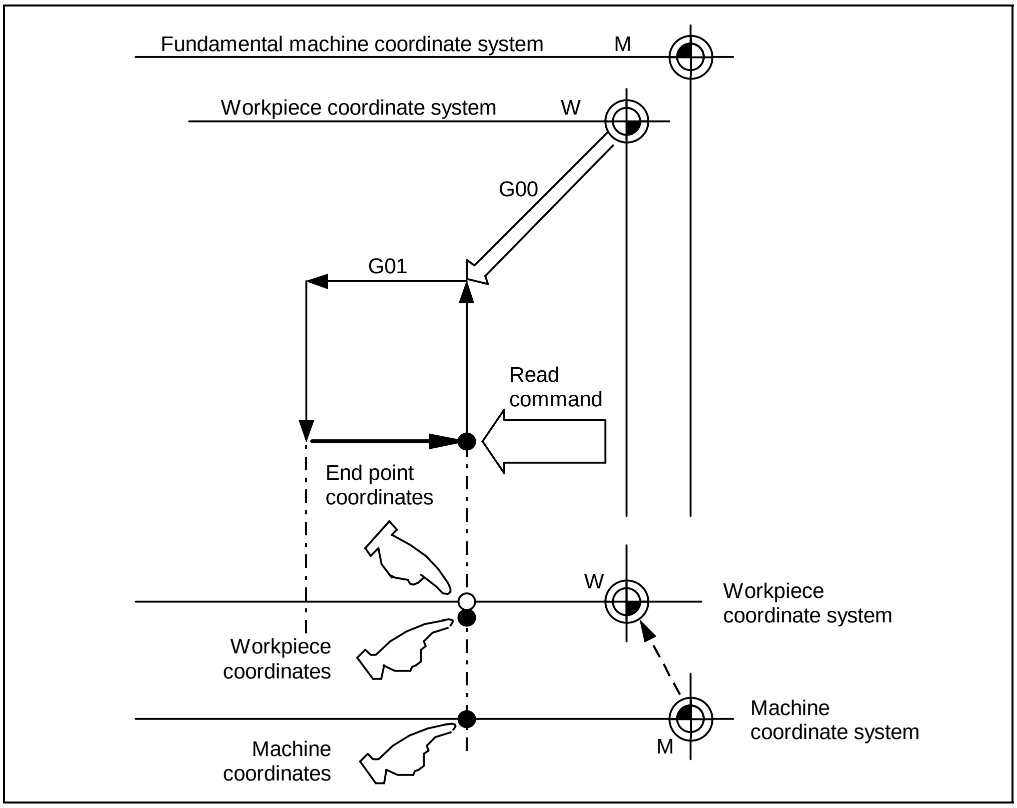
Variable No.	Modal information	Variable No.	Modal information
#4101		#4111	
#4102		#4112	
#4103		#4113	Auxiliary function … M
#4104		#4114	Sequence No. … N
#4105		#4115	Program No. … O
#4106		#4116	
#4107		#4117	
#4108		#4118	
#4109	Feed rate … F	#4119	Spindle function … S
#4110		#4120	Tool function … T

15. Position information

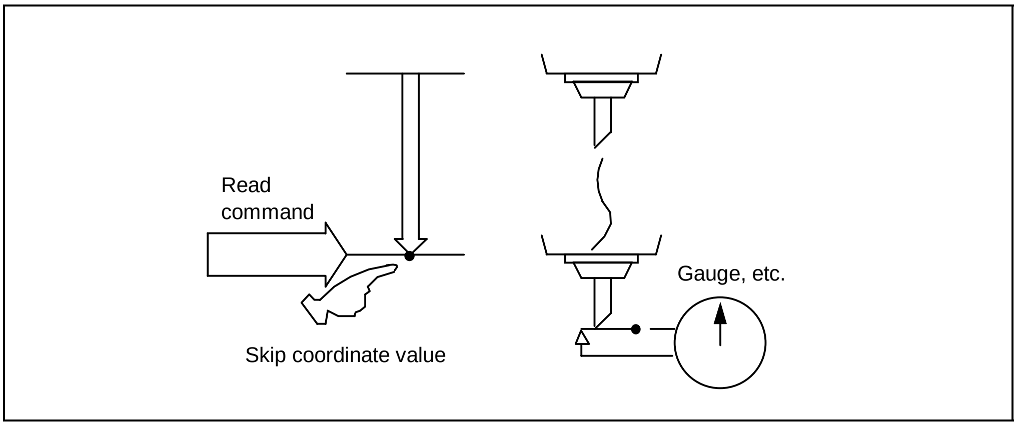
Using variable numbers from #5001 to #5110, you can check the ending-point coordinates of the previous block, machine coordinates, workpiece coordinates, skip coordinates, tool offset amounts, and servo deviations.

Position information Axis. No.	End point coordinates of preceding block	Machine coordinate	Workpiece coordinate	Skip coordinate	Tool offset amount	Servo deviation amount	Turret
1	#5001	#5021	#5041	#5061	#5081	#5101	Selfline
2	#5002	#5022	#5042	#5062	#5082	#5102	
3	#5003	#5023	#5043	#5063	#5083	#5103	
4	#5004	#5024	#5044	#5064	#5084	#5104	
5	#5005	#5025	#5045	#5065	#5085	#5105	
6	—	#5026	#5046	#5066	—	#5106	Counter-part line
7	—	#5027	#5047	#5067	—	#5107	
8	—	#5028	#5048	#5068	—	#5108	
9	—	#5029	#5049	#5069	—	#5109	
10	—	#5030	#5050	#5070	—	#5110	
Remarks (Reading during move)	Possible	Impossible	Impossible	Possible	Impossible	Impossible	—

- The last digit of variable number corresponds to the control axis number.
- Axis numbers 1 to 5 refer to the information of self-line, and 6 to 10, that of counter part line.

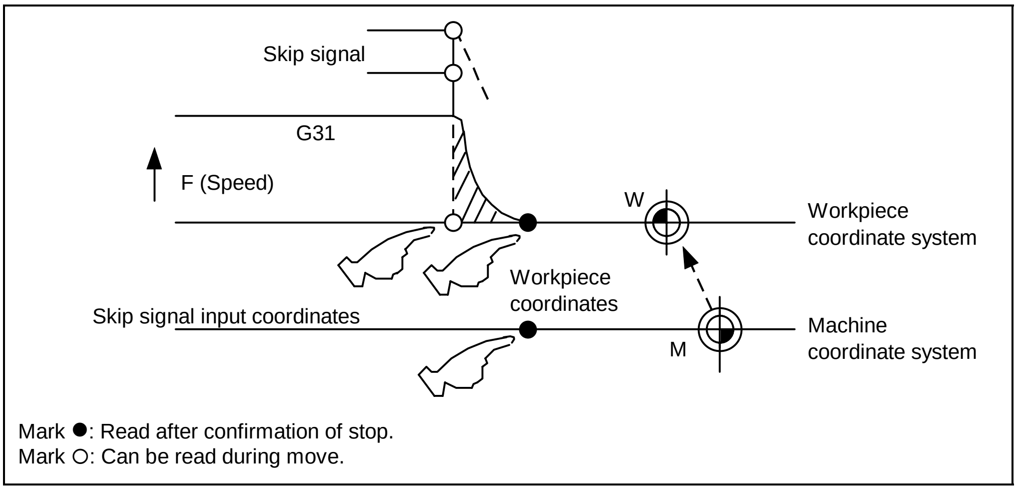


- The ending-point coordinates and skip coordinates read will be those related to the workpiece coordinate system.
- Ending-point coordinates and skip coordinates can be checked even during movement. Machine coordinates, workpiece coordinates, tool offset amount and servo deviation amount must be checked only after movement has stopped.



- 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.

- 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.



Mark ●: Read after confirmation of stop.
Mark ○: Can be read during move.

The input coordinates of a skip signal denote the position within the workpiece coordinate system. The coordinates stored in variables from #5061 to #5070 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 on skip functions for further details.

Return to Main Menu

14-5 Arithmetic Operation Commands

Various operations can be carried out between variables using the following format.

#I = <expression>

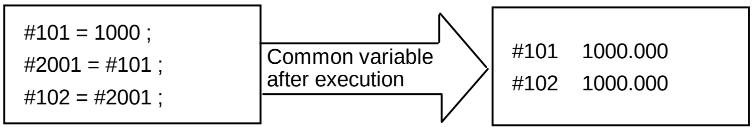
where <expression> must be consist of a constant(s), a variable(s), a function(s), or an operator(s). In the table given below, constants can be used instead of #j and/or #k.

[1] Definition/replacement of variables	#i = #j	Definition/replacement
[2] Additional-type operations	#i = #j+#k #i = #j – #k #i = #j OR #k #i = #j XOR #k	Addition Subtraction Logical additon (For each of 32 bits) Exclusive OR (For each of 32 bits)
[3] Multiplicative-type operations	#i = #j∗ #k #i = #j / #k #i = #j MOD #k #i = #j AND #k	Multiplication Division Surplus Logical product (For each of 32 bits)
[4] Functions	#i = SIN [#k] #i = COS[#k] #i = TAN[#k] #i = ATAN[#j] #i = ACOS [#j] #i = SQRT[#k] #i = ABS[#k] #i = BIN[#k] #i = BCD [#k] #i = ROUND[#k] #i = FIX[#k] #i = FUP[#k] #i = LN[#k] #i = EXP[#k]	Sine Cosine Tangent (tan θ is used as sin θ/cos θ) Arc-tangent (ATN is also available.) Arc-cosine Square root (SQR is also available.) Absolute value BINARY conversion from BCD BCD conversion from BINARY Rounding to the nearest whole number (RND is also available.) Cutting away any decimal digits Counting any decimal digits as 1s Natural logarithm Exponent on 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 #2001, workpiece coordinate system offsets from variable #2501, and other data become data that has a decimal point. If data without a decimal point is defined using these variable numbers, therefore, a decimal point will also be assigned to the data.

Example:



```
#101 = 1000 ;
#2001 = #101 ;
#102 = #2001 ;
```

Common variable after execution

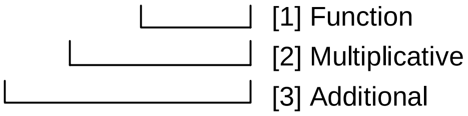
```
#101    1000.000
#102    1000.000
```

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.

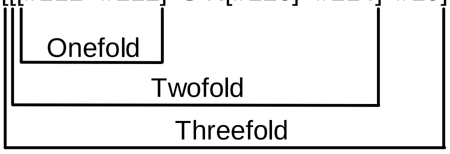
#101=#111+#112∗SIN[#113]



2. Specifying an operational priority leveltext

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] ;



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 } Common variable #4 200.000 #5 –10.000 } Offset amont
[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[173205/100000] #562=ATAN[173205/100.] #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 RND or 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 Error
[21] Exponent EXP	#104=EXP[2] #105=EXP[1] #106=EXP[–2]	#104 7.389 #105 2.718 #106 0.135

Return to Main Menu

4. Operation accuracy

The errors listed in the table below are generated by one arithmetic operation, and the error rate increases each time operations are repeated.

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
			$\|\varepsilon\|$ degrees

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.

14-6 Control Commands

The flow of a program can be controlled using IF ～ GOTO and WHILE ～ DO ～ commands.

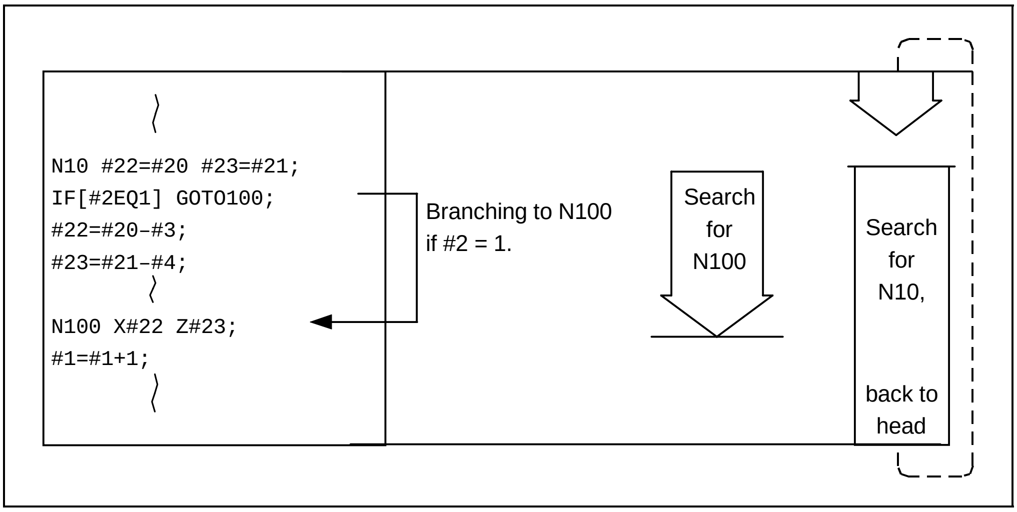
1. Branching

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 #, 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 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 will result. If, however, the block begins with “/” and Nn follows, the branching can be performed to that sequence number.



```
N10 #22=#20 #23=#21;
IF[#2EQ1] GOTO100;
#22=#20-#3;
#23=#21-#4;

N100 X#22 Z#23;
#1=#1+1;
```

Note 1: 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 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.

Note 2: Use only integers for the comparison of EQ and NE. Use GE, GT, LE and LT for the comparison of values with decimal fractions.

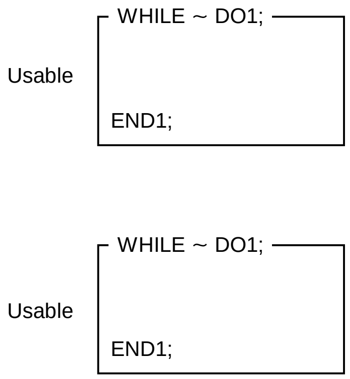
Return to Main Menu

2. Looping

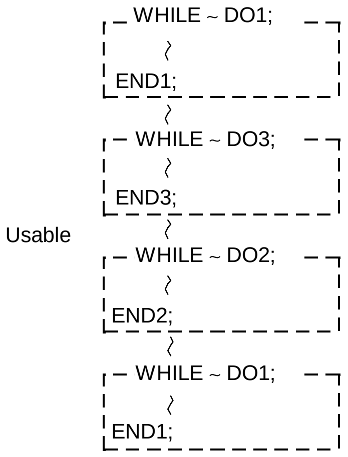
Format:
WHILE [*Condition expression*] DOm; (m = 1,2,3 ···· 127)
⌇
ENDm;

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.

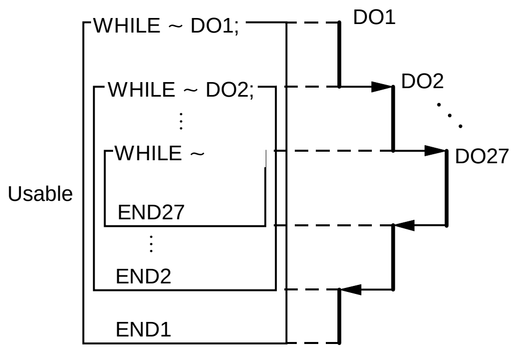
[1] Same identifying No. can be used repeatedly.



[2] The identifying No. of WHILE ~ DOm is arbitraty.

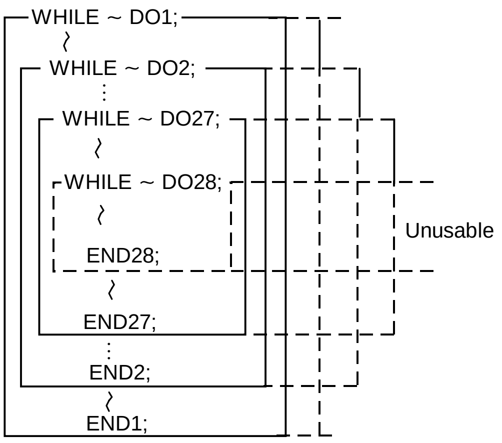


[3] Up to 27 levels of WHILE ~ DOm can be used. m can be 1 to 127, independent of the depth of nesting.



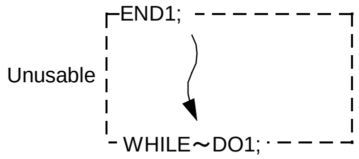
Note: For nesting, m once used cannot be used again.

[4] The total number of levels of WHILE ~ DOm must not exceed 27.

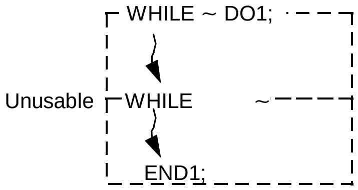


Return to Main Menu

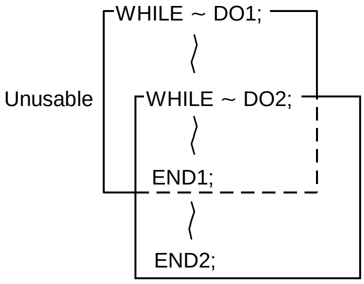
[5] WHILE ~ DOm must precede ENDm.



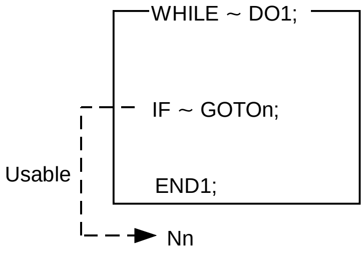
[6] WHILE ~ DOm must correspond to ENDm one-to-one in the same program.



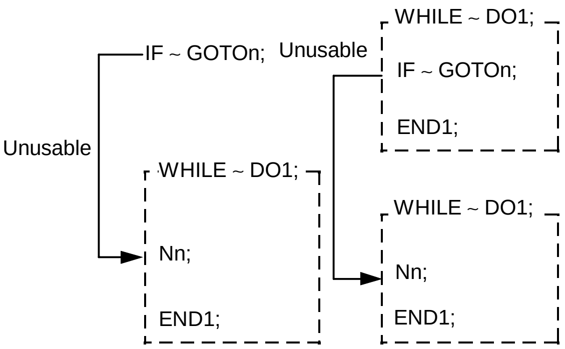
[7] WHILE ~ DOm must not overlap.



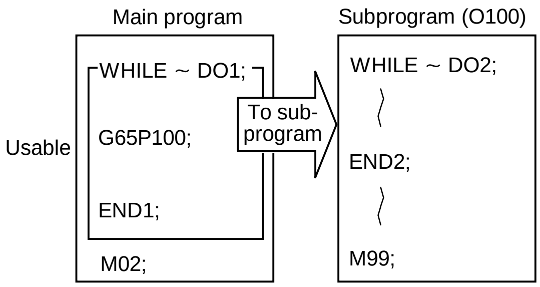
[8] Outward branching from the range of WHILE ~ DOm is possible.



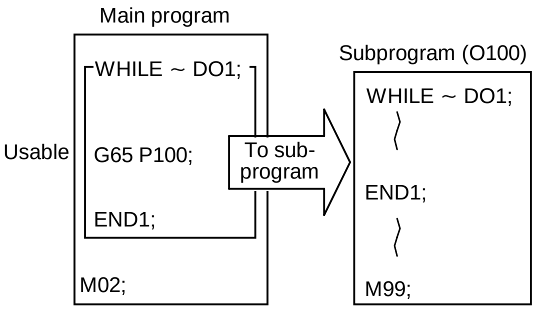
[9] Branching into WHILE ~ DOm is not allowed.



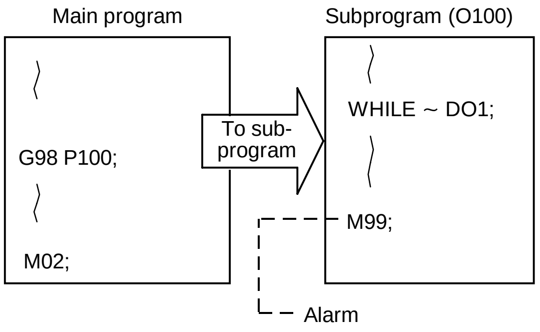
[10] Subprogram can be called using M98, G65, G66, etc. from the midway of WHILE ~ DOm.



[11] The looping can be independently programmed in a subprogram which is called using M65/M66 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.



14-7 External Output Commands

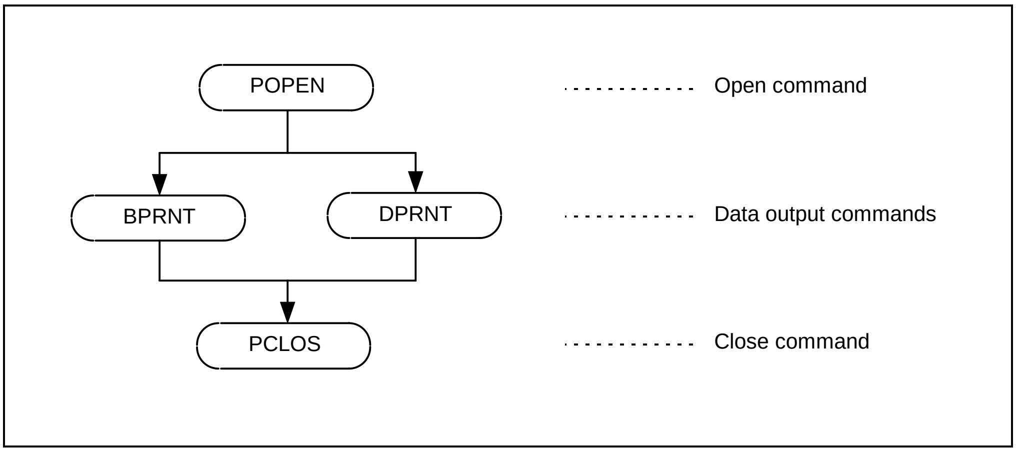
1. Overview

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.

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

2. Programming order



3. Open command POPEN

Programming format: POPEN;

[Detailed description]

- 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.

4. Close command PCLOS

Programming format: PCLOS;

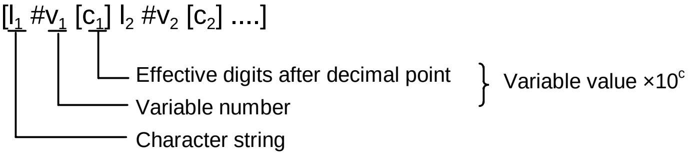
[Detailed description]

- The command code PCLOS must be included after all data output command codes.
- The control code for DC4 and the percentage code % are output 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.

5. Data output command BPRNT

Programming format:

BPRNT [l_1 #v_1 [c_1] l_2 #v_2 [c_2] ....]



[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 output 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

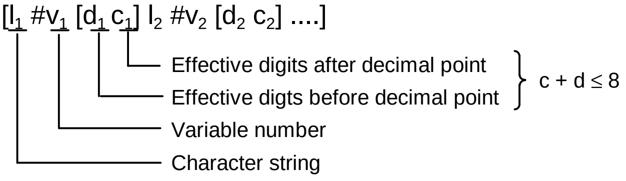
–100 (FFFFFF9C)
will be outputted as binary data.

- 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.

6. Data output command DPRNT

Programming format:

DPRNT [l_1 #v_1 [d_1 c_1] l_2 #v_2 [d_2 c_2] ....]


Effective digits after decimal point
Effective digts before decimal point
Variable number
Character string
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 the data contained in a variable, the necessary number of digits before the decimal point and that of digits after 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.

Return to Main Menu

- Leading zero (0) is omitted.
  A space will take place of an omitted leading zero so that the last digits of data outputted to the printer can be right-justified.

Note: Data output can be commanded also for two-line specifications, but the output channel is used common to both lines. Therefore, avoid executing output command for both lines together.

14-8 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 the macro commands and that of conventional NC commands are taken as a macro instruction and an NC instruction, respectively. The treatment of a macro instruction 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 instruction and the macro instruction becomes possible according to the setting of bit 6 of parameter **P11**.

(It becomes possible to process all macro instructions in batch form by setting the parameter bit to OFF when machining the workpiece, or to execute the macro instructions block-by-block by setting the parameter bit to ON when checking the program. Therefore, set the parameter bit according to your requirements.)

Sample programs

```
N1  G91 G28 X0 Z0;
N2  G92 X0 Z0;
N3  G00 X–100.Z–100.;
N4  #101=100.∗COS[210.];
N5  #103=100.∗SIN[210.];
N6  G01 X#101 Z#103 F800;
```

Macro-instruction

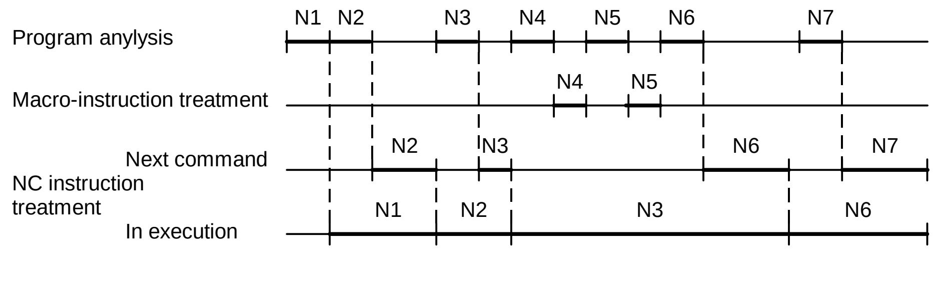
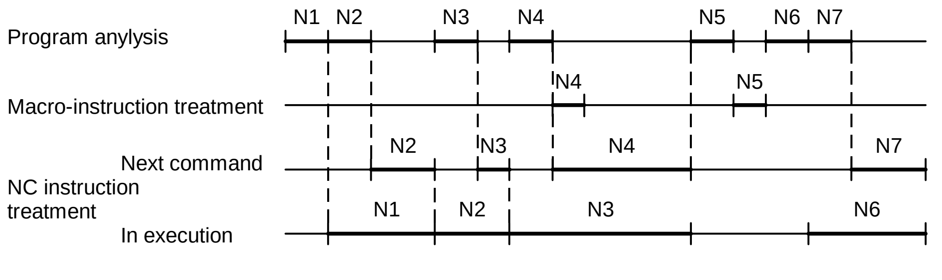
A macro instruction refers to an instruction that includes the following commands:

- Arithmetic operation command (compassing the equal sign =)
- Control command (compassing GOTO, DO ~ END, etc.)
- Macro call command (compassing macro call or cancellation G-code G65, G66, or G67)

An NC instruction refers to a non-macro instruction.

Return to Main Menu

The flow of processing of these two types of instructions is shown below.

Parameter OFF	 Program anylysis N1 N2 N3 N4 N5 N6 N7 Macro-instruction treatment N4 N5 Next command NC instruction treatment N2 N3 N6 N7 In execution N1 N2 N3 N6
Parameter ON	 Program anylysis N1 N2 N3 N4 N5 N6 N7 Macro-instruction treatment N4 N5 Next command NC instruction treatment N2 N3 N4 N7 In execution N1 N2 N3 N6

Machining program data is displayed as follows:

Parameter OFF	(In execution) `N3 G00 X−100.Z−100.;` (Next command) `N6 G01 X#101 Z#103 F800;`	N4, N5 and N6 are treated in parallel with NC execution of N3, and N6 is displayed as next command because it is NC instruction. When N4, N5, and N6 are analyzed during NC execution of N3, machine control continues.
Parameter ON	(In execution) `N3 G00 X−100.Z−100.;` (Next command) `N4 #101 = 100.*COS[210];`	N4 is treated in parallel with the control of NC execution of N3, and is displayed as next command. After N3 is completed, N5 and N6 are analyzed so the machine control is forced to wait by the analyzing time of N5 and N6 before N6 can be executed.

14-9 Macro Interruption

1. Overview

User macro interruption function calls another program to precede the excuting program by inputting a user macro interruption signal (UIT) from the machine side during program execution. Use of the function permits program execution depending on conditions.

2. Function

A. Programming format

```
M96 P_ ;.........................User macro interruption valid
  ⌇    ↑──────────────────────── Interruption program number
M97 ;..............................User macro interruption invalid
```

User macro interruption function is used by putting the interruption signal (UIT) into a valid state or an invalid state by M96 or M97 command in the program. In other words, while the interruption function is valid (from G96 to G97 or resetting), the input of an interruption signal (UIT) from the machine side allows the user macro interruption to be started, resulting in the execution of interruption of a program commanded by “P_” into the program in execution.

Interruption signal (UIT), which is inputted during the interruption of user macro or in a state where user macro interruption is invalidated by M97 or after NC is reset, is ignored until M96 is commanded.

When optional user macro interruption function is selected, M96 and M97 are internally processed as user macro interruption control M-code, but when not selected, they are externally outputted as ordinary M-code.

