FANUC

GE Fanuc Automation

Computer Numerical Control Products

Series 15 / 150 – Model B for Machining Center

Operator's Manual (Programming)

GFZ-62564E/02 January 1997

Warnings, Cautions, and Notes as Used in this Publication

Warning

Warning notices are used in this publication to emphasize that hazardous voltages, currents, temperatures, or other conditions that could cause personal injury exist in this equipment or may be associated with its use.

In situations where inattention could cause either personal injury or damage to equipment, a Warning notice is used.

Caution

Caution notices are used where equipment might be damaged if care is not taken.

Note

Notes merely call attention to information that is especially significant to understanding and operating the equipment.

SAFETY PRECAUTIONS

This section describes the safety precautions related to the use of CNC units. It is essential that these precautions be observed by users to ensure the safe operation of machines equipped with a CNC unit (all descriptions in this section assume this configuration). Note that some precautions are related only to specific functions, and thus may not be applicable to certain CNC units.
Users must also observe the safety precautions related to the machine, as described in the relevant manual supplied by the machine tool builder. Before attempting to operate the machine or create a program to control the operation of the machine, the operator must become fully familiar with the contents of this manual and relevant manual supplied by the machine tool builder.

Contents

1 DEFINITION OF WARNING, CAUTION, AND NOTE

This manual includes safety precautions for protecting the user and preventing damage to the machine. Precautions are classified into Warning and Caution according to their bearing on safety. Also, supplementary information is described as a Note. Read the Warning, Caution, and Note thoroughly before attempting to use the machine.

WARNING

Applied when there is a danger of the user being injured or when there is a damage of both the user being injured and the equipment being damaged if the approved procedure is not observed.

CAUTION

Applied when there is a danger of the equipment being damaged, if the approved procedure is not observed.

NOTE

The Note is used to indicate supplementary information other than Warning and Caution.

❍ **Read this manual carefully, and store it in a safe place.**

2 GENERAL WARNINGS AND CAUTIONS

WARNING

1. Never attempt to machine a workpiece without first checking the operation of the machine. Before starting a production run, ensure that the machine is operating correctly by performing a trial run using, for example, the single block, feedrate override, or machine lock function or by operating the machine with neither a tool nor workpiece mounted. Failure to confirm the correct operation of the machine may result in the machine behaving unexpectedly, possibly causing damage to the workpiece and/or machine itself, or injury to the user.

2. Before operating the machine, thoroughly check the entered data.
Operating the machine with incorrectly specified data may result in the machine behaving unexpectedly, possibly causing damage to the workpiece and/or machine itself, or injury to the user.

3. Ensure that the specified feedrate is appropriate for the intended operation. Generally, for each machine, there is a maximum allowable feedrate. The appropriate feedrate varies with the intended operation. Refer to the manual provided with the machine to determine the maximum allowable feedrate. If a machine is run at other than the correct speed, it may behave unexpectedly, possibly causing damage to the workpiece and/or machine itself, or injury to the user.

4. When using a tool compensation function, thoroughly check the direction and amount of compensation.
Operating the machine with incorrectly specified data may result in the machine behaving unexpectedly, possibly causing damage to the workpiece and/or machine itself, or injury to the user.

5. The parameters for the CNC and PMC are factory–set. Usually, there is not need to change them. When, however, there is not alternative other than to change a parameter, ensure that you fully understand the function of the parameter before making any change.
Failure to set a parameter correctly may result in the machine behaving unexpectedly, possibly causing damage to the workpiece and/or machine itself, or injury to the user.

6. Immediately after switching on the power, do not touch any of the keys on the MDI panel until the position display or alarm screen appears on the CNC unit.
Some of the keys on the MDI panel are dedicated to maintenance or other special operations. Pressing any of these keys may place the CNC unit in other than its normal state. Starting the machine in this state may cause it to behave unexpectedly.

7. The operator’s manual and programming manual supplied with a CNC unit provide an overall description of the machine’s functions, including any optional functions. Note that the optional functions will vary from one machine model to another. Therefore, some functions described in the manuals may not actually be available for a particular model. Check the specification of the machine if in doubt.

WARNING

8. Some functions may have been implemented at the request of the machine–tool builder. When using such functions, refer to the manual supplied by the machine–tool builder for details of their use and any related cautions.

NOTE

Programs, parameters, and macro variables are stored in nonvolatile memory in the CNC unit. Usually, they are retained even if the power is turned off. Such data may be deleted inadvertently, however, or it may prove necessary to delete all data from nonvolatile memory as part of error recovery.
To guard against the occurrence of the above, and assure quick restoration of deleted data, backup all vital data, and keep the backup copy in a safe place.

3 WARNINGS AND CAUTIONS RELATED TO PROGRAMMING

This section covers the major safety precautions related to programming. Before attempting to perform programming, read the supplied operator's manual and programming manual carefully such that you are fully familiar with their contents.

WARNING

1. Coordinate system setting

If a coordinate system is established incorrectly, the machine may behave unexpectedly as a result of the program issuing an otherwise valid move command.
Such an unexpected operation may damage the tool, the machine itself, the workpiece, or cause injury to the user.

2. Positioning by nonlinear interpolation

When performing positioning by nonlinear interpolation (positioning by nonlinear movement between the start and end points), the tool path must be carefully confirmed before performing programming.
Positioning involves rapid traverse. If the tool collides with the workpiece, it may damage the tool, the machine itself, the workpiece, or cause injury to the user.

3. Function involving a rotation axis

When programming polar coordinate interpolation or normal–direction (perpendicular) control, pay careful attention to the speed of the rotation axis. Incorrect programming may result in the rotation axis speed becoming excessively high, such that centrifugal force causes the chuck to lose its grip on the workpiece if the latter is not mounted securely.
Such mishap is likely to damage the tool, the machine itself, the workpiece, or cause injury to the user.

4. Inch/metric conversion

Switching between inch and metric inputs does not convert the measurement units of data such as the workpiece origin offset, parameter, and current position. Before starting the machine, therefore, determine which measurement units are being used. Attempting to perform an operation with invalid data specified may damage the tool, the machine itself, the workpiece, or cause injury to the user.

5. Constant surface speed control

When an axis subject to constant surface speed control approaches the origin of the workpiece coordinate system, the spindle speed may become excessively high. Therefore, it is necessary to specify a maximum allowable speed. Specifying the maximum allowable speed incorrectly may damage the tool, the machine itself, the workpiece, or cause injury to the user.

WARNING

6. Stroke check

After switching on the power, perform a manual reference position return as required. Stroke check is not possible before manual reference position return is performed. Note that when stroke check is disabled, an alarm is not issued even if a stroke limit is exceeded, possibly damaging the tool, the machine itself, the workpiece, or causing injury to the user.

7. Tool post interference check

A tool post interference check is performed based on the tool data specified during automatic operation. If the tool specification does not match the tool actually being used, the interference check cannot be made correctly, possibly damaging the tool or the machine itself, or causing injury to the user.
After switching on the power, or after selecting a tool post manually, always start automatic operation and specify the tool number of the tool to be used.

8. Absolute/incremental mode

If a program created with absolute values is run in incremental mode, or vice versa, the machine may behave unexpectedly.

9. Plane selection

If an incorrect plane is specified for circular interpolation, helical interpolation, or a canned cycle, the machine may behave unexpectedly. Refer to the descriptions of the respective functions for details.

10. Torque limit skip

Before attempting a torque limit skip, apply the torque limit. If a torque limit skip is specified without the torque limit actually being applied, a move command will be executed without performing a skip.

11. Programmable mirror image

Note that programmed operations vary considerably when a programmable mirror image is enabled.

12. Compensation function

If a command based on the machine coordinate system or a reference position return command is issued in compensation function mode, compensation is temporarily canceled, resulting in the unexpected behavior of the machine.
Before issuing any of the above commands, therefore, always cancel compensation function mode.

4 WARNINGS AND CAUTIONS RELATED TO HANDLING

This section presents safety precautions related to the handling of machine tools. Before attempting to operate your machine, read the supplied operator's manual and programming manual carefully, such that you are fully familiar with their contents.

WARNING

1. Manual operation

When operating the machine manually, determine the current position of the tool and workpiece, and ensure that the movement axis, direction, and feedrate have been specified correctly. Incorrect operation of the machine may damage the tool, the machine itself, the workpiece, or cause injury to the operator.

2. Manual reference position return

After switching on the power, perform manual reference position return as required. If the machine is operated without first performing manual reference position return, it may behave unexpectedly. Stroke check is not possible before manual reference position return is performed.
An unexpected operation of the machine may damage the tool, the machine itself, the workpiece, or cause injury to the user.

3. Manual numeric command

When issuing a manual numeric command, determine the current position of the tool and workpiece, and ensure that the movement axis, direction, and command have been specified correctly, and that the entered values are valid.
Attempting to operate the machine with an invalid command specified may damage the tool, the machine itself, the workpiece, or cause injury to the operator.

4. Manual handle feed

In manual handle feed, rotating the handle with a large scale factor, such as 100, applied causes the tool and table to move rapidly. Careless handling may damage the tool and/or machine, or cause injury to the user.

5. Disabled override

If override is disabled (according to the specification in a macro variable) during threading, rigid tapping, or other tapping, the speed cannot be predicted, possibly damaging the tool, the machine itself, the workpiece, or causing injury to the operator.

6. Origin/preset operation

Basically, never attempt an origin/preset operation when the machine is operating under the control of a program. Otherwise, the machine may behave unexpectedly, possibly damaging the tool, the machine itself, the tool, or causing injury to the user.

WARNING

7. Workpiece coordinate system shift

Manual intervention, machine lock, or mirror imaging may shift the workpiece coordinate system. Before attempting to operate the machine under the control of a program, confirm the coordinate system carefully.
If the machine is operated under the control of a program without making allowances for any shift in the workpiece coordinate system, the machine may behave unexpectedly, possibly damaging the tool, the machine itself, the workpiece, or causing injury to the operator.

8. Software operator's panel and menu switches

Using the software operator's panel and menu switches, in combination with the MDI panel, it is possible to specify operations not supported by the machine operator's panel, such as mode change, override value change, and jog feed commands.
Note, however, that if the MDI panel keys are operated inadvertently, the machine may behave unexpectedly, possibly damaging the tool, the machine itself, the workpiece, or causing injury to the user.

9. Manual intervention

If manual intervention is performed during programmed operation of the machine, the tool path may vary when the machine is restarted. Before restarting the machine after manual intervention, therefore, confirm the settings of the manual absolute switches, parameters, and absolute/incremental command mode.

10. Feed hold, override, and single block

The feed hold, feedrate override, and single block functions can be disabled using custom macro system variable #3004. Be careful when operating the machine in this case.

11. Dry run

Usually, a dry run is used to confirm the operation of the machine. During a dry run, the machine operates at dry run speed, which differs from the corresponding programmed feedrate. Note that the dry run speed may sometimes be higher than the programmed feed rate.

12. Cutter and tool nose radius compensation in MDI mode

Pay careful attention to a tool path specified by a command in MDI mode, because cutter or tool nose radius compensation is not applied. When a command is entered from the MDI to interrupt in automatic operation in cutter or tool nose radius compensation mode, pay particular attention to the tool path when automatic operation is subsequently resumed. Refer to the descriptions of the corresponding functions for details.

13. Program editing

If the machine is stopped, after which the machining program is edited (modification, insertion, or deletion), the machine may behave unexpectedly if machining is resumed under the control of that program. Basically, do not modify, insert, or delete commands from a machining program while it is in use.

5 WARNINGS RELATED TO DAILY MAINTENANCE

WARNING

1. Memory backup battery replacement

When replacing the memory backup batteries, keep the power to the machine (CNC) turned on, and apply an emergency stop to the machine. Because this work is performed with the power on and the cabinet open, only those personnel who have received approved safety and maintenance training may perform this work.
When replacing the batteries, be careful not to touch the high–voltage circuits (marked ⚠ and fitted with an insulating cover).
Touching the uncovered high–voltage circuits presents an extremely dangerous electric shock hazard.

NOTE

The CNC uses batteries to preserve the contents of its memory, because it must retain data such as programs, offsets, and parameters even while external power is not applied.
If the battery voltage drops, a low battery voltage alarm is displayed on the machine operator's panel or CRT screen.
When a low battery voltage alarm is displayed, replace the batteries within a week. Otherwise, the contents of the CNC's memory will be lost.
Refer to the maintenance section of the operator's manual or programming manual for details of the battery replacement procedure.

WARNING

2. Absolute pulse coder battery replacement

When replacing the memory backup batteries, keep the power to the machine (CNC) turned on, and apply an emergency stop to the machine. Because this work is performed with the power on and the cabinet open, only those personnel who have received approved safety and maintenance training may perform this work.
When replacing the batteries, be careful not to touch the high–voltage circuits (marked ⚠ and fitted with an insulating cover).
Touching the uncovered high–voltage circuits presents an extremely dangerous electric shock hazard.

NOTE

The absolute pulse coder uses batteries to preserve its absolute position.
If the battery voltage drops, a low battery voltage alarm is displayed on the machine operator's panel or CRT screen.
When a low battery voltage alarm is displayed, replace the batteries within a week. Otherwise, the absolute position data held by the pulse coder will be lost.
Refer to the maintenance section of the operator's manual or programming manual for details of the battery replacement procedure.

WARNING

3. Fuse replacement

For some units, the chapter covering daily maintenance in the operator's manual or programming manual describes the fuse replacement procedure.
Before replacing a blown fuse, however, it is necessary to locate and remove the cause of the blown fuse.
For this reason, only those personnel who have received approved safety and maintenance training may perform this work.
When replacing a fuse with the cabinet open, be careful not to touch the high–voltage circuits (marked and fitted with an insulating cover).
Touching an uncovered high–voltage circuit presents an extremely dangerous electric shock hazard.

CONTENTS

APPENDIX

I. GENERAL

1. GENERAL

The models covered by this manual, and their abbreviations are :

Product Name	Abbreviations	
FANUC Series 15–MB	15–MB	Series 15
FANUC Series 15–MFB	15–MFB	
FANUC Series 15MEK–MODEL B–4 (*)	15MEK	
FANUC Series 15MEL–MODEL B–4 (*)	15MEL	
FANUC Series 150–MB	150–MB	Series 150

(*) The FANUC Series 15MEK/MEL–MODEL B–4 is a software–fixed CNC capable of 4 contouring axes switchable out of 8 axes for milling machines and machining centers.
Further the following functions can not be used in the 15MEK or 15MEL.

- Increment system D/E (Increment system C is an option function)
- Helical interpolation B
- Plane switching
- Designation direction tool length compensation
- 2 axes electric gear box
- Manual interruption of 3–dimensional coordinate system conversion
- 3–dimensional cutter compensation
- Trouble diagnosis guidance
- OSI/ETHERNET function
- High–precision contour control using RISC
- Macro compiler (self compile function)
- MMC–III, MMC–IV
- Smooth interpolation
- Connecting for personal computer by high–speed serial–bus

FANUC Series 15–MFB is a CNC with a conversational function.

Refer to the following manuals for the conversational function.

- Conversational Automatic Programming Function for machining center Operator's manual (B–61264E)
- Function production manual for conversational function (B–61263E)

MMC software is created by the machine tool builder like the programmable machine control (PMC) software. Since the specifications of MMC depend on the specifications of the machine operator's panel, be sure to refer to the MMC manual issued by the machine tool builder.

This manual describes program coding, operation and daily maintenance.

For the parameter information, refer to the parameter Manual (B–62560E).

This manual also describes all the optional functions. For using functions for each CNC, refer to I.2 List of Specifications in the DESCRIPTIONS manual (B–62082E).

For options to be installed in the delivered unit, refer to the manual issued by the machine tool builder.

Manuals related to FANUC Series 15/150–MODEL B are as follows.
This manual is marked with an asterisk (*).

Table Related Manuals

Manual Name	Specification Number	
FANUC Series 15–TB/TFBTTB/TTFB DESCRIPTIONS	B–62072E	
FANUC Series 15/150–MB DESCRIPTIONS	B–62082E	
FANUC Series 15/150–MODEL B CONNECTION MANUAL	B–62073E	
FANUC Series 15–MODEL B CONNECTION MANUAL (BMI Interface)	B–62073E–1	
FANUC Series 15–MODEL B For Lathe OPERATOR’S MANUAL (Programming)	B–62554E	
FANUC Series 15–MODEL B For Lathe OPERATOR’S MANUAL (Operation)	B–62554E–1	
FANUC Series 15/150–MODEL B For Machining Center OPERATOR’S MANUAL (Programming)	B–62564E	*
FANUC Series 15/150–MODEL B For Machining Center OPERATOR’S MANUAL (Operation)	B–62564E–1	
FANUC Series 15–MODEL B PARAMETER MANUAL	B–62560E	
FANUC Series 15/150–MODEL B MAINTENANCE MANUAL	B–62075E	
FANUC Series 15–MODEL B DESCRIPTIONS (Suppelement for Remote Buffer)	B–62072E–1	
FANUC Series 15–MODEL B PROGRAMMING MANUAL (Macro Compiler / Macro Executer)	B–62073E–2	
FANUC Series 15–MB CONNECTION MANUAL (Multi–Teaching Function)	B–62083E–1	
FANUC PMC–MODEL N/NA PROGRAMMING MANUAL (Ladder Language)	B–61013E	
FANUC PMC–MODEL NB/NB2 PROGRAMMING MANUAL (Ladder Language)	B–61863E	
FANUC PMC–MODEL N/NA PROGRAMMING MANUAL (C Language)	B–61013E–2	
FANUC PMC–MODEL NB PROGRAMMING MANUAL (C Language)	B–61863E–1	
FANUC PMC–MODEL N/NA PROGRAMMING MANUAL (C Language – Tool Management Library)	B–61013E–4	
CONVERSATIONAL AUTOMATIC PROGRAMMING FUNCTION FOR MACHINING CENTER (Series 15–MFMFB) PROGRAMMING MANUAL	B–61263E	
CONVERSATIONAL AUTOMATIC PROGRAMMING FUNCTION FOR MACHINING CENTER (Series 15–MF/MFB) OPERATOR’S MANUAL	B–61264E	
CONVERSATIONAL AUTOMATIC PROGRAMMING FUNCTION FOR LATHE (Series 15–TF/TTF/TFB/TTFB) OPERATOR’S MANUAL	B–61234E	
CONVERSATIONAL AUTOMATIC PROGRAMMING FUNCTION II FOR LATHE (Series 15–TFB/TTFB) OPERATOR’S MANUAL	B–61804E–2	

1.1 General Flow of Operation of CNC Machine Tool

When machining the part using the CNC machine tool, first prepare the program, then operate the CNC machine by using the program.

1) First, prepare the program from a part drawing to operate the CNC machine tool.
   – How to prepare the program is described in the Chapter II. PROGRAMMING.
2) The program is to be read into the CNC system. Then, mount the workpieces and tools on the machine, and operate the tools according to the programming.Finally, execute the machining actually.
   – How to operate the CNC system is described in the Chapter III.OPERATION.

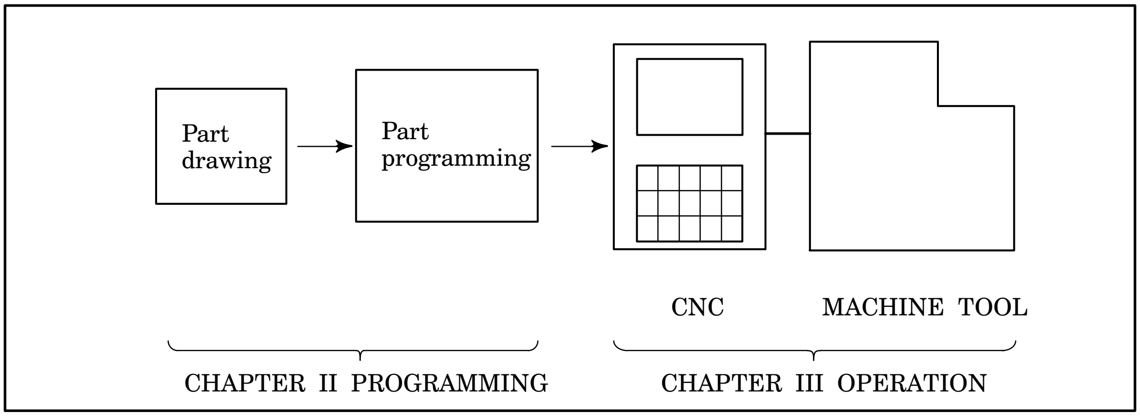


Fig. 1.1 Configuration of the manual

Before the actual programming, make the machining plan for how to machine the part.

Machining plan

1. Determination of workpieces machining range
2. Method of mounting workpieces on the machine tool
3. Machining sequence in every cutting process
4. Cutting tools and cutting conditions

Decline the cutting method in every cutting process.

Cutting process / Cutting procedure	1	2	3
	Face cutting	Side cutting	Hole machining
1.Cutting method : Rough Semi Finish			
2.Cutting tools :Tools			
3.Cutting conditions :Feedrate Cutting depth			
4.Tool path			

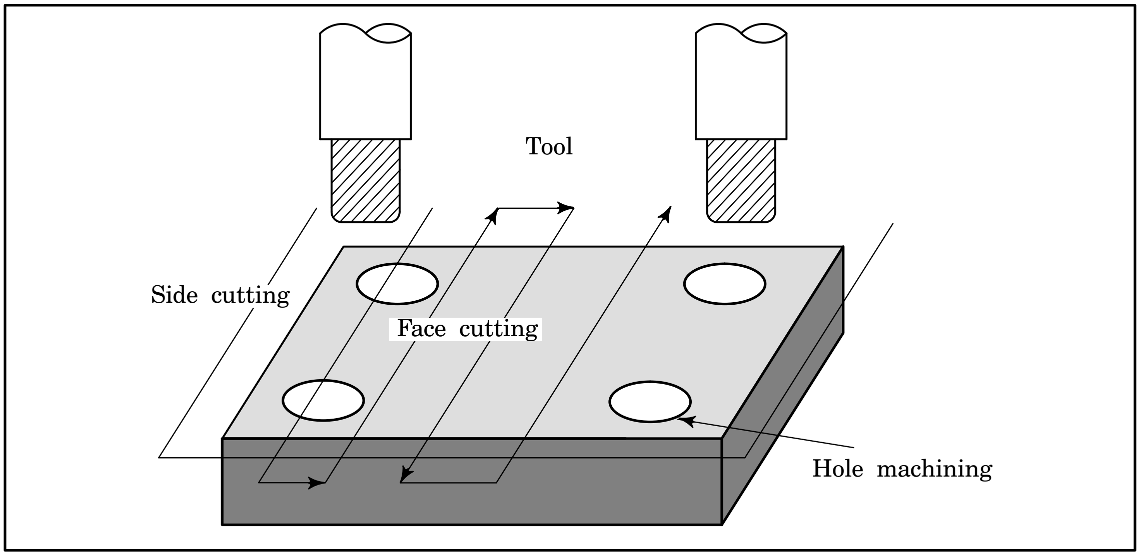


Fig. 1.2 Cutting type

Prepare the program of the tool path and cutting condition according to the workpiece figure, for each cutting.

1.2 Notes on Reading This Manual

1) The function of an CNC machine tool system depends not only on the CNC, but on the combination of the machine tool, its magnetic cabinet, the servo system, the CNC, the operator's panels, etc. It is too difficult to describe the function, programming, and operation relating to all combinations. This manual generally describes these from the stand–point of the CNC. So, for details on a particular CNC machine tool, refer to the manual issued by the machine tool builder, which should take precedence over this manual.
2) Notes refer to detailed and specific items. So, when a note is encountered, terms used in it sometimes are not explained. In this case, first skip the note, then return to it after having read over the manual for details.

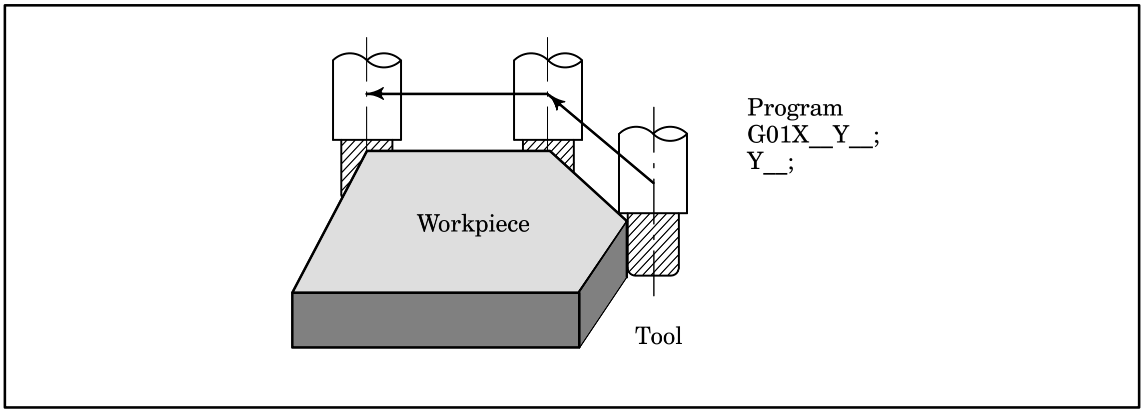
> Machining programs, parameters, variables, etc. are stored in the CNC unit internal non–volatile memory. In general, these contents are not lost by the switching ON/OFF of the power. However, it is possible that a state can occur where precious data stored in the non–volatile memory has to be deleted, because of deletions from a maloperation, or by a failure restoration. In order to restore rapidly when this kind of mishap occurs, it is recommended that you create a copy of the various kinds of data beforehand.

II. PROGRAMMING

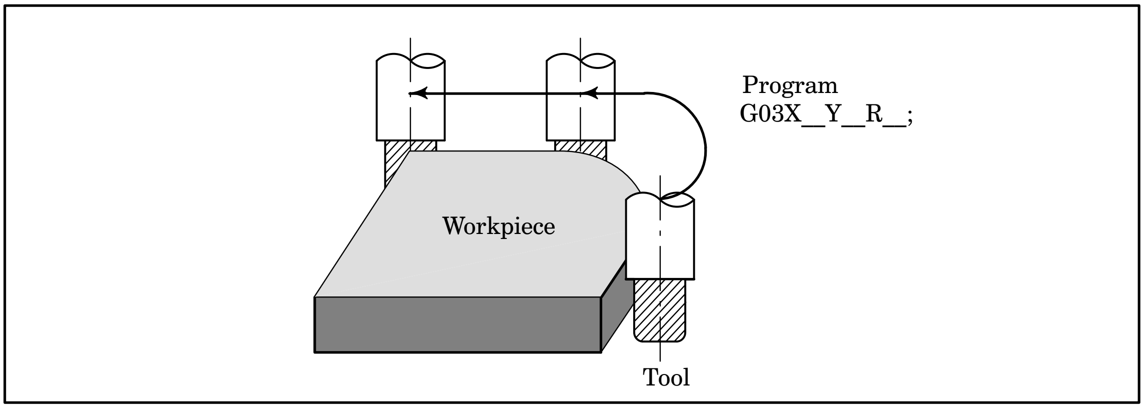
1. INTRODUCTION

1) Tool movement along workpiece parts figure–Interpolation (See II–4)
The tool moves along straight lines and arcs constituting the workpiece parts figure. (See Note)

a) Tool movement along straight line



b) Tool movement along arc



The function of moving the tool along straight lines and arcs is called the interpolation.
Symbols of the programmed commands G01, G02, ... are called the preparatory function and specify the type of interpolation conducted in the control unit.

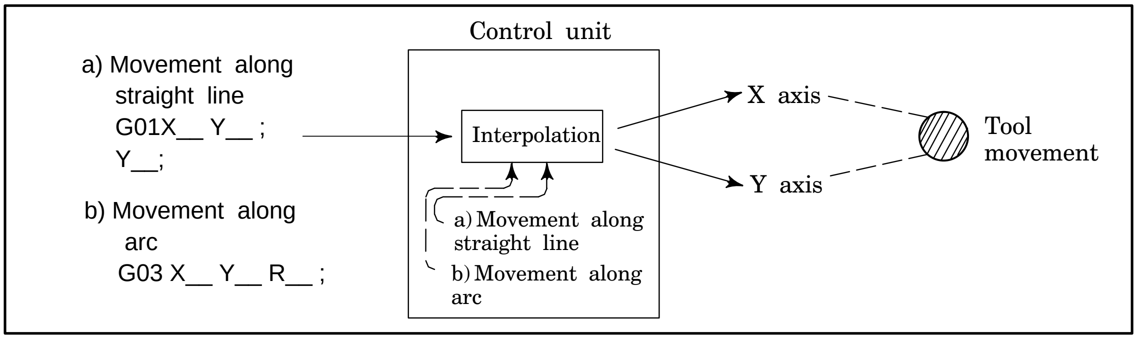


Fig. 1 (a) Interpolation

NOTE	Though the table may be moved without moving the tool in an actual machine, this manual assumes that the tool moves with respect to the workpiece.

2) Feed ________Feed function (See II–6)

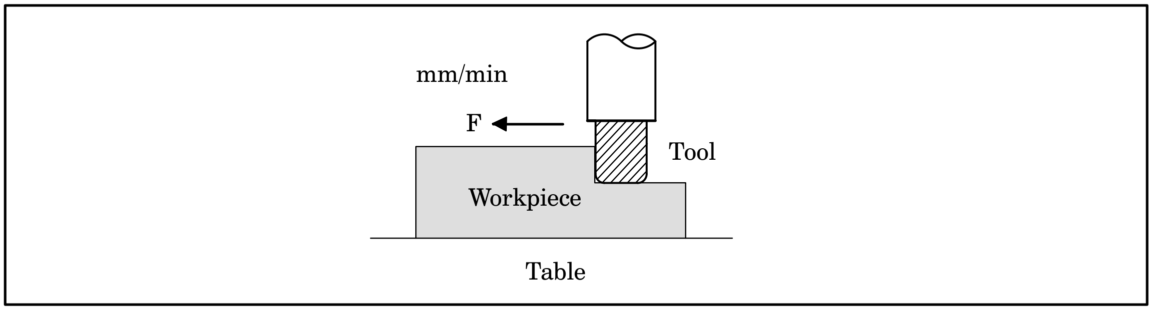


Fig. 1 (b) Feed function

Movement of the tool at a specified speed for cutting a workpiece is called the feed. Feedrates can be specified by using actual numerics. For example, to feed the tool at a rate of 150 mm/min, specify the following in the program:
F150.0
The function of deciding the feed rate is called the feed function.

3) Part drawing and tool movement

a) Reference position (fixed position on machine)

A CNC machine tool is provided with a fixed position. Normally, tool change and programming of absolute zero point as described later are performed at this position. This position is called the reference position.

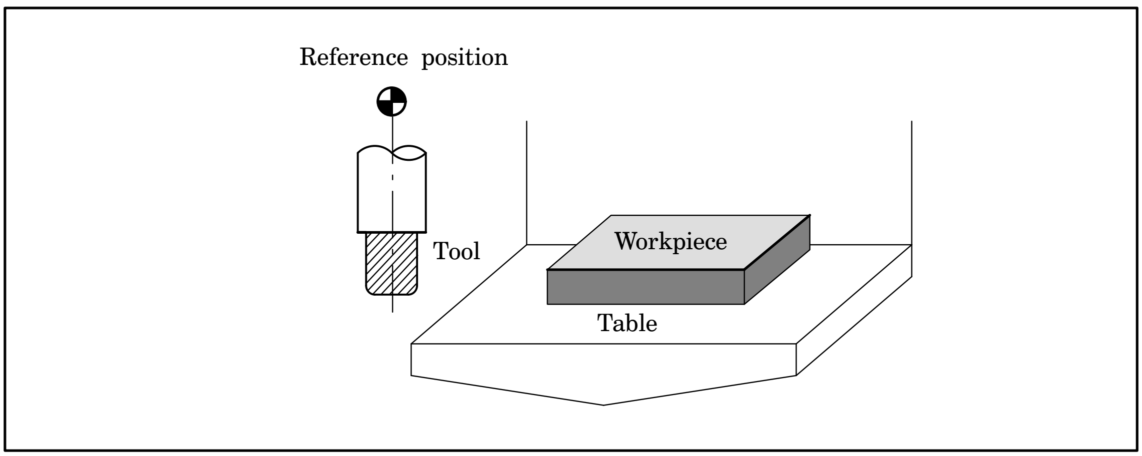


Fig. 1 (c) Reference position

The tool can be moved to the reference position in two ways:

i) Manual reference position return (See III–4.1)

Reference position return is performed by manual button operation.

ii) Automatic reference position return (See II–7.1)

Reference position return is performed in accordance with programmed commands.

In general, manual reference position return is performed first after the power is turned on. In order to move the tool to the reference position for tool change thereafter, the function of automatic reference position return is used.

b) Coordinate system on part drawing and coordinate system specified by CNC––– Coordinate system (See II–8)

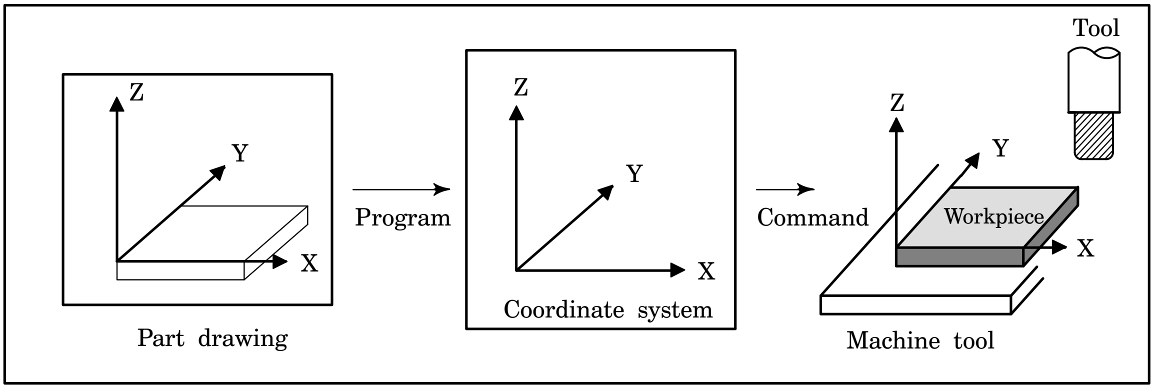


Fig. 1 (d) Coordinate system

There are two types of coordinate systems.

i) Coordinate system on part drawing

The coordinate system is written on the part drawing. As the program data, the coordinate values on this coordinate system are used.

ii) Coordinate system specified by CNC

The coordinate system is prepared on the machine tool table. This can be achieved by programming the distance from the present position of the tool to zero point of the coordinate to be set.

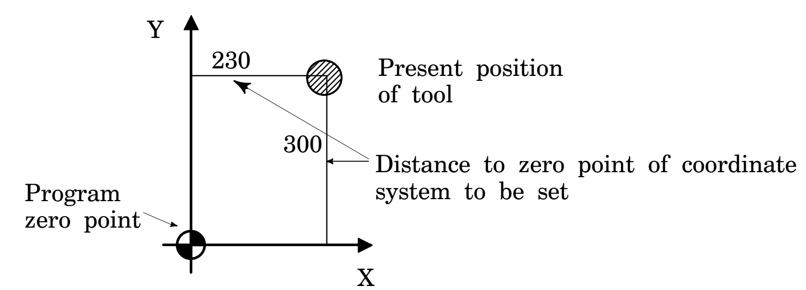
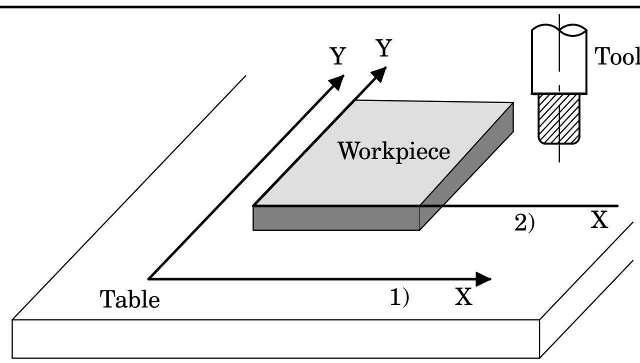


Fig. 1 (e) Coordinate system specified by CNC

When a workpiece is set on the table, these two coordinate systems lay as follows:



1) Coordinate system specified by CNC established on the table
2) Coordinate system on part drawing established on the workpiece

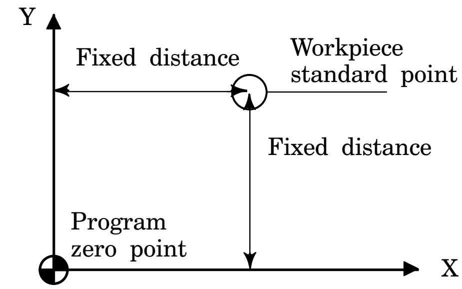
Fig. 1 (f) Coordinate system specified by CNC and coordinate system on part drawing

The tool moves on the coordinate system specified by the CNC in accordance with the command program generated with respect to the coordinate system on the part drawing, and cut a workpiece into a shape on the drawing.

Therefore, in order to correctly cut the workpiece as specified on the drawing, the two coordinate systems must be set at the same position. To set the two coordinate systems at the same position, simple methods shall be used according to workpiece shape, the number of machinings.

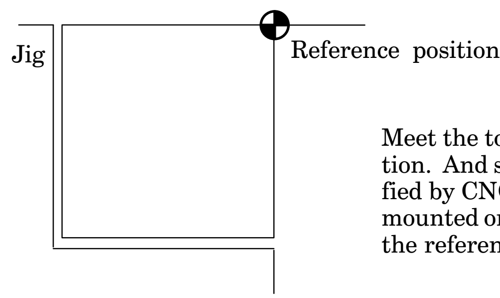
Some examples are shown below:

i) Using a standard point of the workpiece.



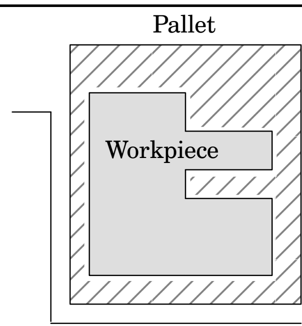
Bring the tool center to the standard point. And set the coordinate system specified by CNC at this position.

ii) Mounting a workpiece directly against the jig.



Meet the tool center to the reference position. And set the coordinate system specified by CNC at this position. (Jig shall be mounted on the predetermined point from the reference position.)

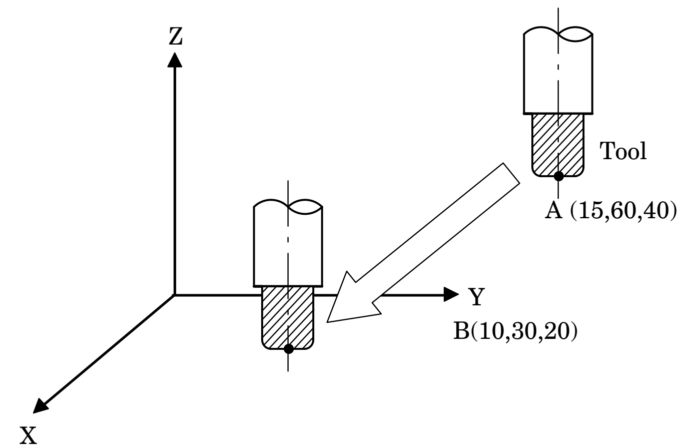
iii) Mounting a pallet with a workpiece against the jig.



Jig and coordinate system shall be specified the same as ii) .

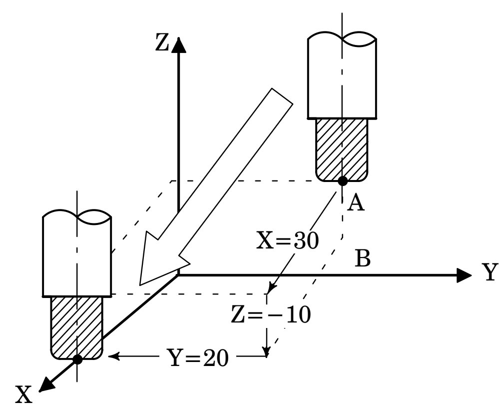
c) How to indicate command dimensions for moving the tool ––– Absolute, incremental commands (See II–9.1)
Coordinate values of command for moving the tool can be indicated by absolute or incremental designation.

i) Absolute coordinate values
The tool moves to a point at “the distance from zero point of the coordinate system,” i.e. to the position of the coordinate values.



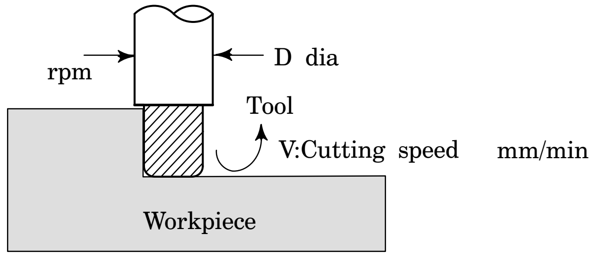
Specify the tool movement from point A to point B by using the coordinate values of point B as follows:
G90X10.0Y30.0Z20.0;

ii) ncremental coordinate values
Specify the distance from the previous tool position to the next tool position.



Specify the tool movement from point A to point B as follows:
G91X40.0Y–30.0Z–10.0;

4) Cutting speed ––––– Spindle speed function (See II–10)

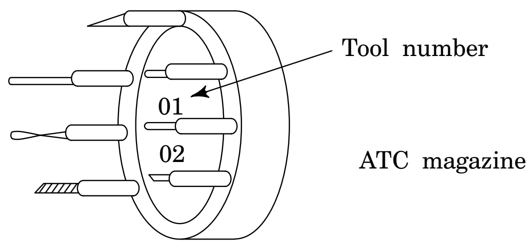


The speed of the tool with respect to the workpiece when the workpiece is cut is called the cutting speed.

As for the CNC, the cutting speed can be specified by the spindle speed in rpm unit.

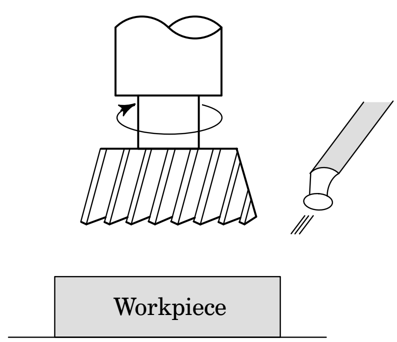
For example, suppose that a workpiece should be machined with a tool 100 mm in diameter at a cutting speed of 80 mm/min. Let N and D the spindle speed and diameter respectively. One revolution of the spindle causes the spindle surface to travel πD mm.
So when the speed is V m/min, $N \times \pi \times D = 1000V$. Therefore, approximately 250 rpm is obtained from $N = 1000V/\pi D$. Hence the following command is required :
S250;
Spindle speed command is called spindle speed function.

5) Selection of tool used for various machining–Tool Function (See II–11)



When drilling, tapping, bearing, milling or the like, is performed, it is necessary to select a suitable tool. When a number is assigned to each tool and the number is specified in the program, the CNC selects the corresponding tool.
For example, when No.01 is assigned to a drilling tool and the tool is stored at No.01 of the ATC magazine, the tool can be selected by specifying:**T01**

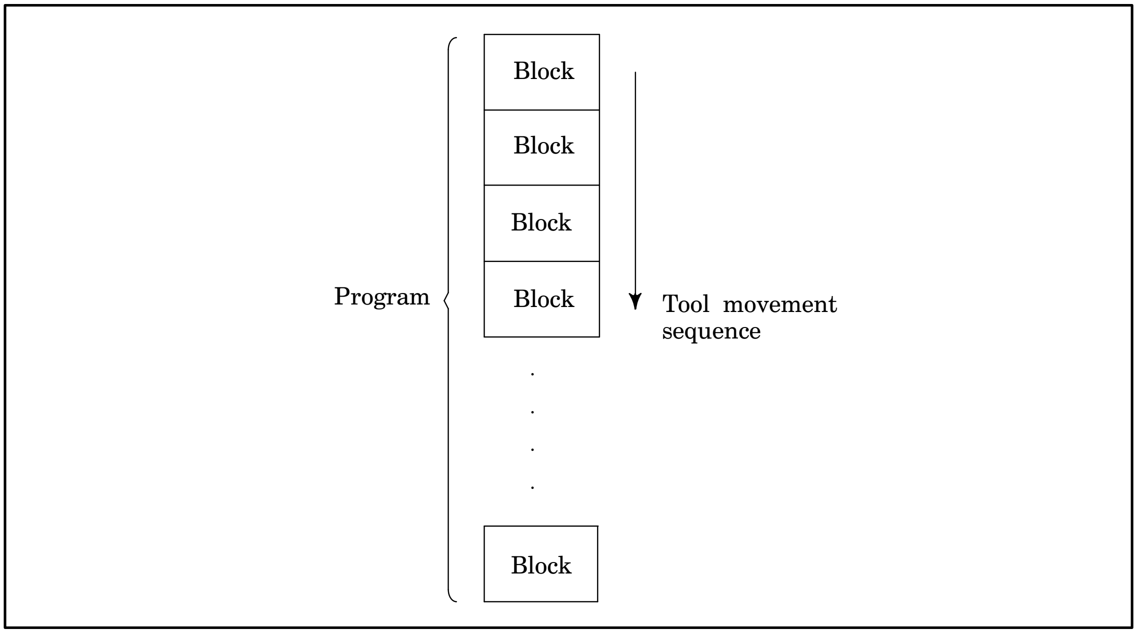
6) Command for machine operations– Miscellaneous function (See II–12)
When machining is actually started, it is necessary to rotate the spindle, and feed coolant. For this purpose, on–off operations of spindle motor and coolant valve should be controlled.



The function of specifying the on–off operations of the components of the machine is called the miscellaneous function. In general, the function is specified by an M code.
For example, when M03 is specified, the spindle is rotated clockwise.

7) Program configuration (See II–13)

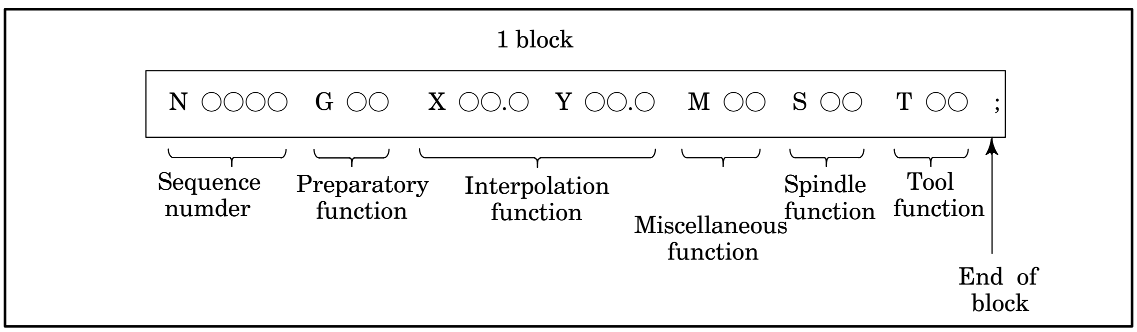
A group of commands given to the CNC for operating the machine is called the program. By specifying the commands, the tool is moved along a straight line or an arc, or the spindle motor is turned on and off. In the program, specify the commands in the sequence of actual tool movements.



A group of commands at each step of the sequence is called the block. The program consists of a group of blocks for a series of machining. The number for discriminating each block is called the sequence number, and the number for discriminating each program is called the program number.