B. Usable conditions

User macro interruption function is used only during program execution. Therefore, usable conditions are:

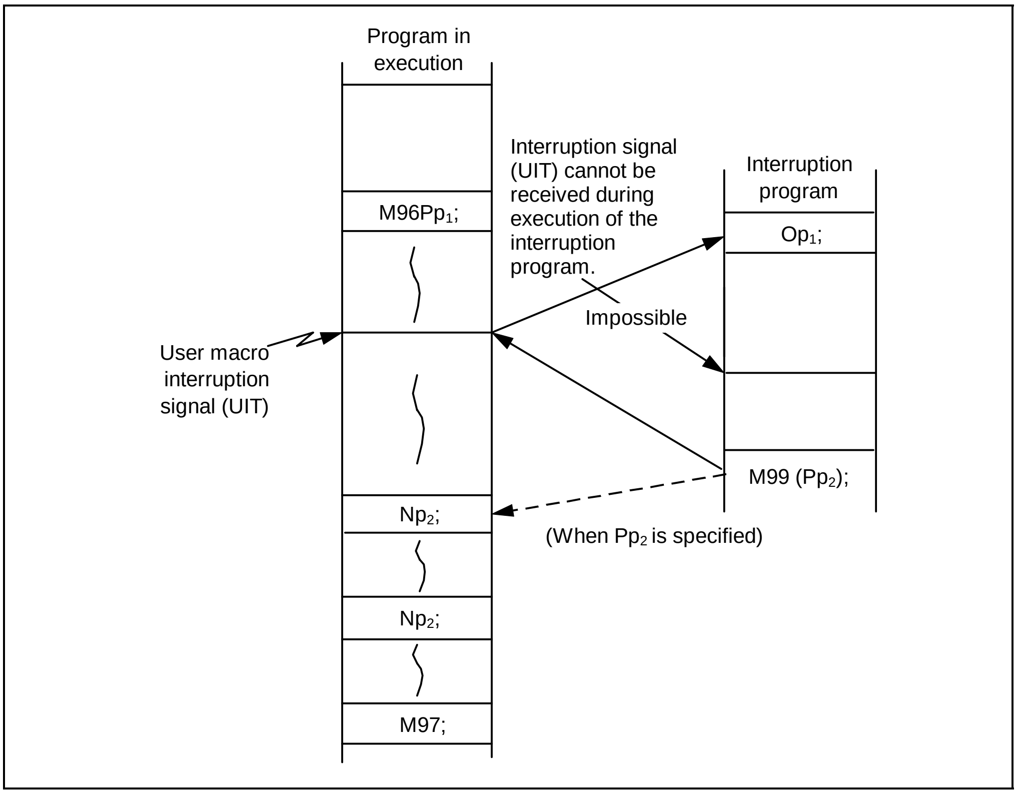
- Either automatic operation of memory or tape, or MDI is selected.
- Automatic operation is being performed.
- User macro interruption is not in process.
- “User macro interruption valid” is selected (**P16** bit 4 = 1).

Macro interruption is invalid during manual operation (jog, step, handle-pulse feed, etc.)

3. Outline of program execution

Input of a user macro interruption signal (UIT) after M96Pp_1; is commanded on the program in execution causes interruption program Op_1 to be executed, resulting in return to the previous program by “M99;” command in the interruption program.

For a command by “M99Pp_2;”, sequence No. p_2 is searched from the block following the interrupted block to the block of the end of the program. And if not found, from the block of the head of the program to the block preceding the interrupted block is searched, and a return is made to the block of sequence number “Np_2;” which has been found first.


Program in execution
M96Pp1;
Np2;
Np2;
M97;
User macro interruption signal (UIT)
Interruption signal (UIT) cannot be received during execution of the interruption program.
Impossible
Interruption program
Op1;
M99 (Pp2);
(When Pp2 is specified)

4. Interruption method

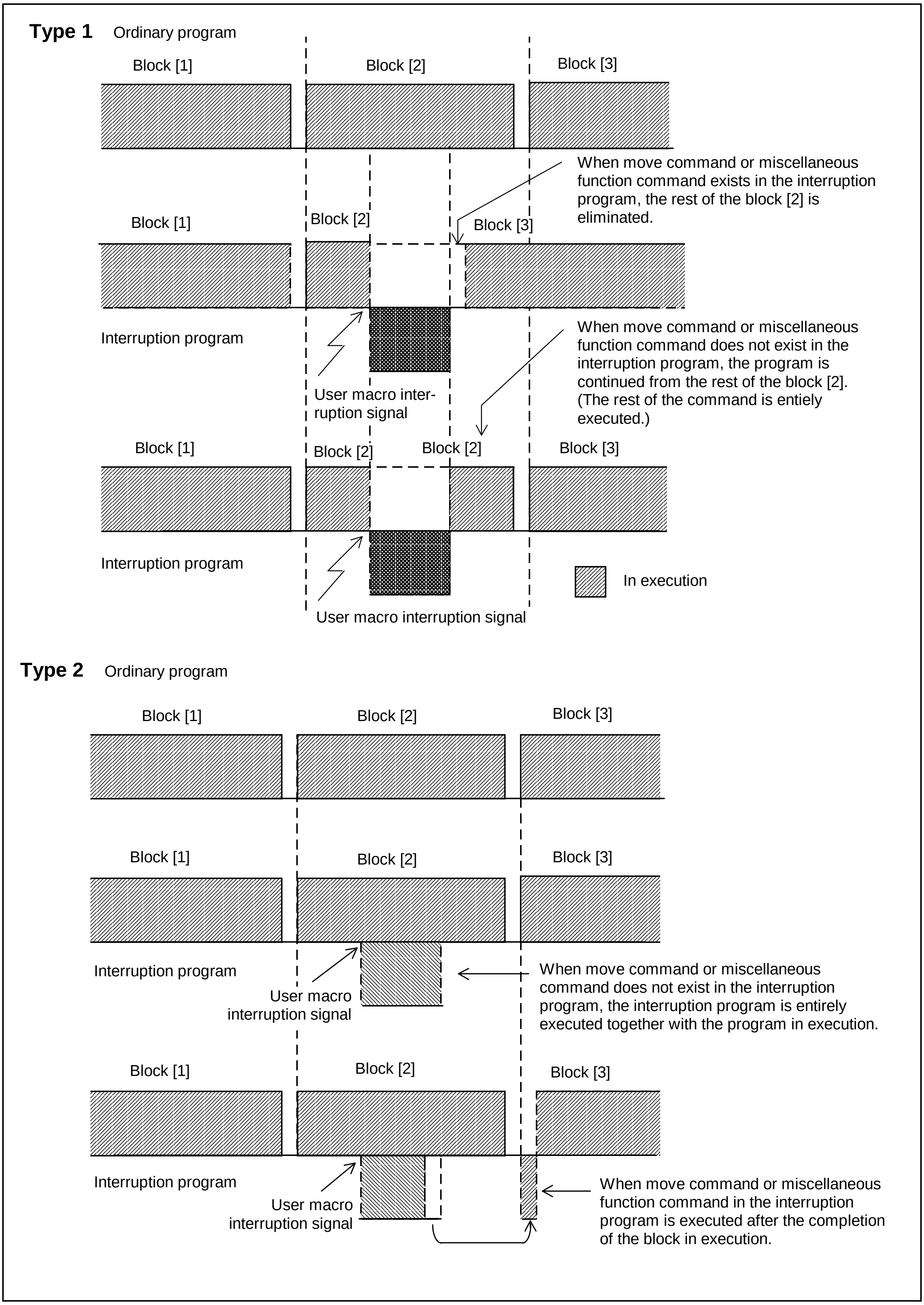
Interruption method includes type 1 and type 2, which depend on machine specifications.

A. Type 1

When an interrutpion signal (UIT) is inputted, the movement or dwell in execution is stopped, and the interruption program is executed.
When move command or miscellaneous fucntion command (M, S, T, B) exists in the interruption program, the command of stopped block is eliminated, and the interruption program is executed.
When the interruption program is completed, a restart is made from the block following the stopped block.
When move command or miscellaneous function command (M, S, T, B) does not exist in the interruption program, a restart is made from the stopped point of the stopped block after the return from the interruption program.
However, when an interruption signal (UIT) is inputted during the execution of miscellaneous function command (M, S, T, B), the move command or miscellaneous command (M, S, T, B) in the interruption program cannot be executed until the FIN signal is inputted because NC is placed in a state of signal waiting (FIN).

B. Type 2

When an interruption signal (UIT) is inputted, the commands of the block in execution are completed first, and then the interruption program is executed.
When move command and miscellaneous function command (M, S, T, B) does not exist in the interruption program, the interruption program is executed simultaneously with the block in execution. However, when the interruption program is not ended even at the end of the respective block, machining may be stopped temporarily.


Type 1
Ordinary program
Block [1]
Block [2]
Block [3]
When move command or miscellaneous function command exists in the interruption program, the rest of the block [2] is eliminated.
Block [1]
Block [2]
Block [3]
Interruption program
User macro inter- ruption signal
When move command or miscellaneous function command does not exist in the interruption program, the program is continued from the rest of the block [2]. (The rest of the command is entiely executed.)
Block [1]
Block [2]
Block [2]
Block [3]
Interruption program
User macro interruption signal
In execution
Type 2
Ordinary program
Block [1]
Block [2]
Block [3]
Block [1]
Block [2]
Block [3]
Interruption program
User macro interruption signal
When move command or miscellaneous command does not exist in the interruption program, the interruption program is entirely executed together with the program in execution.
Block [1]
Block [2]
Block [3]
Interruption program
User macro interruption signal
When move command or miscellaneous function command in the interruption program is executed after the completion of the block in execution.

5. Calling mehthod

User macro interruption includes the following two types depending on the calling method of interruption program.

- Subprogram type interruption
- Macro type interruption

They are selected by a parameter (**P16** bit 5).

Subprogram type interruption

User macro interruption program is called as a subprogram (as with M98 call).
In other words, the level of local variables does not change before and after the interruption.

Macro type interruption

User macro interruption program is called as a user macro (as with G65 call). In other words, the level of local variables changes before and after the interruption. And an argument cannot be transferred from the execution program to the interruption program.
In either case the multiplex degree of call is counted up. And also for the calling of subprogram and user macro executed in the interruption program, each multiplex degree of call is counted up.

6. Method of receiving user macro interruption signal (UIT)

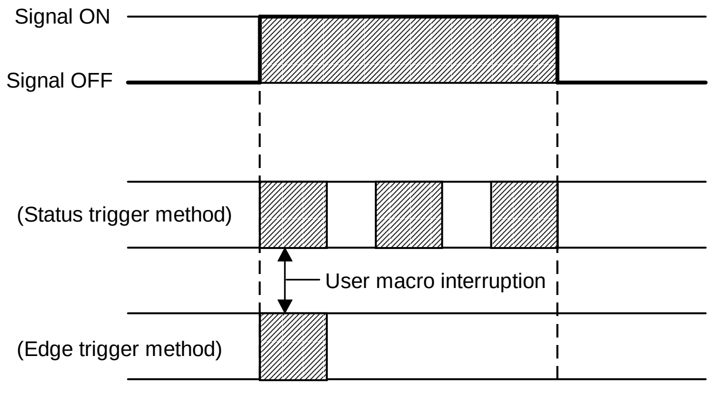
The method of receiving a user macro interruption signal (UIT) includes the following two types depending on machine specifications.

- Status trigger method
- Edge trigger method

Status trigger method
When user macro interruption signal (UIT) is ON, the signal is received as an effective one.
If the interruption signal is ON when user macro interruption becomes valid, the interruption program is executed. The interruption program can be executed repeatedly by keeping the interruption signal ON.

Edge trigger method
When user macro interruption signal (UIT) rises to ON from OFF, the signal is received as an effective one. This method can be used to execute an interruption program only once.

User macro interruption signal (UIT)



Receiving of user macro interruption signal (UIT) D732S0027

7. Return from user macro interruption

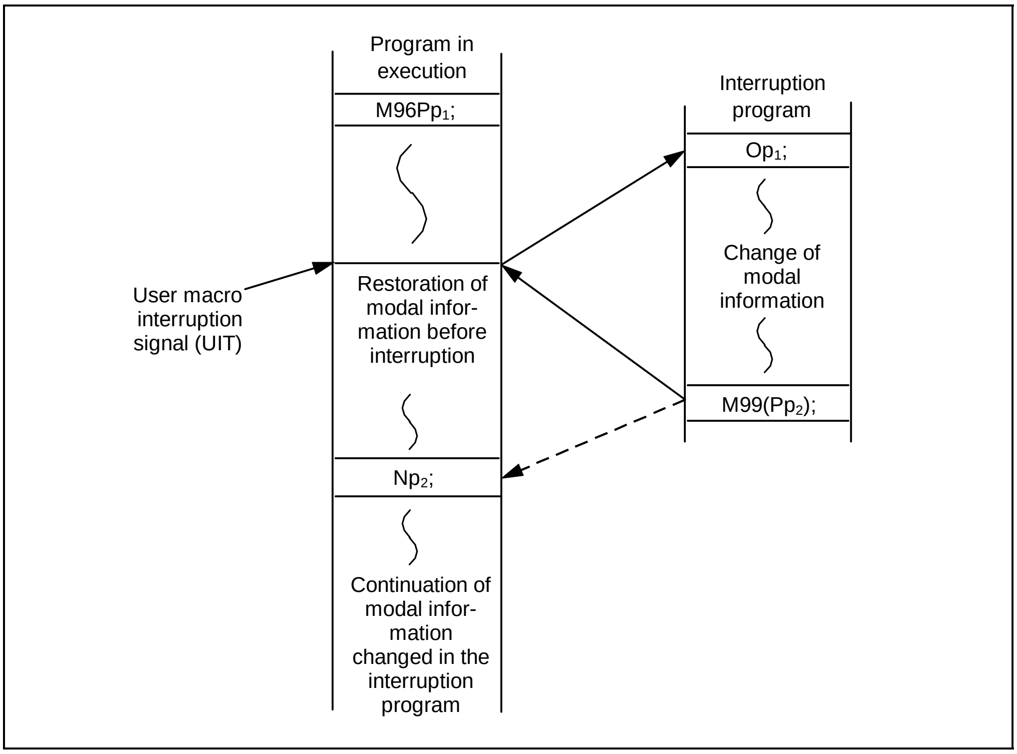
M99 (Pp_2);

For a return to the previous program from user macro, M99 is commanded in the interruption program. The sequence number in the program to be returned can be specified by address P.
In this case from the blocks following the interrupted block to the block of the end of the program is searched. And if not found, from the block of the head of the program to the block preceding the interrupted block is searched, and a return is made to the block of sequence number “Np_2 ;” which has been found first. (As with “M99P_ ;” of M98 call)

8. Modal information in user macro interruption

When modal information is changed in the interruption program, the modal information after return from the interruption program is as follows:

- When returned by “M99 ;”
  Modal information changed in the interruption program becomes invalid, and returns to the modal information before interruption. However, for interruption method type 1, if move command or miscellaneous function command (M, S, T, B) exists in the interruption program, the modal information before interruption cannot be restored any more.
- When returned by “M99P_ ;”
  When modal information is changed in the interruption program, the information changed in the interruption program is continued even after a return from the interruption program. This is as with return by “M99P_ ;” from the program called by M98 or others.


Program in execution
M96Pp1;
User macro interruption signal (UIT)
Restoration of modal infor-mation before interruption
Np2;
Continuation of modal infor-mation changed in the interruption program
Interruption program
Op1;
Change of modal information
M99(Pp2);

9. Modal information

Modal information can be recognized by reading the values of system variables #4001 to #4120 when the control moves to a user macro interruption program. Even if modal information is changed in the interruption program, #4001 to #4120 are not changed.

Return to Main Menu

10. Limitation

- User macro interruption function constituting a part of user macro is not available without user macro specification.
- When system variables #5001 and subsequent (position information) are used to read a coordinate value in the user macro interruption program, they serve as a coordinate value read in the buffer in advance.
- For interruption during the execution of the tool radius compensation, always specify the sequence number (M99P_ ;) to command a return from user macro interruption program. Correct return to the previous program requires specifying the sequence number.

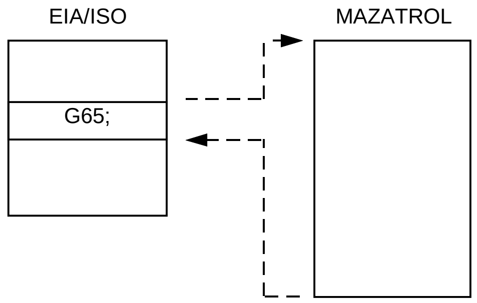
14-10 MAZATROL Program Call by Macro Call Command: G65

1. Overview

MAZATROL machining program can be called as a subprogram from an EIA/ISO machining program.

EIA/ISO → MAZATROL (Program call)

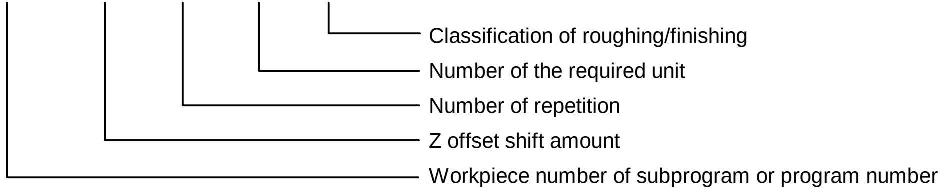
- MAZATROL machining program is called from EIA/ISO program, and it is executed.
  Either the entire machining program or a specified process only (either roughing or finishing) can be executed.



- MAZATROL machining program is executed as is the case with normal start, and not affected by EIA/ISO program at all. When the execution of MAZATROL machining program is ended, 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.
  Either the entire MAZATROL machining program or a specified process only can be executed. For the specified process, either roughing or finishing can be executed.

2. Programming format

G65 P□□□□ Z□□ L□□ A□□ B□□;


Classification of roughing/finishing
Number of the required unit
Number of repetition
Z offset shift amount
Workpiece number of subprogram or program number

Argument P

- It specifies the program number of the MAZATROL machining program to be called.
- When not specified, the alarm 744 “NO DESIGNATED PROGRAM” is displayed.
- When the specified program number does not exist, the alarm 744 “NO DESIGNATED PROGRAM” is displayed.

Argument Z

- It specifies Z offset shift amount.
- It is specified when multiple workpiece machining is performed by MAZATROL machining program.

  –9999.999 to 9999.999 (mm)
  –999.9999 to 999.9999 (inch)
- A value is specified between the above.

Argument L

- The number of repetition of MAZATROL machining program is specified between 1 and 9999.
- When the value is not specified or when 0 is set, the program is executed taking the value as 1.

Argument A

- It specifies a unit number in MAZATROL machining program.
- When the value is not specified or when 0 is set, the entire specified program is executed taking the value as not specified.
- When the specified unit number does not exist, an alarm occurs.

Argument B

- It specifies the classification of roughing and finishing.

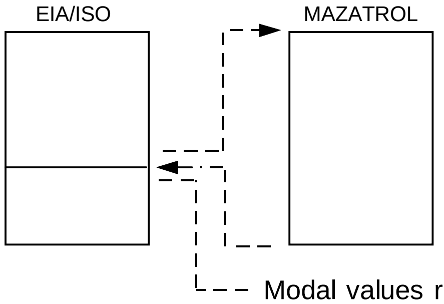
1: Selection of roughing
2: Selection of finishing
3: Selection of re-finishing

- When the value is not specified, it is taken as the selection of roughing.
- When the specified classification does not exist, an alarm occurs.

Example: For G65 P100 A7 B1;
When the process of “work No. 100, unit No. 7, roughing” does not exist, an error occurs.

3. Modal information

For the return from MAZATROL program, the modal value is the same as before MAZATROL program is called. However, it should be noted that used tool, tool position and machine status differ.



Modal values remain unchanged before and after MAZATROL program call. However, used tool, tool position and machine status can be changed. (For example, the spindle which has been rotating forward before the MAZATROL program call is at a stop after the return.)

EIA/ISO program after the return from MAZATROL program must be prepared considering above. (M, S, T or B command is required.)

Return to Main Menu

4. MAZATROL program

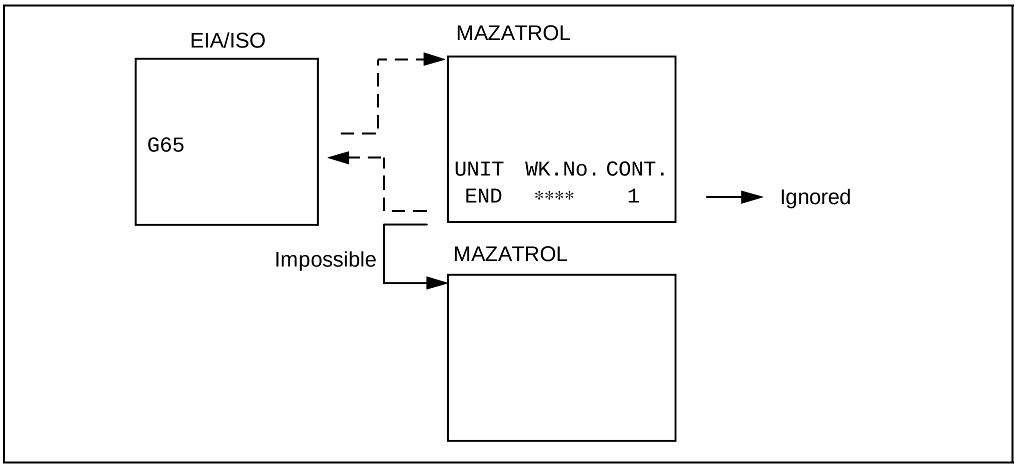
A. End unit

End unit does not have to be specified at the end of MAZATROL machining program.
When end unit is specified:

UNo.	UNIT	COUNTER	RETURN	WK.No.	CONT.	NUM.	SHIFT
□	END			□	□		
(a)		(b)	(c)	(d)	(e)	(f)	(g)

(a) Unit No.
(b) Parts count
0 : Yes 1 : No
(c) 0: No return to zero point
1: Return to zero point
2: Return to fixed point
(d) Work No. or program No. to make program chain
(e) Continuous operation performed or not
(f) Number of repetitions
(g) Z offset shift amount

Even if WK.No. and CONT. are specified, they are ignored. This means that program chain cannot be made with MAZATROL program called from EIA/ISO program.



NUM. and SHIFT are ignored even if they are specified, and L and Z values specified with G65 are valid.

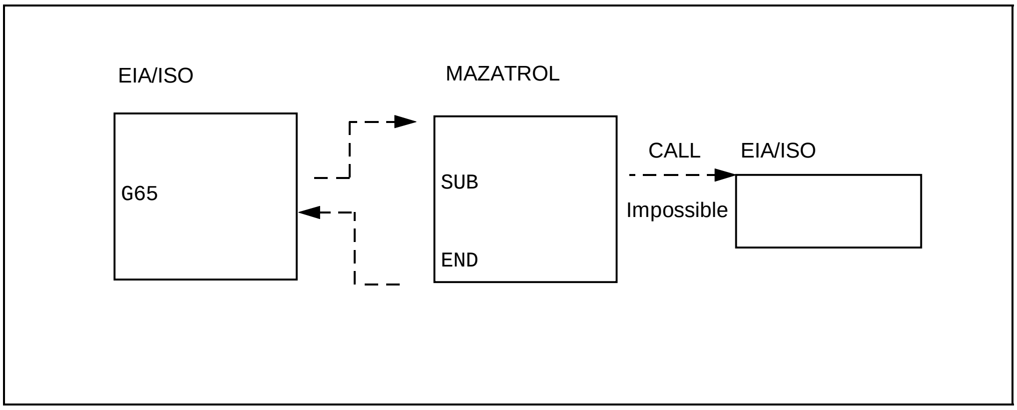
Note: When end unit is not specified, specifying L and Z with G65 means the same as specifying NUM. and SHIFT with the end unit.

B. 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.

C. Nesting

Within a MAZATROL program called from EIA/ISO program, the subprogram unit (SUB) cannot be used.



Refer to the MAZATROL Programming Manual for SUB unit.

Note: As with the case with a SUB 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.

5. Remarks

- MDI interruption and macro interruption signal during MAZATROL program execution are ignored.
- MAZATROL program cannot be restarted halfway.

15 COORDINATE SYSTEM SETTING FUNCTIONS

15-1 Coordinate System Setting Function: G50

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.)

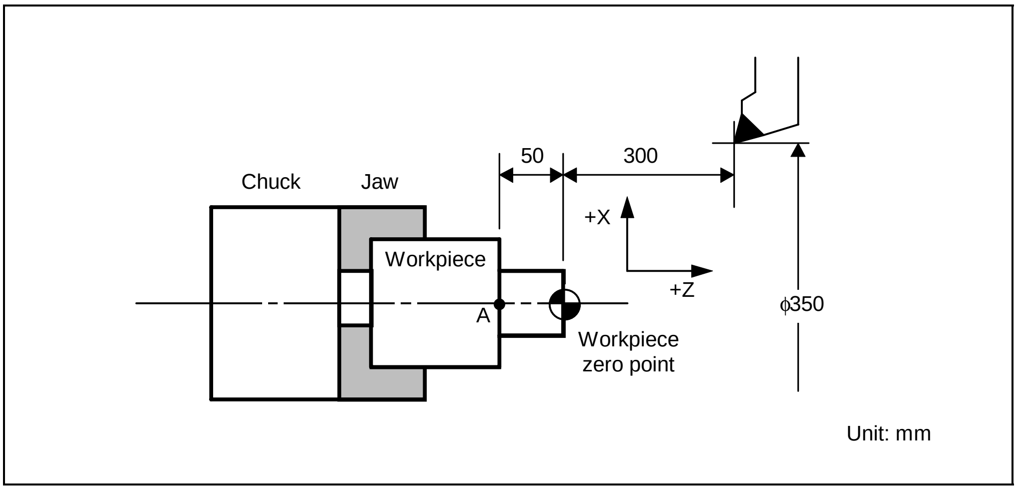
3. Detailed description

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:

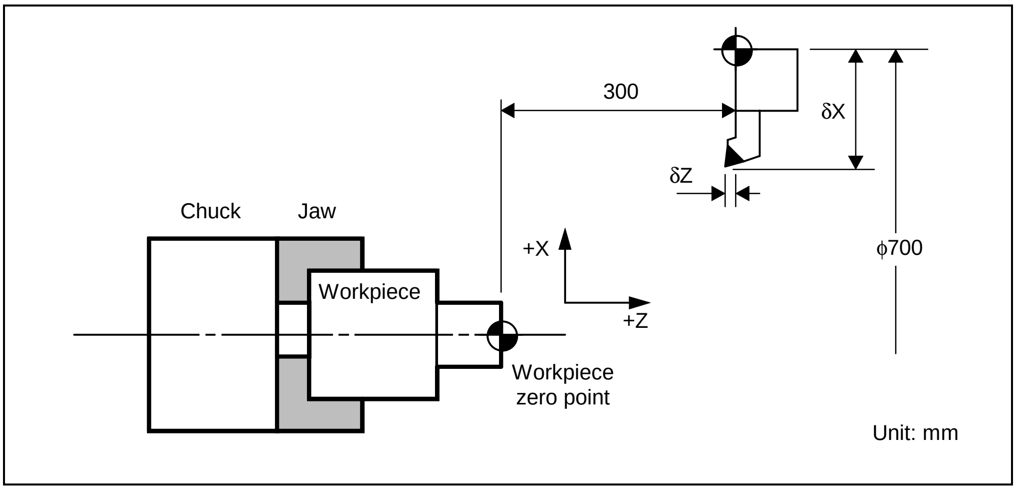


```
G50 X350. Z300.;
```

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 the description of “Tool Position Offset” in this manual.

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.

Return to Main Menu

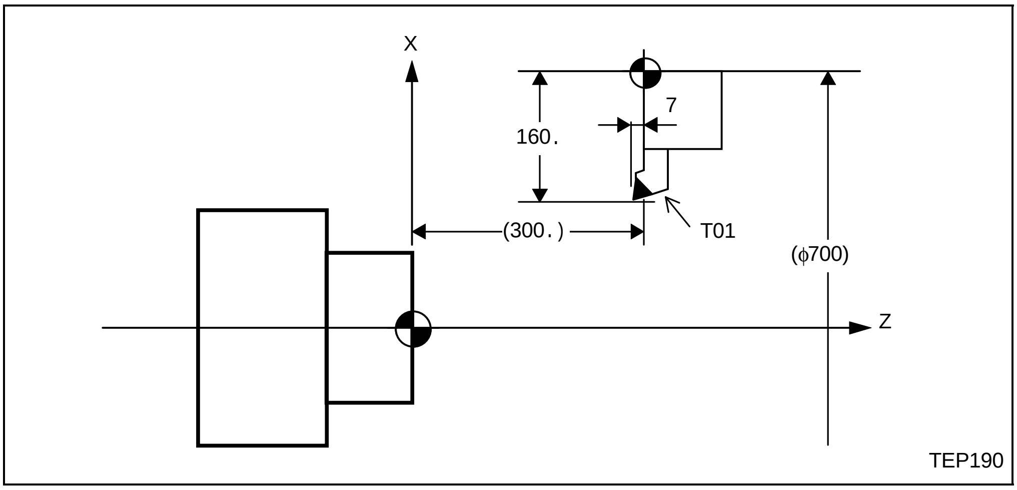
4. Coordinate system shift

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:



(G50 X700. Z300. ;)
(T01)
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 **P13** bit 0 = 0		Parameter **P13** bit 0 = 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 T0101;`	(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 T0100;`	(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)

Example 2:

Offset data (–10.0, –10.0)

Program	Parameter **P13** bit 0 = 0		Parameter **P13** bit 0 = 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. T0101;`	(0, 0)	(50, 50)	(–10, –10)	(50, 50)
`N005 G00 X50. Z50.;`	(40, 40)	(90, 90)	(40, 40)	(90, 90)
`N006 T0100;`	(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 **P13** bit 0 = 0		Parameter **P13** bit 0 = 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. T0101;`	(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 0 of parameter **P13** setting selects whether compensation movement is made on T command (Yes = 1, No = 0).