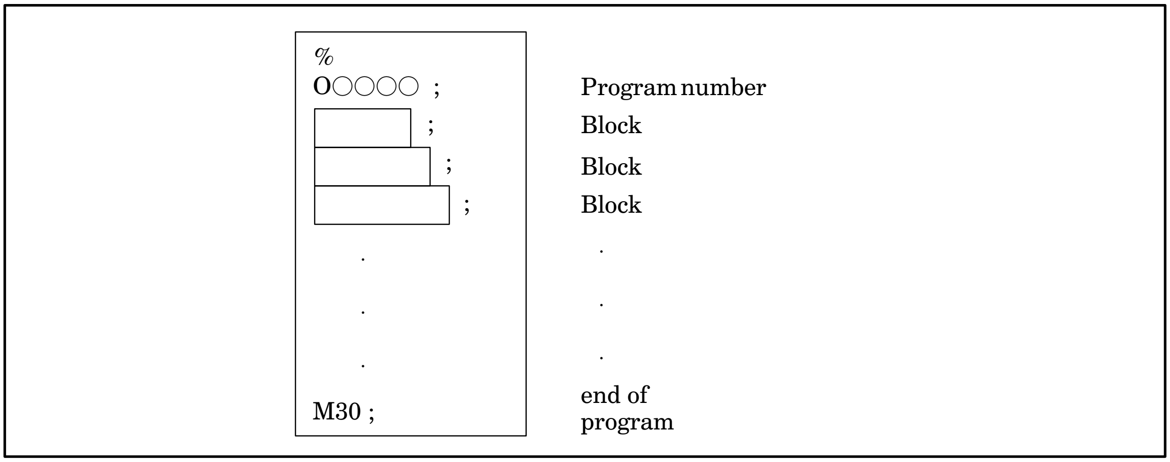
The block and the program have the following configurations.

a) Block



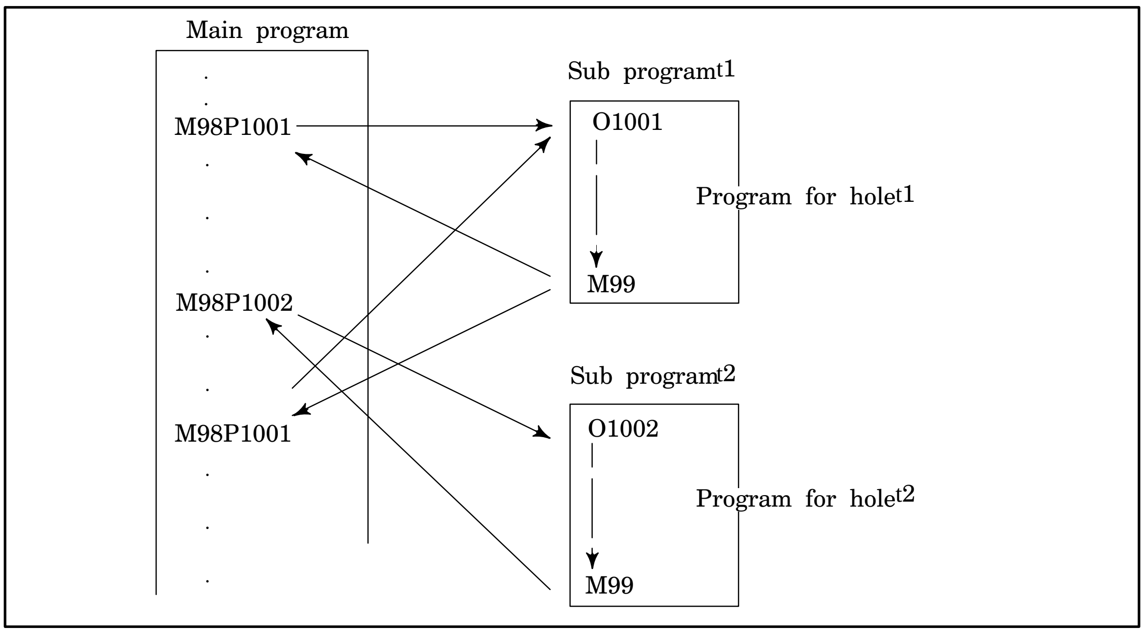
Each block consists of a sequence number for indicating the CNC operation sequence at the beginning of the block, and ; (end of block) for indicating the end of the block.

b) Program

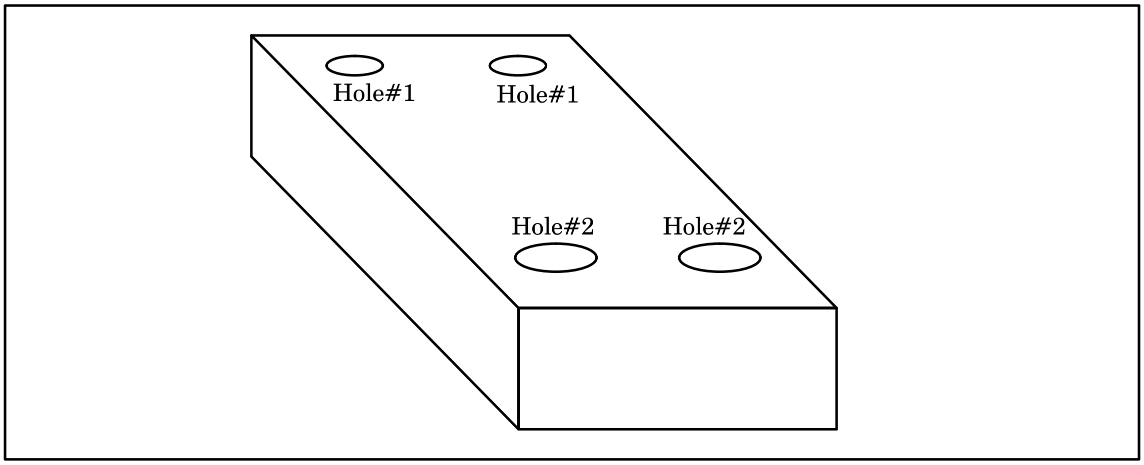


Normally, a program number is specified after the % at the beginning of the program, and a program end code (M02, M30) is specified at the end of the program.

c) Main program and subprogram



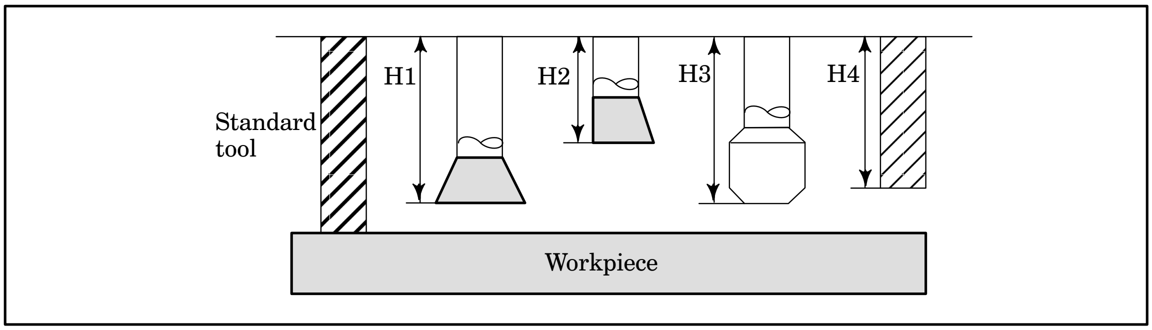
When machining of the same pattern appears at many portions of a program, a program for the pattern is created. This is called the subprogram. On the other hand, the original program is called the main program. When a subprogram execution command appears during execution of the main program, commands of the subprogram are executed. When execution of the subprogram is finished, the sequence returns to the main program.



8) Tool figure and tool motion by program (See II–15.1)

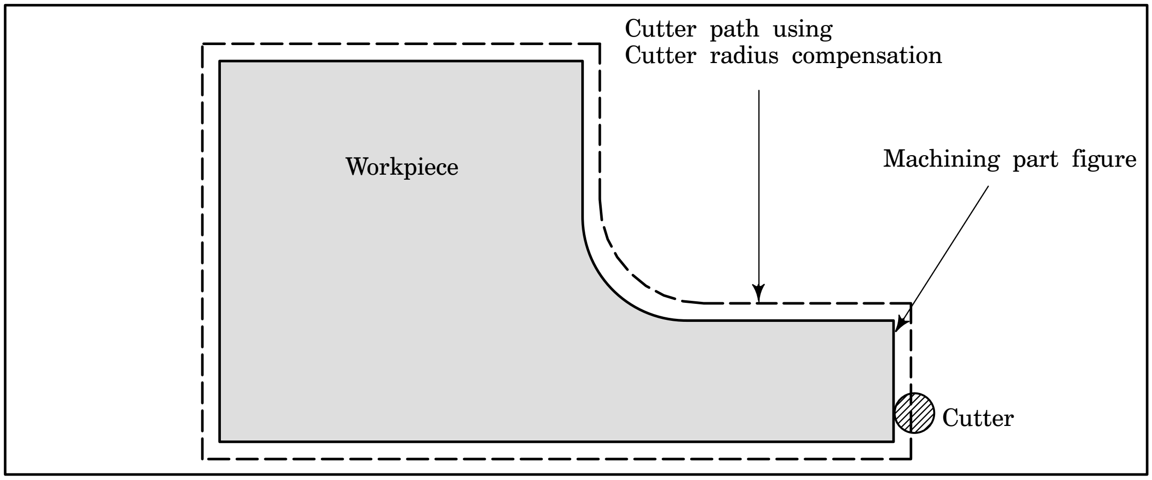
a) Machining using the end of cutter– Tool length compensation function.

Usually, several tools are used for machining one workpiece. The tools have different tool length. It is very troublesome to change the program in accordance with the tools.



Therefore, a standard tool is selected, and the difference between the position of the nose of the standard tool and the position of the nose of each tool used is measured in advance. By setting the measured value in the CNC (data display and setting: see III–10), machining can be performed without altering the program even when the tool is changed.
This function is called the tool length compensation.

b) Machining using the side of cutter ––––– Cutter compensation (See II–15.3)
Because a cutter has a radius, the center of the cutter path goes around the workpiece with the cutter radius deviated.

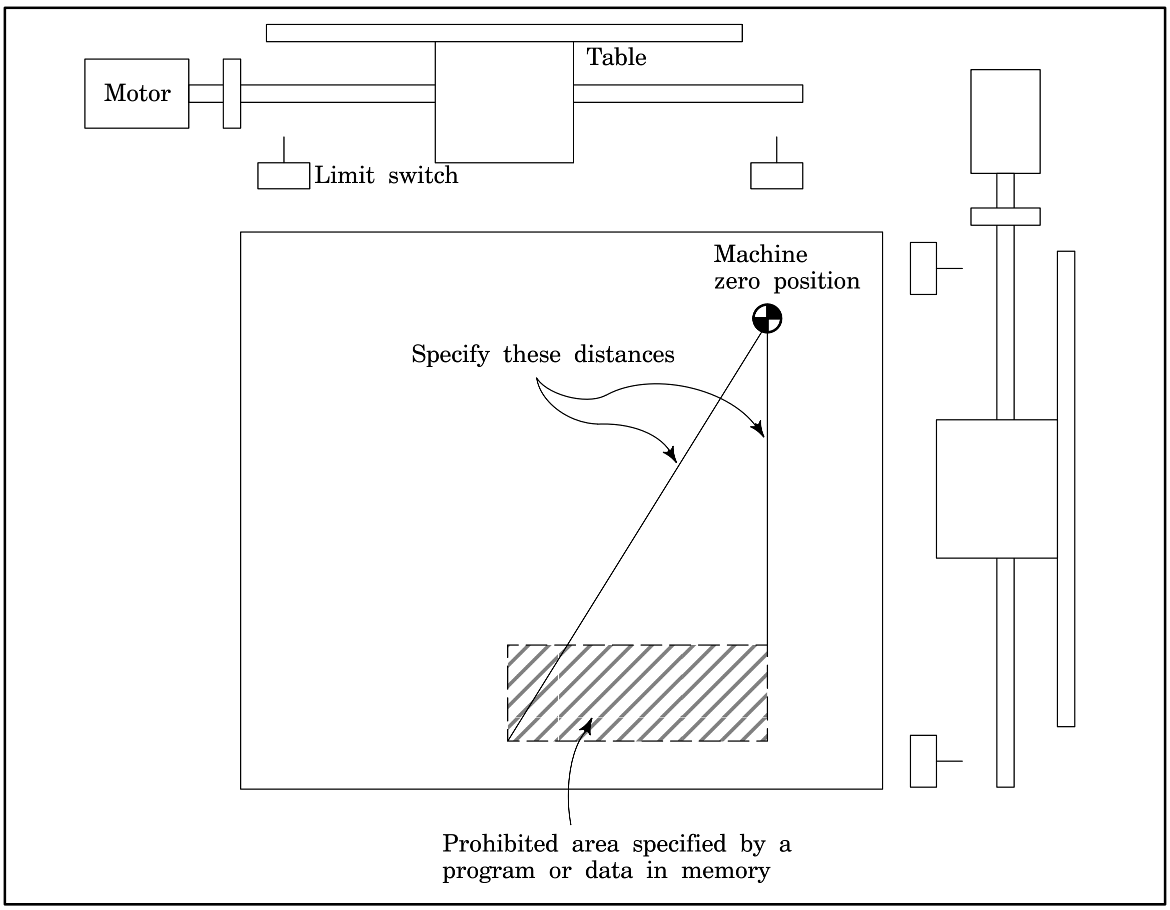


If radius of cutters are stored in memory (data display and setting: See III–10), the tool can be moved by cutter radius apart from the machining part figure. This function is called cutter compensation.

9) Range of tool movement ——— Stroke check (See II–19)

Limit switches are provided at both ends of each machine axis to prevent a tool from moving beyond the ends of the machine.

The range in which a tool is allowed to move is called a stroke.



Besides the stroke determined by the limit switches, an area the tool is not allowed to enter can be defined by a program or data in memory. (See III–10). This function is called the stroke check.

2. CONTROLLED AXES

2.1 Controlled Axes

Table 2.1 Number of controlles axes

No. of basic controlled axes	3 axes
Controlled axes expansion	Max 7. axes (Max. 10 axes (Cs axis is 2 axes) in total)
Basic simultaneously controlled axes	2 axes
Simultaneosly controlled axes expansion	Positioning, linear interpolation, jog feed and incremental feed are controlled all axes at a time.

2.2 Name of Axes

Names of axes can be optionally selected from X, Y, Z, A, B, C, U, V, and W. They can be set by parameter No. 1020.

NOTE	If the optional axis name expansion function is used, I, J, K, and E can also be used as the names of axes. In a multiaxis machine with more than one head, movements along more than nine axes sometimes need to be specified independently at the same time. To do this, more than nine axis names are necessary. In this case, axis names I, J, K, and E can be used in addition to the nine axis names, A, B, C, U, V, W, X, Y, and Z. Therefore, the addresses set for names of axes can be expanded to 13 addresses.

When I, J, K, and E are used as the names of axes, these addresses have the following functions and restrictions:

(1) These addresses are addresses for coordinate words.

Example G17I–K– ; The I–K plane is selected.

(2) The numeric values to be specified must consist of up to 8 digits.

(3) A decimal point can be input.

If a decimal point is omitted, its position is determined according to the increment system of the axis for that address.

Example G00 E0.5 I 100 K 100.0;

(4) A signed value can be input.

Example G01 E–10.5 F100;

(5) When addresses are used for axis names, they are no longer available for other applications.

Address	G code or variable	Normal use	Used for controlled axes	Remarks
I,J,K	G02 G03	Center position of an arc	Coordinate words for I, J, and K	Use an R command to specify the center.
	G41 G42	Three–dimensional offset vector	Coordinate words for I, J, and K	Three–dimensional tool compensation is disabled.
	G76 G87	Canned cycle shift amount	Coordinate words for I, J, and K	An amount of shift cannot be specified.
	G22	Stroke limit coordinates	Stroke limit coordinates	A limit position cannot be specified.
	G65 G66 G66.1	Argument	Argument	The position of the decimal point is determined by the increment system.
E	G33	Screw pitch (number of thread for inch screws)	E–axis coordinate word	The number of threads for inch screws cannot be specified in G33 threading.
	#4108	Macro variable, address E continuous–state information	No meaning	Custom macro variable #4108 is unavailable.

(6) When I, J, K, or E is used as an axis name, the second miscellaneous function cannot be used.

2.3 Increment System

Least input increment	Least command increment	Maximum stroke	Code
0.01 mm	0.01 mm	999999.99 mm	IS–A
0.001 mm	0.001 mm	99999.999 mm	IS–B
0.0001 mm	0.0001 mm	9999.9999 mm	IS–C
0.00001 mm	0.00001 mm	9999.99999 mm	IS–D
0.000001 mm	0.000001 mm	999.999999 mm	IS–E
0.001 inch	0.001 inch	99999.999 inch	IS–A
0.0001 inch	0.0001 inch	9999.9999 inch	IS–B
0.00001 inch	0.00001 inch	999.99999 inch	IS–C
0.000001 inch	0.000001 inch	999.999999 inch	IS–D
0.0000001 inch	0.0000001 inch	99.9999999 inch	IS–E
0.001 deg	0.001 deg	99999.999 deg	IS–A
0.0001 deg	0.0001 deg	9999.9999 deg	IS–B
0.00001 deg	0.00001 deg	999.99999 deg	IS–C
0.000001 deg	0.000001 deg	999.999999 deg	IS–D
0.0000001 deg	0.0000001 deg	99.9999999 deg	IS–E

Five types of increment systems are provided. Increment system IS–A, IS–B, and IS–C can be specified for each axis by setting ISFx and ISRx of parameter No.1004. Increment system IS–D can be specified for each axis by setting ISDx of parameter No.1004. Increment system IS–E can be specified for each axis by setting ISEx of parameter No.1009. Metrlc systems and inch systems, however, cannot be specified for a machine at the same time. Functions, such as circular interpolation and cutter compensation, cannot be used for axes using different increment systems. Increment systems IS–D and IS–E are optional.

For IS–B and IS–C, parameter IPPx (data No. 1004, input unit: multiplied by 10) sets the increment systems as follows. For the settings of the increment systems, refer to the manual provided by the machine tool builder.

Least input increment	Least command increment	Maximum stroke	Code
0.01 mm	0.001 mm	99999.999 mm	
0.001 inch	0.0001 inch	99999.9999 inch	IS–B
0.01 deg	0.001 deg	99999.999 deg	
0.001 mm	0.0001 mm	9999.9999 mm	
0.0001 inch	0.00001 inch	999.99999 inch	IS–C
0.001 deg	0.0001 deg	9999.9999 deg	

2.4 Maximum Stroke

Maximum stroke = minimum command increment x 99999999 (999999999 for IS–D and IS–E)

See Section 2.3.

3. PREPARATORY FUNCTION (G FUNCTION)

A value following address G determines the meaning of the command for the block concerned. G codes are divided into the following two types:

Type	Meaning
One–shot G code	The G code is effective only in the block in which it is specified.
Continuous–state G code	The G code is effective until another G coed of the same group is specified.

Example

G01 and G00 are continuous–state G codes.

```
G 01  X —;  ⎫
      Z —;  ⎬  G01 is effective in this range.
      X —;  ⎭
G00   Z —;
```

NOTE 1 G codes marked ◤ are the initial G codes set when the power is turned on or when the system is reset. For G22 and G23, when the power has been turned on, G22 is set, and when the system is reset, the code immediately before the reset remains. For G00, G01, G17, G18, G43, G44, G49, G94, G95, G90, and G91, desired G codes can be selected by parameter setting of data No. 2401 (G01, G18, G43 or G44, G95, and G90). For G20 and G21, the system enters the state immediately before the power is turned off or the reset button is pressed.

NOTE 2 G codes of the 00 group are not continuous–state G codes. They are effective only within the block in which they are specified.

NOTE 3 If a G code not listed in the G code list is specified, alarm PS010 occurs. If a G code is specified though its associated option is not provided, alarm PS010 also occurs.

NOTE 4 Any number of G codes can be specified in the same block provided the G codes belong to different groups. If two or more G codes that belong to the same group are specified in the same block, the G code specified last is validated.

Table 3 G code list (continued)

Code	Group	Function	
G00	01	Positioning	
G01		Linear interpolation	
G02		Circular interpolation/helical interpolation /spiral interpolation/conical interpolation (clockwise)	
G03		Circular interpolation/helical interpolation/spiral interpolation/conical interpolation (counterclockwise)	
G02.1		Circular threading B (clockwise)	
G03.1		Circular threading B (counterclockwise)	
G02.2		Involute interpolation (clockwise)	
G03.2		Involute interpolation (counterclockwise)	
G02.3		Exponential interpolation (clockwise)	
G03.3		Exponential interpolation (counterclockwise)	
G02.4		3–dimensional circular interpolation	
G03.4		3–dimensional circular interpolation	
G06.1		Spline interpolation	
G04	00	Dwell	
G05.1		Multibuffer	
G07		Hypothetical axis interpolation	
G07.1		Cylindrical interpolation	
G09		Exact stop	
G10		Data setting	
G10.1		PMC Data setting	
G10.3		Start of high–speed machining registration	
G10.6		Tool retraction, retraction data	
G10.9		Programmable diameter/radius specification switching function	
G11		Data setting mode cancel	
G11.3		End of high–speed machining registration	
G12.1	26	Polar coordinate interpolation mode	
G13.1		Polar coordinate interpolation cancel mode	
G12.2	00	Full circle cuttig (clockwise)	
G13.2		Full circle cuttig (counterclockwise)	
G15	17	Polar coordinates command cancel	
G16		Polar coordinates command	
G17	02	Xp–Yp plane	where, Xp :X–axis or a parallel axis
G18		Zp–Xp plane	Yp :Y–axis or a parallel axis
G19		YP–Zp plane	Zp :Z–axis or a parallel axis
G20	06	Inch input	
G21		Metric input	
G22	04	Stored stroke check function on	
G23		Stored stroke check function off	
G25	25	Spindle speed fluctuation detection off	
G26		Spindle speed fluctuation detection on	
G27	00	Reference position return check	
G28		Return to reference position	

Table 3 G code list (continued)

Code	Group	Function
G29	00	Return from reference position
G30		Return to 2nd, 3rd, or 4th reference position
G30.1		Return to floating reference position
G31		Skip function
G31.1		Multistage skip function1
G31.2		Multistage skip function 2
G31.3		Multistage skip function 3
G33	01	Threading
G37	00	Automatic tool length measurement
G38		Cutter compensation C vector retention
G39		Cutter compensation C corner rounding
G40	07	Cutter compensation cancel/three–dimensional cutter compensation cancel
G41		Cutter compensation (left)/three–dimensional cutter compensation
G42		Cutter compensation (right)
G41.2		3 dimensional cutter compensation left
G42.2		3 dimensional cutter compensation right
G41.3		Leading edge offset
G40.1	19	Normal direction control cancel mode
G41.1		Normal direction control (left side on)
G42.1		Normal direction control (right side on)
G43	08	Tool length compensation (+ve)
G43.1		Tool length compensation in tool axis direction
G44		Tool length compensation (–ve)
G45	00	Tool offset increase
G46		Tool offset decrease
G47		Tool offset double increase
G48		Tool offset double decrease
G49	08	Tool length compensation cancel
G50	11	Scaling cancel
G51		Scaling
G50.1	18	Programmable mirror image cancel
G51.1		Programmable mirror image
G52	00	Local coordinate system setting
G53		Machine coordinate system selection
G54	14	Workpiece coordinate system 1 selection
G54.1		Additional workpiece coordinate system selection
G54.2		Fixture offset selection
G55		Workpiece coordinate system 2 selection
G56		Workpiece coordinate system 3 selection
G57		Workpiece coordinate system 4 selection
G58		Workpiece coordinate system 5 selection
G59		Workpiece coordinate system 6 selection
G60	00	Unidirectional positioning

Table 3 G code list (continued)

Code	Group	Function
G61	15	Exact stop mode
G62		Automatic corner override mode
G63		Tapping mode
G64		Cutting mode
G65	00	Macro call
G65.3		High–speed machining program call
G66	12	Macro continuous–state call A
G66.1		Macro continuous–state call B
G67		Macro continuous–state call A/B cancel
G68	16	Coordinate system rotation
G69		Coordinate system rotation cancel
G72.1	00	Rotation copy
G72.2		Linear copy
G73	09	Peck drilling cycle
G74		Counter tapping cycle
G76		Fine boring cycle
G80		Canned cycle cancel/external operation function cancel
		Electronic gear box sgnchronous cancel (Command for hobbing machine or 1 axis)
G81		Drill cycle, stop boring /external operation function
		Electronic gear box sgnchronous start (Command for hobbing machine or 1 axis)
G80.5		Electronic gear box sgnchronous cancel (Command for 2 axes)
G81.5		Electronic gear box sgnchronous start (Command for 2 axes)
G81.1	00	Chopping mode on
G82	09	Drill cycle, counter boring
G83		Peck drilling cycle
G84		Tapping cycle
G84.2		Rigid tapping cycle
G84.3		Reverse rigid tapping cycle
G85		Boring cycle
G86		Boring cycle
G87		Back boring cycle
G88		Boring cycle
G89		Boring cycle
G90	03	Absolute command
G91		Incremental command
G92	00	Change of workpiece coordinate system/maximum spindle speed setting
G92.1		Workpiece coordinate system preset
G93	05	Inverse time feed
G94		Feed per minute
G95		Feed per rotation
G96	13	Constant surface speed control
G97		Constant surface speed control cancel
G98	10	Return to initial level in canned cycle
G99		Return to R–point level in canned cycle

4. INTERPOLATION FUNCTIONS

4.1 Positioning (G00)

G00 specifies positioning.