15-2 MAZATROL Coordinate System Cancellation: G52 (or G52.5)

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 (G52.5)* is ignored in G52 (G52.5) mode.
- When G52 (G52.5) is commanded in G53 (G53.5) mode, the MAZATROL coordinate system currently provided is cancelled. The workpiece coordinate system that has been provided before G53 (G53.5) was commanded is reset. Therefore, display of current position counter is changed.

* In this section the G-codes in parentheses refer to those to be used in the Standard mode.

2. Detailed description

1. When G52 (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 (G52.5) and G50 are commanded in the same block, G50 is executed first and G52 (G52.5) is executed next independently of the order of programs.
3. G52 (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 ;                POSITION counter changed for the G53 system
   G00   X__ Z__ ;
     :
   G00   X__ Z__ ;      POSITION counter with respect to the G53 system
   G52 ;                POSITION counter changed for the preceding G50 system
   ```

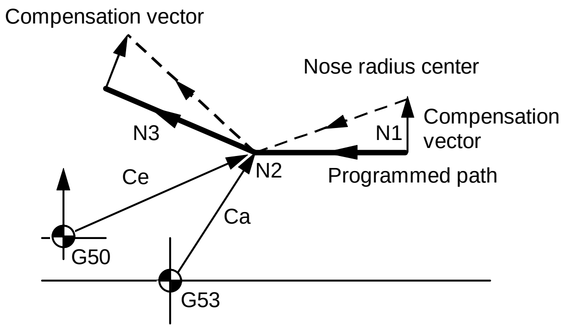
4. It depends on parameter (**P12** bit 2) whether G52 (G52.5) or G53 (G53.5) is selected when turning the power on or resetting.

   P12 bit 2 = 0 : Initial selection G52 (G52.5)
   P12 bit 2 = 1 : Initial selection G53 (G53.5)

   Here, “reset” refers to the following:

   - When the reset key is pressed in G53 (G53.5) mode.
   - When the program is ended in G53 (G53.5) mode. (M30, M198, M199, %)

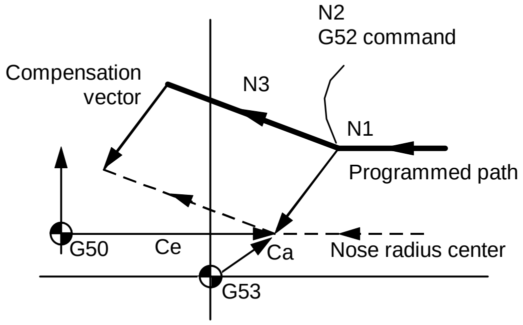
5. When G52 (G52.5) is commanded during nose radius compensation, compensation data will be temporarily cancelled in the movement block just before G52 (G52.5) command. At the time of G52 (G52.5) command, the programmed position and virtual tool tip point are identical.



```
G53;
G41;
 ⋮
N1  G00 Z10.;
N2  G52;
N3  G01 X20. Z20.;
```

TEP183

6. If G52 (G52.5) is commanded during tool position compensation, compensation data will not be cancelled.



```
G53;
 ⋮
N1  G00 Z100.;
N2  G52;
N3  G01 X100. Z100.;
```

TEP184

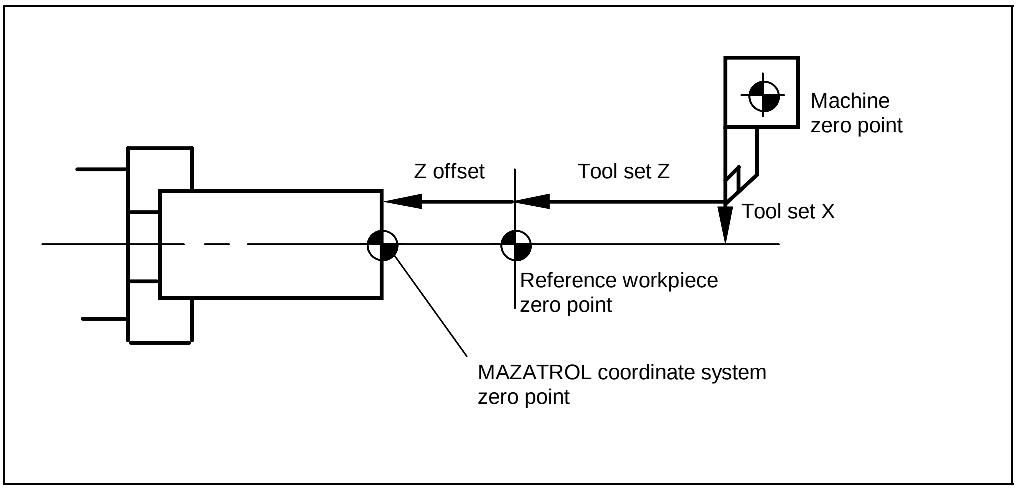
15-3 Selection of MAZATROL Coordinate System: G53 (or G53.5)

1. Function and purpose

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 (G53.5)** is ignored in G53 (G53.5) mode.
- When G53 (G53.5) is commanded in G52 (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 (G53.5), the tool coordinate system is automatically changed.
- When T command and move command are executed to the same block in G53 (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 (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.

** In this section the G-codes in parentheses refer to those to be used in the Standard mode.

2. Detailed description

1. It depends upon the setting of parameter **P12** bit 1 (G50 “valid/invalid” in G53 (G53.5) mode) whether the coordinate system is changed to the coordinate system of G50 when G50 is commanded in G53 (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 (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 (G52.5) in G53 (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 (G53.5) mode is established, and it is not a coordinate system set by G50 in G53 (G53.5) mode.

   For example, in the first (commanded first in the program) G52 (G52.5) at initial selection G53 (G53.5), thellows parameter P1 coordinate system of G54 is selected.
3. G53 (G53.5), T command in G53 (G53.5) mode and G52 (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 (G53.5)	Commands other than G53 (G53.5)	G50
T command in G53 (G53.5) mode	T command, movement command, etc.	G50 by T command
G52 (G52.5)	Commands other than G52 (G52.5)	G50

3. Supplement

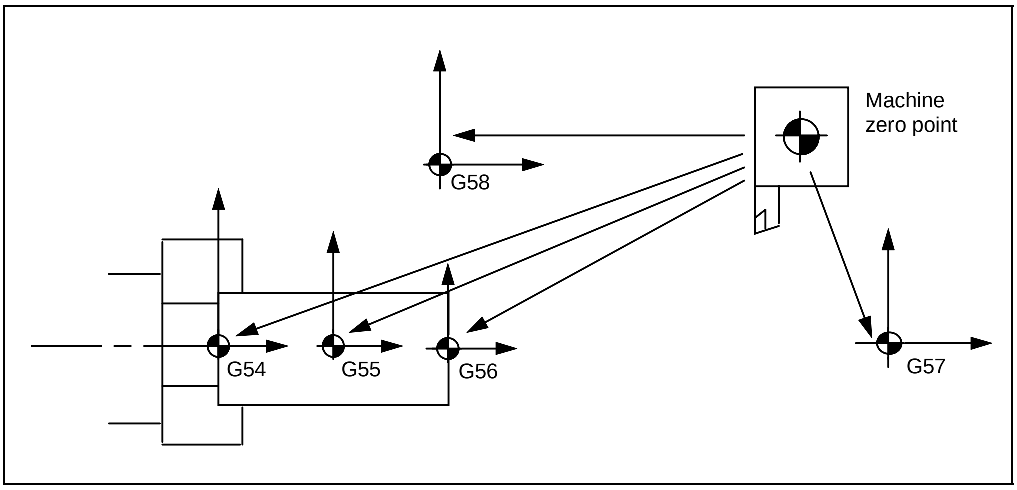
1. Selection of workpiece coordinate system (G54 to G59) is ignored in G53 (G53.5) mode.
2. In the Standard mode, selection of local coordinate system (G52) is ignored in (G53.5) mode.
3. When all the axes other than C-axis have not finished zero point return using watchdog method, G53 (G53.5) command will cause an alarm. However, when G53 (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 (G53.5) and G50 are commanded in the same block, the command is executed in order of G50 to G53 (G53.5) independently of the order of programs.
5. When two or more pieces of G50 exist in a change to G53 (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 (G53.5) mode, the coordinate system of a tool selected after execution of G50 is selected provided that G50 is valid in G53 (G53.5) mode.
7. When T command and G52 (G52.5) are given in the same block in G53 (G53.5) mode, the coordinate system is not changed by T command, and G52 (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 (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.

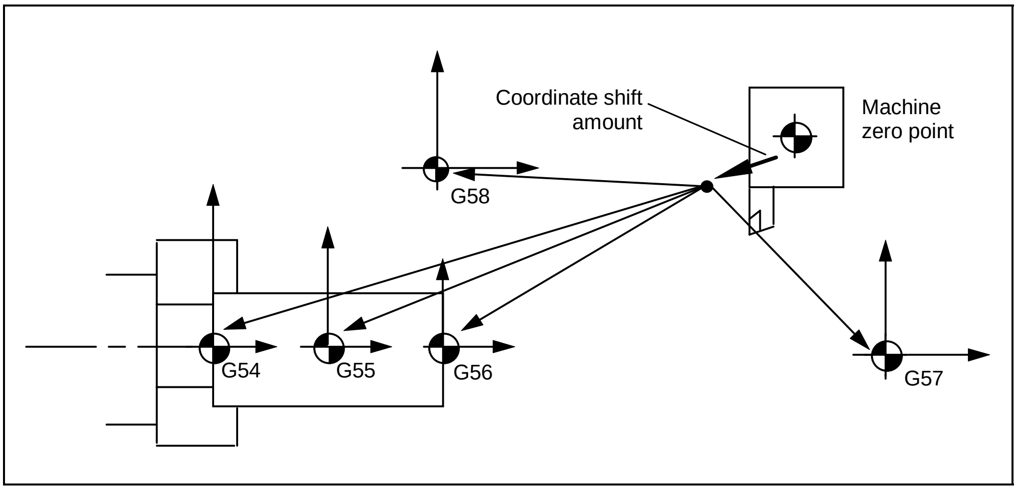
2. Remarks

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.

7. For a machine whose secondary spindle is equipped with the function for the C-axis control or 0.001-degree index (orientation), the C-axis settings on the **WORK OFFSET** display can be used for the 2nd spindle as required with the related parameter (**P77**) being set to "1". Use the preparatory function of G110 to give a command for controlling the C-axis, or indexing in 0.001-degree steps, on the 2nd spindle side.
   See Section 20-2 for details on the programming method.

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.

Return to Main Menu

15-7 Selection of Machine Coordinate System: G53 (Standard Mode)

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 (Standard Mode)

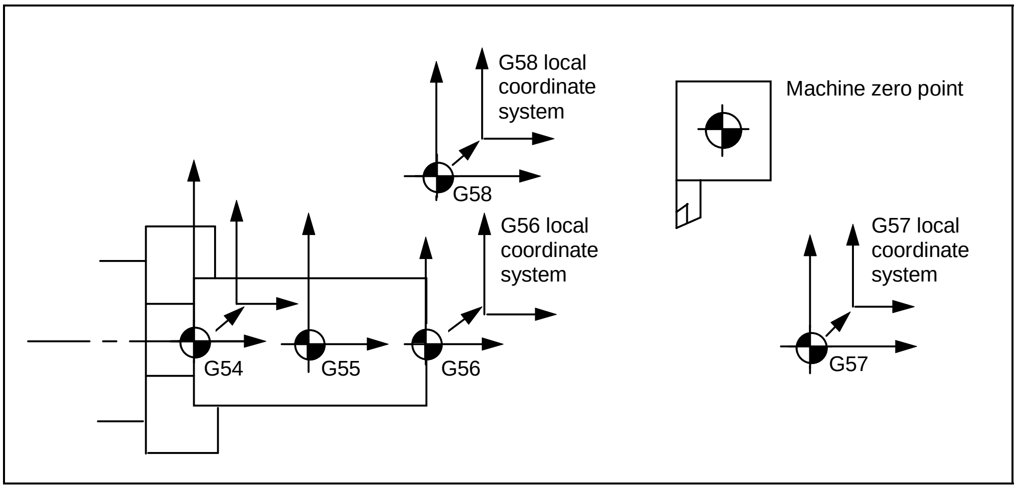
G52 X(U)_ Z(W)_ C(H)_ ;

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.



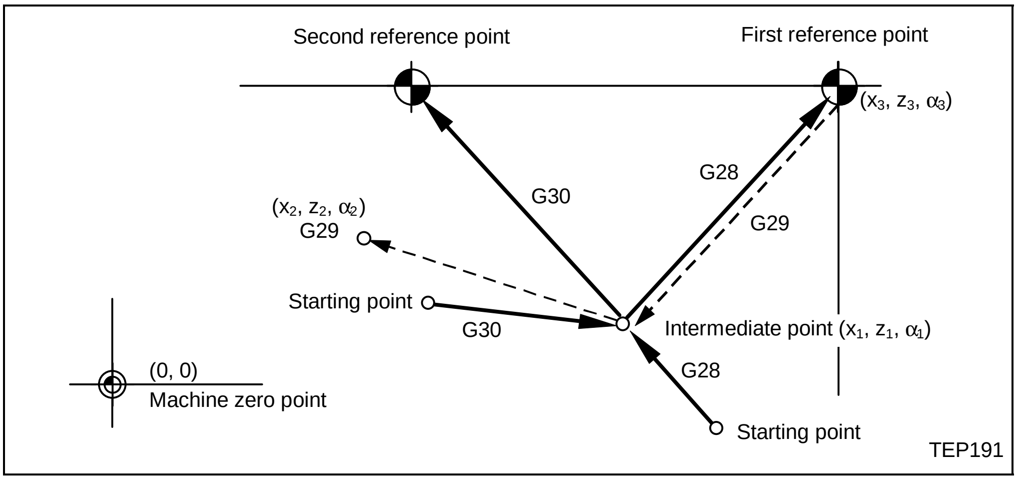
Note: G52 can be used in place of G50 command. However, the distance through which coordinate system is shifted by G52 is not added to other workpiece zero point offset values.

Return to Main Menu

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.



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:

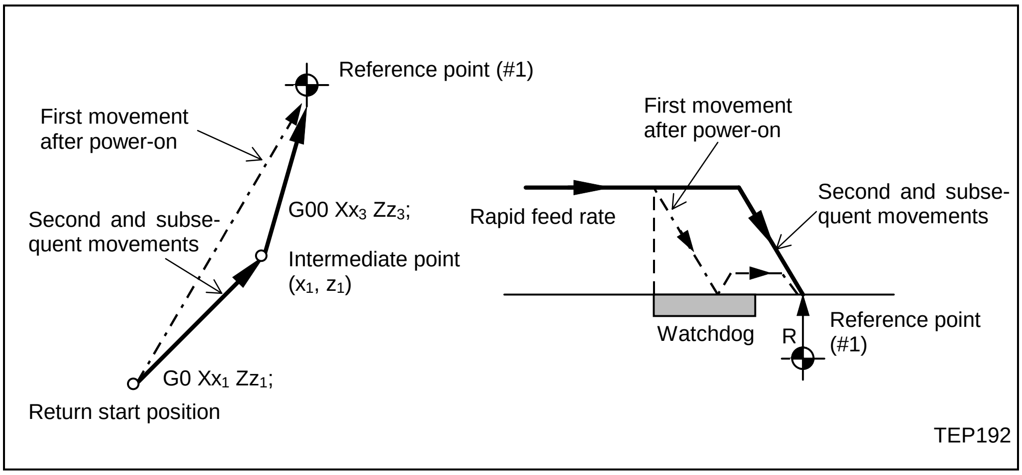
   G00 Xx_1 Zz_1 $\alpha\alpha_1$;
   G00 Xx_2 Zz_2 $\alpha\alpha_2$;
   } This results in independent rapid feed of each axis.

   Where x_1, z_1 and a_1 are the coordinates of the intermediate point specified by G28 or G30.

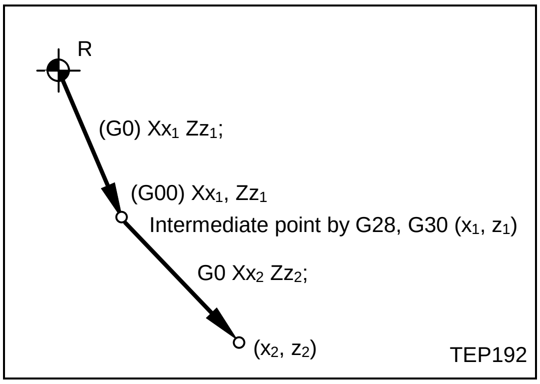
5. A program error will result if G29 is commanded without G28 (automatic return to reference point command) or manual return to zero point being executed after the power has been turned on.
6. The intermediate point coordinates (x_1, z_1, α_1) for the positioning are set by absolute/incremental value.
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. The intermediate point can also be ignored by setting parameter **P11** bit 5.
10. 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.
11. 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.
12. In case of cycle start in the single step mode, stop will be made at the intermediate point.

4. Sample programs

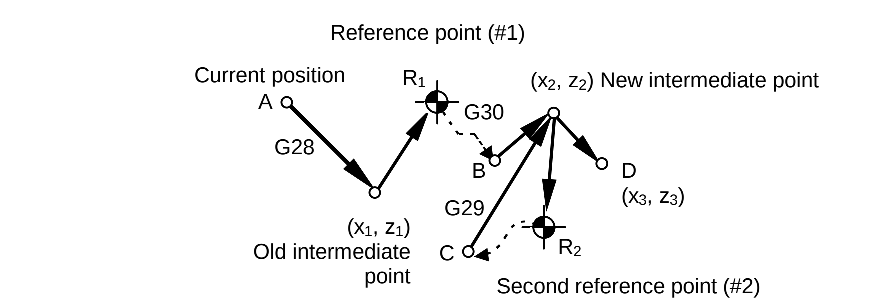
Example 1: G28 Xx_1 Zz_1;


Reference point (#1)
First movement after power-on
Second and subsequent movements
G00 Xx3 Zz3;
Intermediate point (x1, z1)
G0 Xx1 Zz1;
Return start position
First movement after power-on
Rapid feed rate
Second and subsequent movements
Watchdog
R
Reference point (#1)
TEP192

Example 2: G29 Xx_2 Zz_2;



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

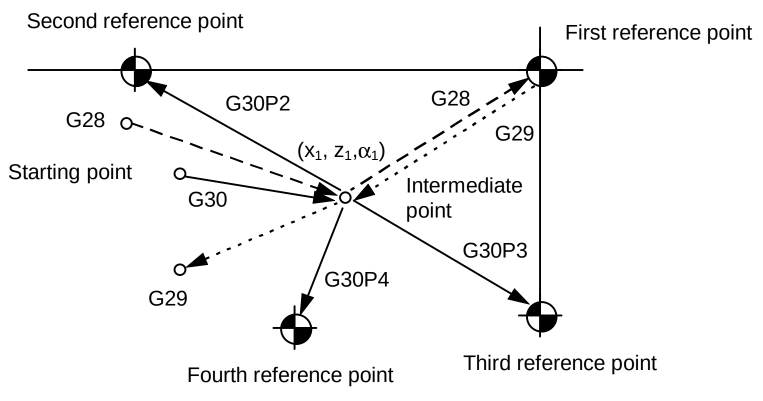


TEP193

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.)

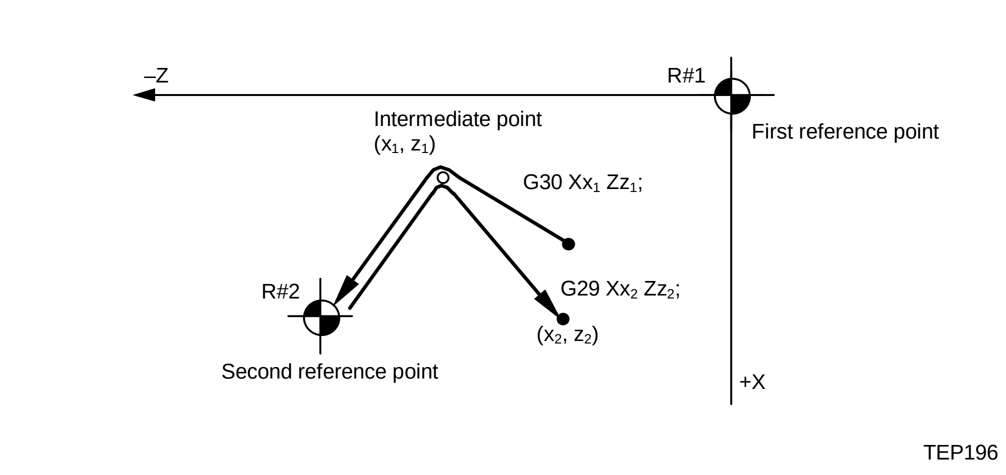
3. Detailed description

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.

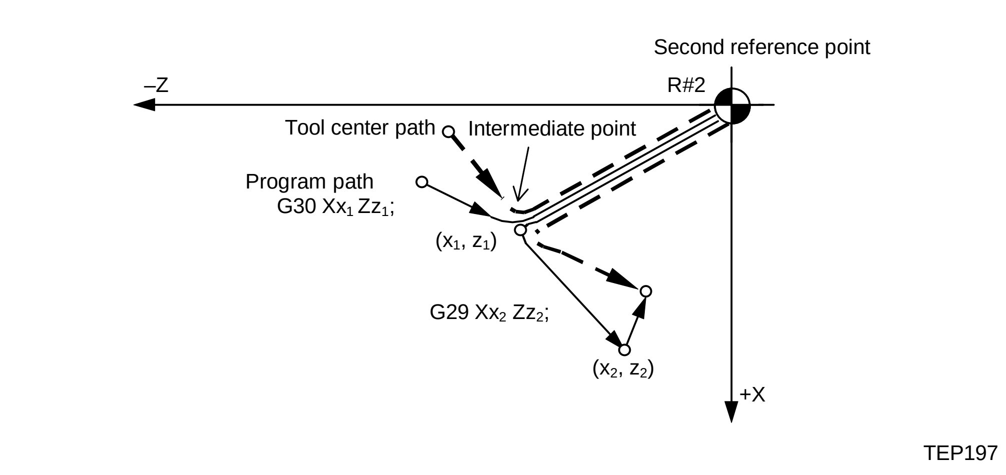
3. The coordinates of the second, third, or fourth reference point represent the position specific to the machine. The coordinates can be checked by using the machine parameter **A5**, **A6**, or **A7**.

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.

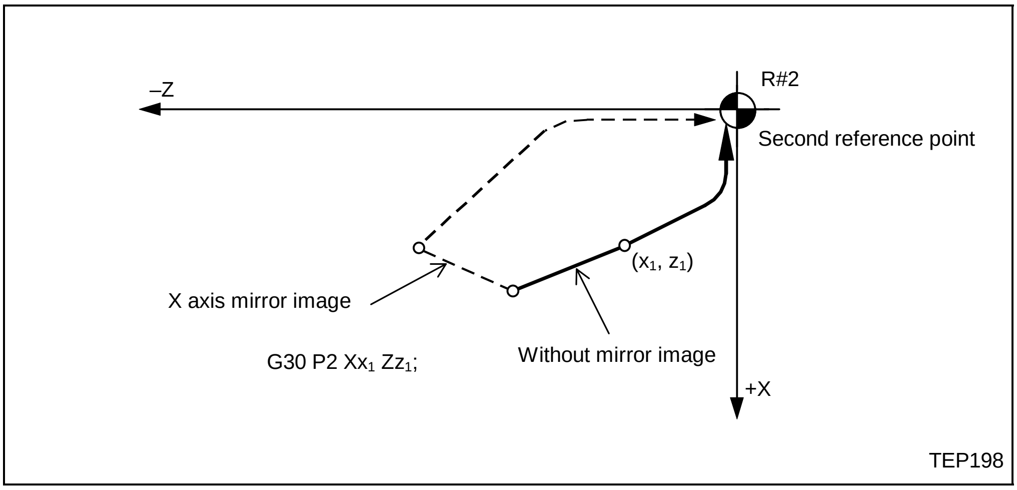


TEP197

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.

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.



9. In case of cycle start in single step mode, no stop will be made at the intermediate point.

15-11 Floating Reference Point Return: G30.1

1. Function and purpose

G30.1 X(U)_ Z(W)_ C(H)_ ;

The above command permits returning to the floating reference point. Floating reference point is a position serving as a reference in operating the machine. It is not always a fixed position. The position may be changed.

Specifically, it is a machine coordinate position stored by pressing the menu key for the floating reference point (POSITION PRESET) on the **POSITION** display, and it is a point established in machine parameter **A15**.

The commanded axis is first positioned at the intermediate point at rapid feed rate, and then positioned at the floating reference point from the intermediate point at rapid feed rate. (There is no difference from G30 in operation in particular.)

Note 1: When G30.1 is commanded, tool offset and tool nose radius compensation must be cancelled.

Note 2: Floating reference point is stored even after the power is turned off.

2. Supplement

Generally a lathe can perform tool changes at any position unless the position comes in contact with the workpiece and tailstock. However, to shorten the cycle time, it is wise to perform tool changes at a position as close as possible from the workpiece. This requires changing the tool position depending on the shape of the workpiece, which can be facilitated by the function.

That is, once a tool position most suited to the workpiece is stored as the floating reference point, return to the tool change position can easily be executed by commanding G30.1.

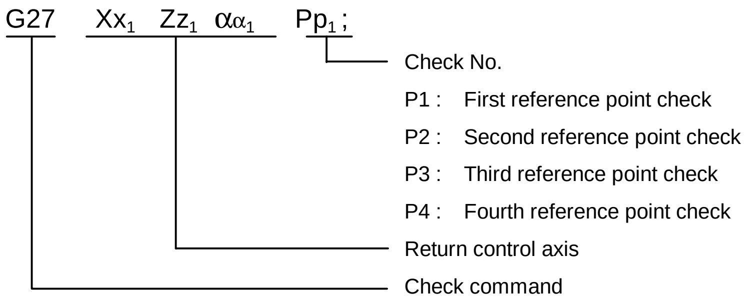
Return to Main Menu

15-12 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 ;
Check No.
P1 : First reference point check
P2 : Second reference point check
P3 : Third reference point check
P4 : Fourth reference point check
Return control axis
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.

15-13 Program Coordinate System Rotation ON/OFF : G68.5/G69.5

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.

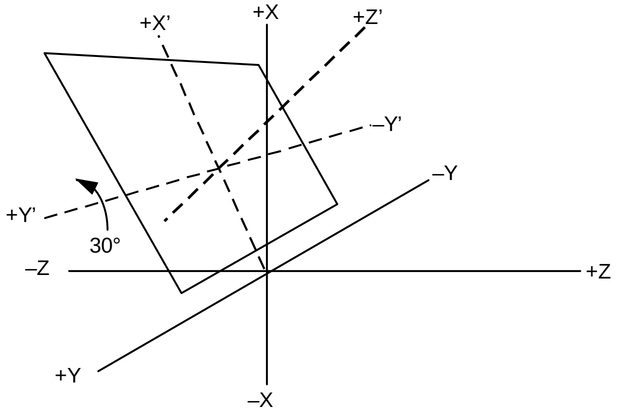
i, j, k : Designation of rotational axis (1: valid, 0: invalid)
I : X-axis
J : Y-axis
K : Z-axis

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.;
```

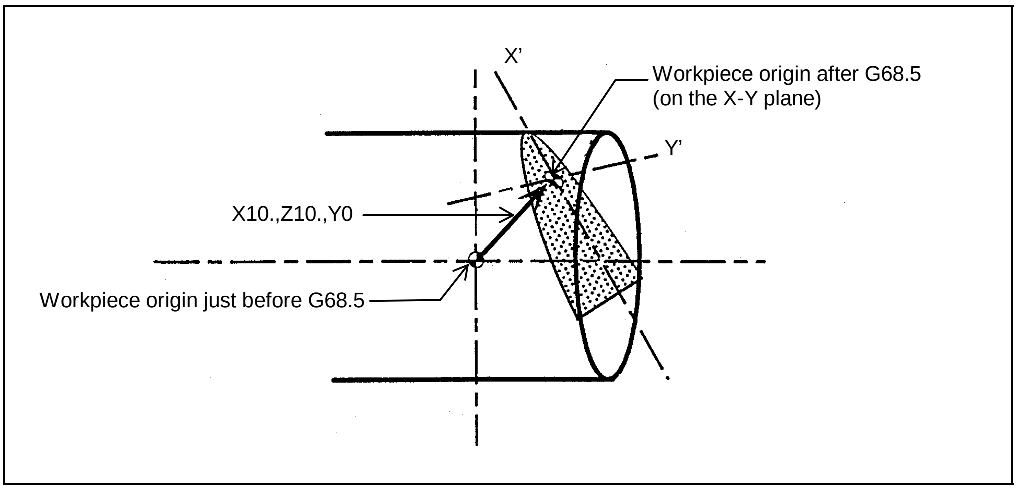


After the selection of the G17 (X-Y) plane, the workpiece origin is shifted to the point
(X, Z, Y) = (10, –5, 0)
and the plane is rotated by 30 degrees on the Y'-axis. The new coordinate system (X', Y', Z') has thus been established.