A tool moves to a certain position in the work coordinate system with an absolute command or to a position specified distance from the current position with an incremental command at a rapid traverse rate.

Format

G00 IP –;

where IP –: Combination of optional axis address (of X, Y, Z, A, B, C, U, V, W) as X–Y–Z–A–...
This manual uses this notation hereinafter

; : End of block (LF for ISO code, CR for EIA code) This manual uses this notation hereinafter.

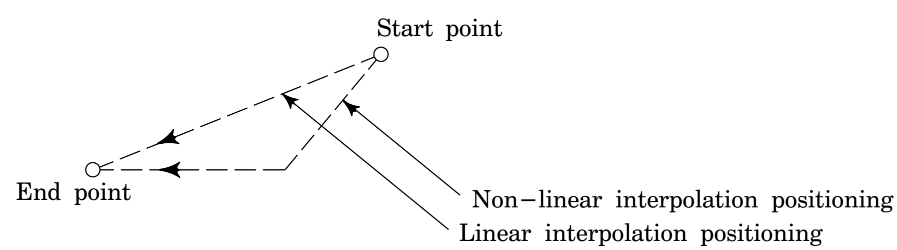
The tool path is determined by selecting one of the following with a parameter (data No. 1400, LRP):

– Linear interpolation type positioning

The tool path for positioning is determined in the same manner as linear interpolation (G01). The tool moves at an appropriate speed so that positioning can be performed in the shortest time without the speed exceeding the rapid traverse feedrate for each axis.

– Non–linear interpolation type positioning

Positioning is performed with each axis independently at the rapid traverse feedrate. Generally, the tool path is not a straight line.



NOTE 1 The rapid traverse rate in the G00 command is set for each axis independently by the machine tool builder (parameter No. 1420).

Accordingly, the rapid traverse rate cannot be specified in the address F.
In the positioning mode actuated by G00, the tool is accelerated to a predetermined speed at the start of a block and is decelerated at the end of a block. Execution proceeds to the next block after confirming the in–position. (See Note 2).

NOTE 2 "In–position" means that the feed motor is within the specified range.
(This range is determined by the machine tool builder) (Parameter No.1827).

4.2 Single Direction Positioning (G60)

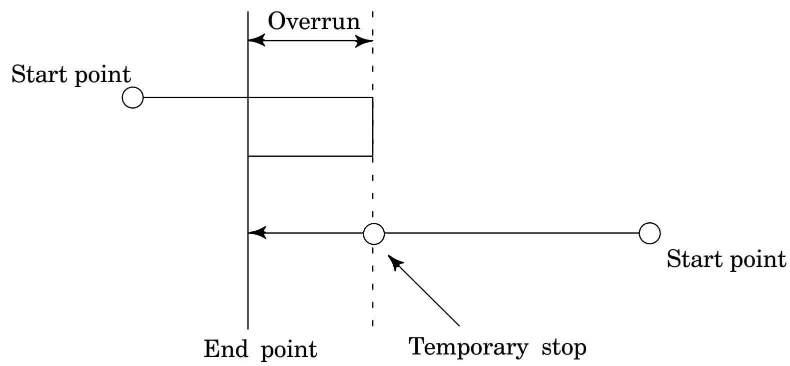
For accurate positioning without play of the machine (backlash), final positioning from only one direction is available.

(When final positioning is done from right to left)

(Direction for final positioning is right to left).
G60 is used insted of G00 as below.

G60IP ;

An overrun and a positioning direction are set by the parameter (No. 6820). Even when a commanded positioning direction coincides with that set by the parameter, the tool stops once before the end point.



NOTE 1 G60, which is a one–shot G–code, can be used as a modal G–code in group 01 by setting a parameter (No. 7616#0 MDL) to 1. Using G60 in such a way eliminates the need to issue it for individual blocks. Other specifications are the same, except for being modal.

Example

–Using G60 as a one–shot G–code

```
:
:
G90              ;
G60  X0.    Y0.  ;
G60  X100.       ;
G60         Y100.;
G04  X10.        ;
G00  X0.    Y0.  ;
:
:
```

–Using G60 as a modal G–code

```
:
:
G90  G60         ;
X0.         Y0.  ;
X100.            ;
            Y100 ;
G04  X10.        ;
G00  X0.    Y0.  ;
:
:
```

NOTE 2 During drilling canned cycle, no Single Direction Positioning is effected in Z.

NOTE 3 No Single Direction Positioning is effected in an axis for which no overrun has been set by the parameter.

NOTE 4 When the move distance 0 is commanded, the Single Direction Positioning is not performed.

NOTE 5 The direction set by the parameter is not effected by mirror image.

NOTE 6 The Single Direction Positioning does not apply to the shift motion in the canned cycles of G76 and G87.

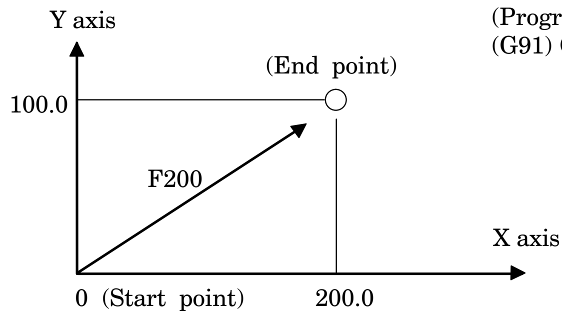
NOTE 7 The single Direction Positioning is always non linear interpolation type positioning.

NOTE 8 The parameter FPSN (data No. 1000) can be used to determine whether the in–position check is to be done or not at the temporary stop point.

4.3 Linear Interpolation (G01)

G01 IP _____F_____;

This command actuates the linear interpolation mode. The values of P define the distance of tool travel which will be conducted in absolute or incremental mode, according to the current status of G90/G91. The feed rate is set to a cutting feed speed commanded by F code and is a modal data.



(Program example)
(G91) G01 X200. 0 Y100. 0 F200. 0 ;

The feed rate commanded by the F code is measured along the tool path. If it is not commanded, the feed rate is regarded as zero.

NOTE 1 The feed rate of each axis direction is as follows.

G01 $\alpha\ \underline{\alpha}\ \beta\ \underline{\beta}\ \gamma\ \underline{\gamma}\ \zeta\ \underline{\zeta} \ldots F\ \underline{f}$;

Feed rate of α axis direction : $F\alpha = \frac{\alpha}{L} \cdot f$

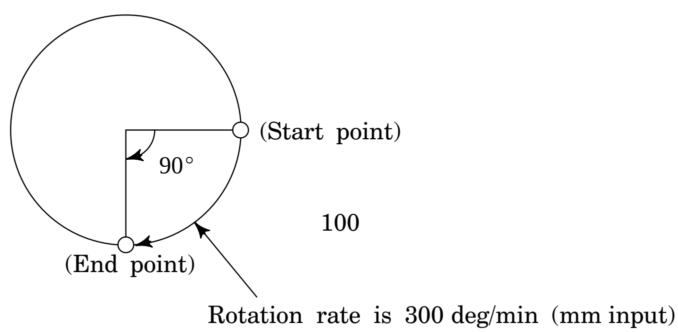
Feed rate of β axis direction : $F\beta = \frac{\beta}{L} \cdot f$

Feed rate of γ axis direction : $F\gamma = \frac{\gamma}{L} \cdot f$

Feed rate of ζ axis direction : $F\zeta = \frac{\zeta}{L} \cdot f$

$$L = \sqrt{\alpha^2 + \beta^2 + \gamma^2 + \zeta^2}$$

NOTE 2 The feed rate of the rotary axis is commanded in the unit of deg/min (the unit is decimal point position).
G91 G01 C–90. 0 F300. ;



NOTE 3 When linear interpolation is performed between a linear axis α (such as X, Y, or Z) and a rotation axis β (such as A, B, or C), the tangential feedrate in the α, β Cartesian coordinate system is specified by F α (mm/min). In this coordinate system, axis A, B, or C is in degrees, and axis X, Y, or Z is in mm or inches. The feedrate on the β axis is obtained by calculating the required time using the equation given in (Note 1) above, then converting the result to deg/min.

Example

G91 G01 X20.0 C40.0 F300.0 ;

When millimeter input is used in the above, 40.0 deg for the C axis is assumed to be 40 mm.

The time required for distribution is expressed as follows:

$$\frac{\sqrt{20^2 + 40^2}}{300} \fallingdotseq 0.14907\text{min}$$

The feedrate on the C axis is

$$\frac{40\,\text{deg}}{0.14907\,\text{min}} \fallingdotseq 268.3\text{deg/min}$$

NOTE 4 When performing linear interpolation between three axes simultaneously, consider the Cartesian coordinate system in the same manner as for simultaneous two–axis linear interpolation.

NOTE 5 For inch input and metric input, the upper limit of the feedrate on the rotation axis is approximately 6000 deg/mm (for the IS–B increment system). If a feedrate exceeding the upper limit is specified, the feedrate is clamped at the upper limit.

4.4 Circular Interpolation(G02, G03)

The command below will move a tool along a circular arc.

Arc on XpYp plane

$$\mathbf{G17}\begin{Bmatrix}\mathbf{G02}\\ \mathbf{G03}\end{Bmatrix}\mathbf{Xp\_\ Yp\_}\begin{Bmatrix}\mathbf{R\_\_}\\ \mathbf{I\_\ J\_}\end{Bmatrix}\quad\mathbf{F\_\ ;}$$

Arc on ZpXp plane

$$\mathbf{G18}\begin{Bmatrix}\mathbf{G02}\\ \mathbf{G03}\end{Bmatrix}\mathbf{Xp\_\ Zp\_}\begin{Bmatrix}\mathbf{R\_\_}\\ \mathbf{I\_\ K\_}\end{Bmatrix}\quad\mathbf{F\_\ ;}$$

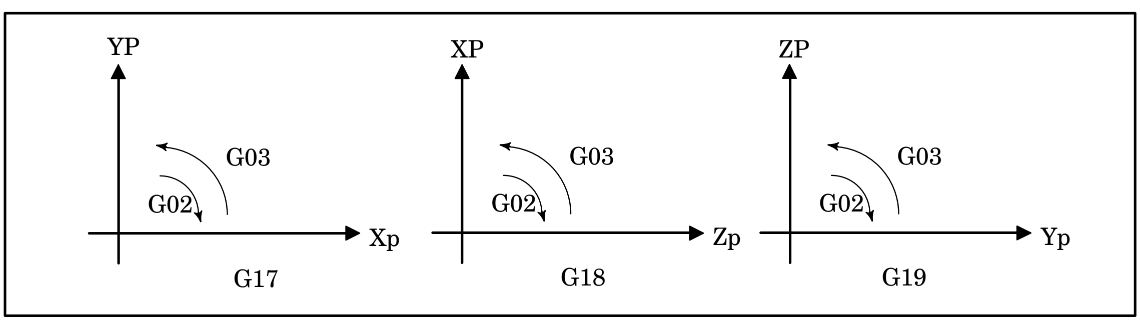
Arc on YpZp plane

$$\mathbf{G19}\begin{Bmatrix}\mathbf{G02}\\ \mathbf{G03}\end{Bmatrix}\mathbf{Yp\_\ Zp\_}\begin{Bmatrix}\mathbf{R\_\_}\\ \mathbf{J\_\ K\_}\end{Bmatrix}\quad\mathbf{F\_\ ;}$$

Xp: X axis or its parallel axis (set by parameter No. 1022)
Yp: Y axis or its parallel axis (set by parameter No. 1022)
Zp: Z axis or its parallel axis (set by parameter No. 1022)

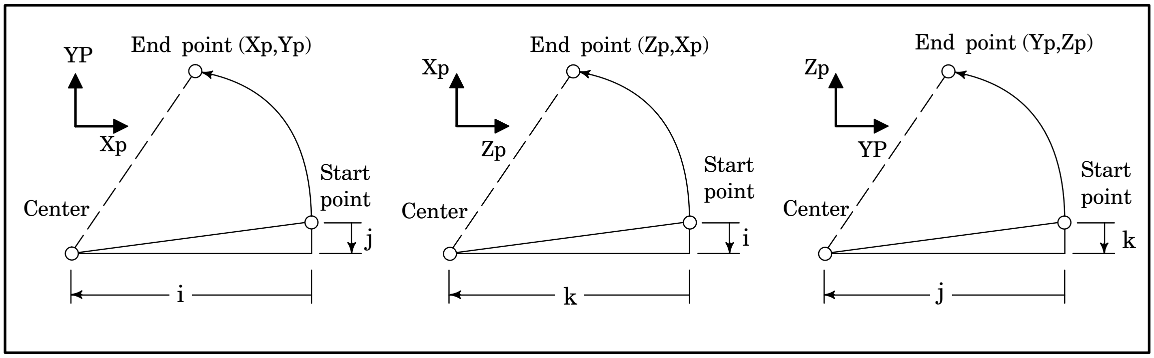
Date to be given			Command	Meaning
1	Plane selection		G17	Specification of arc on XpYp plane
			G18	Specification of arc on ZpXp plane
			G19	Specification of arc on YpZp plane
2	Direction of rotation		G02	Clockwise direction (CW)
			G03	Counterclockwise direction (CCW)
3	End point position	G90 mode	Two of the Xp, Yp, and Zp axes	End point position in the work coordinate system
		G91 mode	Two of the Xp, Yp, and Zp axes	Distance from start point to end-point
4	Distance from start point to center		Two of the I, J, and K axes	The signed distance from start point to center
	Arc radius		R	Arc radius (Fixed to radius designation)
5	Feedrate		F	Velocity along arc

The view is from the positive direction of the Zp axis (Yp axis or Xp axis) to the negative direction on XpYp plane (ZpXp plane or YpZp plane) in the right hand Cartesian coordinate system.



Clockwise and counterclockwise directions

The end point of an arc is specified by address Xp, Yp or Zp, and is expressed as an absolute or incremental value according to G90 or G91. For the incremental value, the coordinate of the end point which is viewed from the start point of the arc is specified. The arc center is specified by addresses I, J, and K for the Xp, Yp, and Zp axes, respectively. The numerical value following I, J, or K, however, is a vector component in which the arc center is seen from the start point, and is always specified as an incremental value irrespective of G90 and G91, as shown below.



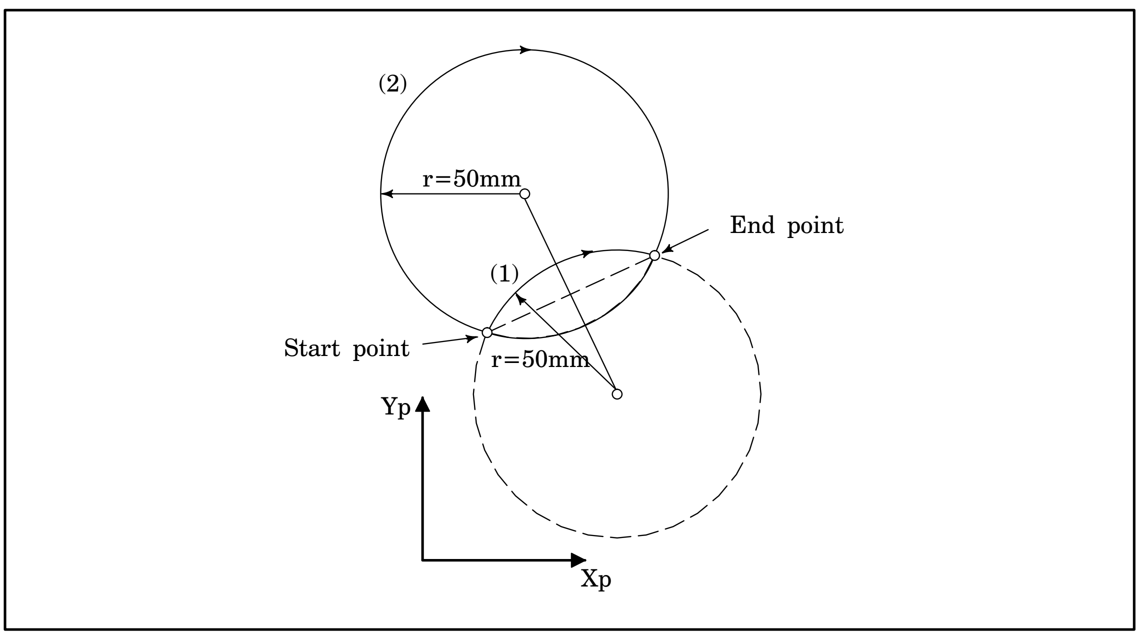
Programming for circular interpolation

I, J, and K must be signed according to the direction.The radius can be specified with address R instead of specifying the center by I, J, or K.The command format is as follows:

$$\left.\begin{matrix}G02\\G03\end{matrix}\right\}\quad Xp\_\_\_\_\ Yp\_\_\_\_\ R\_\_\_\_\ ;$$

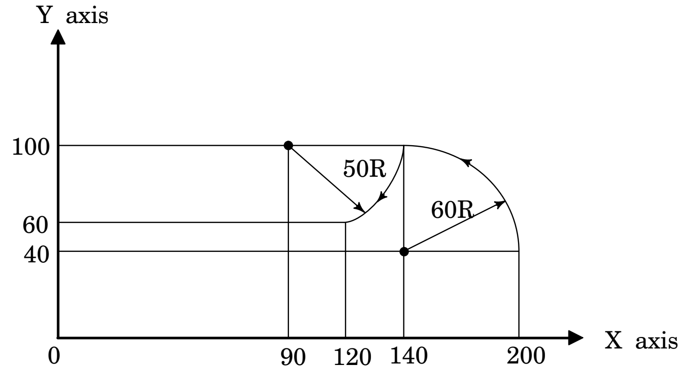
In this case, two types of arcs (one arc is less than 180°, and the other is more than 180°) are considered, as shown in the example figure . When an arc exceeding 180, is commanded, the radius must be specified with a negative value.

For arc (1) (less than 180°)
G91G02Xp60.0Yp20.0R50.0 F300.0;
For arc (2) (greater than 180°)
G91G02Xp60.0Yp20.0R–50.0 F300.0;



An example of Programming

(Program examples)



The above tool path can be programmed as follows:

1) In absolute programming

```
G92   Xp200.0 Yp40.0 Zp0           ;
G90   G03 Xp140.0 Yp100.0 R60.0  F300. ;
      G02 Xp120.0 Yp60.0 R50.0      ;
or
G92   Xp200.0 Yp 40.0 Zp0          ;
G90   G03 Xp140.0 Yp100.0 R60.6 F300. ;
      G02 Xp120.0 Yp 60.0 R50.0     ;
```

2) In incremental programming

```
G91   G03 Xp–60.0 Yp 60.0 R 60.0 F300. ;
      G02 Xp–20.0 Yp–40.0 R 50.0    ;
or
G91   G03 Xp–60.0 Yp 60.0 I–60.0 F300. ;
      G02 Xp–20.0 Yp–40.0 I–50.0     ;
```

The feed rate in circular interpolation is equal to the feed rate specified by the F code, and the feed rate along the arc (the tangential feed rate of the arc) is controlled to be the specified feed rate.

NOTE 1 I0, J0, and K0 can be omitted.

NOTE 2 If Xp, Yp, and Zp are all omitted or if the end point is located at the same position as the start point, and when the center is commanded by I, J, and K, an arc of 360° (a complete circle) is assumed.

G02I____ ; (A complete circle)

When R is used, an arc of 0° is programmed

G02R____; (The cutter does not move.)

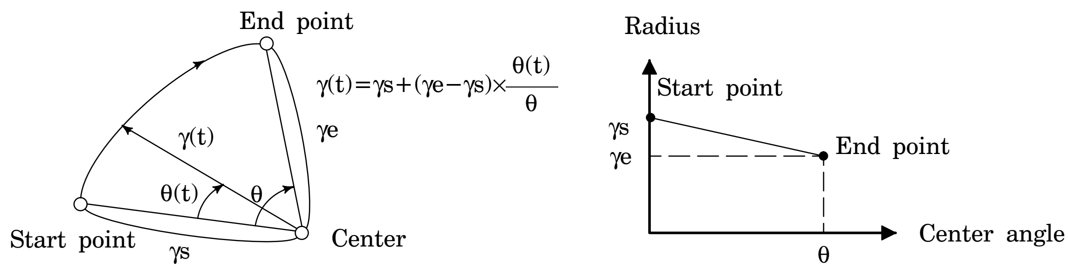
NOTE 3 The error between the specified feed rate and the actual tool feed rate is ±2% or less. However, this feed rate is measured along the arc after the cutter compensation is applied.

NOTE 4 If I, J, K, and R addresses are specified simultaneously, the arc specified by address R takes precedence and the other are ignored.

NOTE 5 If an axis not comprising the specified plane is commanded, an alarm is displayed.
For example, if axis U is specified as a parallel axis to X axis when plane XY is specified, an alarm is displayed.

NOTE 6 If the difference in radius between the start point and end point of the arc is not less than the value set in the parameter (data No. 2410), alarm PS191 occurs.

NOTE 7 If the end point does not lie on the arc, the following spiral results:

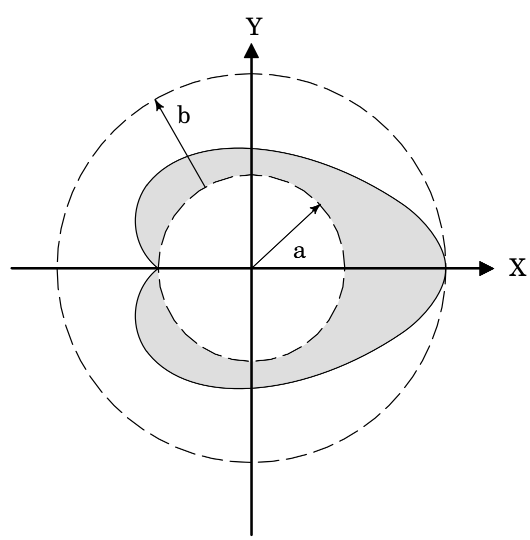


As center angle θ (t) increases, the arc radius changes linearly. Specifying an arc whose radius at the end point differs from that at the start point enables spiral interpolation to be performed. To enable spiral interpolation, specify a large value in the arc radius error limit parameter (data No. 2410).

[Heart cam cutting]

Spiral interpolation enables a heart cam to be machined.

Example



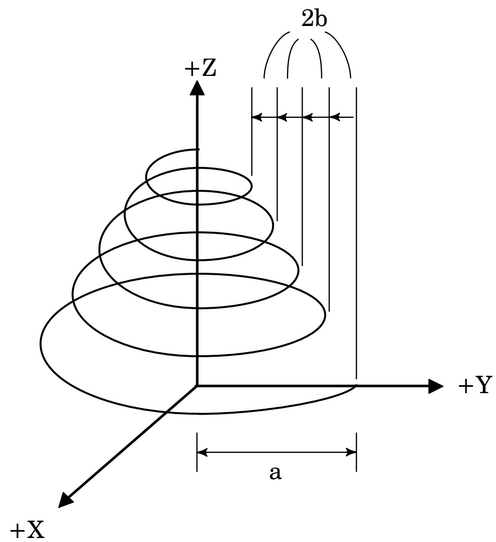
G91G17G02 X(a+b) Ia F1000 ; (Upper side)
G02 X– (a+b) I–b ; (Lower side)

[Conical cutting]

An arc portion in helical interpolation can also be changed to a spiral form.