- 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 T0303.8;
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 T0505.8; ................................. 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;
%
```


X'
Workpiece origin after G68.5
(on the X-Y plane)
Y'
X10.,Z10.,Y0
Workpiece origin just before G68.5

Return to Main Menu

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 inavailable 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 3: `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			Group	Function
A	B	C		
G00	G00	G00	01	Rapid positioning
G01	G01	G01	01	Linear interpolation
G02	G02	G02	01	Circular interpolation, CW
G03	G03	G03	01	Circular interpolation, CCW
G02	G02	G02	01	Helical interpolation, CW
G03	G03	G03	01	Helical interpolation, CCW
G04	G04	G04	00	Dwell
G09	G09	G09	00	Exact stop
G10	G10	G10	00	Data setting mode
G11	G11	G11	00	Data setting mode cancel
G17	G17	G17	02	X-Y plane section
G18	G18	G18	02	Z-X plane section
G19	G19	G19	02	Y-Z plane section
G20	G20	G20	06	Inch command
G21	G21	G21	06	Metric command
G22	G22	G22	04	Stored stroke check ON
G23	G23	G23	04	Stored stroke check OFF
G32	G33	G33	01	Thread cutting
G34	G34	G34	01	Variable lead thread cutting
G40	G40	G40	07	Tool diameter/nose radius compensation OFF
G41	G41	G41	07	Tool diameter/nose radius compensation, left
G42	G42	G42	07	Tool diameter/nose radius compensation, right
G60	G60	G60	00	Unidirectional positioning
G61	G61	G61	13	Exact stop mode
G62	G62	G62	13	Automatic corner override mode
G64	G64	G64	13	Cutting mode
G65	G65	G65	00	Macro call
G66	G66	G66	14	Modal macro call
G67	G67	G67	14	Modal macro call cancel
G69.5	G69.5	G69.5	16	Program coordinate system rotation mode cancel
G80	G80	G80	09	Hole machining cycle cancel
G83	G83	G83	09	Face drilling cycle
G84	G84	G84	09	Face tapping cycle
G84.2	G84.2	G84.2	09	Face synchronous tapping cycle
G85	G85	G85	09	Face boring cycle
G87	G87	G87	09	Longitudinal drilling cycle
G88	G88	G88	09	Longitudinal tapping cycle
G88.2	G88.2	G88.2	09	Longitudinal synchronous tapping cycle
G89	G89	G89	09	Longitudinal boring cycle
G90	G77	G20	09	Outside/inside diameter turning cycle
G92	G78	G21	09	Thread cutting cycle
G94	G79	G24	09	Face turning cycle
G96	G96	G96	17	Constant peripheral speed control ON
G97	G97	G97	17	Constant peripheral speed control OFF
G98	G94	G94	05	Feed per minute
G99	G95	G95	05	Feed per revolution
—	G90	G90	03	Absolute dimension
—	G91	G91	03	Incremental dimension
—	G98	G98	10	Return to initial level
—	G99	G99	10	Return to R-point level

Return to Main Menu

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
G36: Preparation for measurement
G37: Measurement computing

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)

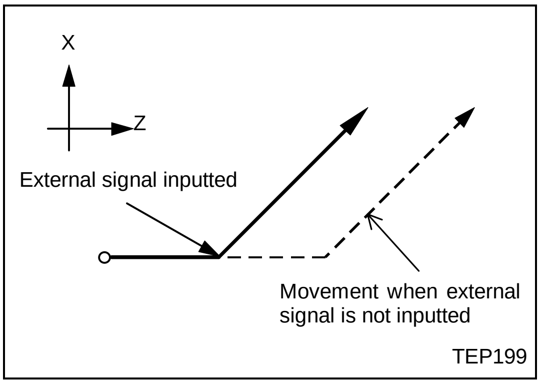
3. Detailed description

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/deceleration 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

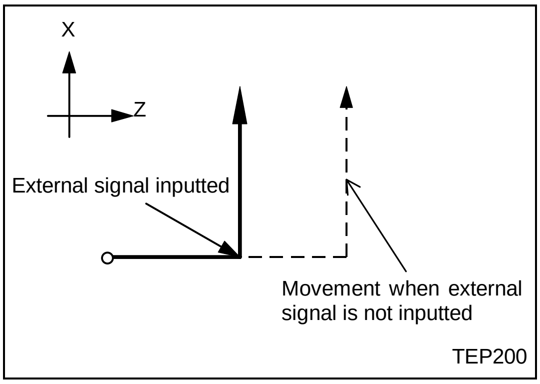
Example 1: When the next block is an incremental value command

G31 Z1000 F100;
G01 U2000 W1000;



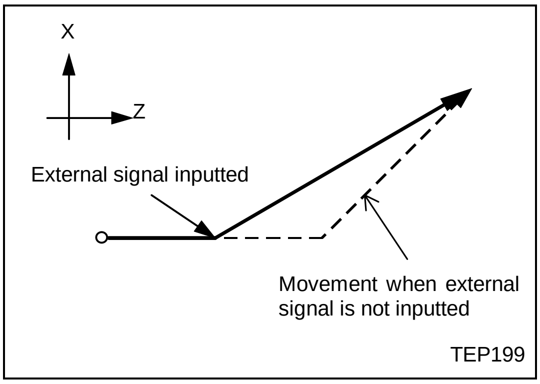
Example 2: When the next block is a one axis move command with absolute value

G31 Z1000 F100;
G01 X1000;



Example 3: When the next block is a two axes move command with absolute value

G31 Z1000 F100;
G01 X1000 Z2000;



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:

$$\delta_0 = \frac{F}{60} \times Tp + \frac{F}{60}(t_1 \pm t_2) = \underbrace{\frac{F}{60} \times (Tp + t_1)}_{\delta_1} \pm \underbrace{\frac{F}{60} \times \quad t_2}_{\delta_2}$$

δ_0 : Amount of coasting (mim)

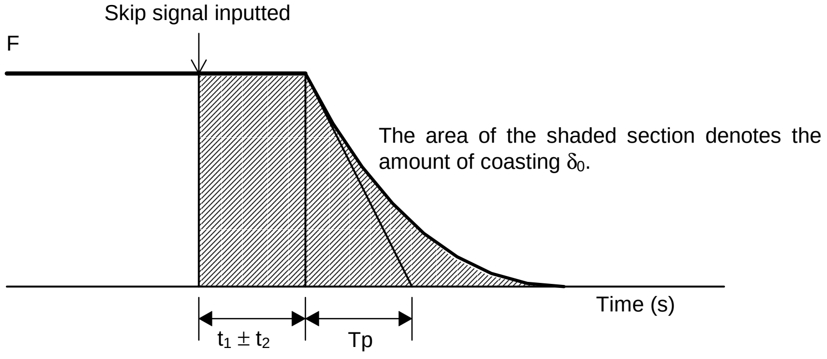
F : G31 skip rate (mm/min)

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)

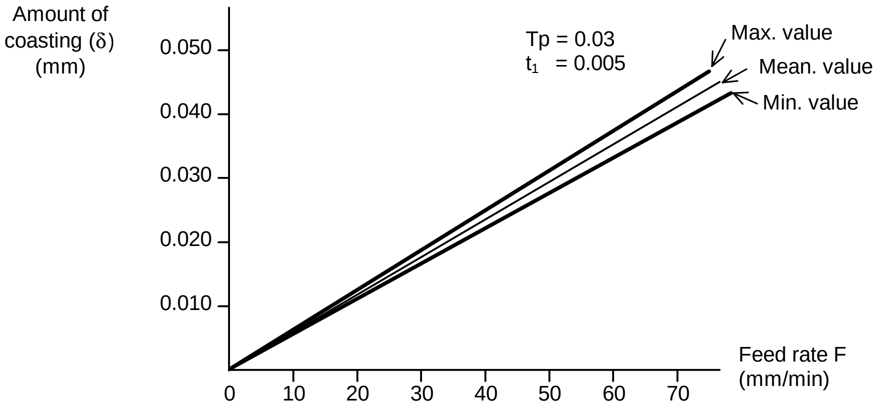
When using command G31 for measurement purposes, measured data δ_1 can be corrected. Such corrections, however, cannot be performed for δ_2.


Skip signal inputted
F
The area of the shaded section denotes the amount of coasting δ0.
Time (s)
t1 ± t2
Tp

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

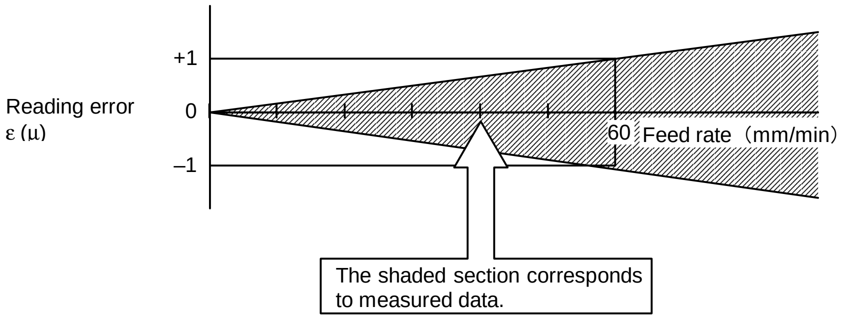
16-1-3 Skip coordinate reading error

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)



The shaded section corresponds to measured data.

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 previously in section 16-1-2. 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 Main Menu

16-2 Preparation for Measurement: G36

1. Function and purpose

It gives measuring mode, measurement target value and offset number (tool number).

2. Programming format

G36 (G36.5) X(Z)_ R_ K_ P_ (T_) Q_ A_ I_ D_ L_ ;

X(Z) : Target value X (Z)
R : Sensor radius
K : Tolerance
P : Offset No.
Q : Measuring mode
A : Change of the sign of offset data
A = 0: Sign is not reversed.
A ≠ 0: Sign is reversed.
I : Reference position (diametral value)
T : Tool number
D : Tool breakage detection data of tool tip measurement
D = 0: Tool breakage detection is not performed.
D ≠ 0: Tool breakage detection is performed.
L : The number of measrurements in C/Z offset measurement
1: One point is measured.
2: Two points are measured.

3. Detailed description

1. G36.5 is used for Standard mode.
2. Sensor radius is given by its radius value. It should be noted that a sign is put or that the sensor radius is not specified depending on measuring mode.
3. Tolerance gives the range of ± error for the target value.

Tolerance K	—— Target dimension

4. Measuring mode includes the following six types.
   Q = 0: Measuring mode of difference between two points
   Q = 1: Measuring mode of difference from reference point
   Q = 2: Measuring mode of tool tip
   Q = 3: External measuring mode
   Q = 4: Measuring mode of Z offset
   Q = 5: Measuring mode of C axis phase
5. G36 (G36.5) does not involve movement.
6. Addresses X and Z cannot be commanded together except in the measuring mode of tool tip. If commanded together, the command entered later is valid.
7. Target value must be given by absolute command. Executing incremental command causes an alarm (746 “ABSOLUTE INPUT REQUIRED (G36)”).
8. ommanding address T and address P together causes an alarm (707 “ILLEGAL FORMAT”).
9. Commanding a tool number with address T causes the measuring result to be added to the wear compensation amount in the tool data of the specified number.

10. When address D is specified in tool tip measurement, it is concluded that the tool has served its life (the tool is broken) if it does not come in contact with the sensor. (In this case an alarm is not given.)

4. Supplement

A. Comparison between EIA - G36 (G36.5) and MAZATROL measurement

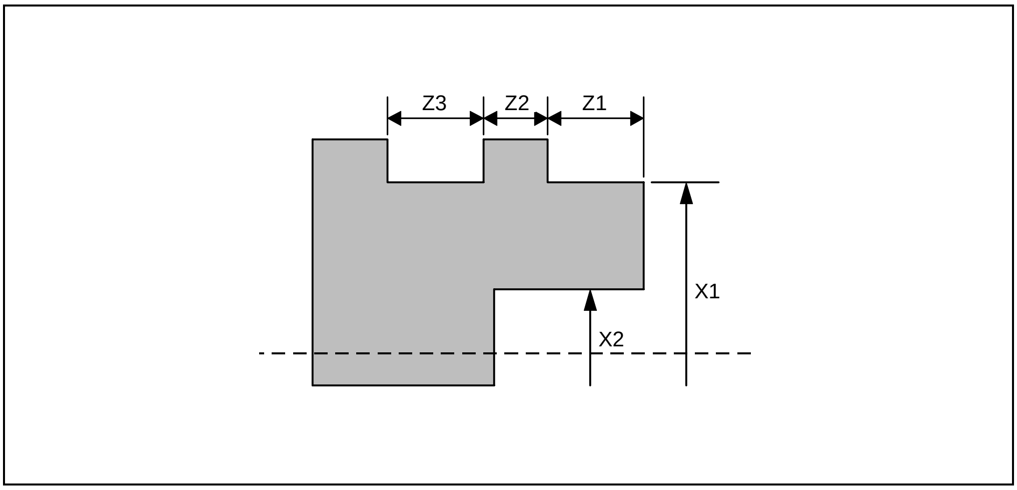
MAZATROL		EIA/ISO program											
Measuring mode	Measuring part	G	X	Z	C	I	K	R	P or T	Q	L	D	A
DIA	OUT	36 (36.5)	○	\	\	\	○	+R	○	0			◊
	IN	36 (36.5)	○	\	\	\	○	–R	○	0			◊
STR	OUT	36 (36.5)	\	○	\	\	○	\	○	0			◊
	IN	36 (36.5)	\	○	\	\	○	\	○	0			◊
GRV	OUT	36 (36.5)	\	○	\	\	○	–R	○	0			◊
	IN	36 (36.5)	\	○	\	\	○	–R	○	0			◊
WID	OUT	36 (36.5)	\	○	\	\	○	+R	○	0			◊
	IN	36 (36.5)	\	○	\	\	○	+R	○	0			◊
DIS	OUT	36 (36.5)	○	\	\	○	○	\	○	1			◊
	IN	36 (36.5)	○	\	\	○	○	+R	○	1			◊
TOL	#1	36 (36.5)	○	○	\	\	○	\	○	2		◊	◊
	#2	36 (36.5)	○	○	\	\	○	\	○	2		◊	◊
	#3	36 (36.5)	○	○	\	\	○	\	○	2		◊	◊
	#4	36 (36.5)	○	○	\	\	○	\	○	2		◊	◊
EXT	X	36 (36.5)	○*1	\	\	\	○	\	○	3			◊
	Z	36 (36.5)	\	○*2	\	\	○	\	○	3			◊
ZOF	Z	36 (36.5)	\	○	\	\		\	\	4	2		◊
COF	OUT#2	36 (36.5)	\	\	○	\		+R	\	5	2		◊
	FCE#2	36 (36.5)	\	\	○	\		–R	\	5	2		◊
	OUT#1	36 (36.5)	\	\	○	\		+R	\	5	1		◊
	FCE#1	36 (36.5)	\	\	○	\		–R	\	5	1		◊

- The blanks in EIA/ISO program column indicates that specifying the addresses is pointless.
- The value of address marked with ◊ is taken as 0 if omitted.
- indicates the data for offset direction change, and the offset data is normally compensated if “A” is not specified. If a value other than 0 is specified, the sign of measuring result is reversed for compensation.
- indicates the data for tool breakage detection, and tool breakage detection is not performed if “D” is not specified. If a value other than 0 is specified, tool breakage detection is performed.
- For +R, the sensor radius is set by “+”, and for –R, it is set by “–”.
- For *1, the data of X gives only information about the axis, and for *2, that of Z gives only information about the axis.
- For the axis setting of TOL, only one axis can be specified. However, in that case the specified axis is only compensated.
- For address L of COF, the number of measurements (1 or 2) is specified. For the method of internal identification, L<2 is treated as 1 and L ≥ 2 is treated as 2.

Return to Main Menu

B. Method of giving a target value in each measuring mode

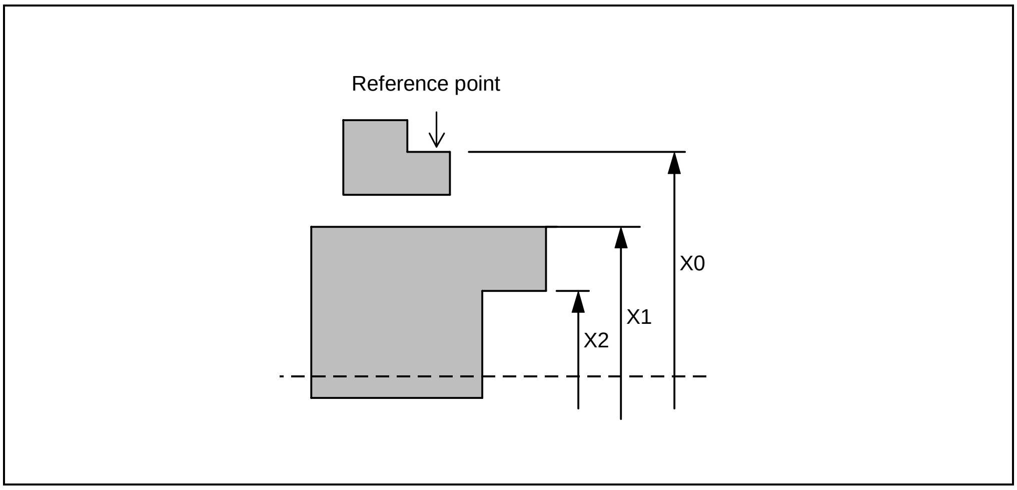
1) Q = 0: Distance calculation between two points



X = (X1, X2) : Specified by diameter value
Z = (Z1, Z2, Z3) : Specified by radius value

- X and Z cannot be specified together.

2) Q = 1: Distance calculation from reference point

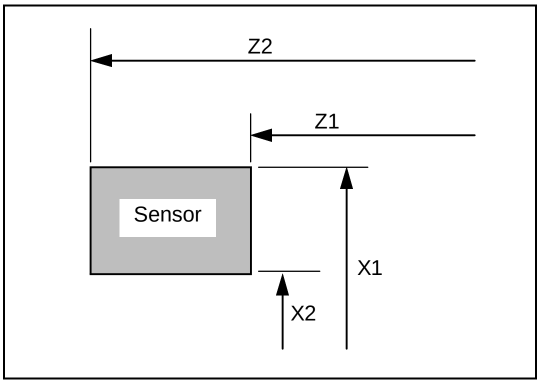


X = (X1, X2) : Specified by diameter value
I = X0 : Specified by diameter value

- This mode cannot be used for Z-axis.

3) Q = 2 : Tool tip measurement

The machine position (X1, X2, Z1, Z2) of a sensor brought into contact depending on the type of measuring tools is given as a target value.



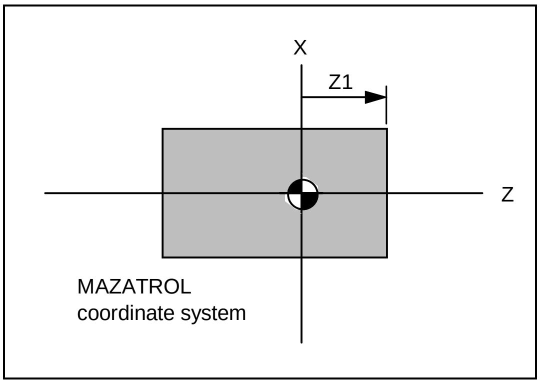
X = (X1, X2) : Specified by diameter value (machine position)
Z = (Z1, Z2) : Specified by radius value (machine position)

- X and Z can be specified together.

4) Q = 3 : External measurement

X and Z give only information about axis respectively, and a command value is pointless.

5) Q = 4 : Z offset measurement

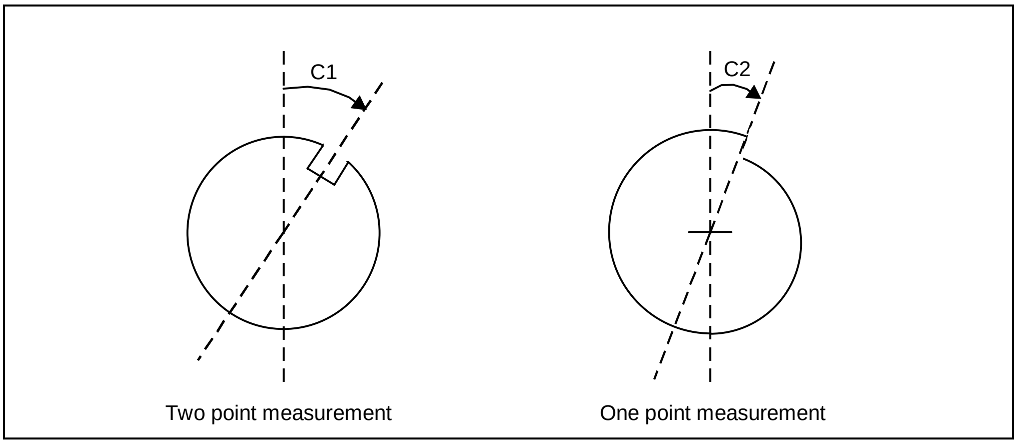


Z = Z1 (Target value): Z coordinate of the measured point.

- Z1 refers to the Z coordinate of the measured point in the MAZATROL coordinate system to be established. Setting of “0” establishes a system where the origin of Z coordinates is set on the measured position.
- One-point measurement:
  Setting of Z offset from the measured point.
- Two-point measurement:
  Setting of Z offset from the point of the second measurement.

6) Q = 5: C axis phase measurement

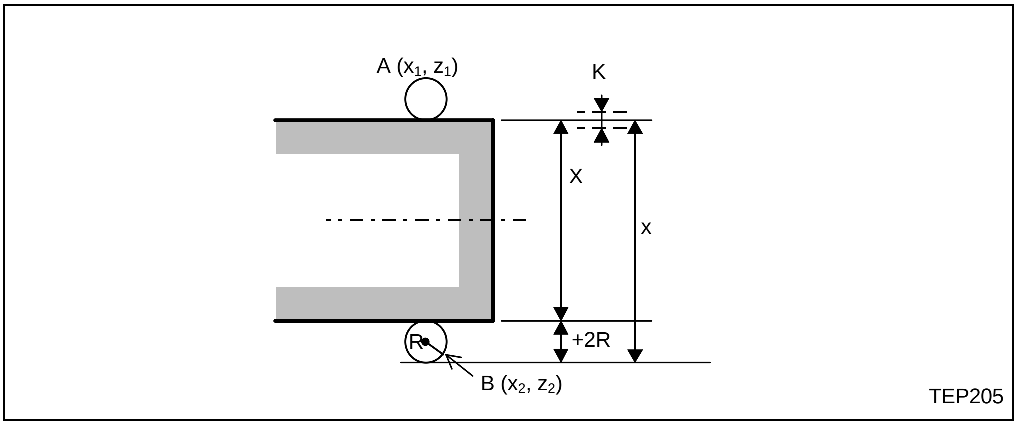
C1 and C2 give the shift amount from program coordinate zero point with a radius value.



C. G36 setting examples

1) Both end contact measurement mode (Q0)

```
G36 X100. K0.3 P8 Q0 R3. A0;
```

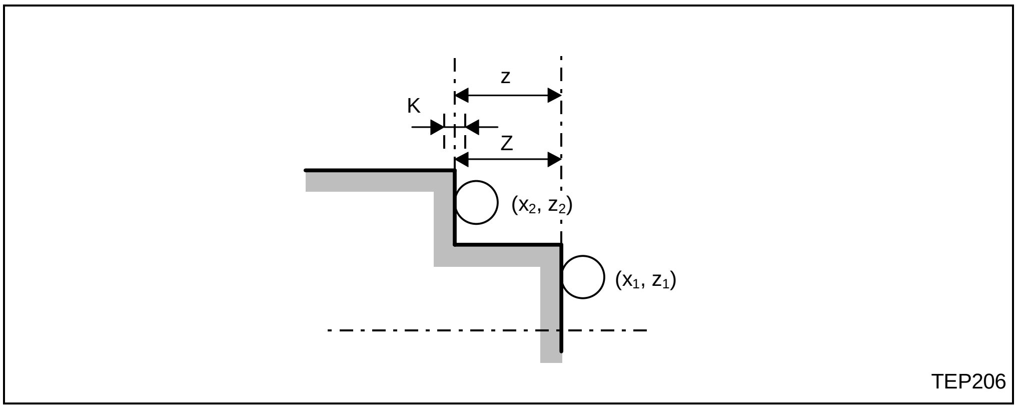


$x = | x_1 – x_2 |$: Actual measurement data

Measurement data = x – 2R

where 2R should be calculated by diametral data (2 × R) for program preparation.

```
G36 Z20. K0.4 P28 Q0 R0. A0;
```

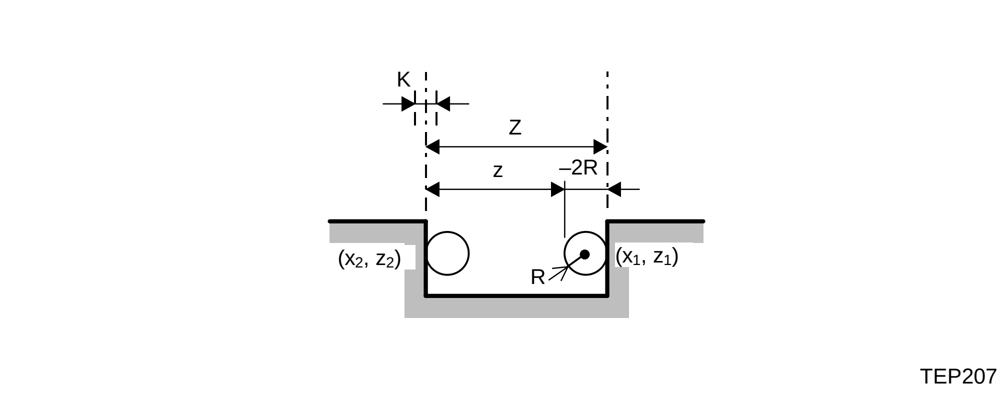


$z = | z_1 – z_2 |$: Actual measurement data

Measurement data = z

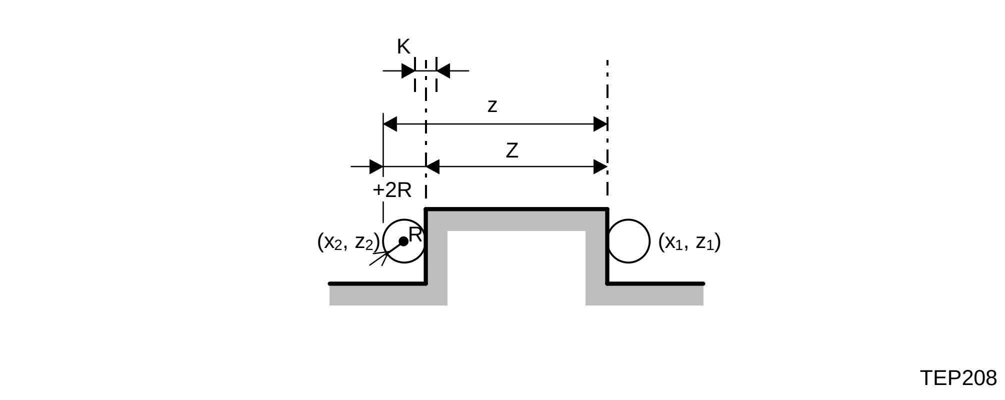
where sensor R is set at 0.

```
G36 Z20. K0.4 P38 Q0 R–3. A0;
```



$z = | z_1 – z_2 |$: Actual measurement data
Measurement data = z – (–2R)
where sensor R is set at minus data.

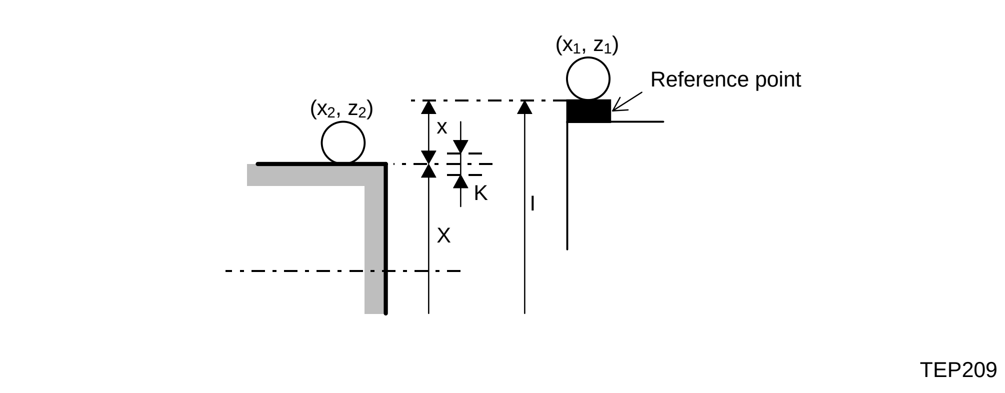
```
G36 Z20. K0.4 P38 Q0 R3. A0;
```



$z = | z_1 – z_2 |$: Actual measurement data
Measurement data = z – (+2R)

2) Reference point contact measurement mode (Q1)

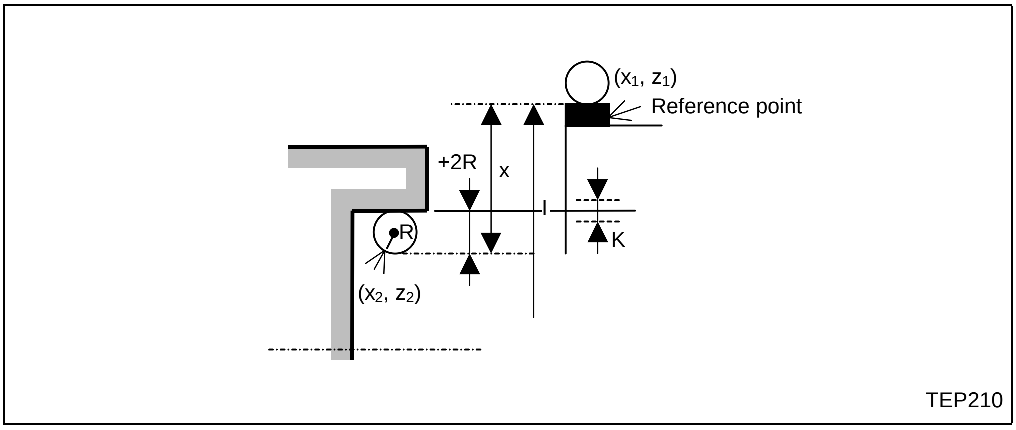
```
G36 X100. I200. K0.3 P23 Q1 R0. A0;
```



I: Reference position (position of reference point on workpiece coordinate system)
$x = x_1 – x_2$: Actual measurement data (diametral data)
Measurement data = I – x

- Set R (sensor radius) at 0 (R = 0) in the example shown above because no sensor radius error will be caused.

```
G36 X100. I200. K0.3 P53 Q1 R3. A0;
```



TEP210

I: Reference position (position of reference point on workpiece coordinate system)

$x = x_1 - x_2$: Actual measurement data (diametral data

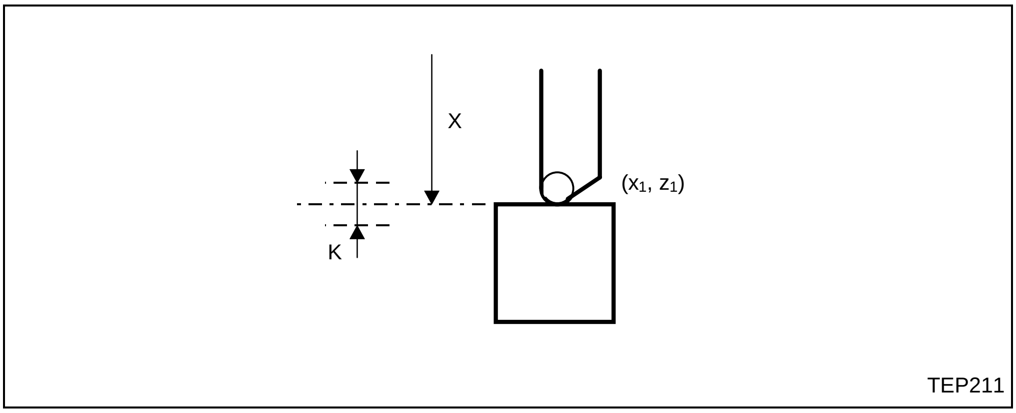
Measurement data = I – (x + 4R)

- Set R (sensor radius) in the example shown above because sensor radius error would be caused.

3) Tool tip contact measurement mode (Q2)

[TOL-X]

```
G36 X–100. K0.3 P30 Q2 A0;
```



TEP211

X : Target data (machine coordinate value)

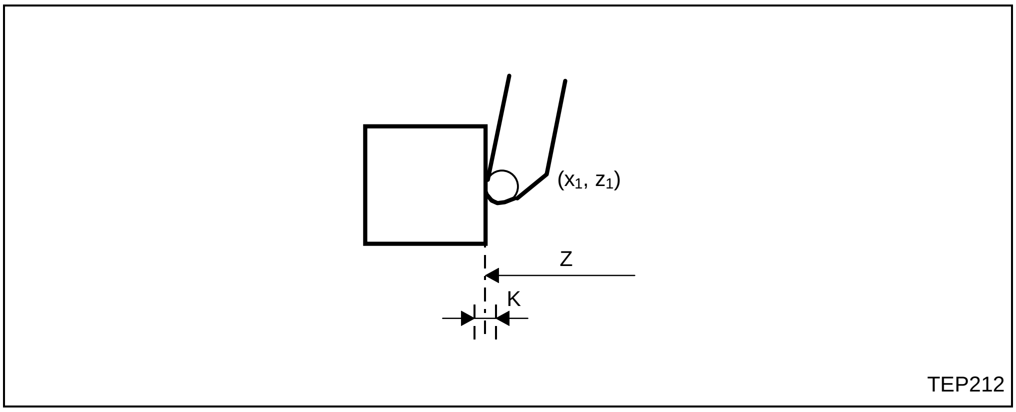
K : Tolerance

T : Offset No. for the tool to be measured

Measurement data = x_1: Actual measurement data (diametral data)

[TOL-Z]

```
G36 Z-20. K0.4 P32 Q2 A0;
```

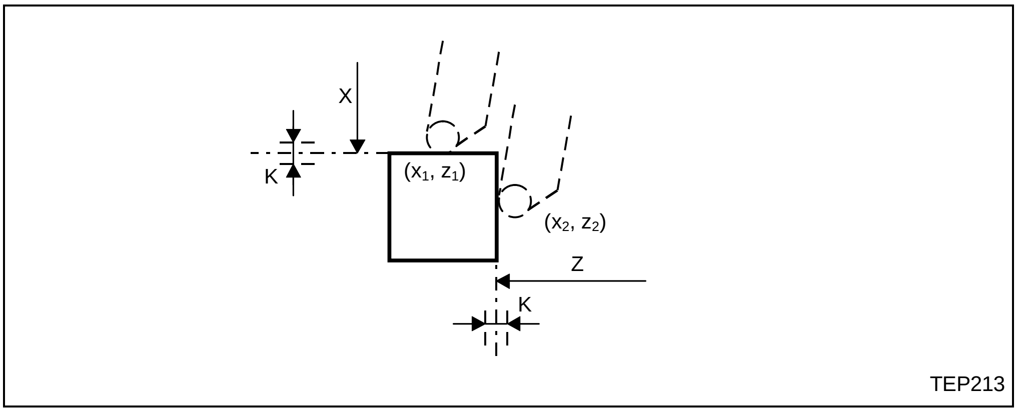


Z : Target data (machine coordinate value)
K : Tolerance
P : Offset No. for the tool to be measured

Measurement data = x_1: Actual measurement data (diametral data)

[TOL-XZ]

```
G36 G36 X–100. Z–20. K0.4 P61 Q2 A0;
```



X : Target data (machine coordinate value)
Z : Target data (machine coordinate value)
K : Tolerance
P : Offset No. for the tool to be measured

Measurement data (X) = x_1 : Actual measurement data
Measurement data (Z) = z_2 : Actual measurement data

<u>Remarks on tool-tip measurement</u>

- For tool tip measurement, compensation No. of the tool to be measured and compensation No. specified by program must be identical.
- Tool tip must be contacted with the sensor in the order X and Z in the program for simultaneous tool tip measurement in X and Z axes. If contact is made in the order of Z and X, measurement data may not be correct.

Return to Main Menu

16-3 Measurement Computing : G37

1. Function and purpose

The measurement data is calculated by G37 (G37.5) command, and an error from the target data given by G36 (G36.5) is determined. When the amount of error is within the tolerance, it is added to the offset data of the specified offset No. or the setup data of work No. in execution.

2. Programming format

G37 (G37.5) ;   It is usually given independently.

3. Operation

Axis is not moved.

When P is specified by G36 (G36.5):
The offset data of the specified offset No. is compensated.
When T is specified by G36 (G36.5):
The wear compensation amount of the specified tool is compensated.
Z offset, C-axis phase :
Z offset of the setup data of work No. in execution and C-axis phase are rewritten.

4. Remarks

1. G36.5 and G37.5 are used in Standard mode.
2. When tool path is checked, offset data and setup data are not changed.
3. Next block is not processed until the operation of G37 (G37.5) is completed.

<Arithmetic operation>

Measure-ment mode	Measurement axis X	Measurement axis Z
Q 0	Actual measurement data x = | $x_1 – x_2$ | Measurement data = x –2R Compensation data = Target data X – measurement data New offset data = Previous offset data + compensation data* * When A = 0 Add compensation data. When A = 1 Substract compensation data.	Actual measurement data z = | $z_1 – z_2$ | Measurement data = z – 2R Compensation data = Target data Z – measurement data New offset data = Previous offset data – compensation data* * When A = 0 Substract compensation data. When A = 1 Add compensation data.
Q 1	Actual measurement data x = $x_1 – x_2$ Measurement data = I (Reference position) – (x – 2R) Compensation data = Target data X – measurement data New offset data = Previous offset data + compensation data* * When A = 0 Add compensation data. When A = 1 Substract compensation data.	————
Q 2	Actual measurement data x = x_1 Measurement data = Actual measurement data Compensation data = Target data X – measurement data New offset data = Previous offset data – compensation data* * When A = 0 Substract compensation data. When A = 1 Add compensation data.	Actual measurement data z = z_1 (z_2 when X and Z are measured simultaneously) Measurement data = Actual measurement data Compensation data = Target data Z – measurement data New offset data = Previous offset data – compensation data* * When A = 0 Substract compensation data. When A = 1 Add compensation data.
Q 3	Compensation data = Input data from external measuring unit New offset data = Previous offset data + compensation data* * When A = 0 Add compensation data. When A = 1 Substract compensation data.	Compensation data = Input data from external measuring unit New offset data = Previous offset data + compensation data* * When A = 0 Add compensation data. When A = 1 Substract compensation data.

(x_1, z_1): First sensor contact measurement point position.

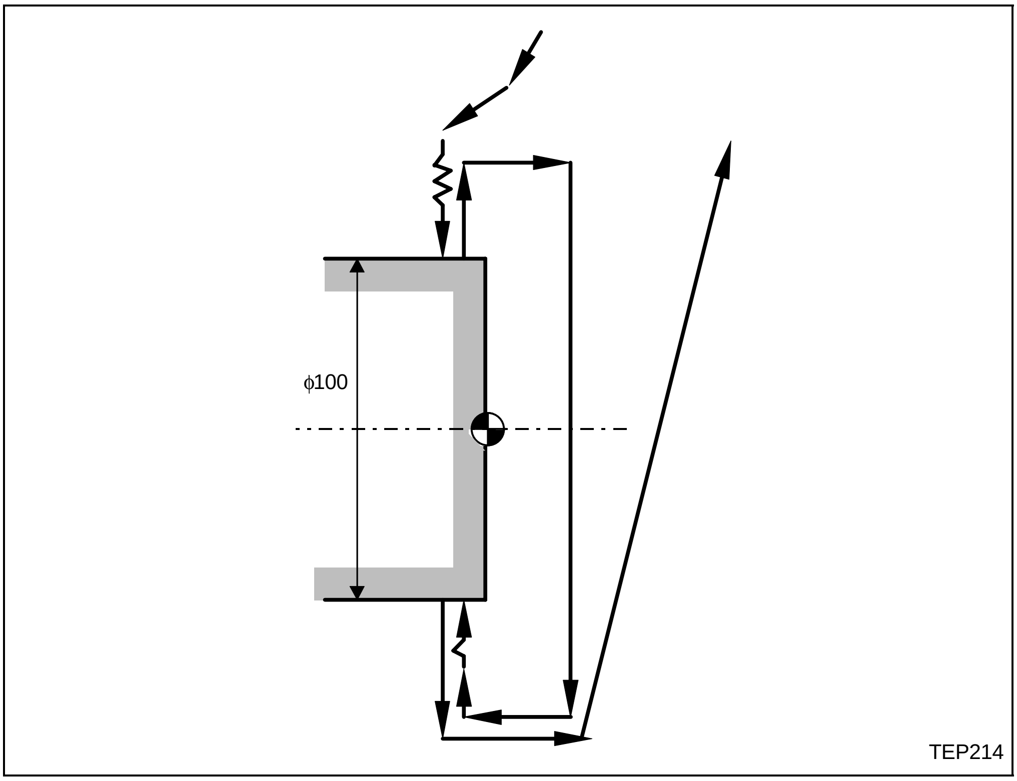
(x_2, z_2): Second sensor contact measurement point position.

Return to Main Menu

16-4 Measurement Program Example

Example 1: Outside diameter measurement

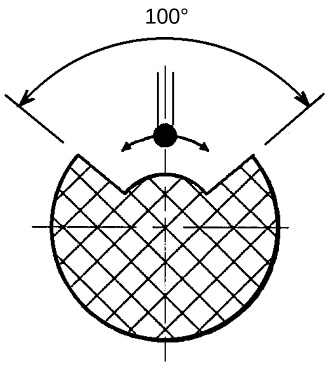
`N001 G50 X200.Z300.;`	Setting of measurement coordinate system
`N002 T0100;`	Selection of measurement sensor
`N003 G36 X100.K0.3 P8 R3.Q0 A0;`	Setting of target data
`N004 G00 X110.Z0.;`	Approach to measurement point
`N005 G01 G98 X108.Z–10.F120;`	
`N006 G31 X104.F25;`	Contact with measurement point
`N007 G00 X110.;`	Approach to next measurement point
`N008 Z5.;`	
`N009 X–110.;`	
`N010 Z–10.;`	
`N011 G01 X–108.F120;`	
`N012 G31 X–104.F25;`	Contact with measurement point
`N013 G00 X–110.;`	Escape to safety position
`N014 Z5.;`	
`N015 G37;`	Calculation and compensation of measurement
`N016 G28 X200.Z300.;`	



Return to Main Menu

Example 2: C-axis offset measurment

```
G00 G98;
M200;                     Selection of milling mode
G28 UWH;                  Zero-point return
T0100;
G00 X100.;
G31 X72.F200;           ⎧ Setting of C-axis measurement mode.
G01 U2.F50;             ⎨ Target value: 0°, Sensor radius: 1.5 mm,
G36.5 C0.R1.5 Q5 L2;    ⎪ Number of measurement points: 2.
                        ⎩ (C for absolute, and H for incremental C-axis position)
G31 H–180.F500;           Trial measurement
G01 H2.F50;
G31 H–3.F40;              Measurement (1st point)
G01 H2.F50;
G01 H70.F3500;            Transition
G31 H30.F40;              Measurement (2nd point)
G01 H–3.F50;
G00 X200.;
G37.5;                    Calculation
G28 UW;
M202;
M30;
%
```



16-5 Automatic Tool Offset: G36/G37 (Standard Mode)

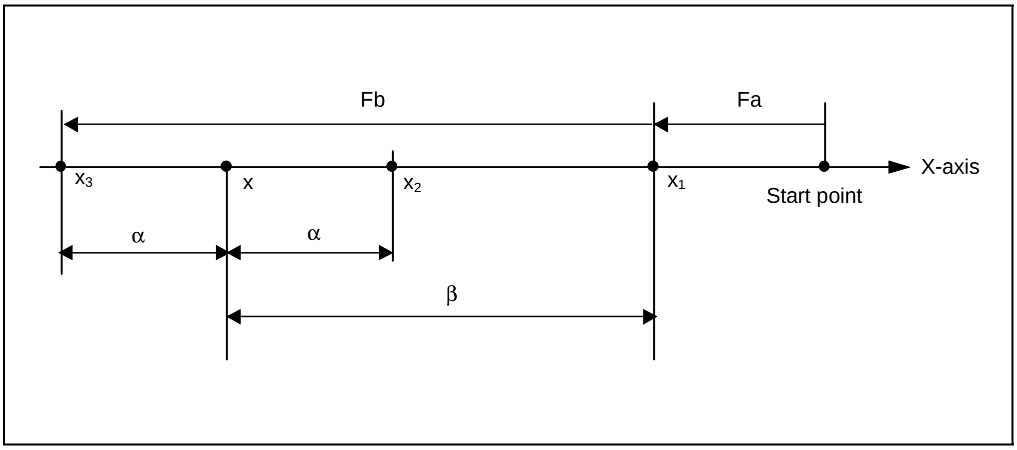
1. Function and purpose

A command value from measuring start position to measuring position is given to move the tool in the direction of the measuring position, and after the tool reaches the sensor, the machine is stopped. The difference between the coordinate value and the coordinate value of commanded measuring position is automatically calculated by NC, which is used as the offset data of the tool. When offset is already applied, the tool is moved in the direction of measuring position with offset applied. When offset is still required as the result of measurement and calculation, offset is given further to the current tool offset data.

2. Programming format

G36 Xx;
G37 Zz;

x : Target data of X-axis (Absolute data is specified)
z : Target data of Z-axis (Absolute data is specified)



α	:	Measuring range (Set by parameter **B237**. For HD2 side, **C237**.)
β	:	Decelerating range (Set by parameter **B239**. For HD2 side, **C239**.)
Fa	:	Rapid feed rate
Fb	:	Measuring speed (Set by parameter **B148**. For HD2 side, **C148**.)
x	:	Target point
x_1	:	x + β
x_2	:	x + α
x_3	:	x – α

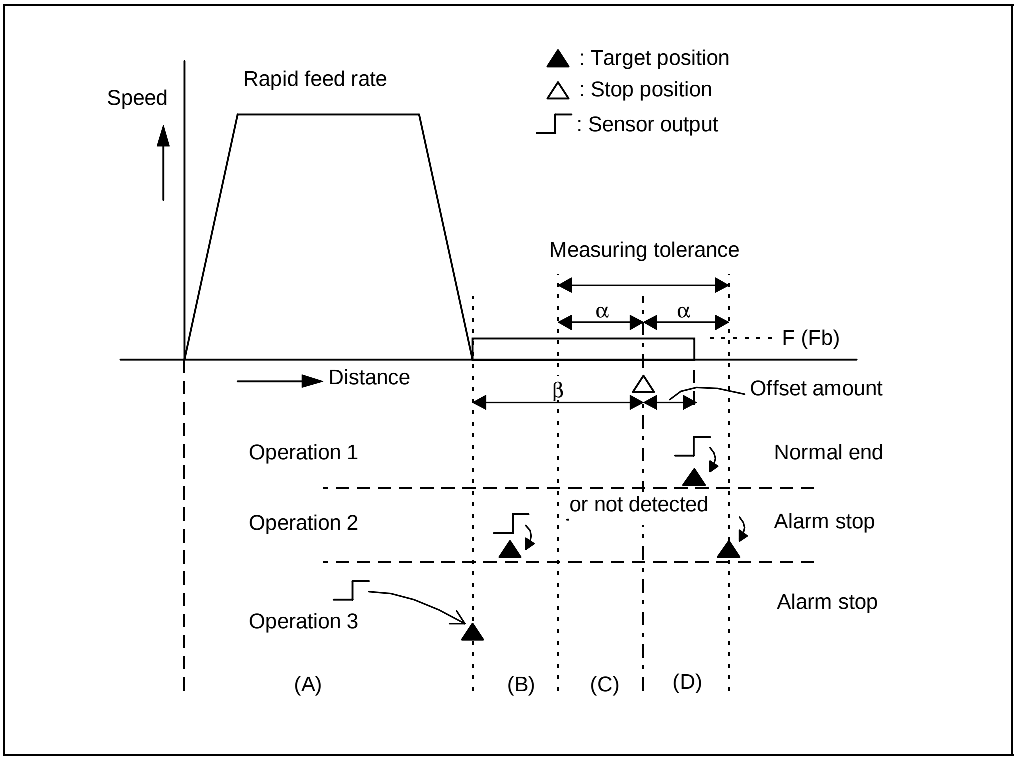
The tool moves from the start point to point x_1 at a rapid feed rate (Fa), and it decelerates from point x_1 at a rate of measurement (Fb). And when the tool reaches the sensor while it moves between point x_2 and point x_3, the machine is stopped to set the difference between the target data and the measured data at the tool shape offset amount.
And when the tool reaches the sensor while it moves between point x_1 and x_2 or the tool moves over x_3, it causes an alarm.

It is the same with Z-axis, and the parameters are as follows:

α	:	Measuring range (Set by parameter **B238**. For HD2 side, **C238**.)
β	:	Declerating range (Set by parameter **B240**. For HD2 side, **C240**.)
Fb	:	Measuring speed (Set by parameter **B148**. For HD2 side, **C148**.)

3. Detailed description

1. Operation by G36/G37 command



2. Sensor signal (measuring position arrival signal) is used in common with skip signal.
3. When F command and measuring speed of parameter are 0, it provides a feed rate of 1 mm/min (or 0.1 in./min).
4. The tool moves at synchronous feed (mm/rev or in./rev) in synchronous feed mode.
5. Renewed offset amount is valid from the block following G36/G37 command.
6. The machine position coordinate is read by the detection of a sensor signal. The machine overshoots by servo dloop, and stops.

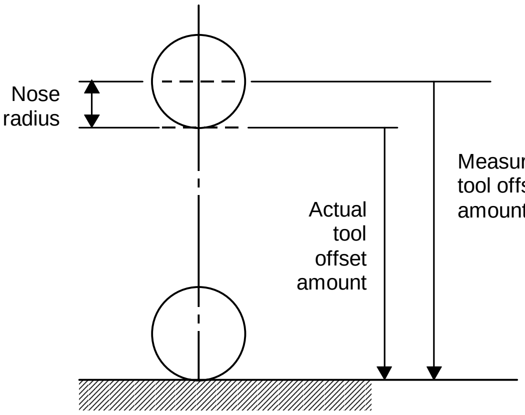
$$\text{Maximum overshoot [mm]} = \text{Measuring speed [mm/min]}\cdot\frac{1}{60}\cdot\frac{1}{\text{Position loop gain [sec}^{-1}\text{]}}$$

4. Notes

1. When G36/G37 is commanded by NC, which is not provided with optional automatic tool offset, it causes an alarm.
2. When T-code is commanded in the block of G36/G37, it causes an alarm. And when the offset No. of the T-code is 0, it also causes the alarm of item 3.
3. When T-code is not commanded prior to the block of G36/G37, it causes an alarm. And when the offset No. of the T-code is 0 even if the T-code is commanded, it causes an alarm.
4. When sensor signal is inputted beyond the measuring tolerance or when sensor signal is not detected, it causes an alarm. However, even if the sensor signal has turned on during rapid feed as shown in “operation 3” in the operating example of the above figure, it is taken as normal measurement if the area (B) is not set.
5. When manual interruption is executed during movement at measuring speed, the tool must always be returned to the position before the interruption for restart.
6. Data commanded in G36/G37 block and parameter setting data must fill the following condition.

   | Target point – start point | > β > α

7. When α is 0, the operation is normally ended only when the commanded measuring point is aligned with the detecting point of sensor signal. An alarm is given in all other cases.
8. When α and β are 0, it causes an alarm independently of sensor signal after positioning is accomplished at the commanded measuring point.
9. When | target point – start point | is less than measuring tolerance, it entirely provides the measuring tolerance.
10. When | target point – start point | is less than measuring speed movement distance, the tool moves entirely at the measuring speed.
11. When measuring tolerance is greater than measuring speed movement distance, the tool moves within the measuring tolerance at the measuring speed.
12. Nose radius compensation must always be cancelled before G36/G37 is commanded.
13. Even when provided with optional nose radius compensation, tool offset amount is calculated without considering the value of nose radius and tool tip number.
    Subtract the value corresponding to nose radius from the measured tool offset amount.



Return to Main Menu

- NOTE -

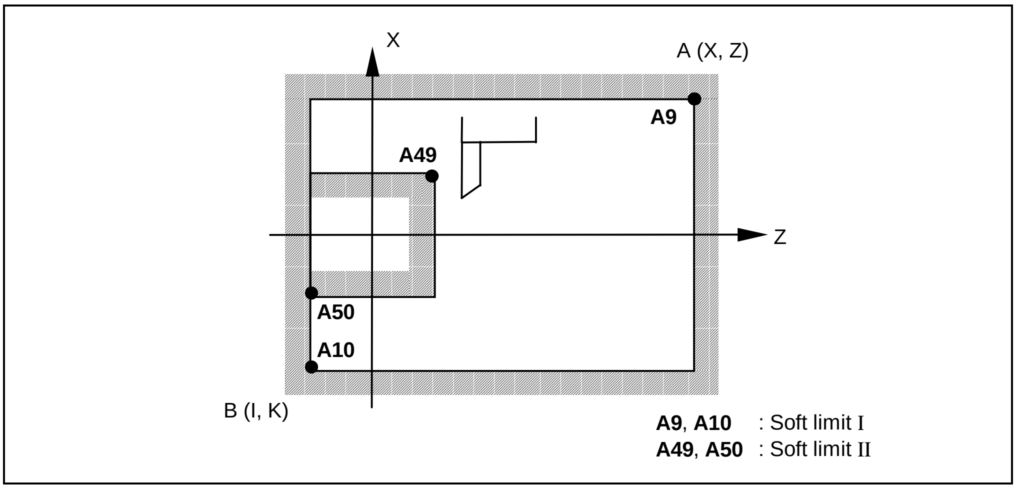
Return to Main Menu

17 PROTECTIVE FUNCTION

17-1 Stored Stroke Limit ON/OFF: G22/G23

1. Function and purpose

G22 X_ Z_ I_ K_;	Limit ON (X > I, Z > K)
G23;	Limit OFF


X
A (X, Z)
A9
A49
Z
A50
A10
B (I, K)
A9, **A10** : Soft limit I
A49, **A50** : Soft limit II

The above command specifies the area where a tool is forbidden to enter, and it stops the machine automatically before the boundary to the move command beyond the area. Values X, Z, I and K established by program provide the distance (least input increment) to points A and B using the machine zero point as 0. The value established by program is converted to least input increment, and it replaces the value established already. (Machine parameter **A49**/**A50**)
The inside or outside of the specified boundary provides the forbidden area. Parameter **B81** bit 6 selects either inside or outside as the forbidden area. (0: Inside 1: Outside)

2. Detailed description

1. When the machine goes nearly beyond the boundary (soft limit II), the function stops the machine and also gives an alarm display. To cancel the alarm, NC must be reset.
2. Soft limit II is effective also for machine lock.
3. Soft limit II function does not become effective immediately after NC is turned on. It becomes effective after reference point return is performed.
   However, if absolute position detecting system is provided as an option and if the absolute position detection is valid, the soft limit function becomes effective immediately after the power is turned on.
4. If any axis gives an alarm in automatic operation, a deceleration stop is applied to all the axes. The stop position is always placed before the forbidden area. The distance between forbidden area and stop position depends on feed rate.
5. Commanding G22 requires moving the tool beyond the forbidden area beforehand.

- NOTE -

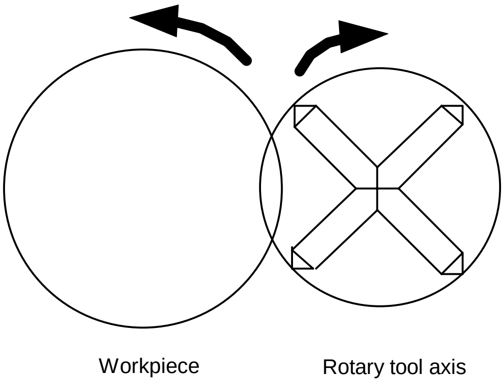
18 POLYGONAL MACHINING AND HOBBING

18-1 Polygonal Machining ON/OFF: G51.2/G50.2

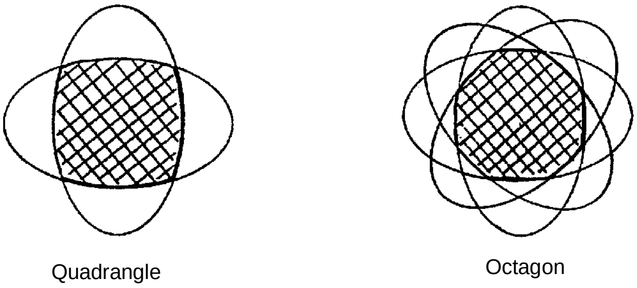
1. Function and purpose

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 ~ 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;
T1100;           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
```

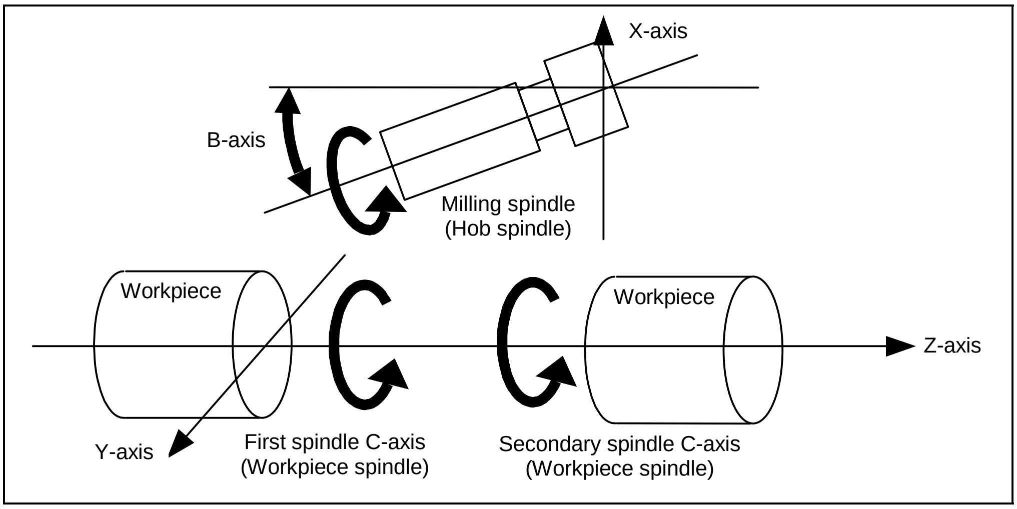
4. Notes

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).