Helical interpolation with a spiral allows conical cutting.

Example




```
G91G17G02   Y–(2a–b) J–aZpF1000 ;
            Y(2a–3b) J(a–b) Zp
            Y–(2a–5b) J–(a–2b) Zp ;
            Y(2a–7b) J(a–3b) Zp
            ..........
            ..........
```

4.5 3–Dimensional Circular Interpolation Function

3–dimensional circular interpolation can be performed by specifying an intermediate point and end point of an arc.

G02.4 Xx1 Yy1 Zz1 Aa1 Bb1 ;
Xx2 Yy2 Zz2 Aa2 Bb2 ;

Or

G03.4 Xx1 Yy1 Zz1 Aa1 Bb1 ;
Xx2 Yy2 Zz2 Aa2 Bb2 ;

3-dimensional circular interpolation is performed by specifying one of the above commands.

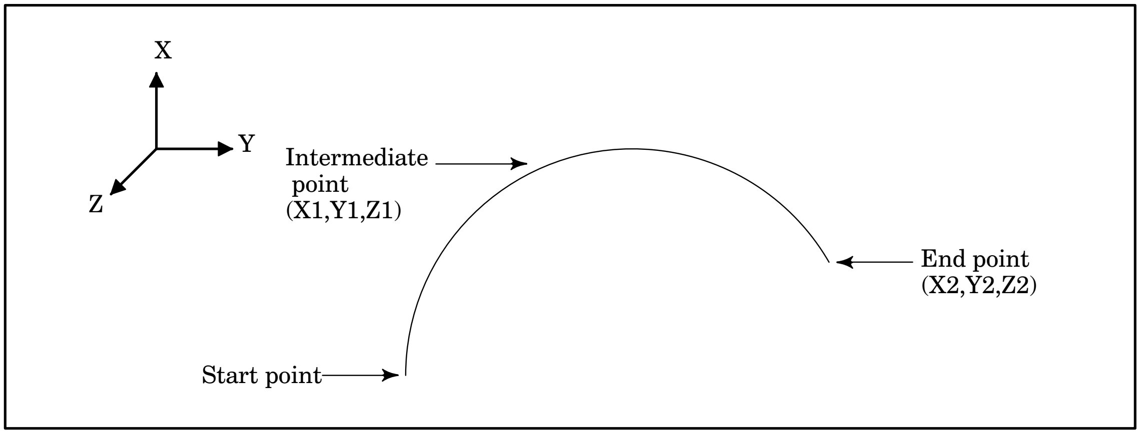
In the above commands, the first block designates the intermediate point of an arc and the second block designates the end point.

As required, up to two axes (and) can be specified in addition to the three–dimensional interpolation axes.

In incremental specification, the intermediate point specified in the first block must be specified as a position relative to the start point. The end point specified in the second block must be specified as a position relative to the intermediate point.

Since this function does not distinguish between the directions of rotation, either G02.4 or G03.4 can be specified.

G02.4 and G03.4 fall within G code group 01. These commands are continuous–state commands. Therefore, Once G02.4 or G03.4 is specified, it is valid until another group 01 G code is specified.



As shown in the figure, an arc ending at a certain point cannot be obtained unless both an intermediate and end point are specified. Specify the intermediate point and end point in separate blocks.

In MDI operation, 3–dimensional circular interpolation starts when the start button is pressed after the blocks for the intermediate and end points are entered. If the start button is pressed immediately after the intermediate point block is entered, the end point of the arc is still unknown so only buffering is performed. In this case, to start 3–dimensional circular interpolation, enter the end point block, then press the start button again.

When the commands for spatial circular interpolation are specified successively, the end point is used as the start point for the next interpolation operation.

(1) When this function is used, the following functions cannot be used.

· Three–dimensional coordinate system rotation
· Coordinate rotation
· Scaling
· Programmable mirror image
· Polar coordinate interpolation
· Polar coordinate command (G16)
· Normal direction control (G41.1)
· Hypothetical axis interpolation (G07)
· Cyrindrical interpolation (G07.1)
· Canned cycle for drilling
· Modal calling (G65)
· Exact stop mode (G61)
· Automatic cornar override (G62)
· Tapping mode (G63)
· Chopping function
· Interruption macro
· Single direction positioning (G60)
· Dwell (G04)
· Auxiliary function
· Data setting, PMC data setting (G10, G10.1)
· Reference position return
· Skip function
· Automatic tool length measurement
· Macro calling
· Parallel copy
· Rotation copy
· Exact stop
· Workpiece coordinate system preset

When any of the following functions are used, this function cannot be used.

· Three–dimensional coordinate system rotation
· Coordinate rotation
· Scaling
· Programmable mirror image
· Polar coordinate interpolation
· Polar coordinate command
· Normal direction control
· Hypothetical axis interpolation
· Cyrindrical interpolation
· Canned cycle for drilling
· Modal calling
· Exact stop mode (G61)
· Automatic cornar override (G62)
· Tapping mode (G63)
· Chopping function
· Interruption macro
· Data setting, PMC data setting

(2) A full circle (360 arc) cannot be specified.

(3) When the start, middle, and end points are located on a straight line, linear interpolation is applied.

(4) When the start point is the same as the middle point, or the middle point is the same as the end point, linear interpolation is applied to the end point.

(5) Before using this function, cancel tool length compensation and cutter compensation.

(6) When G01 is specified without an end point, an are cannot be obtained. In this case, linear interpolation is applied from the start point to the middle point.

(7) When this function is used, do not perform manual intervention while the manual absolute signal is on.

(8) This function is not supported by systema with the parallel–axis control function.

(9) This function cannot be used together with chamfering or corner rounding.

(10) This function cannot be used while the chamfering or corner rounding function is being used.

4.6 Helical Interpolation (G02, G03)

Helical interpolation is enabled by specifying up to two other axies which move synchronously with the circular interpolation by circular commands. That is, the can be moved helically.

$$\mathbf{G17}\begin{Bmatrix}\mathbf{G02}\\ \mathbf{G03}\end{Bmatrix}\mathbf{Xp\_\_Yp\_\_}\begin{Bmatrix}\mathbf{R\_\_}\\ \mathbf{I\_\_\ J\_\_}\end{Bmatrix}\quad \alpha\_\ (\beta\_\ )\ \mathbf{F\_\ ;}\ \text{(Xp–Yp plane)}$$

$$\mathbf{G18}\begin{Bmatrix}\mathbf{G02}\\ \mathbf{G03}\end{Bmatrix}\mathbf{Xp\_\_\ Zp\_\_}\begin{Bmatrix}\mathbf{R\_\_}\\ \mathbf{I\_\_\ K\_\_}\end{Bmatrix}\quad \alpha\_\ (\beta\_\ )\ \mathbf{F\_\ ;}\ \text{(Zp–Xp plane)}$$

$$\mathbf{G19}\begin{Bmatrix}\mathbf{G02}\\ \mathbf{G03}\end{Bmatrix}\mathbf{Yp\_\_\ Zp\_\_}\begin{Bmatrix}\mathbf{R\_\_}\\ \mathbf{J\_\_\ K\_\_}\end{Bmatrix}\quad \alpha\_\ (\beta\_\ )\ \mathbf{F\_\ ;}\ \text{(Yp–Zp plane)}$$

α, β : Any one axis where circular interpolation is not applied

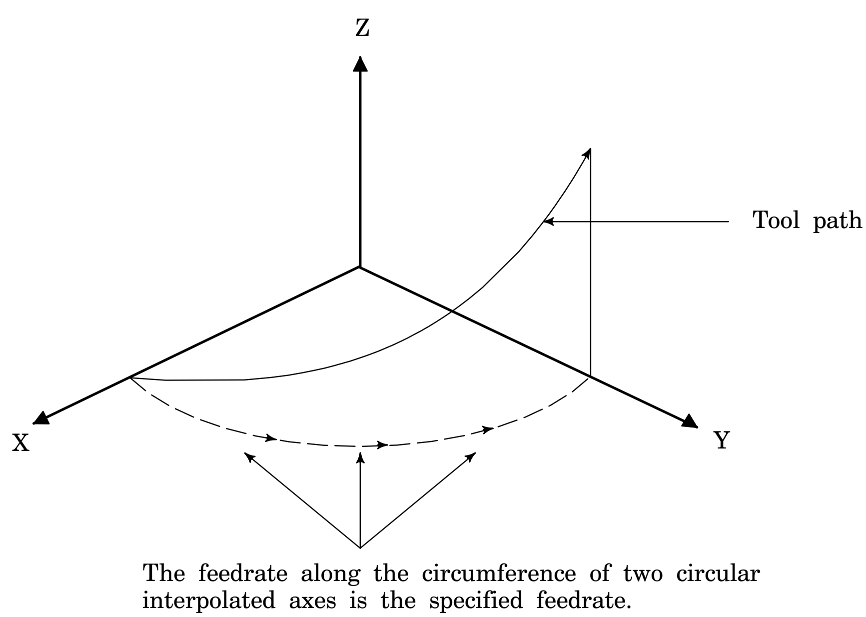
The command method is to simply add a move command axis which is not circular interpolation axes.

Bit 2 (HTG) of parameter No.1401 can be used to specify whether the speed command specifies the feedrate along the tangential line of the arc on the plane, or the feedrate along the tangential line of the actual tool path, including movement along the linear axis.

When parameter HTG (No.1401#2) is 1, an F command specifies a feed rate along a circular arc. Therefore, the feed rate of the linear axis is as follows:

$$F \times \frac{\text{Length of linear axis}}{\text{Length of circular arc}}$$

Determine the feed rate so the linear axis feed rate does not exceed any of the various limit values.
The feedrate along the circumference of two circular interpolated axes is the specified feedrate.



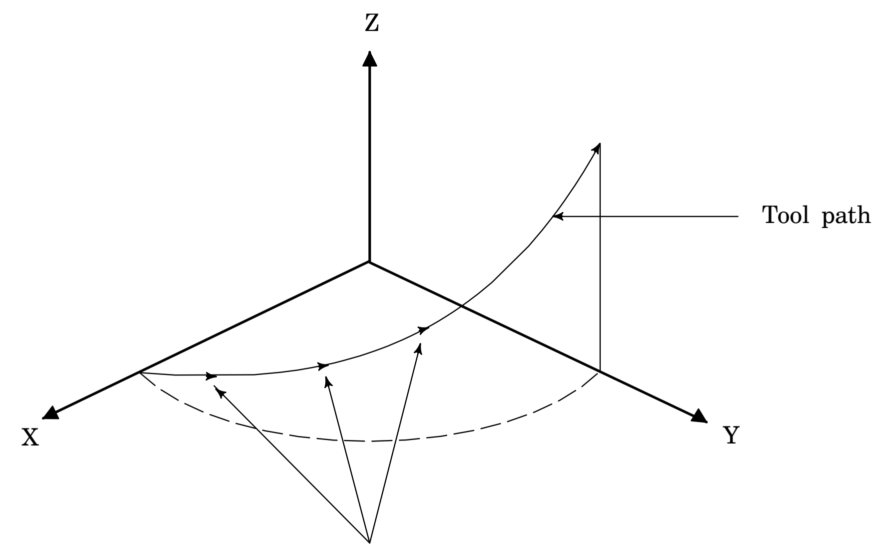
When bit 2 (HTG) of parameter No. 1401 is set to 1, the speed command specifies the feedrate along the actual tool path, including movement along the linear axis.

In this case, the feedrate along the arc on the plane is:

$$F \times \frac{\text{Length of the arc}}{\sqrt{(\text{Length of the arc})^2 + (\text{Travel along the linear axis})^2}}$$

The feedrate along the linear axis is:

$$F \times \frac{\text{Travel along the linear axis}}{\sqrt{(\text{Length of the arc})^2 + (\text{Travel along the linear axis})^2}}$$



NOTE 1	Cutter compensation is applied only for a circular arc.
NOTE 2	Tool offset and tool length compensation cannot be used in a block in which a helical cutting is commanded.

4.7 Helical Interpolation B (G02, G03)

Normal helical interpolation allows circular interpolation for two arbitrary axes and linear interpolation for two other axes to be performed simultaneously. Helical interpolation B, on the other hand, allows circular interpolation for two arbitrary axes and linear interpolation for four other axes to be performed simultaneously. In this case, the four axes in linear interpolation must be in a plane other than that for circular interpolation.

The command format for helical interpolation B consists of the command format for normal helical interpolation and move commands for two axes. As with normal helical interpolation, the feedrate of helical interpolation B is controlled so that the feedrate of circular interpolation can achieve the specified feedrate.

Bit 2 (HTG) of parameter No. 1401 can be used to specify whether the speed command specifies the feedrate along the tangential line of the arc on the plane, or the feedrate along the tangential line of the actual tool path, including movement along the linear axis.

Format

$$\mathbf{G17}\left\{\begin{matrix}\mathbf{G02}\\ \mathbf{G03}\end{matrix}\right\}\mathbf{Xp\_\_Yp\_\_}\left\{\begin{matrix}\mathbf{R\_\_}\\ \mathbf{I\_\_\ J\_\_}\end{matrix}\right\}\ \alpha\_\ \beta\_\ \gamma\_\ \zeta\_\ \mathbf{;}\ \text{(Xp–Yp plane)}$$

$$\mathbf{G18}\left\{\begin{matrix}\mathbf{G02}\\ \mathbf{G03}\end{matrix}\right\}\mathbf{Xp\_\_\ Zp\_\_}\left\{\begin{matrix}\mathbf{R\_\_}\\ \mathbf{K\_\_\ I\_\_}\end{matrix}\right\}\ \alpha\_\ \beta\_\ \gamma\_\ \zeta\_\ \mathbf{;}\ \text{(Zp–Xp plane)}$$

$$\mathbf{G19}\left\{\begin{matrix}\mathbf{G02}\\ \mathbf{G03}\end{matrix}\right\}\mathbf{Yp\_\_\ Zp\_\_}\left\{\begin{matrix}\mathbf{R\_\_}\\ \mathbf{J\_\_\ K\_\_}\end{matrix}\right\}\ \alpha\_\ \beta\_\ \gamma\_\ \zeta\_\ \mathbf{;}\ \text{(Yp–Zp plane)}$$

where α, β, γ, ζ : Arbitrary axes other than the axes for circular interpolation

4.8 Hypothetical Axis Interpolation (G07)

In helical interpolation, when pulses are distributed with one of the circular interpolation axes set to a hypothetical axis, sine interpolation is enabled.

When one of the circular interpolation axes is set to a hypothetical axis, pulse distribution causes the speed of movement along the remaining axis to change sinusoidally. If the major axis for threading (the axis along which the machine travels the longest distance) is set to a hypothetical axis, threading with a fractional lead is enabled. The axis to be set as the hypothetical axis is specified with G07.

Format

G07 α **0 ;** Hypothetical axis setting
G07 α **1 ;** Hypothetical axis cancel

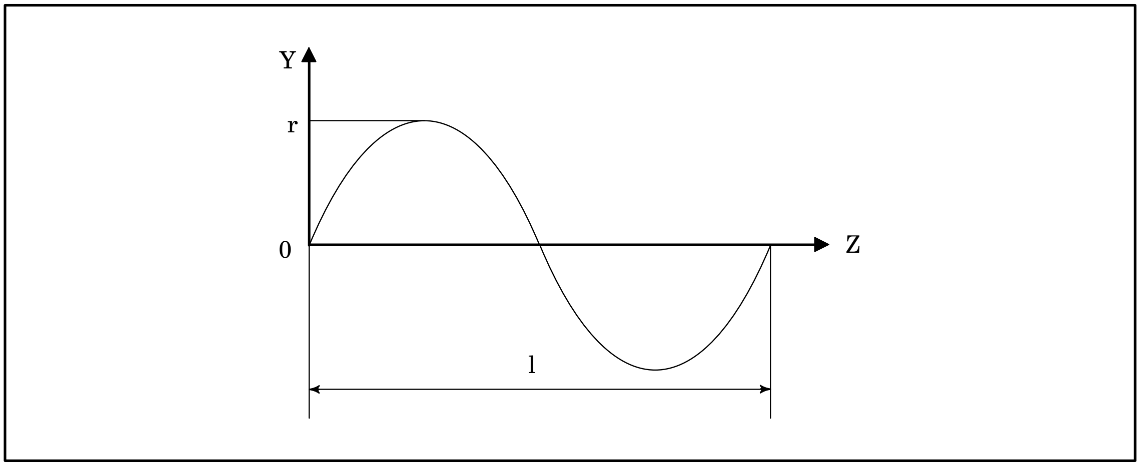
where, α is any one of the addresses of the controlled axes

The α axis is regarded as a hypothetical axis for the period of time from the G07 α 0 command until the G07 α 1 command appears.

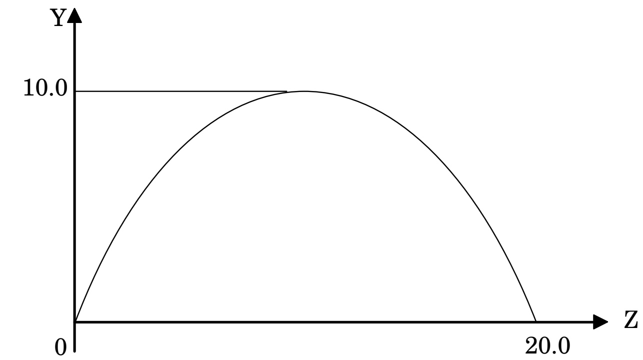
Suppose sine interpolation is performed for one cycle in the YZ plane. The hypothetical axis is then the X axis.

$X^2 + Y^2 = r^2$ (r is the radius of an arc.)

$Y = r\mathrm{SIN}\left(\frac{2\pi}{l} Z\right)$ (l is the distance traveled along the Z–axis in one cycle.)



Sample program (sine interpolation)



N001 G07 X0 ;
N002 G91 G17 G03 X–20.2 Y0.0 I–10.0 Z20.0 F100 ;
N003 G01 X10.0 ;
N004 G07 X1 ;

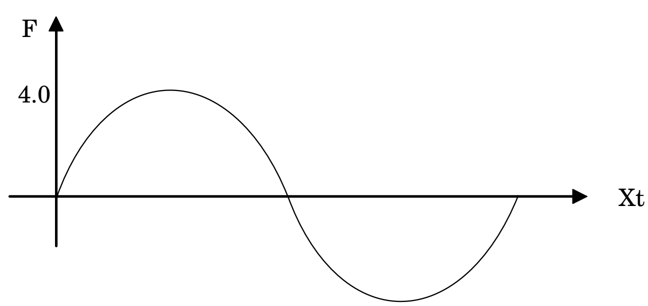
From the N002 to N003 blocks, the X–axis is set to a hypothetical axis.

The N002 block specifies helical cutting in which the Z–axis is the linear axis. Since no movement takes place along the X–axis, movement along the Y–axis is performed while performing sine interpolation along the Z–axis.

In the N003 block, there is no movement along the X–axis, and so the machine dwells until interpolation terminates.

(Sample program)

G07Z0 ;	The Z–axis is set to a hypothetical axis.
G02X0Z0I10.0F4. ;	The feedrate on the X–axis changes sinusoidally.
G07Z1 ;	The use of the Z–axis as a hypothetical axis is canceled.



NOTE 1 The hypothetical axis can be used only in automatic operation. In manual operation, it is not used, and movement takes place.

NOTE 2 Interlock, stroke limit, and external deceleration can also apply to the hypothetical axis.

NOTE 3 An interrupt caused by the handle also applies to the hypothetical axis. This means that movement for a handle interrupt is performed.

NOTE 4 Specify hypothetical axis interpolation only in the incremental mode.

4.9 Polar Coordinate Interpolation (G12.1, G13.1)

Polar coordinate interpolation is a function that exercises contour control in converting a command programmed in a cartesian coordinate system to the movement of a linear axis (movement of a tool) and the movement of a rotary axis (rotation of a workpiece).

This function is useful for grinding a cam shaft.

Whether to perform polar coordinate interpolation is specified using the following G codes (group 25):

G12.1: Polar coordinate interpolation mode (for performing polar coordinate interpolation)

G13.1: Polar coordinate interpolation cancel mode (for not performing polar coordinate interpolation)

Each G code is to be specified in a single block.

Upon power–up or reset, the polar coordinate interpolation cancel mode (G13.1) is set.

A linear axis and rotary axis subject to polar coordinate interpolation must be set in the parameters (Nos. 1032, 1033) beforehand.

By specifying G12.1, the polar coordinate interpolation mode is set; a plane (referred to as a polar coordinate interpolation plane) is selected where the linear axis serves as the first plane axis, and a virtual axis that intersects the linear axis orthogonally serves as the second plane axis, with the origin of a local coordinate system (or the origin of a workpiece coordinate system if no local coordinate system is specified (G52)) set as the origin

(*3). Polar coordinate interpolation is carried out on this plane.

For program commands in the polar coordinate interpolation mode, cartesian coordinates on a polar coordinate interpolation plane are used. For a second (virtual) plane axis address for a command, a rotary axis (parameter No. 5461) address is used. However, the command unit is not degrees, but the same unit (mm or inches) as for the first plane axis (command based on linear axis address) is used. Whether to use a diameter specification or radius specification is not based on the first plane axis, but the same specification as for the rotary axis applies.

In the polar coordinate interpolation mode, the linear interpolation command (G01) and circular interpolation commands (G02, G03) can be used. In addition, the absolute command (G90) and incremental command (G91) can be used.

For a program command, cutter compensation can be specified; polar coordinate interpolation is performed for a path after cutter compensation. Note, however, that polar coordinate interpolation mode (G12.1, G13.1) switching cannot be performed in a cutter compensation mode (G41, G42). G12.1 or G13.1 is usable in the G40 mode (cutter compensation cancel mode).

For a feedrate, a tangential speed (relative speed between the workpiece and tool) on the polar coordinate interpolation plane (cartesian coordinate system) is to be specified with F (in mm/minute or inch/minute).

When G12.1 is specified, the coordinate of the virtual axis is 0. That is, polar coordinate interpolation is started, assuming degree = 0 at the position where G12.1 is specified.

WARNING 1 Before G12.1 is specified, a local coordinate system (or workpiece coordinate system) where the center of the rotary axis is the origin of the coordinate system must be set. In the G12.1 mode, the coordinate system must not be changed (G92, G52, G53, relative coordinate reset, G54 through G59, etc.).

WARNING 2 Cutter compensation must be specified in the polar coordinate interpolation cancel mode before G12.1 is specified. It cannot be specified in the polar coordinate interpolation mode. Furthermore, no offset values can be changed in the polar coordinate interpolation mode.

WARNING 3 Polar coordinate interpolation converts a figure programmed in a cartesian coordinate system to the movement of the rotary axis (C axis) and the movement of the linear axis (X axis). Near the workpiece center, the speed component of the C axis becomes greater. As shown in the figure below, assume three straight lines L1, L2, and L3, and let ΔX be the amount of travel based on a feedrate per unit time in a cartesian coordinate system specified with address F. Then for a straight line closer to the center (from L1 to L2 to L3), the amount of travel of the C axis becomes greater (from θ1 to θ2 to θ3) with respect to the amount of travel (ΔX) per unit time in the cartesian coordinate system. A greater amount of travel of the C axis per unit time means a greater speed component of the C axis near the workpiece center. As a result of conversion from a cartesian coordinate system to the movements of the C axis and X axis, an alarm (OT512) may be raised if the speed component of the C axis exceeds the maximum cutting feedrate (parameter No. 1422) of the C axis. Accordingly, to prevent the speed component of the C axis from exceeding the maximum cutting feedrate, a slower feedrate must be specified with address F, or such a program that gets closer to the workpiece center (or such a program that allows the tool center to get closer to the workpiece center if cutter compensation is applied) must not be used.