18-2 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.
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.

Return to Main Menu

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`
`Q2.5R0;`
	Selection of the hob milling mode (with phase matching for zero shift angle).
	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

`M200;`	Selection of the milling mode for the 1st spindle.
`G112M200;`	Selection of the milling mode for the 2nd spindle.
`G112M203S0;`	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.
`G112S50;`	Specification of the hob spindle rotation at 50 min^{-1}.
`G00X18.;`	
`G01Z20.F10;`	
`G00X40.;`	
`   Z-5.;`	
`G113;`	Cancellation of the hob milling mode.
`G112M205;`	Milling spindle stop for the 2nd spindle.
`G112M202;`	Cancellation of the milling mode for the 2nd spindle.
`M202;`	Cancellation of the milling mode for the 1st 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.

- NOTE -

Return to Main Menu

19 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.

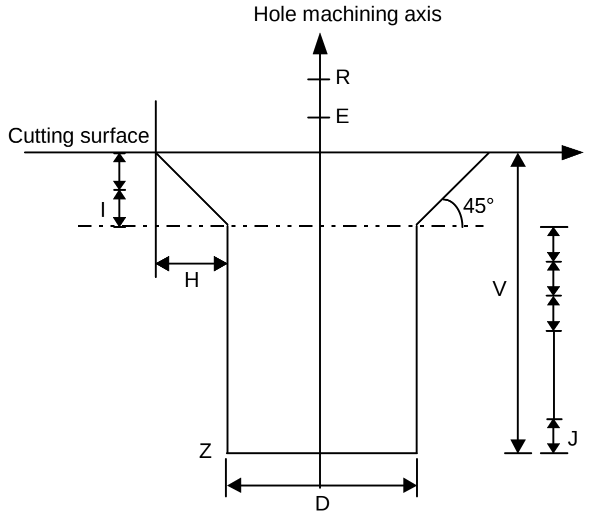
This cycle function is only available on machines equipped with the Y-axis control facility.

Note: Set one of the parameters (**K81** to **K88**) for macro-call G-codes as follows:
CODE = 130, NUMBER = 100009114.

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;
```



R:	Position of R-point
Z:	Position of hole bottom
D:	Hole diameter
T:	Tool diameter
V:	Hole depth
F:	Feed rate
H:	Chamfering amount
I:	Pitch 1
J:	Pitch 2
K:	Bottom finishing (0: No, 1: Yes, Others: Yes)
Q:	Machining direction (0: CW, 1: CCW)
E:	Position of 2nd R-point
M:	Hole machining axis (0: X, 1: Z or oblique)

TEP300

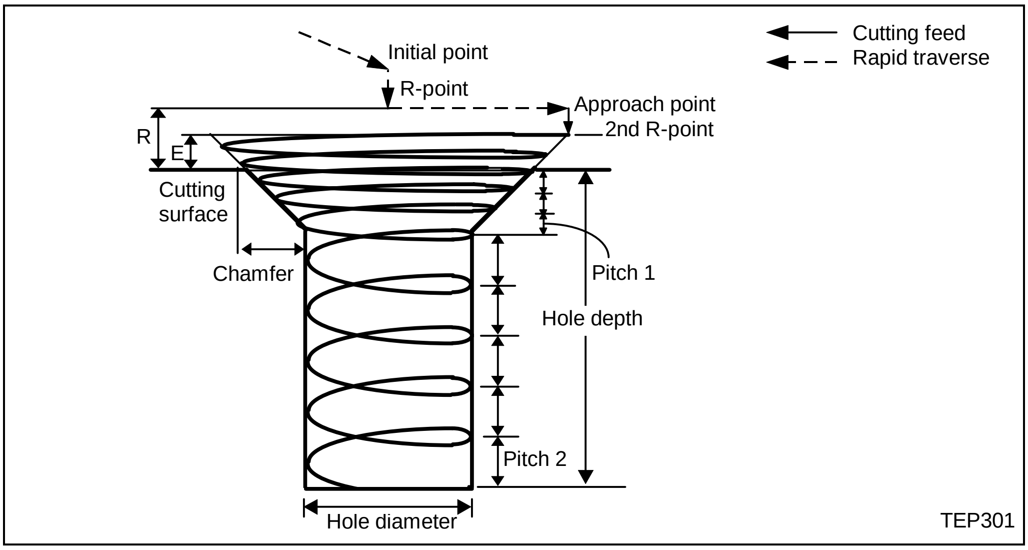
- The chamfering angle is fixed at 45°.
- Arguments D (hole diameter) and T (tool diameter) must satisfy the following condition:
  D ≥ T ≥ D/2.
- Argument K is used to select whether finishing is to be (K1) or not to be (K0) executed on the bottom of the hole.
- Set the hole position separately from 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.

3. Description of movement

A. Hole machining

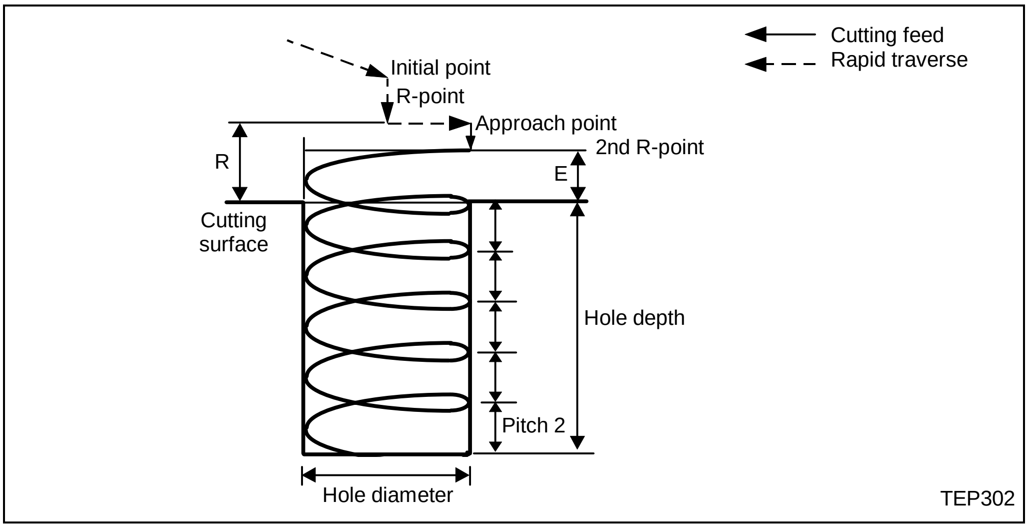
1. With chamfering

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

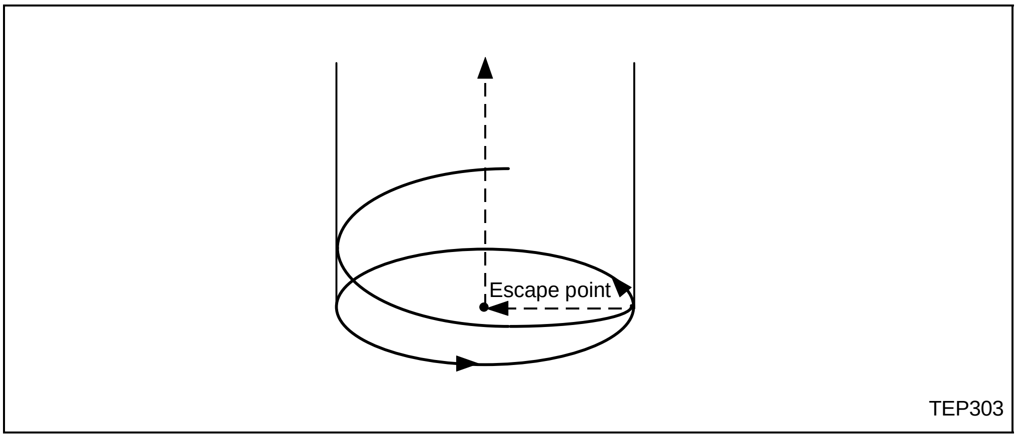
After moving from the current position to the R-point on the hole axis and then approaching through the hole radius and to a point on the 2nd R-point level, cylindrical machining is carried out from the top to the bottom by a circular-helical interpolation.



B. Movement on the bottom

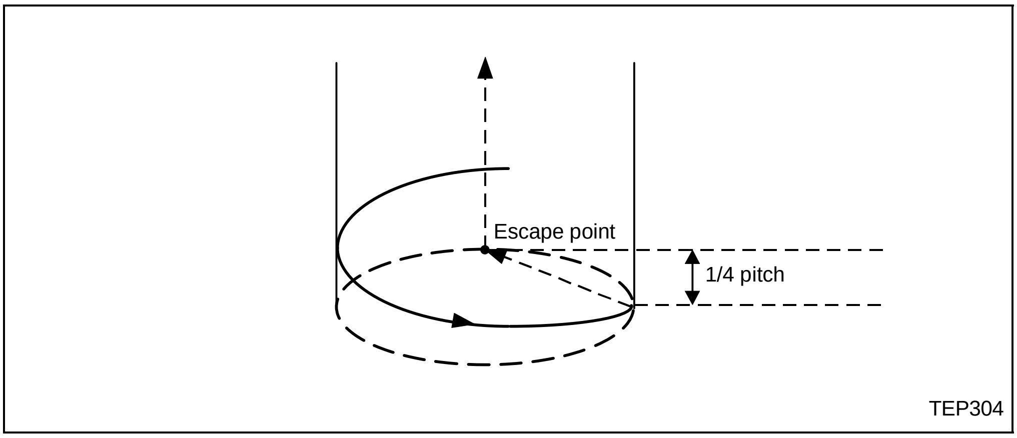
1. With bottom finishing

   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.



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.



- NOTE -

Return to Main Menu

20 TWO-LINE CONTROL FUNCTION

20-1 Two Process One Program: G109

1. Outline

When machining of different processes are performed by respective lines on a machine with two-lines of HD1 and HD2, two-lines can be controlled by a program.
The section from “G109L○;” to “%” or to “G109L*;” is used as a program of ○-line.

2. Programming format

G109 L_;

L = 1 : HD1
  2 : HD2

A line number is 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 single block operation, the stop can be performed after execution of G109 block. However, when the number specified by L belongs to that of counterpart line such as L2 in HD1 operation, the machine does not stop.
3. G109 can be specified in the same block as G code other than group 0. When specified in the same block as G code of group 0, 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.

Example:

```
G28 U W;   ] Common to HD1 and HD2
G109 L1;   ]
 ⋮         ] HD1
G109 L2;   ]
 ⋮         ] HD2
M30;       ]
G109 L1;   ]
 ⋮         ] HD1
M30;       ]
%          ] Common to HD1 and HD2
```

5. One block including more than 128 characters causes an alarm 707 “ILLEGAL FORMAT”.

20-2 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.
Axis address following G110 and HD number controlling the axis are specified.

2. Programming format

```
G110 X_ Z_ C_; .........................Cross machining control axis and HD number are specified,
                                     1 or 2
                                          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.
```

Specify the Z-axis on the 2. headstock side (INTEGREX) as follows:

Example:

```
G110 Z2;        ———— Selection of the 2nd headstock’s Z-axis
G00 Z100.;      ———— All the Z-axial commands between G110 and G111 are processed
   ⋮                 as those for the 2nd headstock side.
G111;           ———— Cancellation of G110
```

Specify the C-axis, or orientation, with an index capacity of 0.001° on the 2. headstock side (INTEGREX, SQT) 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 Main Menu

For a machine (INTEGREX, SQT) whose secondary spindle is equipped with the function for the C-axis control or 0.001-degree index (orientation), prepare a program as follows to use the C-axis settings on the **WORK OFFSET** display for the 2nd spindle (with the parameter **P77** being set to “1”):

Example:

```
G52.5;      ———————— MAZATROL coordinate system cancellation
M200;       ———————— Milling mode selection for 1st spindle
G28UWH;
T0101.1;
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
M300;       ———————— 2nd spindle selection
G112M200;   ———————— Milling mode selection for 2nd spindle
G110 C2;    ———————— Selection of 2nd spindle C-axis (or 0.001° index)
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
G112M202;   ———————— Milling mode cancellation for 2nd spindle
```

For a machine (INTEGREX, SQT) whose secondary spindle is equipped with the function for the C-axis control or 0.001-degree index (orientation), prepare a program as follows to use a fixed cycle for hole machining on the 2nd spindle side:

Example:

```
M300;                      ———————— 2nd spindle selection
G112M200;                  ———————— Milling mode selection for 2nd spindle
G110 C2;                   ———————— Selection of 2nd spindle C-axis (or 0.001° index)
G00 C0.;                   ———————— HD2 C-axis positioning
G112G87Z–5.0X5.0P0.2M210;— Clamping; Deep-hole drilling cycle
C45.;                      ———————— Unclamping, positioning, clamping; Deep-hole drilling cycle
C90.;                      ———————— Unclamping, positioning, clamping; Deep-hole drilling cycle
G112M212;                  ———————— Unclamping
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 a machine whose secondary spindle is of the 0.001-degree index (orientation) specifications or of the C-axis specifications are shown below. (INTEGREX-SY)

```
01234
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
```

Return to Main Menu

```
（MAIN SPINDLE SIDE)             1st spindle side machining program
M302                            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)
T0100.0                         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)
M300                            2nd spindle selection
G112M202                        2nd spindle turning mode
M200 (MAIN C-ON)                1st spindle mill-point machining mode
G00C#101                        1st spindle C-axis positioning (angle indexing)
G112M200(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)
G112M6                          2nd spindle chuck open
M540                            TRS-CHK mode
G110Z2                          2nd spindle side Z-axis selection
G00Z-686.                       2nd spindle side Z-axis positioning
G112M508                        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
G112M509                        2nd spindle M508 cancellation
G111                            2nd spindle side Z-axis selection revoking
M541                            TRS-CHK mode cancellation
G112M7                          2nd spindle chuck close
M6                              1st spindle chuck open
G112M202                        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)
M300                            2nd spindle selection (enter for machining at the 2nd spindle side)
T0300.1                         Tool selection
G98G97                          Feed per minute and cancellation of constant peripheral speed control
G112M200                        2nd spindle mill-point machining mode
G112S3184M203                   2nd spindle milling speed selection and milling spindle normal rotation
G110C2                          2nd spindle C-axis selection
G0C#102                         2nd spindle C-axis positioning (angle indexing)
G112M210                        2nd spindle C-axis clamping
G00X25.Z-5.                     Positioning
G87Z-5.X5.Q5000P0.2F200         Longitudinal deep-hole drilling cycle
G112M212                        2nd spindle C-axis unclamping
G80                             Cancellation of fixed hole-drilling cycle
G00C[#102+180.]                 2nd spindle C-axis positioning (angle indexing)
G112M210                        2nd spindle C-axis clamping
G87Z-5.X5.Q5000P0.2F200         Longitudinal deep-hole drilling cycle
G112M212                        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 can be used together with G codes other than group 0. However, for G codes other than group 0 using axis address, the modal value of the appropriate G code is only renewed.

   Example: G110 X2 G00; The modal value is only changed to G00 and axis movement is not made.
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 single block operation, the stop is performed after execution of G110 and G111 blocks.
5. After the axis is changed by G110, the tool of counterpart line must be used, which requires giving instructions of the change of compensation amount to the self-line. Therefore, execute T command after G110 and G111. (MULTIPLEX)
6. When the axis is changed by G110, the counterpart line must be in a state of automatic starting and standby.

   State of standby

   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.

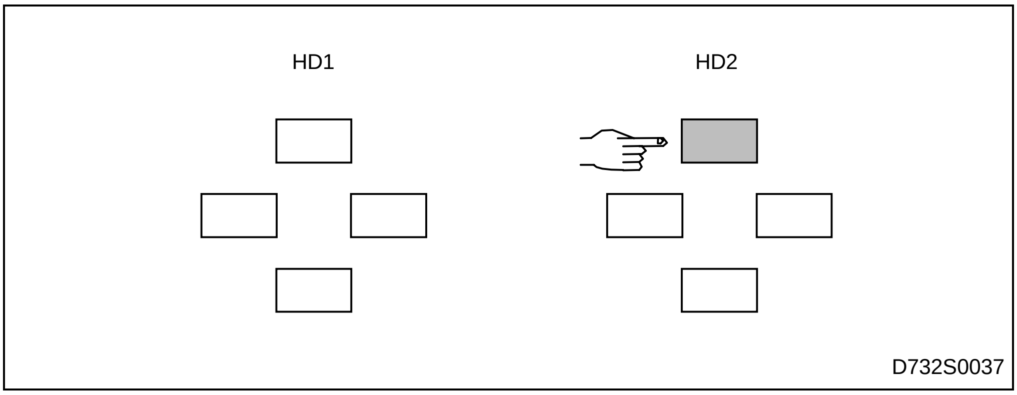
   For example, when M950 is outputted from HD1, HD1 is in a state of standby until M950 is outputted from HD2. (HD1 does not execute blocks subsequent to M950.) When M950 is outputted from HD2, HD1 executes the block following M950.

   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;      ⎭
```

7. Even after the axis is changed by G110, manual interruption can be executed. In this case, press JOG/HANDLE button of the axis to be moved without considering an axis change by program.

   For example, when X-axis of HD2 is moved by manual interruption with X-axis of HD2 controlled by HD1 (That is, the state where “G110X2;” is commanded by program at HD1 side), press X-axis button for HD2 side.



8. 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.

9. Synchronous feed with, or control of feed per, revolution of the milling spindle is not available during cross machining control.

10. For a machine whose secondary spindle is equipped with the function for the C-axis control or 0.001-degree index (orientation), 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			Group	Function
A	B	C		
G00	G00	G00	01	Rapid positioning
G01	G01	G01	01	Linear interpolation (*)
G02	G02	G02	01	Circular interpolation CW (*)
G03	G03	G03	01	Circular interpolation CCW (*)
G10	G10	G10	00	Data setting/change
G27	G27	G27	00	Reference point return check
G28	G28	G28	00	Reference point return
G29	G29	G29	00	Return from reference point
G30	G30	G30	00	Return to 2nd/3rd/4th reference point
G30.1	G30.1	G30.1	00	Return to floating reference point
G36 (G36.5)	G36 (G36.5)	G36 (G36.5)	00	Measurement target data setting
G50	G92	G92	00	Coordinate system setting/Spindle limit speed setting
G53	G53	G53	00	MAZATROL coordinate system selection
G65	G65	G65	00	Macro call
G66	G66	G66	14	Macro modal call
G83	G83	G83	09	Face drilling cycle
G84	G84	G84	09	Face tapping cycle
G84.2	G84.2	G84.2	09	Face synchronous tapping cycle
G85	G85	G85	09	Face boring cycle
G87	G87	G87	09	Side drilling cycle
G88	G88	G88	09	Side tapping cycle
G88.2	G88.2	G88.2	09	Side synchronous tapping cycle
G89	G89	G89	09	Side boring cycle
G110	G110	G110	00	Cross machining control axis selection
G111	G111	G111	00	Cross machining control axis cancellation
G112	G112	G112	00	M-, S-, T-, and B-code output to counterpart

(*) G01, G02, G03, and G07.1 are not available to a machine of the 0.001-degree index.

Return to Main Menu

20-3 M, S, T, B Output Function to Counterpart: G112

1. Outline

The function outputs M, S, T and B commanded after G112 to the counterpart line or secondary spindle side. And M, S, T and B before G112 are outputted to the self-line.

2. Programming format

G112 M_ M_ M_ M_ S_ T_ B_;

These are outputted to the counterpart line.

M_ M_ S_ T_ B_ G112 M _ M_;

These are outputted to the counterpart line.

These are outputted to the self-line.

3. Note

1. For T commanded to be outputted to the counterpart line, the offset number is incorporated in the self-line, and only the tool number is outputted to the counterpart line.
2. The quantity of M, S, T and B, which can be specified in the block of G112, are 4 of M, 4 of S, 1 of T and 1 of B. For commands exceeding them, the last command is effective.

   Example: M03 M04 M05 G112 M03 M04 M05 ;

   In this case, they are given as follows:

Self-line	M05
Counterpart line	M03, M04 and M05

3. G112 can be specified in the same block as another G code of group 0. And in single block operation, the stop can be performed after execution of G112 block.
4. For the INTEGREX, designate the auxiliary functions for the 2. headstock side as follows:

   Example: G112 M03 S100; Normal rotation of the secondary spindle at 100 min^{-1}
   G112 M19 S90; Oriented stop of the secondary spindle at 90°

   This also applies to an SQT model whose secondary spindle is designed to be indexed in units of 0.001 degrees.

20-4 Waiting for Turrets

To perform synchronized machining for turret 1 (HD1 side) and turret 2 (HD2 side), waiting is required.

M-codes for waiting should be commanded beforehand. Then, if an M-code is commanded for turret 1, the turret 1 waits until the same M-code is commanded for the turret 2. After that M-code is commanded for the turret 2, next block is executed for the turret 1.

M950 to M997 are used as M-codes for waiting.

20-5 Common Macro Variable between Turrets

Variable #100 of HD1 side is different from #100 of HD2 side in general. However, the specified number of variables among #100 to #149 and #500 to #531 can be used as common variables capable of reading and writing in common between turrets.

Parameter **K68**: It sets the number of variables used as common macro variables between turrets among #100 to #149.

Example:

Setting of **K68** = 20 causes 20 common variables (#100 to #119) to act as common macro variables between turrets.

Parameter **K69**: It sets the number of variables used as common macro variables between turrets among #500 to #531.

Example:

Setting of **K69** = 10 causes 10 common variables (#500 to #509) to act as common macro variables between turrets.

Return to Main Menu

21 FUNCTIONS PROPER TO SQR SERIES

The SQR series machine is distinguished by two turrets (upper and lower) which can move independently from each other. This chapter describes the functions proper to the SQR series machine.

21-1 Programming for the SQR Series Machine

1. Outline

The movement of the upper and lower turrets of the SQR series machine is to be controlled in a single program as follows:

G109 L1;....................................Selection of the upper turret

Commands for the upper turret

M950;..Waiting command

M30;

G109 L2;....................................Selection of the lower turret

Commands for the lower turret

M950; ························ Waiting command

M30;

2. Remarks

1. Enter the turret selection command G109L1 or G109L2 even if only one turret is used. Both turrets will be moved in default of the selection command.
2. The spindle function (S-code), the miscellaneous function for the spindle rotation (M03, M04, M05) and the preparatory function for the constant peripheral speed control (G96, G97) must be identical for both turrets. Otherwise, an alarm will be caused.
3. Enter the waiting command before the end of the machining commands for each turret in a program using both turrets. Otherwise, the other turret will be brought to a stop upon execution of the commands for one turret.

21-2 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

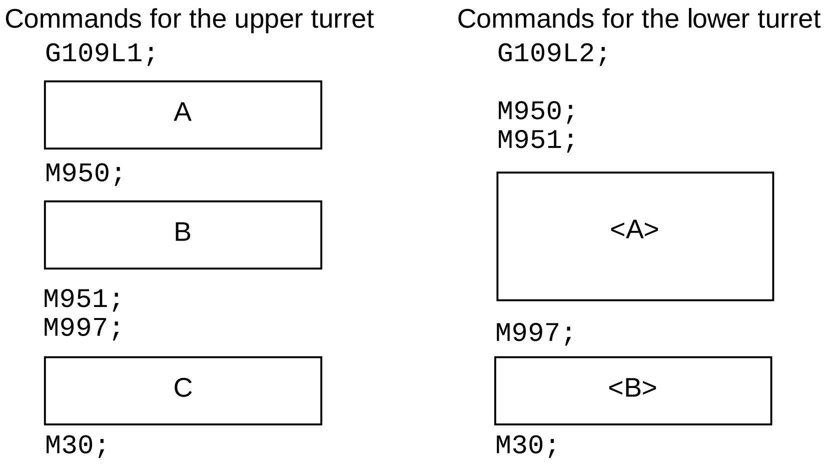
A. M-codes for waiting

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

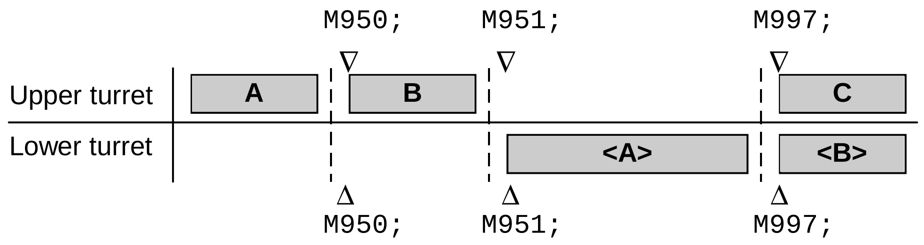

Commands for the upper turret

```
G109L1;
  A
M950;
  B
M951;
M997;
  C
M30;
```

Commands for the lower turret

```
G109L2;
M950;
M951;
  <A>
M997;
  <B>
M30;
```

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.

Return to Main Menu

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.

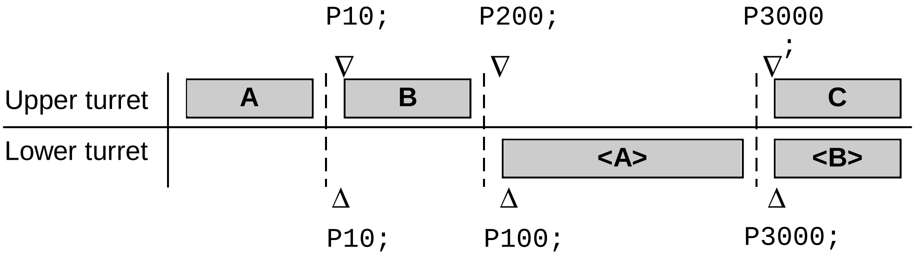
<u>Programming format</u>

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;
                                        P100;
   P10;
   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.

21-3 Selection of Turret for Spindle Speed Function: M560/M561

1. Outline

The codes M560 and M561 are used to specify the turret whose spindle speed function (S-code) is to be made valid.

2. Programming format

M560;.......Selection of the S-code specified for the upper turret
M561;.......Selection of the S-code specified for the lower turret

3. Program structure

Commands for the upper turret   Commands for the lower turret

```
G109L1;                     G109L2;

P1000;                      P1000;
T0103M560;                  T0508;
G96M03S500;                 G96M03S500;

        A                          <A>

P2000;                      P2000;
T0207;                      T0309M561;
G96M03S850;                 G96M03S850;

        B                          <B>

G97;                        G97;

M30;                        M30;
```

- The spindle rotates according to the S-code specified for the upper turret during execution of the machining commands given in sections A and <A>.
- The spindle rotates according to the S-code specified for the lower turret during execution of the machining commands given in sections B and <B>.

21-4 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)
- M550; .... Coupling command for the two turrets
- M551; .... 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 M550 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.

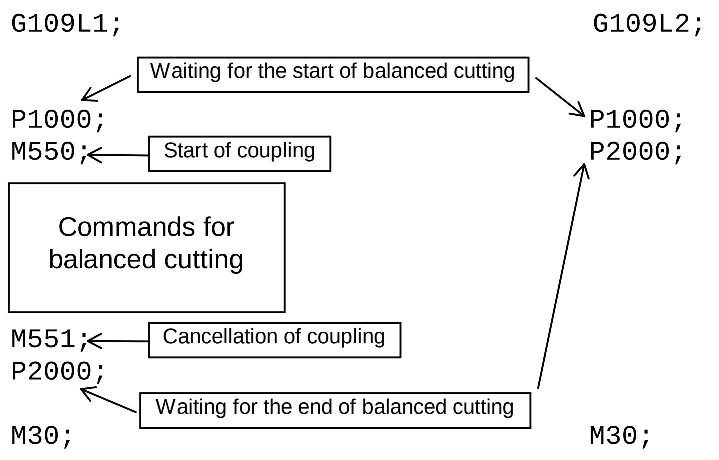
4) Enter the command M551 after the movement commands for the master turret to cancel the coupling.