NOTE 1 The plane (selected with G17, G18, or G19) before G12.1 is specified is once cancelled, and the plane is restored when G13.1 (polar coordinate interpolation cancel) is specified. At the time of reset, the polar coordinate interpolation mode is cancelled; a plane selected with G17, G18, or G19 is used.

NOTE 2 The method of specifying an arc radius (which of the I, J, and K addresses to use) when circular interpolation (G02, G03) is to be performed on a polar coordinate interpolation plane depends on which axis in the basic coordinate system represents the first plane axis (linear axis) (parameter No. 1022):

- When the linear axis is the X axis or is parallel with the X axis, the Xp–Yp plane is assumed, and I and J are used for arc radius specification.
- When the linear axis is the Y axis or is parallel with the Y axis, the Yp–Zp plane is assumed, and J and K are used for arc radius specification.
- When the linear axis is the Z axis or is parallel with the Z axis, Zp–Xp plane is assumed, and K and I are used for arc radius specification.

R can be used for arc radius specification.

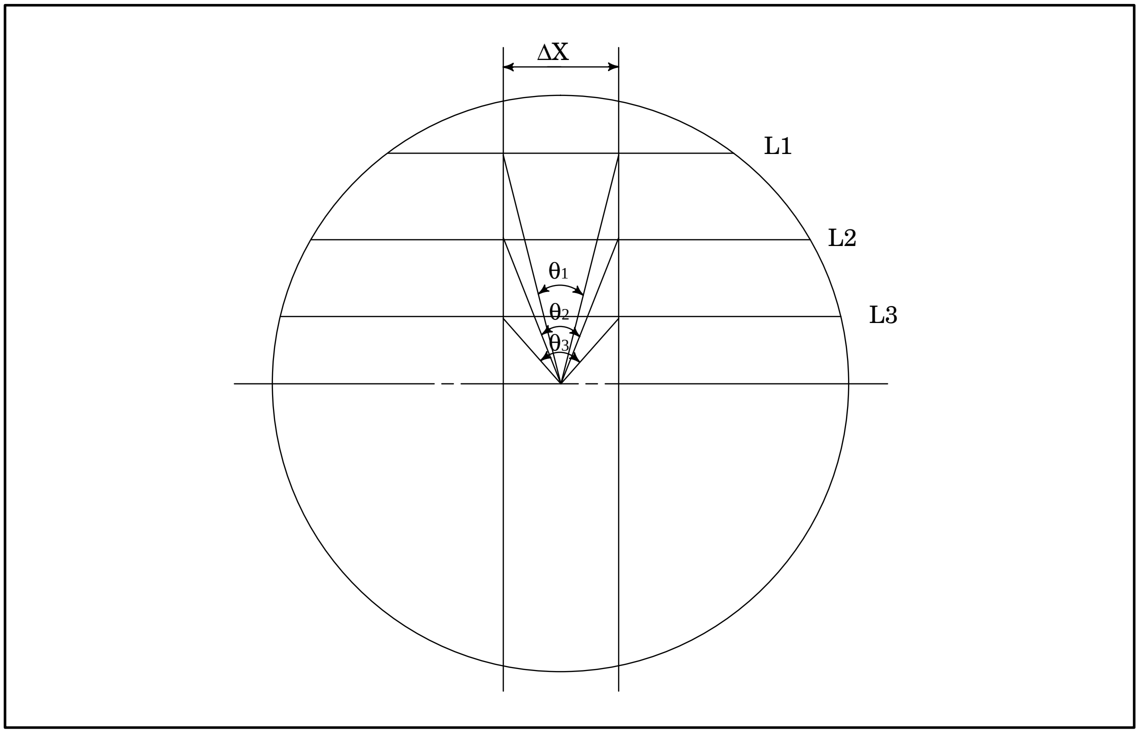
NOTE 3 The G codes specifiable in the G12.1 mode are only those listed below:

G04, G65, G66, G67, G01, G02, G03, G90, G91, G94, G95, G40, G41, and G42

NOTE 4 The movement of an axis other than those on the plane in the G12.1 mode is independent of polar coordinate interpolation.

NOTE 5 Current position indications in the G12.1 mode:All indications represent actual coordinates, but the “remaining amount of travel” indicates the remaining amount of travel of a block on the polar coordinate interpolation plane (cartesian coordinate system).

NOTE 6 For a block in the G12.1 mode, the program and the block cannot be restarted.



Let L and R be as follows:

L: Distance (in mm) between the tool center and workpiece center when the tool center is the nearest to the workpiece center

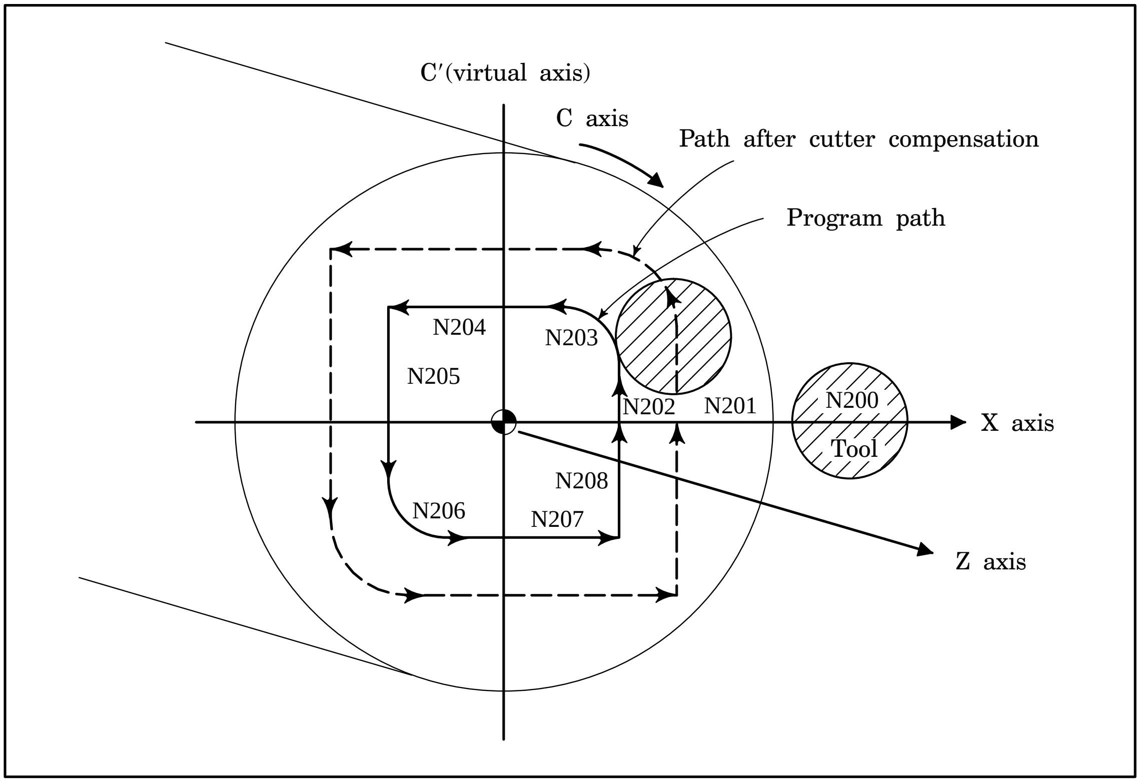
R: Maximum cutting feedrate (deg/minute) of the C axis

Then, a speed specifiable with address F in polar coordinate interpolation can be given by the formula below. Specify a speed allowed by the formula. The formula provides a theoretical value; in practice, a value slightly smaller than a theoretical value may need to be used due to a calculation error.

$$F < L \times R \times \frac{\pi}{180} \text{ (mm/min)}$$

The speed can be controlled so as to prevent the issue of alarm OT512 (excessive speed). See Section 6.6.2 for details.

Example of Polar Coordinate Interpolation Program Based on X Axis (Linear Axis) and C Axis (Rotary Axis)




```
O0001 :
   |
N010 T0101
   |
N0100   G90 G00 X60.0 C0 Z __;   Positioning to start position
N0200   G12.1 ;                  Start of polar coordinate interpolation
N0201   G42 G01 X20.0 F__ ;     ┐
N0202   C10.0 ;                  |
N0203   G03 X10.0 C20.0 R10.0 ;  |
N0204   G01 X–20.0 ;             |   Geometry program (program based on
N0205         C–10.0 ;           ├──    cartesian coordinates on X–C’ plane)
N0206   G03 X–10.0 C–20.0 110.0 J0 ; |
N0207   G01 X20.0 ;              |
N0208         C0 ;               |
N0209   G40 X60.0 ;             ┘
N0210   G13.1 ;                   Cancellation of polar coordinate interpolation
N0300   Z__ ;
N0400   X__ C__  ;
   |
   |
M30 ;
```

4.10 Cylindrical Interpolation (G07.1)

In the cylindrical interpolation mode, the amount of travel of a rotary axis specified by an angle is once internally converted to a distance along a linear axis on the outer surface so that linear interpolation or circular interpolation can be performed with another axis. After interpolation, such a distance is converted back to the amount of travel of the rotary axis.

The cylindrical interpolation function allows the side of a cylinder to be developed for programming. So programs such as a program for cylindrical cam grooving can be created very easily.

G7.1 Name of rotary axis Cylinder radius; (1)

G7.1 Name of rotary axis 0; (2)

The cylindrical interpolation mode can be set with command (1). The name of a rotary axis subject to cylindrical interpolation is to be specified as a word address. The radius of a cylinder to be machined is to be specified as a command value.

The cylindrical interpolation mode can be cancelled with command (2).

Example

```
O0001 ;
N1  G28  X0  Z0  C0  ;
N2 . . . . . . . . . .     ;
  . . . . . . . . . . . .  ;
N6 G7.1  C125.67     ;   The rotary axis subject to cylindrical interpolation is the C axis, and
                         the cylinder radius is 125.67 mm.
  . . . . . . . . . . . .  ;
N9  G7.1  C0         ;   The cylindrical interpolation mode is cancelled.
  . . . . . . . . . . . .  ;
```

Cylindrical Interpolation Mode and Other Functions

(1) Feedrate specification

A feedrate specified in the cylindrical interpolation mode is a speed on the developed cylindrical surface.

(2) Circular interpolation (G02, G03)

(a) Plane selection

To perform circular interpolation between a rotary axis and linear axis in the cylindrical interpolation mode, a plane selection command (G17, G18, G19) is required.

Example Circular interpolation between the Z axis and C axis

For the C axis of parameter 1022, 5 (axis parallel with the X axis) is to be set. In this case, the command for circular interpolation is

G18 Z___C___;
G02 (G03) Z___ C___ R___ ;

For the C axis of parameter 1022, 6 (axis parallel with the Y axis) may be specified instead. In this case, however, the command for circular interpolation is

G19 C___ Z___;
G02 (G03) Z___ C___ R___ ;

(b) Radius specification

In the cylindrical interpolation mode, an arc radius cannot be specified with word address I, J, or K. Instead, word address R is used to specify an arc radius. A radius must not be specified in degrees but in mm (for metric input) or inches (for inch input).

(3) Cutter compensation

For cutter compensation in the cylindrical interpolation mode, a plane is to be specified in the same way as for circular interpolation. However, cutter compensation must be started up or cancelled while the cylindrical interpolation mode is set. Cutter compensation cannot be performed correctly if the cylindrical interpolation mode is set after cutter compensation is started.

(4) Positioning

In the cylindrical interpolation mode, positioning operations (including those that produce rapid traverse cycles such as G28, G53, G80 through G89). Before positioning can be performed, the cylindrical interpolation mode must be cancelled.

(5) Coordinate system setting

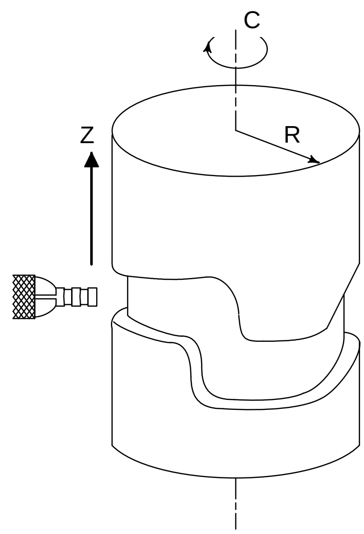
In the cylindrical interpolation mode, a workpiece coordinate system (G92, G54 through G59) or local coordinate system (G52) cannot be specified.

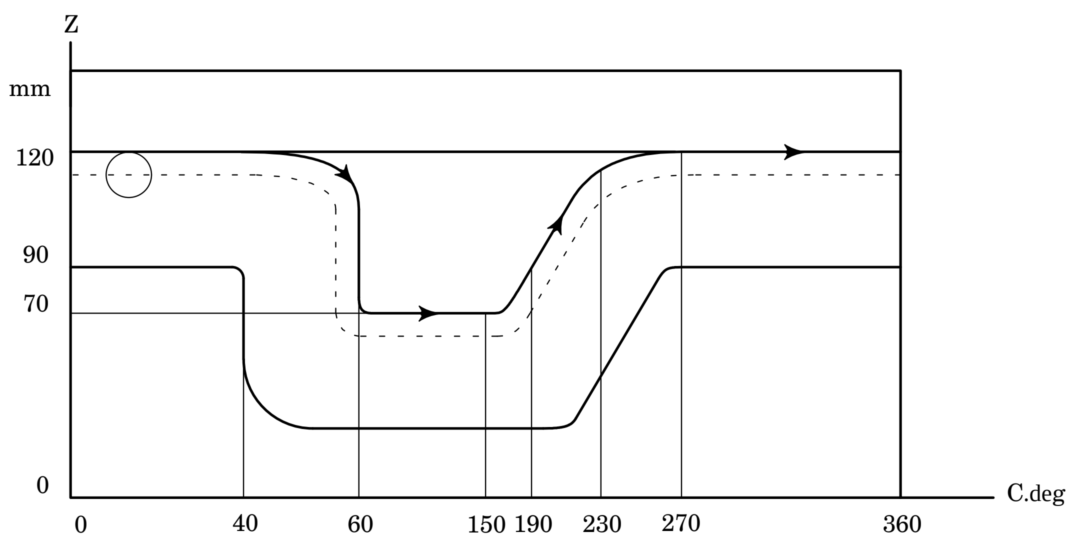
(6) Block restart

A block cannot be restarted in the cylindrical interpolation mode. A program including a cylindrical interpolation command, however, can be restarted.

Example of Cylindrical Interpolation Program

```
O0001 (CYLINDRICAL INTERPOLATION) ;
N01 G00 G90 Z100.0 C0 ;
N02 G01 G91 G18 Z0 C0 ;
N03 G07.1 C57299 ;
N04 G90 G01 G42 Z120.0 D01 F250 ;
N05 C30.0 ;
N06 G02 Z90.0 C60.0 R30.0 ;
N07 G01 Z70.0 ;
N08 G03 Z60.0 C70.0 R10.0 ;
N09 G01 C150.0 ;
N10 G03 Z70.0 C190.0 R75.0 ;
N11 G01 Z110.0 C230.0 ;
N12 G02 Z120.0 C270.0 R75.0 ;
N13 G01 C360.0 ;
N14 G40 Z100.0 ;
N15 G07.1 C0 ;
N16 M30 ;
```





NOTE 1 Specify G7.1 in a single block.

NOTE 2 In the cylindrical interpolation mode, the cylindrical interpolation mode cannot be reset. The cylindrical interpolation mode must be cancelled before the cylindrical interpolation mode can be reset.

NOTE 3 Only one rotary axis can be set for cylindrical interpolation. So two or more rotary axes cannot be specified with G7.1.

NOTE 4 In the positioning mode (G00), cylindrical interpolation (G7.1) cannot be specified.

NOTE 5 The rotation axis subject to cylindrical interpolation must not be a parallel axis.

NOTE 6 Precision

(i) Automatic operation

In the cylindrical interpolation mode, the amount of travel of a rotary axis specified by an angle is once internally converted to a distance along a linear axis on the outer surface so that linear interpolation or circular interpolation can be performed with another axis. After interpolation, such a distance is converted back to an angle. For this conversion, the amount of travel is rounded to a minimum setting unit.

So when the radius of a cylinder is small, the actual amount of travel can differ from a specified amount of travel. Note, however, that such an error is not cumulative.

$$\text{Actual amount of travel} = \left[\frac{\text{MOTION REV}}{2 \times 2\pi R} \times \left[\text{specified} \times \frac{2 \times 2\pi R}{\text{MOTION REV}} \right] \right]$$

MOTION REV : Amount of travel per rotary axis rotation (1260°)

R : Workpiece radius

[ ] : Rounded to minimum setting unit

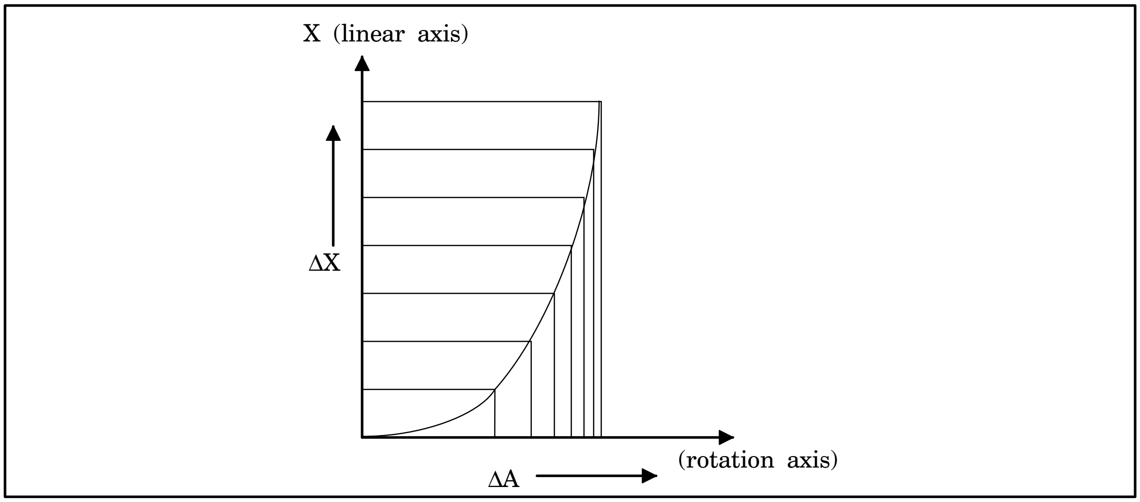
(ii) Manual operation

If manual operation is performed in the cylindrical interpolation mode with manual absolute on, an error can occur for the reason described above.

NOTE 7 In the cylindrical interpolation mode, canned cycles (G81 through G89) for drilling cannot be specified.

4.11 Exponential Interpolation (G02.3, G03.3)

A linear axis (the X–axis) is subjected to exponential interpolation and synchronized with movement around a rotation axis. Other axes are subjected to linear interpolation with the X–axis. The exponential function method uses linear approximations. In this method, the distance traveled around the rotation axis is obtained with respect to a certain degree of change along the X–axis. This function is useful for helix machining of a taper in a tool grinder.



The exponential relationship between the linear axis and rotation axis is defined as follows:

$X(\theta) = R \times (e^{\frac{\theta}{k}}-1) \times \frac{1}{\tan(1)}$ .... (Movement along the linear axis) (1)

$A(\theta) = (-1)^{\omega} \times 360 \times \frac{\theta}{2\pi}$ .... (Movement around the rotation axis) (2)

where,

$K = \frac{\tan(J)}{\tan(I)}$ .... (Direction of rotation)

$\omega = 0/1$

R, I, and J are constants, and θ is an angle in radians.

From equation (1) :

$\theta(X) = K \times \mathrm{1n} \left(\frac{X \times \tan(I)}{R} + 1 \right)$

Suppose that equations (1) and (2) are specified in the following format:

For positive rotation (ω = 0)

G02.3X –– Y –– Z –– I –– J –– K –– R –– F –– Q –– ;

For negative rotation (ω = 1)

G03.3X –– Y –– Z –– I –– J –– K –– R –– F –– Q –– ;

X –– ; Specify the end point in the absolute or incremental mode.

Y –– ; Specify the end point in the absolute or incremental mode.

Z –– ; Specify the end point in the absolute or incremental mode.

I –– ; Specify angle I. (The unit depends on the reference axis (see parameter No. 1031). (A value from1° to ±89° can be specified.)

J –– ; Specify angle J. (The unit depends on the reference axis. A value from 1° to ±89° can be specified.)

K __ ; Division unit of the linear axis in exponential interpolation (span) (The unit depends on the reference axis. Any positive value can be specified.)

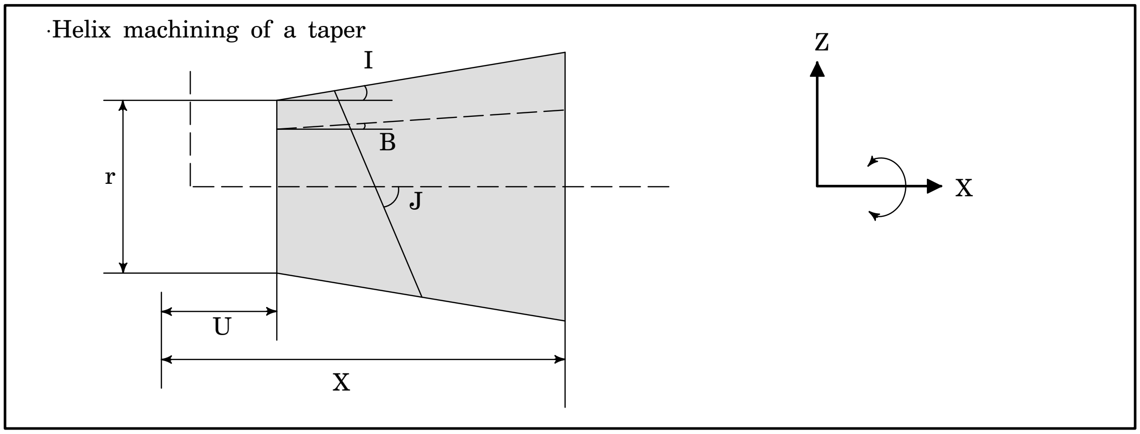
R __ ; Specify constant R in exponential interpolation. (The unit depends on the reference axis.)

F __ ; Specify the initial feedrate. The command is the same as a normal F code. Specify a composite speed including the speed about the rotation axis.

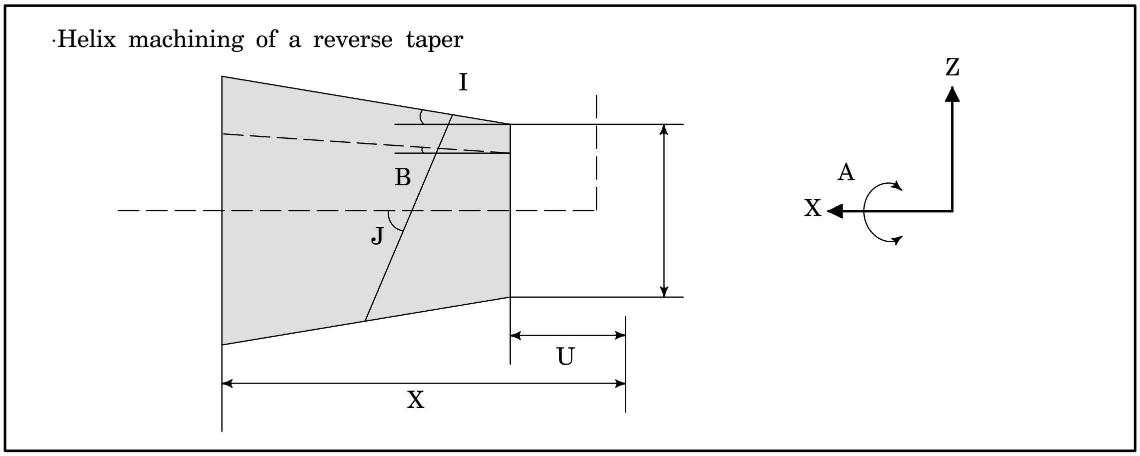
Q __ ; Specify the feedrate at the end point. The unit depends on the reference axis. Internally, the CNC performs interpolation between the initial feedrate (F ––) and the terminal feedrate (Q ––), depending on the distance traveled along the X–axis.

Example of use (Tool grinder, end mill grinder)

– Helix machining of a taper



– Helix machining of a reverse taper



Relation

$$Z(\theta) = \{-\frac{\gamma}{2} - U \times \tan(I) \times (e^{\frac{\theta}{K}}-1) \times \frac{\tan(B)}{\tan(I)}\} + Z(0) \quad \ldots \quad (3)$$

$$X(\theta) = \frac{\gamma}{2} - U \times \tan(I) \times (e^{\frac{\theta}{K}}-1) \times \frac{1}{\tan(I)} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots \ (4)$$

$$A(\theta) = (-1)^{\omega} \times 360 \times \frac{\theta}{2\pi}$$

where,

$$K = \frac{\tan(J)}{\tan(I)}$$

$X(\theta)$, $Z(\theta)$, and $A(\theta)$ are absolute coordinates along the X– and Z–axis and about the A–axis from the zero point, respectively.

r : Radius at left end
U : Allowance
I : Gradient
B : Gradient at the bottom of the groove
J : Helix angle
X : Distance traveled along linear axis
ω : Helix direction (0: Positive rotation, 1: Negative rotation)
θ : Workpiece rotation

From equations (3) and (4):

$Z(\theta) = \tan(B) \times X(\theta) + Z(0)$ ……………………………………… (5)

Equation (5) gives the gradient (B) at the bottom of the groove from the end positions along the X– and Z–axis. Then, the gradient (B) at the bottom of the groove and X determine the distance traveled along the Z–axis.

From equations (1) and (4) :

R = r/2–U*tan(I) ……………………………………………………… (6)

From equation (6), the radius at the left end (r) and the length of allowance (U) determine constant R. The gradient (I) is given by address I, and the helical angle (J) is given by address J. For helix machining of a reverse taper, etc., the gradient (I) is given a negative value. The helix direction is changed by G02.3 or G03.3.

In this way, helix machining for a taper or reverse taper is enabled.

WARNING In the G02.3 or G03.3 mode, tool length compensation, cutter compensation, and three–dimensional tool compensation cannot be used.

NOTE 1 The division unit (span) of the linear axis in exponential interpolation affects the accuracy of the figure to be formed. If the set value is too small, the machine may stop in the middle of interpolation. So an appropriate division unit must be set on the machine used.

NOTE 2 Initial feedrate command F in exponential interpolation is treated as if a normal F code were specified.

NOTE 3 G02.3 and G03.3 are group 1 continuous–state G codes. Therefore, these G codes and the other G codes of the same group are mutually exclusive.

NOTE 4 In the following cases, linear interpolation is performed even in the G02.3 or G03.3 mode:

– The linear axis specified with parameter No. 7636 is not specified, or the distance traveled along the linear axis is 0.

– The rotation axis specified with parameter No. 7637 is specified. The division unit (span) of the linear axis in exponential interpolation is set to 0.

NOTE 5 See parameters of data Nos. 7610, 7636, and 7685.

NOTE 6 In exponential interpolation, the X coordinate and angular displacement θ of the A axis to X are expressed by equation (1).

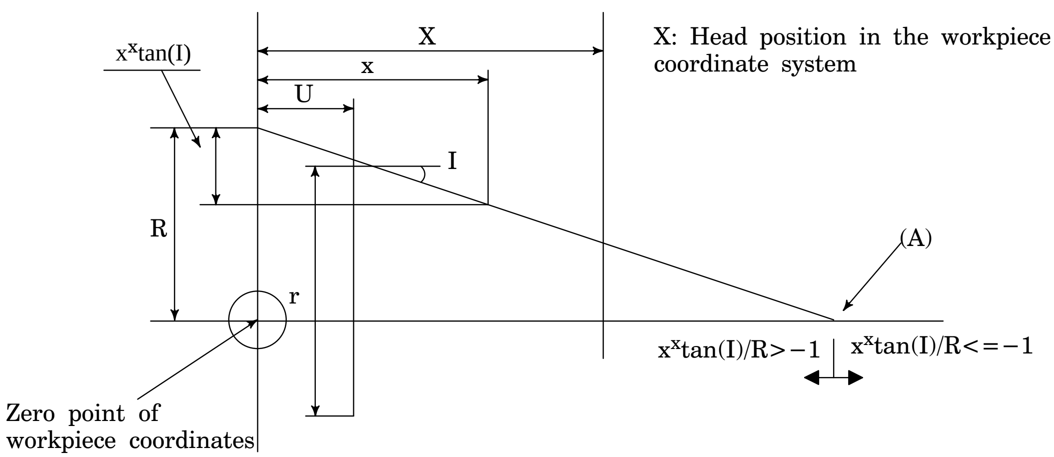
$$\theta(X) = K \times 1n \left(\frac{x \times \tan(I)}{R} + 1\right) \quad \cdots\cdots\cdots\cdots\cdots\cdots \quad (1)$$

where, I is the gradient.

In equation (1), the portion of the natural logarithm l n in parentheses must satisfy equation (2). Because it is impossible to find the logarithm of a negative number, which means that the value in the parentheses must be positive.

$$\frac{x \times \tan(I)}{R} > -l \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (2)$$

As illustrated below, if the value of xtan(I)/R becomes smaller than –1, the position moves beyond point (A), which is not realistic. In this case, alarm PS897 occurs.



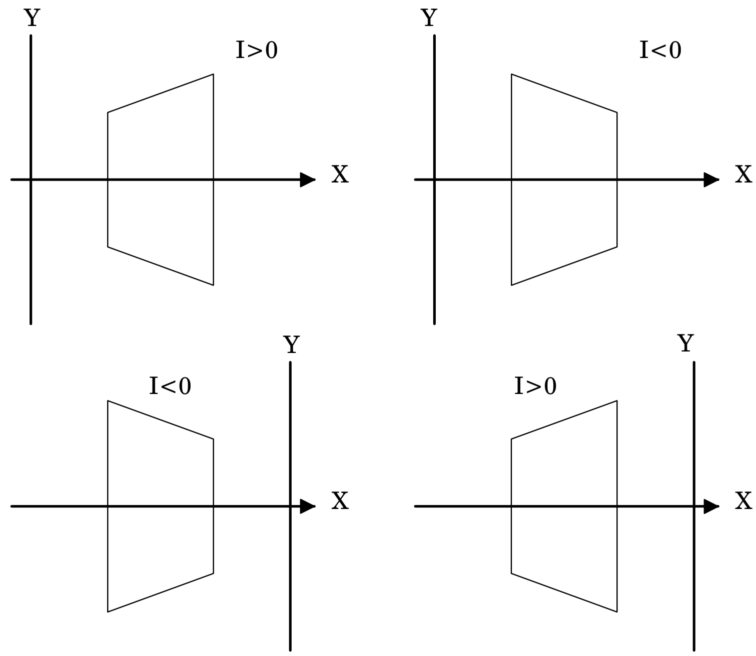
The values of X, Y, Z, and U in the equations for exponential interpolation (refer to the operator’s manual) are treated using coordinates in the workpiece coordinate system. Therefore, even when a positive value is specified in an incremental command, it is regarded as negative in the equations if it is positioned at a negative coordinate in the workpiece coordinate system.

NOTE 7 The relationship between the machining profile and the sign of the gradient I is as follows:

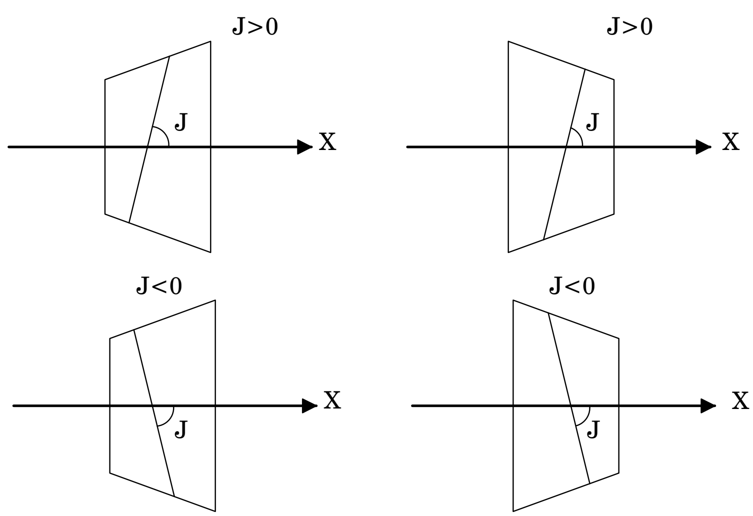
(1) For a slope going upward from left to right, I is a positive value.

(2) For a slope going downward from left to right, I is a negative value.

Example



NOTE 8 The sign of helix angle J is determined in the same manner as I as shown below.



4.12 Circular Threading B (G02.1, G03.1)

In circular threading B, circular interpolation is applied to two axes. Linear interpolation can be also applied to the major axis to which circular interpolation is applied and two other arbitrary axes.

This circular threading function does not move a tool in synchronism with the rotation of the spindle (workpiece) using the spindle motor. This function controls workpiece rotation using a servo motor (rotation axis) to perform threading at equal pitches along cylindrical material, grooving, and tool grinding.

Format

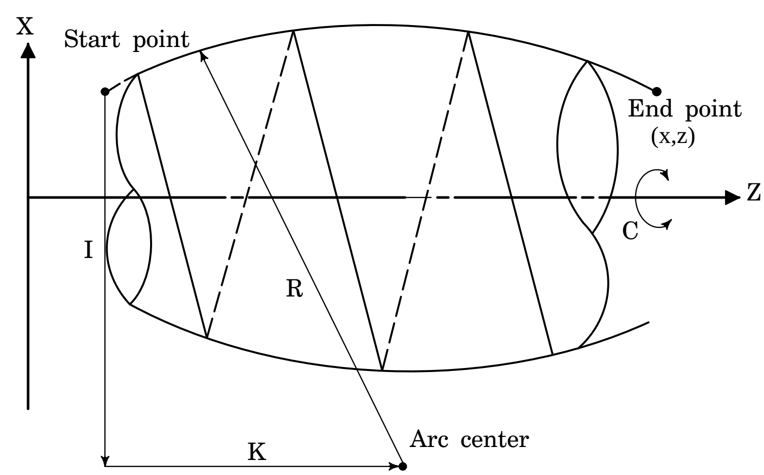
$$\textbf{G17} \begin{Bmatrix} \textbf{G02.1} \\ \textbf{G03.1} \end{Bmatrix} \textbf{Xp--Yp--}\alpha\textbf{--}\beta\textbf{--} \begin{Bmatrix} \textbf{I–J–} \\ \textbf{R–} \end{Bmatrix} \textbf{F-- :}\ \text{Xp–Yp plane}$$

$$\textbf{G18} \begin{Bmatrix} \textbf{G02.1} \\ \textbf{G03.1} \end{Bmatrix} \textbf{Zp--Xp--}\alpha\textbf{--}\beta\textbf{--} \begin{Bmatrix} \textbf{K–I–} \\ \textbf{R–} \end{Bmatrix} \textbf{F-- :}\ \text{Zp–Xp plane}$$

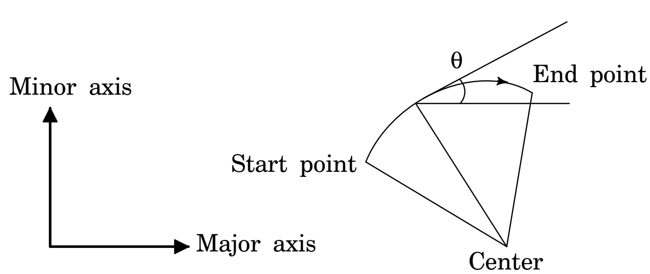
$$\textbf{G19} \begin{Bmatrix} \textbf{G02.1} \\ \textbf{G03.1} \end{Bmatrix} \textbf{Yp--Zp--}\alpha\textbf{--}\beta\textbf{--} \begin{Bmatrix} \textbf{J–K–} \\ \textbf{R–} \end{Bmatrix} \textbf{F-- :}\ \text{Yp–Zp plane}$$

where, α and β are two arbitrary axes other than the axes to which circular interpolation is applied.

WARNING In a block in which circular threading B is specified, tool offset or tool length compensation cannot be specified.

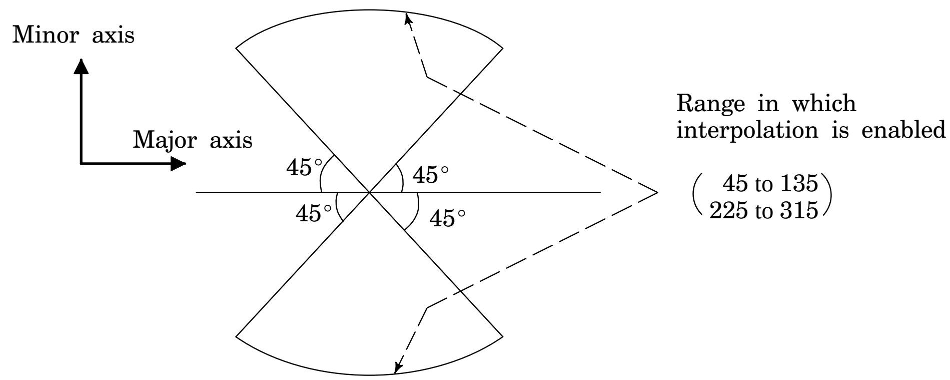


NOTE 1 The relationship between the major axis and minor axis is as follows:



If $|\Delta X| > |\Delta Y|$, the major axis is the X–axis, and the minor axis is the Y–axis.
If $|\Delta X| < |\Delta Y|$, the major axis is the Y–axis, and the minor axis is the X–axis.

NOTE 2 If an arc extends beyond the following range, PS alarm 201 occurs:

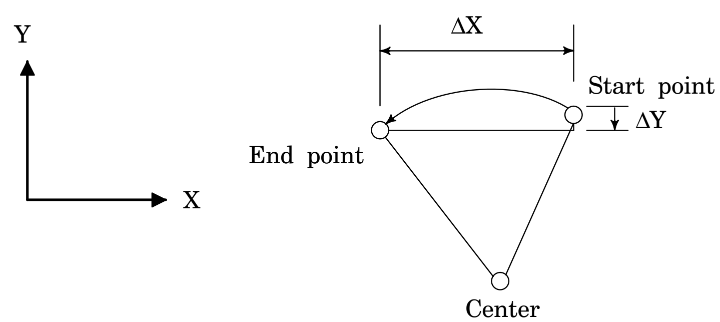


NOTE 3 If the specified feedrate for the major axis is F, the feedrate for the minor axis Fs and feedrate for the α axis F α are expressed as follows:

$$F\alpha = F \times \frac{\text{Length of } \alpha \text{ axis}}{\text{Length of major axis}}$$

$$Fs = F \times TAN\theta$$

θ: Angle of a tangent to the major axis



NOTE 4 Cutter compensation applies only to an arc.

4.13 Involute Interpolation

An involute curve can be machined using involute interpolation. Cutter compensation is also possible. Use of involute interpolation eliminates the need to approximate minute straight lines or arcs to a involute curve. This eliminates interruption in pulse distribution during high–speed operation of minute blocks. As a result, fast and smooth operation is enabled. In addition, machining tape can be created easier, and the required length of tape decreases.

(1) Involute curve

An involute curve in the X–Y plane is defined as follows:

$$\mathbf{X(\theta) = R\ [\cos\theta + (\theta-\theta_0)\ \sin\theta\ ] + X_0}$$

$$\mathbf{Y(\theta) = R\ [\ \sin\theta-(\theta-\theta_0)\ \cos\theta\ ] + Y_0}$$

where,

X_0, Y_0 : Center coordinates of a base circle
R : Radius of a base circle
θ_0 : Angle of the start point of an involute curve
θ : Angle of the contact point of a base circle and its tangent line drawn from the current position
$X(\theta)$, $Y(\theta)$: Current position on the X– and Y–axes

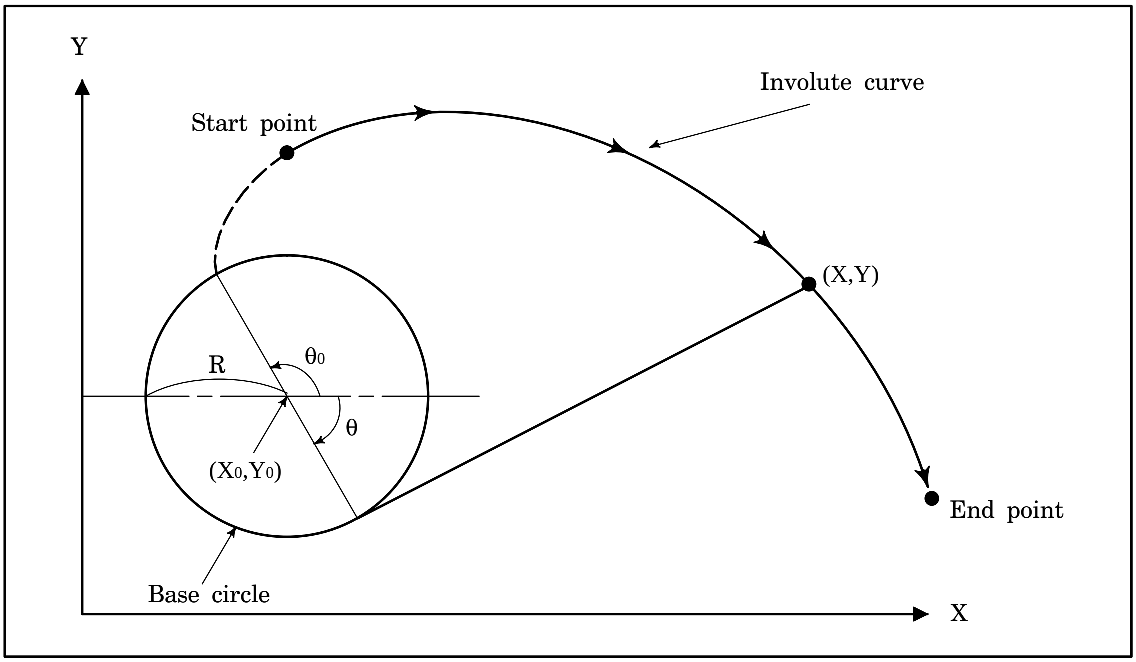


Fig. 4.13 (a) Involute curve

When an involute curve is produced in a plane other than the X–Y plane, X and Y in the above equations are replaced by the corresponding axis names of the target plane.

(2) Command format

Specify involute interpolation as follows:

Involute interpolation in the Xp–Yp plane

G17G02.2 Xp–Yp– I — J — R — F — ;
G17G03.2 Xp–Yp– I — J — R — F — ;

Involute interpolation in the Zp–Xp plane

G18G02.2 Zp–Xp– K — I — R — F — ;
G18G03.2 Zp–Xp– K — I — R — F — ;

Involute interpolation on the Yp–Zp plane

G19G02.2 Yp–Zp— J — K — R — F — ;
G18G03.2 Yp–Zp— J — K — R — F — ;

where,

G02.2	:	Clockwise involute interpolation
G03.2	:	Counterclockwise involute interpolation
G17/G18/G19	:	Selection of the Xp–Yp/Zp–Xp/Yp–Zp plane
Xp, Yp, Zp	:	Coordinates of the end point Pe of an involute curve
Xp	:	X–axis or a parallel axis (parameter setting, data No. 1022)
Yp	:	Y–axis or a parallel axis (parameter setting, data No. 1022)
Zp	:	Z–axis or a parallel axis (parameter setting, data No. 1022)
I, J, K	:	Center position of the base circle of an involute curve with respect to the start point Ps
R	:	Radius of the base circle
F	:	Cutting feedrate

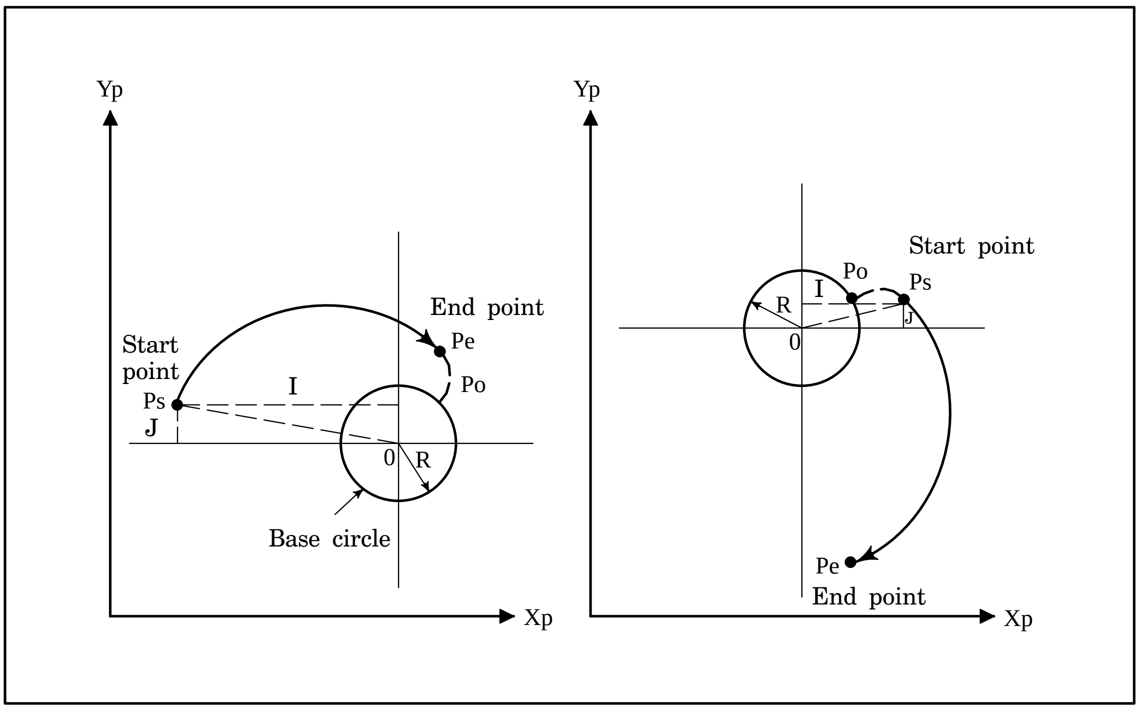


Fig. 4.13 (b) Clockwise involute interpolation (G02.2)

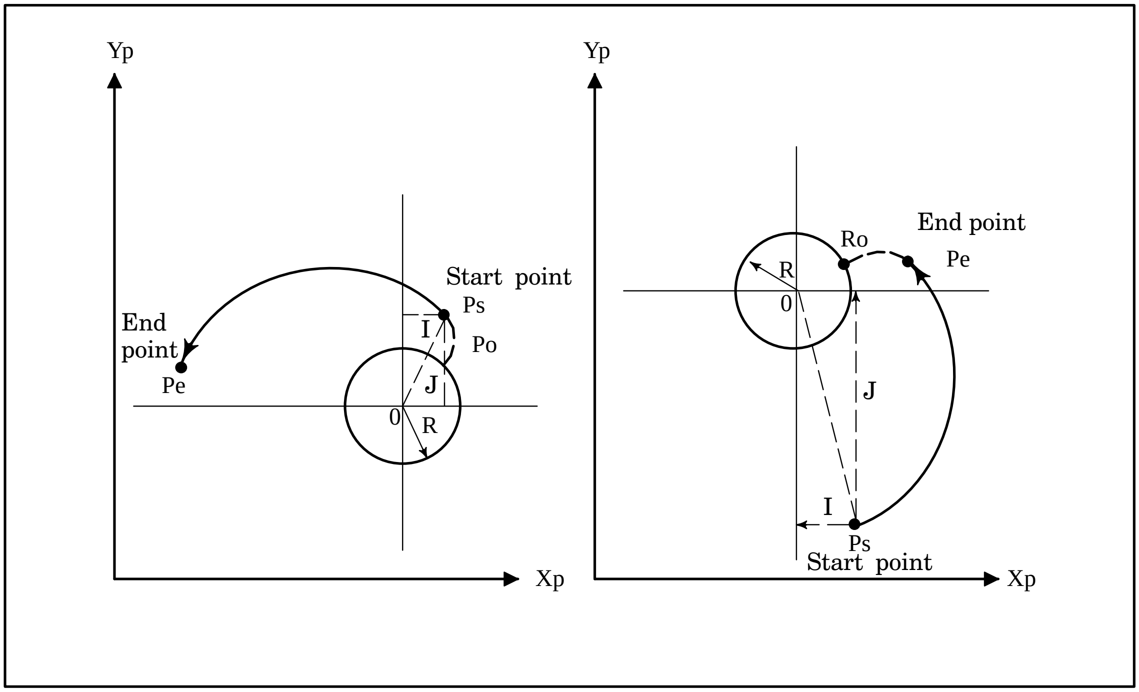


Fig. 4.13 (c) Counterclockwise involute interpolation (G03.2)

The end point of an involute curve is specified by address Xp, Yp, or Zp, and is indicated as an absolute value or incremental value, depending on the setting of G90 or G91. If an incremental value is used, the coordinates of the end point are specified with respect to the start point of the involute curve.