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



```
G109L1;                            G109L2;

P1000;                             P1000;
M550;                              P2000;

M551;
P2000;

M30;                               M30;
```

Waiting for the start of balanced cutting
Start of coupling
Commands for balanced cutting
Cancellation of coupling
Waiting for the end of balanced cutting

4. Sample program

```
N000 G109 L1;
N001 G00 X800.Z70.;
P10;
M42 S250 T0101 M03;
N002 X132.Z60.M08;
M950;
M550;
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.;
M551;
P20;
N015 M09 M05;
N100 G109 L2;
N101 G00 X800.Z200.;
P10;
M42 S250 T0101 M03;
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

P109 L2: Lower turret selection

P20: Single command for waiting. Required for both turrets

Return to Main Menu

22 DIFFERENCES OF PROGRAMMING FORMAT BETWEEN T32 COMPATIBLE MODE AND STANDARD MODE

No.	Function	T32 compatible mode (Parameter **P16** bit 3 = 0)	Standard mode (Parameter **P16** bit 3 = 1)	Remarks
1	Cylindrical interpolation	During G12.1 mode G16 C Cylinder radius;	G07.1 C Cylinder radius;	
	(Cancellation)	G13.1;	G07.1 C0;	
	(Virtual axis address)	Y Coordinate value;	C Coordinate value;	
2	Polar coordinate interpolation	During G12.1 mode G17;	G12.1;	
	(Virtual axis address)	Y Coordinate value;	C Coordinate value;	
3	Measurement target data setting	G36;	G36.5;	
	Measurement result compensation start	G37;	G37.5;	
4	Automatic tool offset, X-axis	——	G36;	
	Automatic tool offset, Z-axis	——	G37;	
5	MAZATROL coordinates cancel	G52;	G52.5;	
	MAZATROL coordinates setting	G53;	G53.5;	
6	Local coordinate system	——	G52;	
7	Machine coordinate system	——	G53;	
8	Spindle rotational speed clamp	G50 S Maximum speed Q Minimum speed;		
9	Peripheral speed constant control	G96 S Peripheral speed P Control axis No.		
10	Programmable parameter setting	G10 L50; N Parameter No. R Setting value P Axis No.;		
	(Cancellation)	G11;		
11	Hole machining fixed cycle (G80 to G89)	G8□ .....L Repeat times Q Cutting depth;	G8□ .....K Repeat times Q Cutting depth;	Decimal point for Q available in T32 compatible mode, but not in Standard mode.
12	Single form fixed cycle (G90/G92)	G9□ I Taper value;	G9□ .....R Taper value;	
	(G94)	G94 ......K Taper value;	G94 ......R Taper value;	
13	Roughing cycle (G71/G72)	G7□ .....D Cutting depth;	G7□ .....U Cutting depth; G7□ ....;	
	Imitative cutting cycle (G73)	G73 I Cutting allowance X K Cutting allowance Z D Division times;	G73 U Cutting allowance X W Cutting allowance Z R Division times; G73 ....;	
	Cutting-off cycle (G74/G75)	G7□ .....IMovement distance K Cutting depth D Escape distance;	G7□ P Movement distance Q Cutting depth R Escape distance;	
	Threading cycle (G76)	G76 I Taper height K Thread height D Cutting depth A Tool nose angle;	G76 ......P○○○○●●; Tool nose angle G76 ...... R Taper height P Thread height Q Cutting depth;	

- NOTE -

Return to Main Menu

23 EIA/ISO PROGRAM DISPLAY

This chapter describes general procedures for and notes on constructing an EIA/ISO program newly, and then editing functions.

23-1 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.

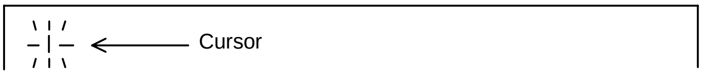
Remark: Refer to the Operating Manual for the window of work number list.

(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 (⎆).

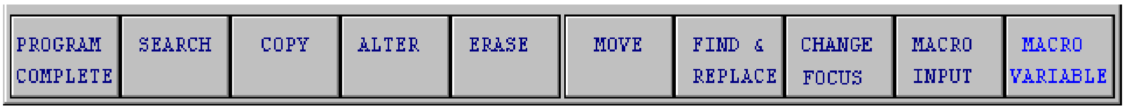
- When INPUT key is pressed, the cursor is moved to the top of the next line, and then the data of the next block can be entered.

(7) Press the **PROGRAM COMPLETE** menu key to end the editing.

23-2 Editing Function of EIA/ISO PROGRAM Display

23-2-1 General

Establishing a constructing mode on the EIA/ISO **PROGRAM** display allows the following menu to be displayed as an initial one.



[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.

23-2-2 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.

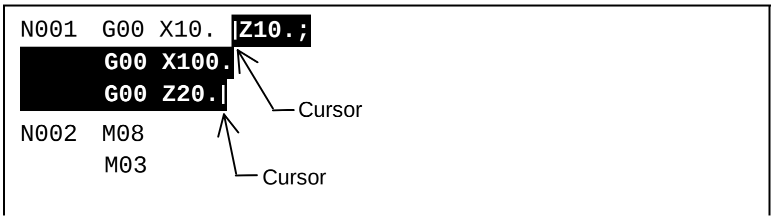
(2) Press the **ERASE** menu key.

➔ 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:

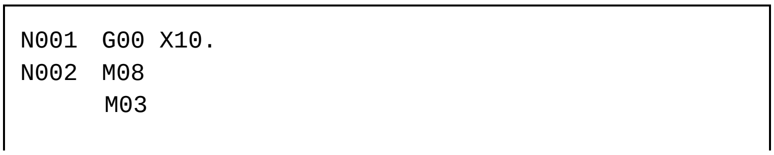


```
N001  G00 X10. Z10.;
      G00 X100.
      G00 Z20.
N002  M08
      M03
```

(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.

Return to Main Menu

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.

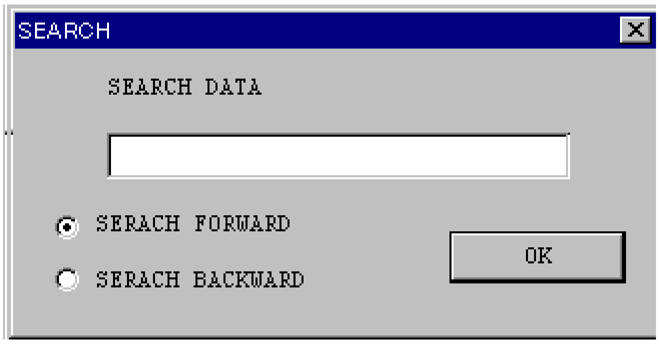
➔ The cursor moves to the specified line.

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.



● SERACH FORWARD

○ SERACH BACKWARD

- 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.

E. Searching for G109 Commands

This is a special function provided for the SQR series machine.

(1) Press the **SEARCH** menu key.

(2) Press the **SEARCH G109** menu key.

➔ The cursor moves to the line of the G109 command.

F. Searching for waiting commands

This is a special function provided for the SQR series machine.

(1) Press the **SEARCH** menu key.

(2) Press the **WAITING COMMAND** menu key.

➔ The cursor moves to the line of the waiting command (M950 to M997 or P1 to P99999999).

Return to Main Menu

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.

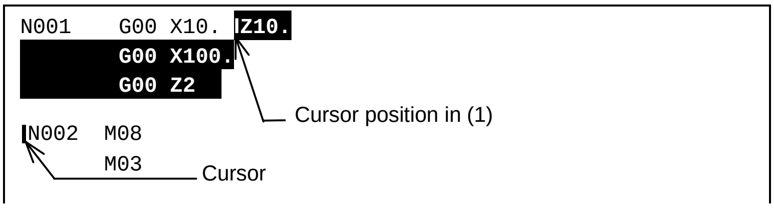
(3) Press the **LINE(S) COPY** menu key.

➔ 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

N002   M08
       M03
```

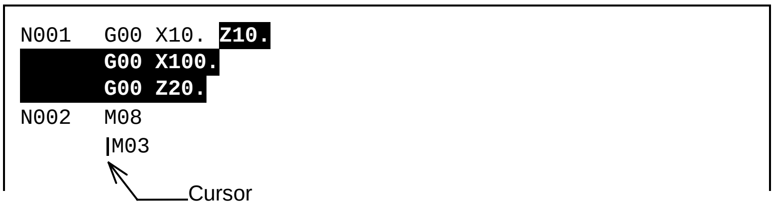
(5) Press the input key.

➔ 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:

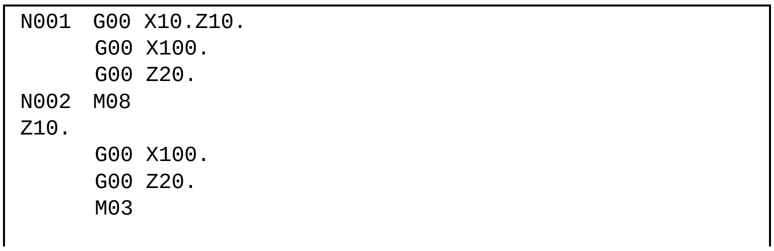


```
N001   G00 X10. Z10.
       G00 X100.
       G00 Z20.
N002   M08
       M03
```

(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

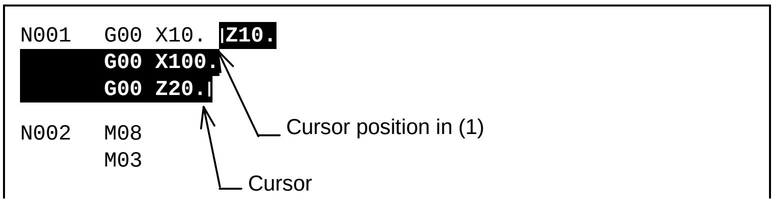
(1) Move the cursor to the head of the character string to be moved.

(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
```

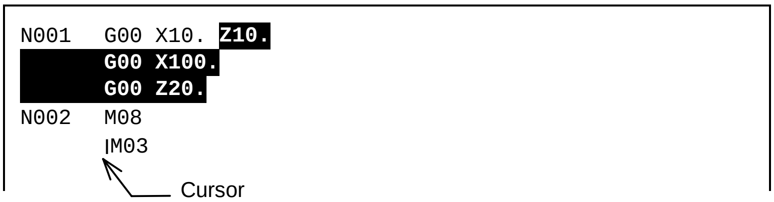
(4) Press the input key.

➔ 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
```

Return to Main Menu

(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.

- Pressing the **PROGRAM FILE** menu key allows the window of program list to be displayed.

➔ The character string is moved to the new program.

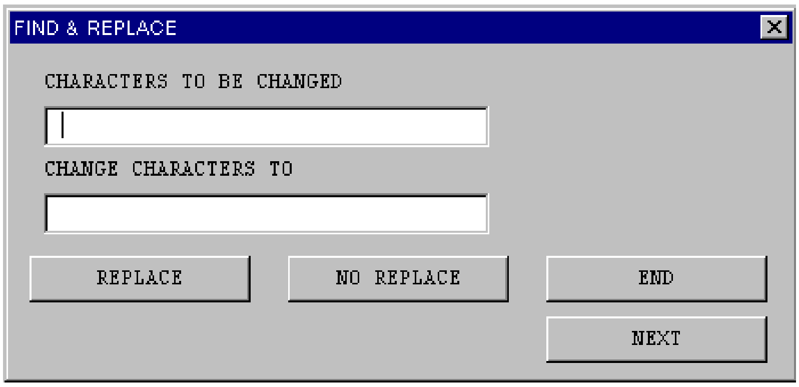
7. Replacing the data

(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.

Remark 1: 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.

Return to Main Menu

Remark 2: To stop the replacement, press the **END** menu key.

Remark 3: To replace all the character strings in the program, press the **NEXT** menu key.

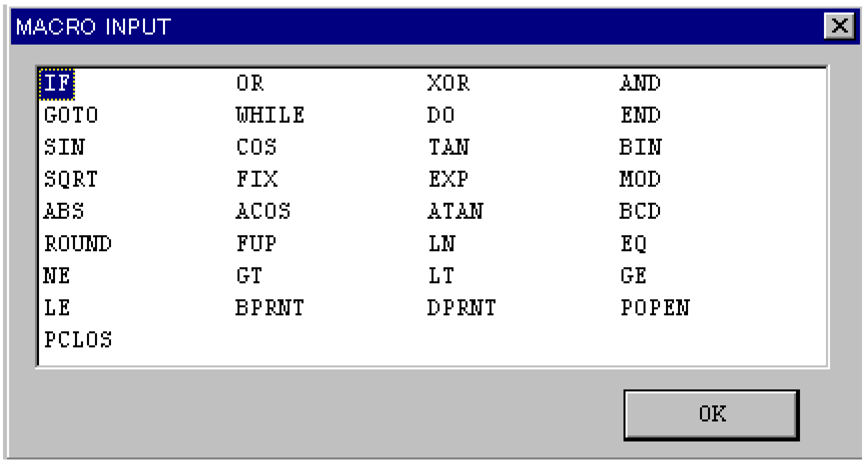
Remark 4: Press the data cancellation key (CANCEL) to stop halfway the total replacement by the NEXT menu function, whose running state is indicated by the message **CNC BUSY** on the display.

23-3 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.



IF	OR	XOR	AND
GOTO	WHILE	DO	END
SIN	COS	TAN	BIN
SQRT	FIX	EXP	MOD
ABS	ACOS	ATAN	BCD
ROUND	FUP	LN	EQ
NE	GT	LT	GE
LE	BPRNT	DPRNT	POPEN
PCLOS			

- The characters selected with the cursor are 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.

Press the menu selector key to display the menu for normal data input, and continue program editing.

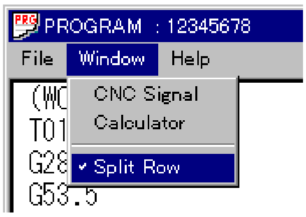
Return to Main Menu

23-4 Division of Display (Split Screen)

This function can only be used with the SQR series machine.

The function permits the screen to be divided into two parts, arranged left and right or up and down, of various contents within the same program in order for the operator to input data while comparing the related part.

The initial division is made vertically (into the left and right parts).
Select the option “Split Row” from the listing under “Window” on the menu bar to divide the screen horizontally (into the upper and lower parts). The selection is indicated by a check-off mark (✓) on the left of the option name as shown below:



This section describes the procedure for (vertically) dividing the screen, cancelling the division and changing the active part.

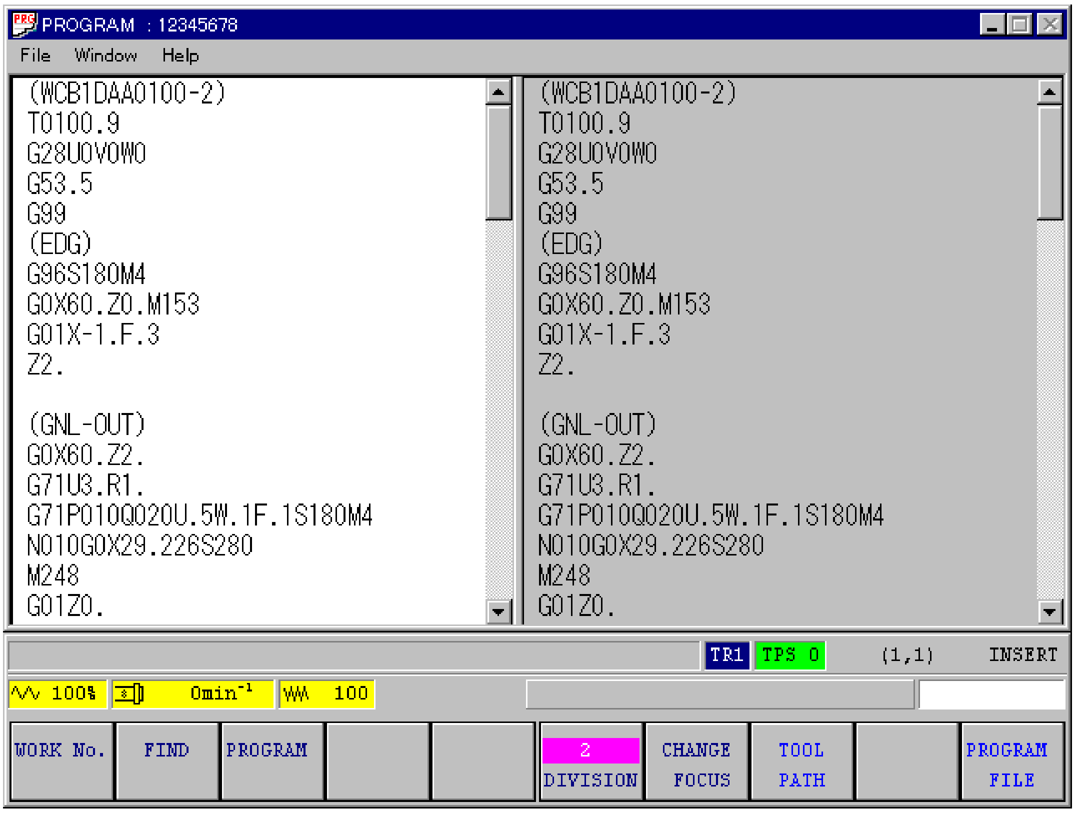
1. Dividing the screen

(1) Temporarily cancel the editing mode, if selected, by pressing the **PROGRAM COMPLETE** menu key.

(2) Press the **2 DIVISION** menu key.

➔ The display of the menu item will be reversed and the screen divided into the left and right part.

- One and the same section of the program is initially displayed in both parts.



```
(WCB1DAA0100-2)
T0100.9
G28U0V0W0
G53.5
G99
(EDG)
G96S180M4
G0X60.Z0.M153
G01X-1.F.3
Z2.

(GNL-OUT)
G0X60.Z2.
G71U3.R1.
G71P010Q020U.5W.1F.1S180M4
N010G0X29.226S280
M248
G01Z0.
```

```
(WCB1DAA0100-2)
T0100.9
G28U0V0W0
G53.5
G99
(EDG)
G96S180M4
G0X60.Z0.M153
G01X-1.F.3
Z2.

(GNL-OUT)
G0X60.Z2.
G71U3.R1.
G71P010Q020U.5W.1F.1S180M4
N010G0X29.226S280
M248
G01Z0.
```

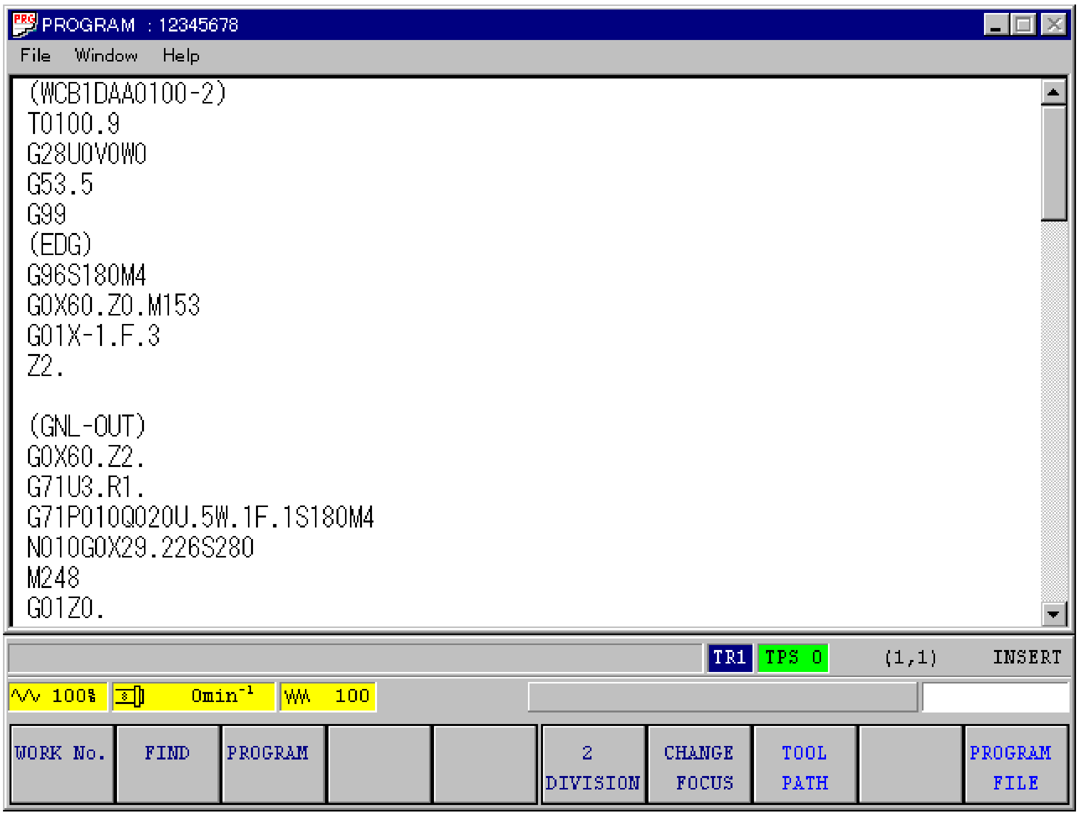
- The editing operation can only be carried out in the part the background of which is white.
- The display contents in the other part the background of which is gray (inactive part) will remain unchanged even after the editing in the active part. Press the **CHANGE FOCUS** menu key to change the display in the other part according to the editing operation.

2. Cancelling the division

(1) Temporarily cancel the editing mode, if selected, by pressing the **PROGRAM COMPLETE** menu key.

(2) Press anew the **2 DIVISION** menu key.

➔ The reversed display of the menu item will be released and the division of the screen cancelled.



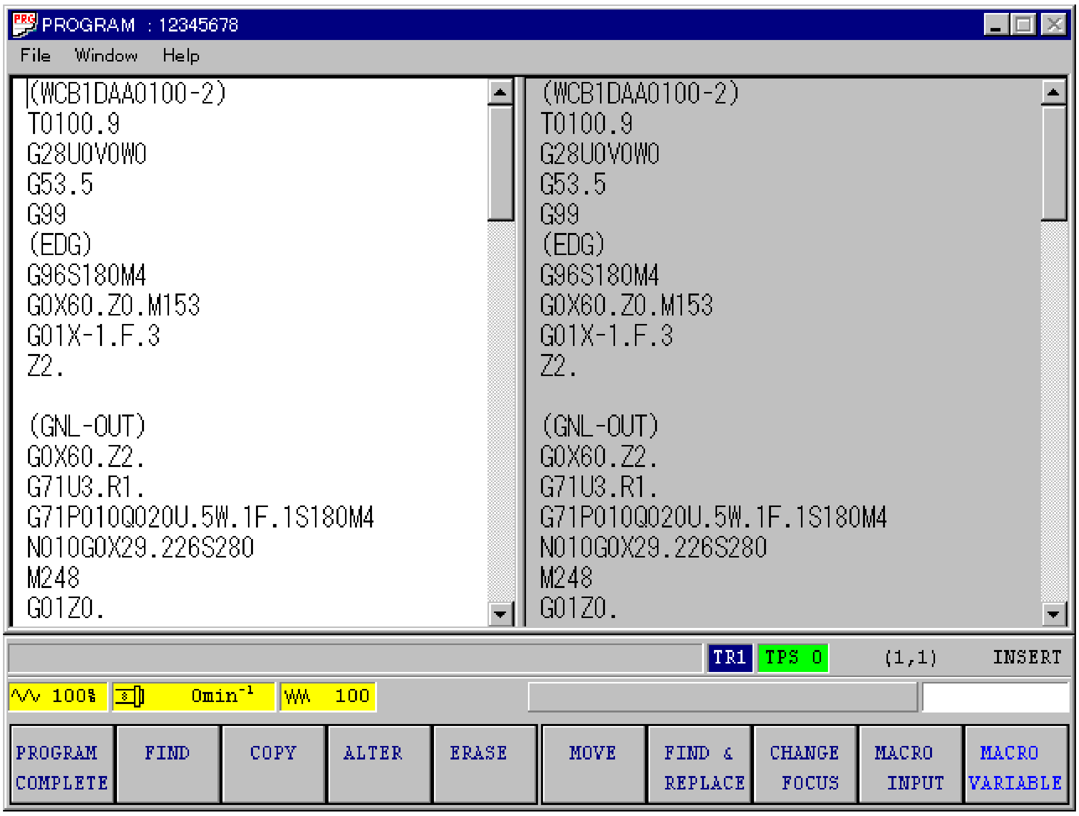
```
(WCB1DAA0100-2)
T0100.9
G28U0V0W0
G53.5
G99
(EDG)
G96S180M4
G0X60.Z0.M153
G01X-1.F.3
Z2.

(GNL-OUT)
G0X60.Z2.
G71U3.R1.
G71P010Q020U.5W.1F.1S180M4
N010G0X29.226S280
M248
G01Z0.
```

3. Changing the active part

The editing is only possible for the part currently displayed on the white background. The method to change the active part is indicated here below.
The data after the editing will not be displayed in the other part unless this changing operation is carried out.

Return to Main Menu

In the example hereafter, the left part is currently active.



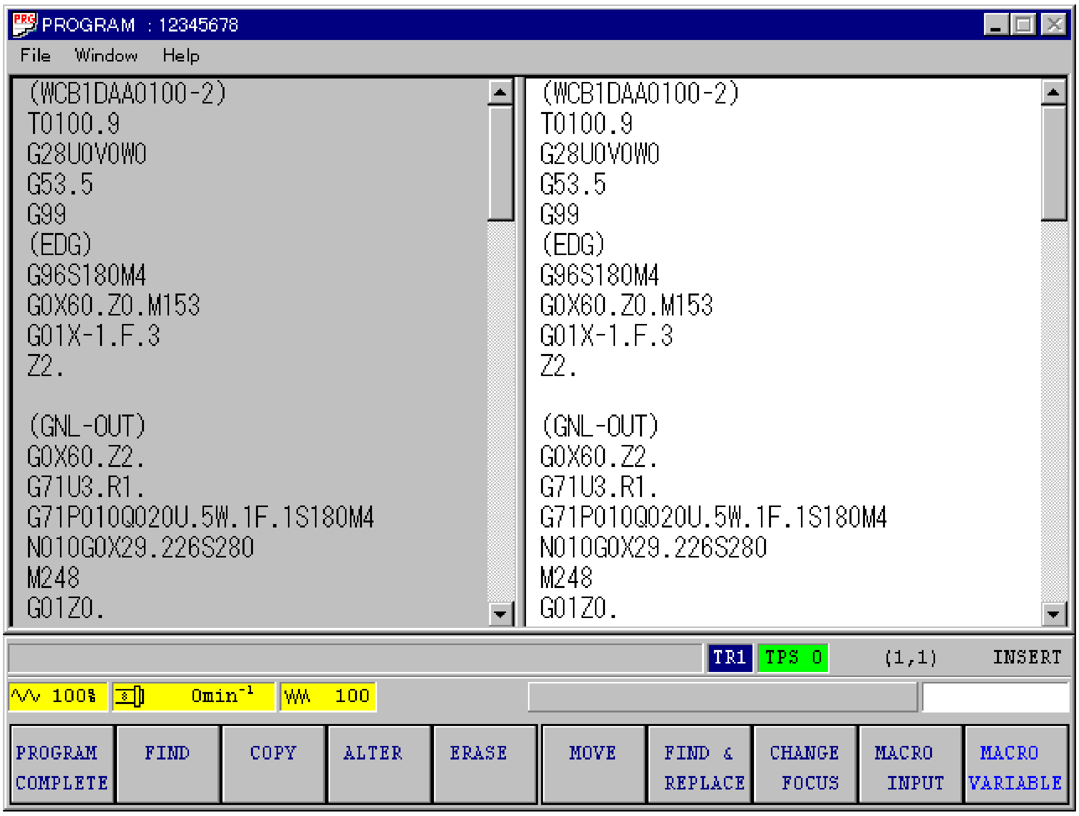
```
(WCB1DAA0100-2)
T0100.9
G28U0V0W0
G53.5
G99
(EDG)
G96S180M4
G0X60.Z0.M153
G01X-1.F.3
Z2.

(GNL-OUT)
G0X60.Z2.
G71U3.R1.
G71P010Q020U.5W.1F.1S180M4
N010G0X29.226S280
M248
G01Z0.
```

```
(WCB1DAA0100-2)
T0100.9
G28U0V0W0
G53.5
G99
(EDG)
G96S180M4
G0X60.Z0.M153
G01X-1.F.3
Z2.

(GNL-OUT)
G0X60.Z2.
G71U3.R1.
G71P010Q020U.5W.1F.1S180M4
N010G0X29.226S280
M248
G01Z0.
```

(1) Press the **CHANGE FOCUS** menu key.

➔ The white background will be transferred from the left to the right part to indicate that the latter has been made active.

- The contents in the right part will have been modified at the same time according to the editing operation performed for the left part.



```
(WCB1DAA0100-2)
T0100.9
G28U0V0W0
G53.5
G99
(EDG)
G96S180M4
G0X60.Z0.M153
G01X-1.F.3
Z2.

(GNL-OUT)
G0X60.Z2.
G71U3.R1.
G71P010Q020U.5W.1F.1S180M4
N010G0X29.226S280
M248
G01Z0.
```

```
(WCB1DAA0100-2)
T0100.9
G28U0V0W0
G53.5
G99
(EDG)
G96S180M4
G0X60.Z0.M153
G01X-1.F.3
Z2.

(GNL-OUT)
G0X60.Z2.
G71U3.R1.
G71P010Q020U.5W.1F.1S180M4
N010G0X29.226S280
M248
G01Z0.
```

- NOTE -

Return to Main Menu

24 ADVANCED INPUT SENSITIVITY (FOR 2-AXIS MODELS ONLY)

1. Function and purpose

This function (exclusive for 2-axis models) allows the smallest input/control capacity to be enhanced from 0.001 mm (or 0.0001 in.) to 0.0001 mm (or 0.0.00001 in.).

This function, however, is only available for EIA/ISO programs and execution of MAZATROL programs is not allowed with the advanced sensitivity being selected. Changeover of the capacity, moreover, must be followed by corresponding setup operations.

2. Selection parameter

Parameter	Description	Meaning	Unit	Setting range	Effective condition
P19 bit 0	Inch/Metric (mm) selection	1: Inch 0: mm	—	0, 1	Power OFF→ ON
P19 bit 1	Number of decimal digits available (EIA)	1: Advanced 0: Standard	—	0, 1	

3. Detailed description

A. Data of the TOOL DATA, TOOL OFFSET and WORK OFFSET displays

The external data to be used for execution of EIA/ISO programs on the above-mentioned displays must be prepared separately in accordance with the standard and the advanced input sensitivity. Do not use one and the same set of data commonly for both input sensitivities.

B. Data of the PARAMETER display

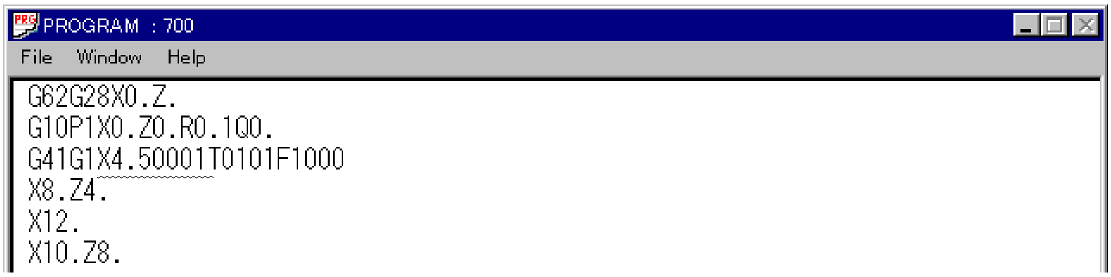
Referring to the table given in Item 5, modify the data of the related parameters with respect to the selected input sensitivity.

Example: **B33** (Chuck O.D.) = 10000 <for Inch system>
Parameter **B33** requires modification since the above setting (without decimal point) is processed as

1 in. (10000 × 0.0001 in.) for “standard sensitivity”, or
0.1 in. (10000 × 0.00001 in.) for “advanced sensitivity”.

C. Data for advanced input sensitivity

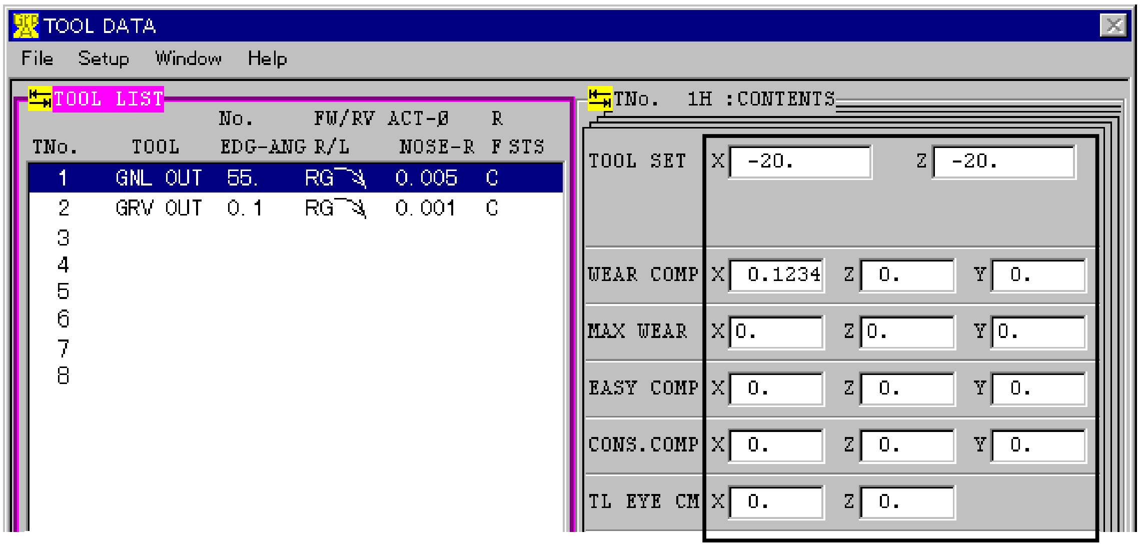
- Specific data in EIA/ISO programs



```
G62G28X0.Z.
G10P1X0.Z0.R0.1Q0.
G41G1X4.50001T0101F1000
X8.Z4.
X12.
X10.Z8.
```

Return to Main Menu

- Data of the **TOOL DATA** display (Tool-setting and compensation data)

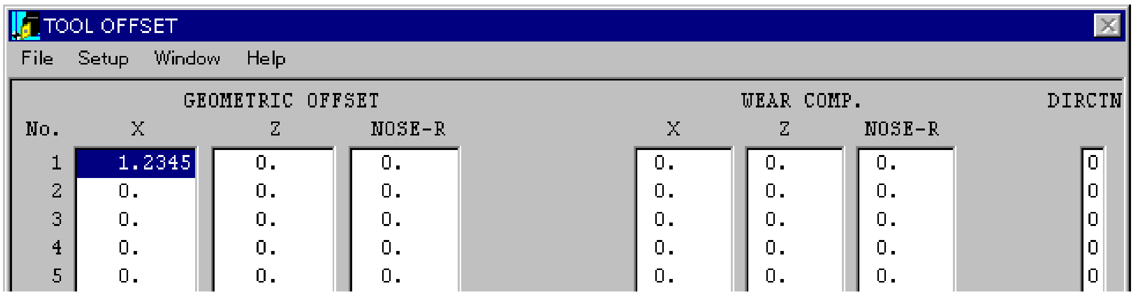


TNo.	TOOL	No. EDG-ANG	FW/RV R/L	ACT-Ø NOSE-R	R F STS
1	GNL OUT	55.	RG	0.005	C
2	GRV OUT	0.1	RG	0.001	C
3					
4					
5					
6					
7					
8					

TNo. 1H :CONTENTS

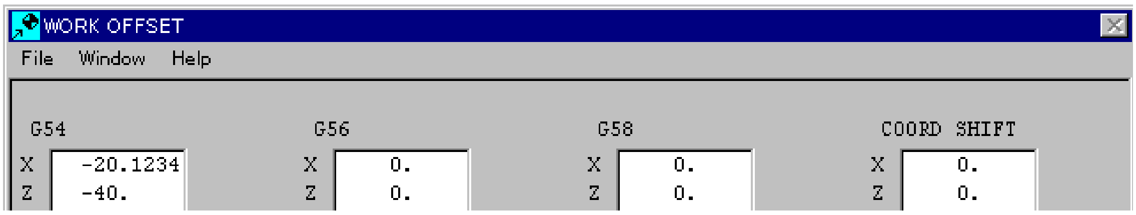
TOOL SET	X	-20.	Z	-20.		
WEAR COMP	X	0.1234	Z	0.	Y	0.
MAX WEAR	X	0.	Z	0.	Y	0.
EASY COMP	X	0.	Z	0.	Y	0.
CONS.COMP	X	0.	Z	0.	Y	0.
TL EYE CM	X	0.	Z	0.		

- Data of the **TOOL OFFSET** display



	GEOMETRIC OFFSET			WEAR COMP.			DIRCTN
No.	X	Z	NOSE-R	X	Z	NOSE-R	
1	1.2345	0.	0.	0.	0.	0.	0
2	0.	0.	0.	0.	0.	0.	0
3	0.	0.	0.	0.	0.	0.	0
4	0.	0.	0.	0.	0.	0.	0
5	0.	0.	0.	0.	0.	0.	0

- Data of the **WORK OFFSET** display



	G54	G56	G58	COORD SHIFT
X	-20.1234	0.	0.	0.
Z	-40.	0.	0.	0.

4. Selection procedure

The following description refers to the changeover from Standard to Advanced sensitivity.

(1) Save the data for “Standard input sensitivity” by the aid of output functions of the **DATA I/O** display.

(2) Change the selection parameter (**P19** bit 1) from “0” to “1”.

(3) Temporarily turn the power off and then turn it back on.

(4) Modify the related data tabulated below according to the advanced input sensitivity.

- Initialize the displays **TOOL DATA**, **TOOL OFFSET** and **WORK OFFSET** by the aid of the related function of the **DIAGNOSIS** display.
- Data of the **TOOL DATA**, **TOOL OFFSET** and **WORK OFFSET** displays are to be entered by the user.
- Modification on the related parameters, if required, is to be performed carefully in accordance with their respective setting units as indicated in the table in Item 5.

Return to Main Menu

(5) Temporarily turn the power off and then turn it back on.

Program type Input sensitivity	MAZATROL/EIA Standard sensitivity	EIA only Advanced sensitivity	Setting method
Data	Machine parameters (for Standard sensitivity)	Machine parameters (for Advanced sensitivity)	Default values (FD)
	User parameters (for Standard sensitivity	User parameters (for Advanced sensitivity)	Default values (FD) By the user
	TOOL DATA display's data (for Standard sensitivity)	**TOOL DATA** display's data (for Advanced sensitivity)	By the user
	TOOL OFFSET display's data (for Standard sensitivity)	**TOOL OFFSET** display's data (for Advanced sensitivity)	
	WORK OFFSET display's data (for Standard sensitivity)	**WORK OFFSET** display's data (for Advanced sensitivity)	
	CUTTING CONDITION display's data	Not provided	

Note: The machine is delivered with a floppy disk (FD) containing the default values for the **PARAMETER** display in separate sets for the standard and the advanced input sensitivity.

5. Related parameters

The following table enumerates the parameters that are to be modified correspondingly to the selection of the smallest input capacity.

Address	Description	Setting unit				Setting range	Effective condition
		Metric		Inch			
		Std. sns.	Adv. sns.	Std. sns.	Adv. sns.		
B33	Chuck outside diameter	0.001	0.0001	0.0001	0.00001	99999999	Instant
B34	Chuck width	0.001	0.0001	0.0001	0.00001	99999999	Instant
B35	Chuck inside diameter	0.001	0.0001	0.0001	0.00001	99999999	Instant
B37	Tailstock body outside diameter	0.001	0.0001	0.0001	0.00001	99999999	Instant
B38	Tailstock body length	0.001	0.0001	0.0001	0.00001	99999999	Instant
B39	Tail spindle outside diameter	0.001	0.0001	0.0001	0.00001	99999999	Instant
B40	Tail spindle length as retracted	0.001	0.0001	0.0001	0.00001	99999999	Instant
B41	Tail head outside diameter	0.001	0.0001	0.0001	0.00001	99999999	Instant
B42	Tail head length	0.001	0.0001	0.0001	0.00001	99999999	Instant
B44	Tail head biting diameter	0.001	0.0001	0.0001	0.00001	99999999	Instant
B45	Tool post radius	0.001	0.0001	0.0001	0.00001	99999999	Instant
B46	Tool post width	0.001	0.0001	0.0001	0.00001	99999999	Instant
B47	Tool post reference position X	0.001	0.0001	0.0001	0.00001	±99999999	Instant
B48	Tool post reference position Z	0.001	0.0001	0.0001	0.00001	±99999999	Instant
B73	Tool path drawing start position X (HD1)	0.001	0.0001	0.0001	0.00001	±99999999	Instant
B74	Tool path drawing start position Z (HD1)	0.001	0.0001	0.0001	0.00001	±99999999	Instant
B229	Distance between opposite turrets	0.001	0.0001	0.0001	0.00001	0 - 99999999	Instant
B235	Sensor radius correction data for protrusion width measurement	0.001	0.0001	0.0001	0.00001	±99999999	Instant
B236	Sensor radius correction data for groove width measurement	0.001	0.0001	0.0001	0.00001	±99999999	Instant
B237	Measuring range X for tool offset mesurement	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
B238	Measuring range Z for tool offset mesurement	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant

Address	Description	Setting unit				Setting range	Effective condition
		Metric		Inch			
		Std. sns.	Adv. sns.	Std. sns.	Adv. sns.		
B239	Decelerating range X for tool offset mesurement	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
B240	Decelerating range Z for tool offset mesurement	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
B241	Jaw dimension A (for chuck barrier [EIA])	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
B242	Jaw dimension B (for chuck barrier [EIA])	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
B243	Jaw dimension C (for chuck barrier [EIA])	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
B244	Jaw dimension D (for chuck barrier [EIA])	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
B245	Chucking diameter (for chuck barrier [EIA])	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
B246	Holder dimension 1 (for tool barrier [EIA])	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
B247	Holder dimension 2 (for tool barrier [EIA])	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
B248	Holder dimension 3 (for tool barrier [EIA])	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
B249	Tool nose position X (for tool barrier [EIA])	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
B250	Tool nose position Z (for tool barrier [EIA])	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
B251	Sub-chuck position Z (for tool barrier [EIA])	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
B252	Protrusion length Z (for tail barrier [EIA])	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
B253	Retract position Z (for tail barrier [EIA])	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
B254	Workpiece length Z (for tail barrier [EIA])	0.001	0.0001	0.0001	0.00001	1 - 99999999	Instant
BS10	Creep speed for homing	1 mm/min	0.1 mm/min	1 mm/min	0.1 mm/min	—	Instant
BS14	Backlash in G0	0.001 mm	0.0001 mm	0.001 mm	0.0001 mm	—	Instant
BS15	Backlash in G1	0.001 mm	0.0001 mm	0.001 mm	0.0001 mm	—	Instant

Return to Main Menu

APPENDIX

1. List of Function Codes

Function code		Significant/ insignificant information	Function inside NC unit	Control panel key code yes/no
EIA	ISO			
Blank	NUL	×	In the case of an EIA code, parity H applies to a significant/insignificant code in a single block; always ignored in the case of an ISO code.	×
Del	DEL	×		×
BS	BS	×		×
Tab	HT	×		×
CR	LF/NL	○	End of block (EOB) (**Note 4**)	○
–	CR	×		×
SP	SP	×		○
ER	%	○	Rewind stop and search rewind start	○
2+4+5	(	○	Control out (**Note 6**)	○
2+4+7	)	○	Control in (**Note 6**)	○
+	+	○		○
–	–	○		○
(**Note 5**)	:	○		×
/	/	○	Block delete (optional block skip)	○
–	SI	×		×
UC	–	×		×
LC	–	×		×
.	.	○	Denotes decimal point.	○
,	,	○		○
(**Note 5**)	]	○		○
(**Note 5**)	[	○		○
(**Note 5**)	#	○		○
(**Note 5**)	*	○		○
(**Note 5**)	=	○		○
(**Note 5**)	!	○		○
(**Note 5**)	$	○		○
0 to 9	0 to 9	○	Numbers	○
A to Z	A to Z	○	Letters of alphabet (addresses)	○

Note 1: NC unit ignores the codes marked “×” in the significant/insignificant information column and they are not stored in the memory.

Note 2: Significant information codes are marked with a “○” in the above table.

Note 3: The ALL MARK in EIA codes is also ignored.

Note 4: EOBs are denoted with “ ; ”.

Note 5: When converting ISO codes into EIA codes, this pattern can be designated by parameter.

Note 6: The control out and control in EIA codes are combinations of channels defined inside the NC unit. Consequently, they are not provided by ordinary typewriters.

Return to Main Menu

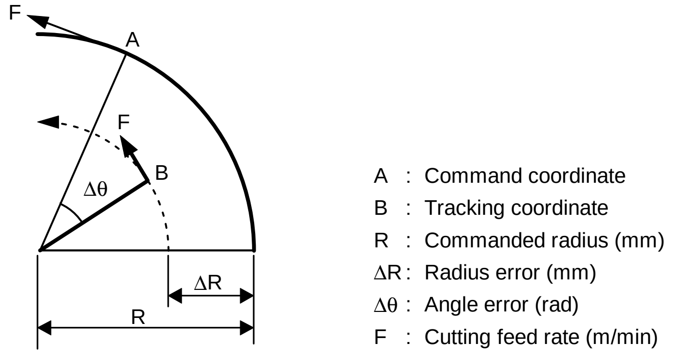
2. List of Command Values and Setting Ranges

	Linear axis		Rotational axis
	Input unit (mm)	Input unit (inch)	Degree (deg)
Units of minimum data setting	0.001 mm	0.0001 inch	0.001
Maximum stroke (value for machine coordinate system)	±99999.999 mm	±9999.9999 inch	±99999.999°
Maximum programmable dimension	±99999.999 mm	±9999.9999 inch	±99999.999°
Rapid feed rate	1 – 60000 mm/min	1 – 6000 inch/min	1 – 60000°/min
Cutting feed rate	1 – 60000 mm/min	1 – 6000 inch/min	1 – 60000°/min
Second zero point offset (value in machine coordinate system)	±99999.999 mm	±9999.9999 inch	±99999.999°
Tool offset amount (tool position)	±999.999 mm	±99.9999 inch	
Incremental feed amount	0.001 mm/P	0.0001 inch/P	0.001 /P
Handle feed amount	0.001 mm/P	0.0001 inch/P	0.001 /P
Soft limit range (value in machine coordinate system)	–99999.999 – +99999.999 mm	–9999.9999 – +9999.9999 inch	1 – 359.999
Dwell time	0 – 99999.999 sec 0 – 99999 rev	0 – 99999.999 sec 0 – 99999 rev	
Backlash compensation amount	0 – ±511 pulses	0 – ±511 pulses	0 – ±511 pulses
Pitch error compensation amount	0 – ±127 pulses	0 – ±127 pulses	0 – ±127 pulses
Dry run speed	0 – 3600 mm/min	0 – 360 inch	0 – 3600°/min
Manual jog rapid feed	0 – 60000 mm/min	0 – 60000 inch/min	0 – 60000°/min
Thread lead	0.001 – 999.99999 mm	0.0001 – 99.999999 inch	
Synchronous feed	0.001 – 999.999 mm/rev	0.0001 – 99.9999 inch/rev	

Return to Main Menu

3. Arc Cutting Radius Error

When arc cutting is performed, an error is caused between the command coordinate and the tracking coordinate due to the tacking delay in the smoothing cirucit and servo system, and the workpiece ends up with a radius smaller than the commanded value.



TEP215

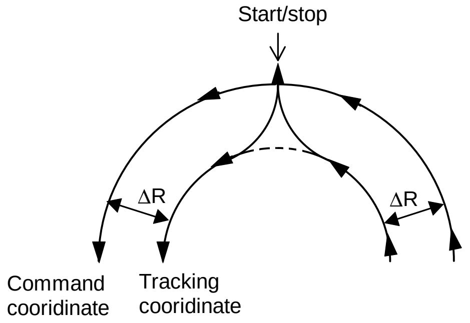
The radius error ΔR and angle error Δθ are calculated from the following formula.

$$\Delta R = \frac{1}{2R} \bullet (Ts^2 + Tp^2) \bullet \left(\frac{F \times 10^3}{60}\right)^2 \qquad \text{(mm)}$$

$$\Delta\theta = \tan^{-1}\left(Ts \bullet \frac{F}{R}\right) + \tan^{-1}\left(Tp \bullet \frac{F}{R}\right) \qquad \text{(rad)}$$

Ts: Time constant of smoothing circuit (sec)
Tp: Position loop time constant
Tp = 0.03 sec for semi-closed loop system
Tp = 0.04 sec for closed loop system

- When the radius error ΔR during arc cutting is not within the allowable value, proceed to reduce the cutting feed rate F, set Ts to a lower value or review the program.
- In the normal state, ΔR is constant. However, it is not constant at command start and stop transient time. At command start and stop time, therefore, the tracking coordinate is as shown in the figure below.

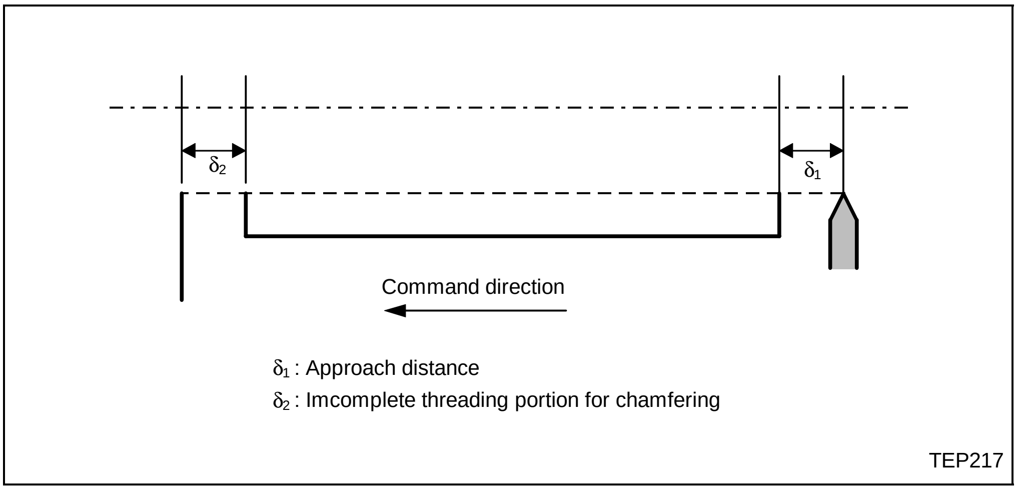


TEP216

Return to Main Menu

4. Additional Explanation About Incomplete Thread Portion in Threading

In threading, pitches may become irregular at positions close to the starting point and ending point owing to delay due to automatic acceleration/deceleration or delay due to position loop in the servo system. In preparing programs, therefore, extra distance may be commanded for approach distance δ_1 and incomplete threading portion for chamfering distance δ_2.


δ_2
δ_1
Command direction
δ_1 : Approach distance
δ_2 : Imcomplete threading portion for chamfering
TEP217

Approach distance (δ_1)

- When Ts $\neq$ 0

$$\delta_1 = \frac{F}{60} \bullet t_1 - \frac{F}{60} \left(Ts + Tp - \frac{Tp^2 \bullet e^{-\frac{t_1}{Tp}} - Ts^2 \bullet e^{-\frac{t_1}{Tp}}}{Tp - Ts}\right) \quad [\mathrm{mm}]$$

where F : Threading speed (mm/min)
Ts : Acceleration/deceleration time constant (sec)
Tp : Position loop time constant (sec)
t_1 : Time required for pitch error to be reduced within tolerated accuracy “a” (sec)

Tolerance accuracy, however, may be expressed by equation:

$$a = \frac{\Delta P}{P} = \frac{1}{Tp - Ts} \left(Tp^2 \bullet e^{-\frac{t_1}{Tp}} - Ts^2 \bullet e^{-\frac{t_1}{Tp}}\right) \quad [\mathrm{mm}]$$

where P : Pitch
ΔP : Pitch error

- When Ts = 0

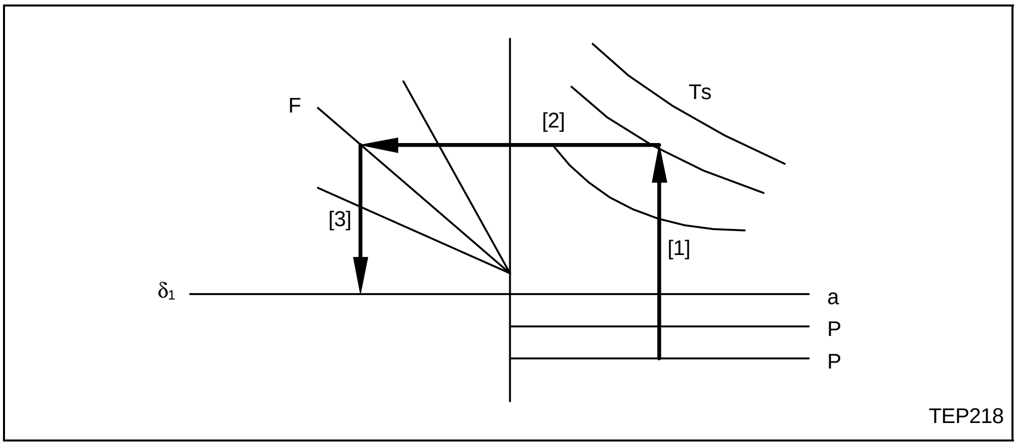
$$\delta_1 = \frac{F}{60} \bullet t_1 - \frac{F}{60} \left(Tp - Tp \bullet e^{-\frac{t_1}{Tp}}\right) \quad [\mathrm{mm}]$$

$$a = e^{-\frac{t_1}{Tp}}$$

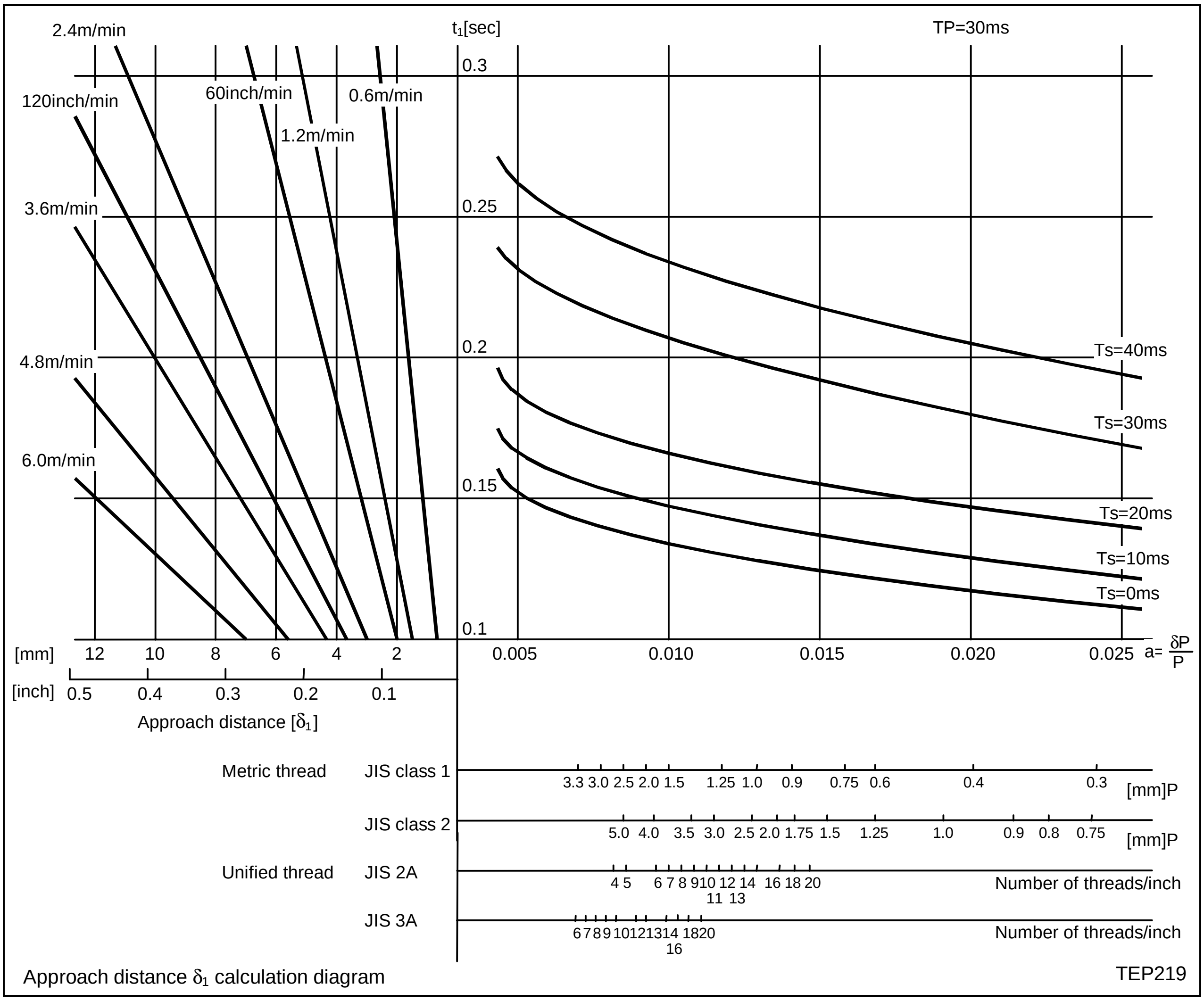
Return to Main Menu

Usually approach distance δ_1 is determined from the calculation diagram because its calculation is complicated. The diagram is used as below.

1) First obtain a crossing point on the acceleration/deceleration time constant curve (Ts) along the line [1] from a point on the P-axis determined by thread class and pitch (P).
2) Obtain a crossing point on thread cutting speed line (F) along the line [2].
3) Obtain approach distance (δ_1) by crossing point on the δ_1 axis along the line [3].



Note: The calculation diagram shown below is for position loop time constant Tp = 30 msec.



Approach distance δ_1 calculation diagram

TEP219

5. Additional Explanation About Y-axis

For a machine equipped with inclined Y-axis, the Y-axis is constructed by combination of an axis called Yt-axis and X-axis. However, when Y is commanded in EIA/ISO programming, movement will be made in direction of Y-axis perpendicular to X- and Z-axes.

When movement is made in direction not of Y-axis but of Yt-axis only, M348 is used as follows.

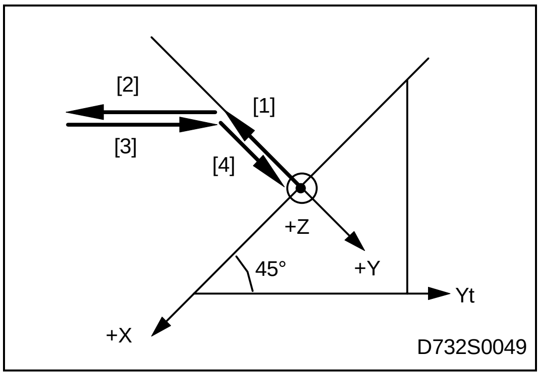
When the power is turned on, M348 mode is established. When X-axis and Yt-axis zero point return have been completed respectively, M349 mode is automatically established.

M348: Yt-axis mode (Y-axis mode cancel)

M349: Y-axis mode

Example:

```
G00  Y–10.;....................[1]
     M348;
     Y–10.;....................[2]
     Y+10.;....................[3]
     M349;
     Y+10.;....................[4]
M30;
```


[2]
[1]
[3]
[4]
+Z
45°
+Y
Yt
+X
D732S0049