Both the start point and end point must be within the range of 100 rotations from the start of the involute curve. An involute curve involving one or more rotations can be specified with a single block. The center of the base circle is specified by I, J, and K which correspond to Xp, Yp, and Zp, respectively. Note that the numeric values following I, J, and K are vector components viewed from the start point of the involute curve to the center of the base circle, and they must always be specified with incremental values, regardless of whether G90 or G91 is specified. Add a sign to I, J, and K according to the direction.

When only start point, I, J, and K is provided, two involute curves are possible. One involute curve approaches the base circle, and the other goes away from the base circle. If the end point is nearer to the center of the base circle than the start point, an involute curve approaching the base circle is produced. If the end point is farther than the start point, an involute curve going away from the base circle is produced. The feedrate of involute interpolation is set to the cutting feedrate specified by an F code. The feedrate along the involute curve (feedrate along the tangent line to the involute curve) is controlled so that it matches the specified feedrate.

Select the target plane for involute interpolation by using G17, G18, or G19 in the same manner as circular interpolation.

(3) Involute interpolation and cutter compensation

Machining can be performed with cutter compensation applying to an involute curve. Cutter compensation is specified by using G40, G41, and G42 in the same manner as for a straight line or an arc. Specify an offset memory number with a D code.

G40: Cutter compensation cancel
G41: Offset at the left of a tool path
G42: Offset at the right of a tool path

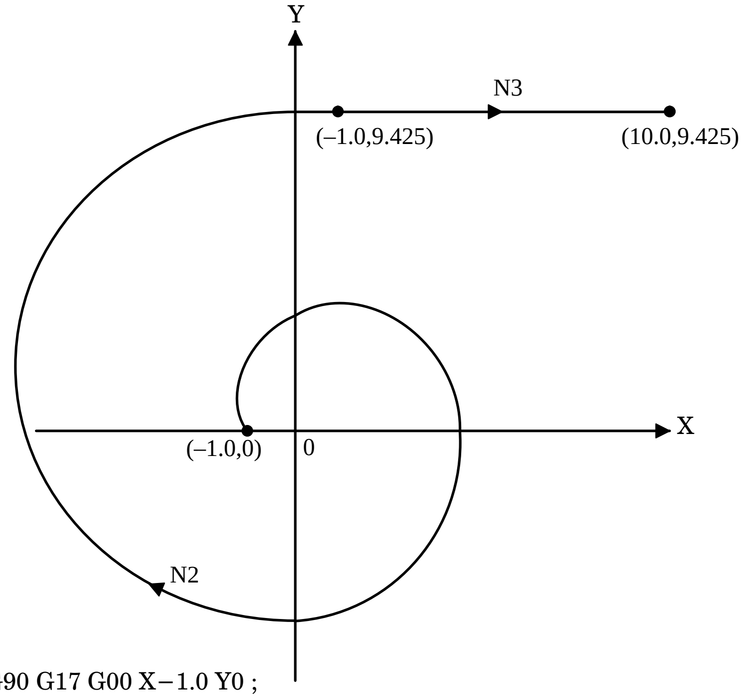
At the start and end points of the involute curve, intersections with a straight line or an arc are obtained by approximate calculation. An involute curve passing through the obtained intersections at the start point and end point is the path of the center of the tool.

In the involute interpolation mode, it is not possible to specify starting or canceling cutter compensation.

(4) Automatic speed control during involute interpolation

To improve the machining precision, an override can be automatically applied to the specified feedrate during involute interpolation. See Subsection 6.6.1 for details.

(5) Sample program




```
N1G90 G17 G00 X−1.0 Y0 ;
N2G02.2 X1.0 Y9.425 I1.0 J0 R1.0 F100 ;
N3G01 X10.0 ;
```

WARNING 1 In the involute interpolation mode, optional–angle chamfering/corner rounding cannot be specified.

WARNING 2 In the cylindrical interpolation mode, involute interpolation cannot be specified.

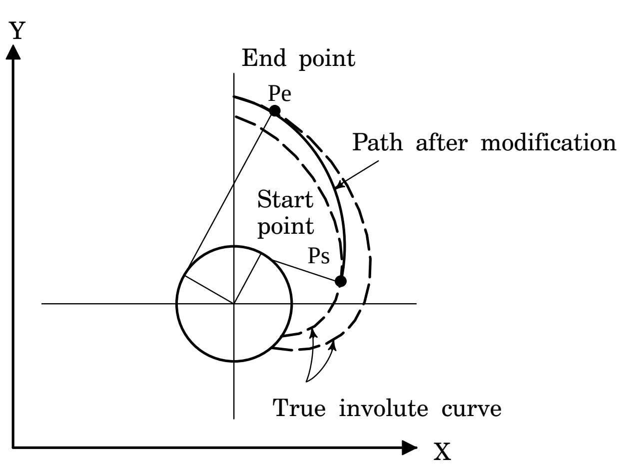
NOTE 1 If no end point is specified, an alarm occurs.

NOTE 2 If neither, I, J, nor K, are specified, or if I0J0K0 is specified, an alarm occurs.

NOTE 3 If R is not specified, or if R0 is specified, an alarm occurs.

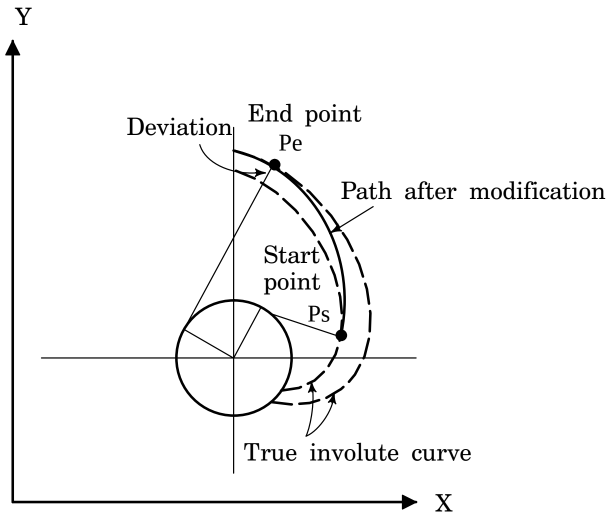
NOTE 4 If the start point or end point is located within the base circle, an alarm occurs.

NOTE 5 If the end point does not lie on an involute curve that passes through the start point, the following curve is produced:



Counterclockwise involute interpolation (G03.2)

NOTE 6 If the deviation between an involute curve passing through the start point and an involute curve passing through the end point is not smaller than the value set by a parameter (data No. 2510), an alarm occurs.



Counterclockwise involute interpolation (G03.2)

NOTE 7 The following G codes can be specified in the involute interpolation mode:
G04, G10, G11, G17, G18, G19, G65, G66, G67, G90, G91

NOTE 8 Even in the cutter compensation mode (G41, G42), programmable mirror image mode (G51.1), or coordinate system rotation mode, involute interpolation is enabled.

4.13.1 Involute interpolation with a linear axis and rotation axis

In the polar coordinate interpolation mode, an involute curve can be machined using involute interpolation. The involute curve to be machined is drawn in the plane of the linear axis and rotation axis.

In the polar coordinate interpolation mode, positions are indicated with an angle and distance from the center. The coordinates of the command end point, however, must be specified as Cartesian coordinates in the polar coordinate interpolation plane.

(1) How to specify involute interpolation

Specify involute interpolation (G02.2, G03.2) in a polar coordinate interpolation plane as follows. Specify the end point coordinates of an involute curve as Cartesian coordinates by using the axis addresses of the first plane axis (linear axis) and the second plane axis (rotation axis). The center position of the base circle of the involute curve viewed from the start point is determined by the axis of the base coordinate system to which the first plane axis corresponds.

– If the linear axis is the X–axis or a parallel axis, the Xp–Yp plane is assumed, and I and J are used.
– If the linear axis is the Y–axis or a parallel axis, the Yp–Zp plane is assumed, and J and K are used.
– If the linear axis is the Z–axis or a parallel axis, the Zp–Xp plane is assumed, and K and I are used.

The first plane axis (linear axis) and the second plane axis (rotation axis) are set with parameters 1032 and 1033.

G02.2 X–– C –– I –– J –– R –– F ––; X is the linear axis.
G03.2 X–– C –– I –– J –– R –– F ––;

where, G02.2 : Clockwise involute interpolation
G03.2 : Counterclockwise involute interpolation
X, C : End point linear axis coordinate of the involute curve, rotation axis
I, J : Center position of the base circle of the involute curve viewed from the start point
R : Radius of the base circle
F : Cutting feedrate

G02.2 Y–– C –– J –– K –– R –– F –– ; Y is the linear axis
G03.2 Y–– C –– J –– K –– R –– F –– ;
G02.2 Z–– C –– K –– I –– R –– F –– ; Z is the linear axis
G03.2 Y–– C –– J –– K –– R –– F –– ;

(2) Notes

In involute interpolation in a polar coordinate interpolation plane, only the G codes that can be used in both involute interpolation and polar coordinate interpolation are available. Therefore, the following G codes cannot be specified:

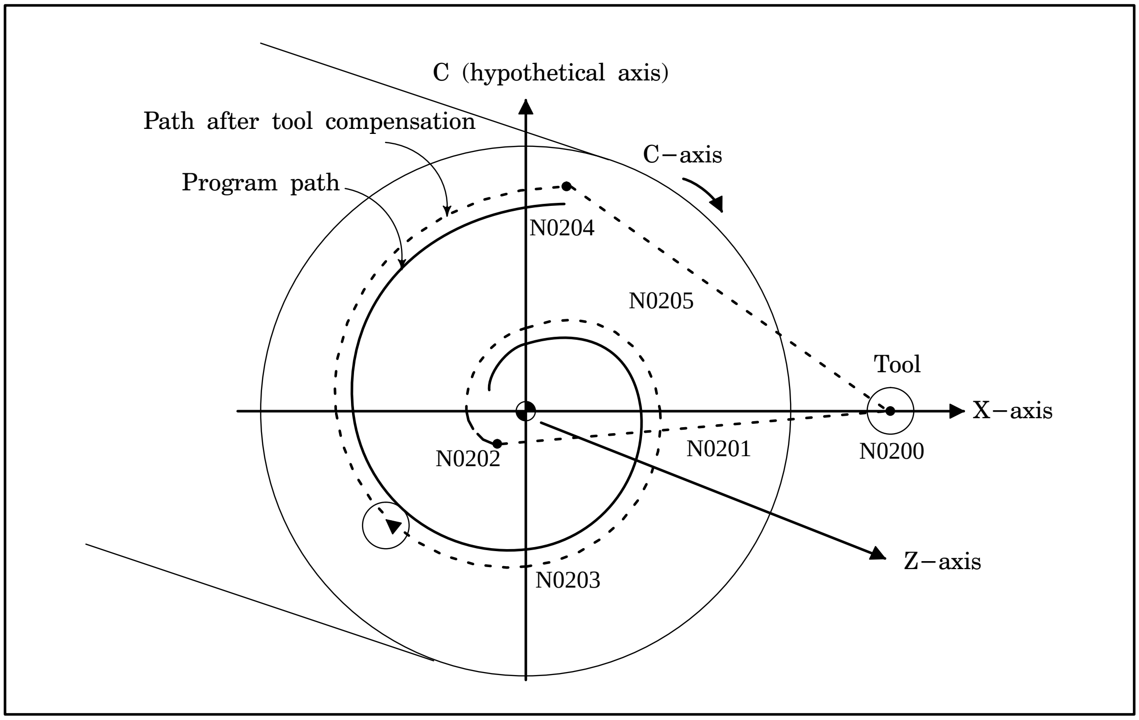
G17 : Xp–Yp plane selection
G18 : Zp–Xp plane selection
G19 : Yp–Zp plane selection
G10 : Data setting

Before specifying this function, cancel the following modes:

G51.1: Programmable mirror image
G68: Coordinate system rotation

(3) Sample program

Sample program for involute interpolation performed during polar coordinate interpolation using the X–axis (linear axis) and C–axis (rotation axis)




```
O0001 ;
  |
N0010   T0101 ;
  |
N0100   G90 G00 X15.0 C0 Z0 ;            Positioning to the start position
N0200   G12.1  ;                         Start of polar coordinate interpolation
N0201   G41 G00 X–1.0 ;
N0202   G01 Z–2.0 F_ ;
N0203   G02.2 X1.0 C9.425 I1.0 J0 R1.0 ; Involute interpolation during polar
N0204   G01 Z0 ;                         coordinate interpolation
N0205   G40 G00 X15.0 C0 ;
N0206   G13.1 ;                          Polar coordinate interpolation cancel
N0300   Z_  ;
N0400   X_C_;
  |
M30;
```

4.14 Helical Involute Interpolation

This interpolation function applies involute Interpolation to two axes and directs movement for up to four other axes at the same time. This function is similar to the helical function used in circular interpolation.

(1) Commands

Specify commands for helical involute interpolation as follows:

(1) For involute interpolation in the Xp–Yp plane,

G17 G02.2 Xp_ Yp_α_ β_ γ_ δ_ I_ J_ R_ F_;
G17 G03.2 Xp_ Yp_α_ β_ γ_δ_ I_ J_ R_ F_;

(2) For involute interpolation in the Zp–Xp plane,

G18 G02.2 Zp_ Xp_α_ β_ γ_ δ_ K_ I_ R_ F_;
G18 G03.2 Zp_ Xp_α_ β_ γ_ δ_ K_ I_ R_ F_;

(3) For involute interpolation in the Yp–Zp plane,

G19 G02.2 Yp_ Zp_α_ β_ γ_ δ_ J_ K_ R_ F_;
G19 G03.2 Yp_ Zp_α_ β_ γ_ δ_ J_ K_ R_ F_;

Addresses α, β, γ, and δ must incicate axes other than Yp and Zp. Other addresses must be the same as those used in the conventional involute interpolation.

(2) Notes

1. Cutter compensation is only applied to involute curves.
2. This function can be used for involute interpolation for a linear axis and rotation axis.

4.15 Spline Interpolation

Spline interpolation produces a spline curve connecting specified points. When this function is used, the tool moves along the smooth curve connecting the points. The spline interpolation command eliminates the need to approximate the smooth curve with minute straight lines or arcs. A machining program coded with this command requires less tape than that including the approximation.

(1) Command format

The G06.1 command puts the system into the spline interpolation mode. The mode is canceled when the G code of another interpolation mode is specified. In the block containing the G06.1 command, specify the first–order and second–order differential vectors at the start point. According to the specified data, the CNC unit produces the first spline curve, then more spline curves in succession. The spline interpolation mode uses three basic axes. Specify the components of the first–order and second–order differential vectors with floating–point data.

G06.1 Xx Yy Zz Ii Jj Kk Pp Qq Rr Ff ;

Xx : Mantissa of the X component of the first–order differential vector (Specify this without a decimal point.)

Yy : Mantissa of the Y component of the first–order differential vector (Specify this without a decimal point.)

Zz : Mantissa of the Z component of the first–order differential vector (Specify this without a decimal point.)

Ii : Mantissa of the X component of the second–order differential vector (Specify this without a decimal point.)

Jj : Mantissa of the Y component of the second–order differential vector (Specify this without a decimal point.)

Kk : Mantissa of the Z component of the second–order differential vector (Specify this without a decimal point.)

Pp : Exponents of the X components of the first–order and second–order differential vectors. Specify four digits without a decimal point. The first two digits represent the exponent of the second–order differential vector and the second two digits represent the exponent of the first–order differential vector. When values P_1 and P_2 are specified as shown below, the components of the vectors are expressed as follows:

$$p = \underbrace{\square\,\square}_{p2}\ \underbrace{\square\,\square}_{p1}$$

X component of the first–order differential vector = $x \times 10^{-P1}$

X component of the second–order differential vector = $i \times 10^{-P2}$

Example When G06.1 X12345 I9876 P0302; is specified, the components of the vectors are:

X component of the first–order differential vector = 123.45

X component of the second–order differential vector = 9.876

Qq : Exponents of the Y components of the first–order and second–order differential vectors. Specify the values like P described above.

Rr : Exponents of the Z components of the first–order and second–order differential vectors. Specify the values like P described above.

Ff : Feedrate at which the tool moves along a tangent. The feedrate is the tangential velocity in Cartesian coordinates formed by the specified axes. In the spline interpolation mode, each axis component of the velocity changes continually. With axis components of velocity Fx, Fy, and Fz, the velocity at a point is expressed as:

$$f = \sqrt{Fx^2 + Fy^2 + Fz^2}$$

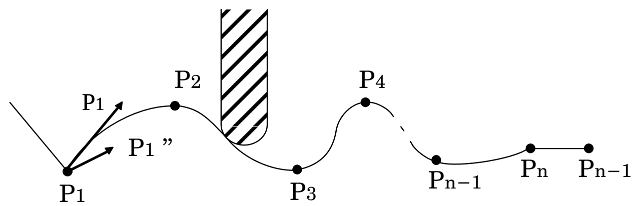
(Sample program)

The system is in the spline interpolation mode from N120 to N500 of the program below:

N110 G01	X __	Y __	Z __ ;		P^1
N120 G06.1	X __	Y __	Z __	I __ J __ K __ F__ ;	
N130	X __	Y __	Z __ ;		P^2
67N140	X __	Y __	Z __ ;		P^3
N150	X __	Y __	Z __ ;		P^4
. . .					
. . .					
N500	X __	Y __	Z __ ;		Pn
N510 G01	X __	Y __	Z __ ;		Pn + 1

P_1' : First–order differential vector

P_1" : Second–order differential vector



NOTE 1 In the block containing the G06.1 command, specify the components of the first–order and second–order differential vectors for spline interpolation. If the exponents are not specified with P, Q, or R, the system assumes that the value of the corresponding axis is entered in the usual CNC command format.

Example When G06.1 X12345 I23456 P0201; is specified, the components of the vectors are:

X component of the first–order differential vector = 1234.5
X component of the second–order differential vector = 234.56

When G06.1 X12345 I23456; is specified in the IS–B unit, the components of the vectors are:

X component of the first–order differential vector = 12.345
X component of the second–order differential vector = 23.456

NOTE 2 Specify the first–order and second–order differential vectors with the highest possible precision.

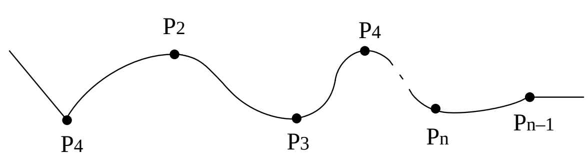
(2) Spline curve

A spline curve connecting n + 1 points consists of n parametric cubic curves. The general formula f(t) of the cubic curve is expressed as:

$P = f(t) = At^3 + Bt^2 + Ct + D$,

where P is the position vector at a point on the curve, t is a parameter, and A, B, C, and D are vector coefficients.

The CNC unit performs spline interpolation by calculating the coefficients according to the specified points and changing t.



In the spline interpolation mode, two curves joining at a point satisfy the following conditions at that point. This results in a smooth spline curve.

(a) The two curves have an identical joining point (specified point).

(b) The two curves have an identical tangential vector at the joining point. (The two curves have an identical first–order differential vector obtained for t.)

(c) The two curves have an identical curvature at the joining point. (The two curves have an identical second–order differential vector obtained for t.)

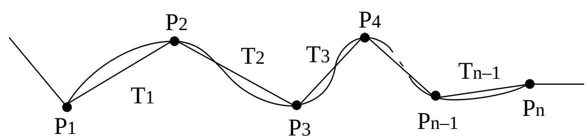
(3) Range of parameter t

When the range of parameter t for the spline curve is $0 \leqq t \leqq Ti$ and the length of the chord between points Pi and Pi+1 is represented as $\overline{Pi\ Pi{+}1}$ (i = 1 to n), T values are expressed as:

$T_1 = \overline{P1\ P2} / \overline{P1\ P2} = 1$
$T_2 = \overline{P2\ P3} / \overline{P1\ P2}$
. . .
$Tn - 1 = \overline{Pn - 1\ Pn} / \overline{P1\ P2}$

If NCOD is set to 1 in parameter 1001, the T values are:

T1 = T2 = . . . = Tn–1 = 1



(4) Initial conditions of the spline curve

With the first–order and second–order differential vectors specified in the block containing the G06.1 command, P'1 and P''1 respectively, the following four expressions determine the vector coefficients A1, B1, C1, and D1 of the first spline curve:

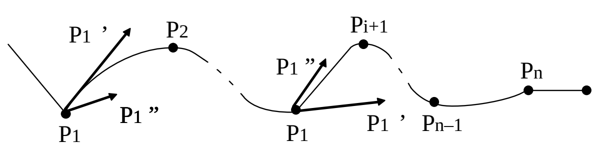
$f_1 \quad (0) = D_1 \qquad = P_1$
$f_1 \quad (1) = A_1 + B_1 + C_1 + D_1 = P_2$
$f'_1 \quad (0) = C_1 \qquad = P'_1$
$f''_1 \quad (0) = 2B_1 \qquad = P''_2$

The second and subsequent spline curves are generated in such a manner that P' and P'' at the end point of a curve are used as P' and P'' at the start point of the next curve.

The i–th spline curve is determined by the following four expressions:

fi (0) = Di = Pi
fi (Ti) = Ai Ti3 + Bi + Ti2 + Ci Ti + Di = Pi + 1
f'i (0) = Ci = f'i – 1 (Ti – 1)
f''i (0) = 2Bi = f''i – 1 (Ti – 1)

P_1' : First–order differential vector at point P_1
P_1'' : Second–order differential vector at point P_1
Pi' : First–order differential vector at point Pi
Pi'' : Second–order differential vector at point Pi



NOTE 1 When calculating the first–order and second–order differential vectors to be specified in the block containing the G06.1 command, use the following procedure as a guide:
The first–order differential vector at a point on the spline curve defined by n specified points is expressed as:

$$\begin{bmatrix} T_3 & 2(T_2+T_3) & T_2 & 0 & 0 & 0 \cdots\cdots 0 \\ 0 & T_4 & 2(T_3+T_4) & T_3 & 0 & 0 \cdots\cdots 0 \\ 0 & 0 & T_5 & 2(T_4+T_5) & T_4 & 0 \cdots\cdots 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 \cdots 0 & Tn^2 & (Tn-1+Tn) & Tn-1 & & \end{bmatrix} \begin{bmatrix} P_1' \\ P_2' \\ P_3' \\ \\ P'n-1 \\ Pn' \end{bmatrix}$$

$$= \begin{bmatrix} \dfrac{3}{T_2\,T_3}\,[T_2^2\,(P_3-P_2)+T_3^2\,(P_2-P_1)] \\ \dfrac{3}{T_3\,T_4}\,[T_3^2\,(P_4-P_3)+T_4^2\,(P_3-P_2)] \\ \dfrac{3}{T_4\,T_5}\,T_4^2\,[P_5-P_4)+T_5^2\,(P_4-P_3)] \\ \cdots\cdots \\ \dfrac{3}{Tn-1\;Tn}[Tn-1^2\,(Pn-Pn-1)+Tn^2\,(Pn-1-Pn-2)] \end{bmatrix} \qquad \text{Expression (1)}$$

where

Tn : Parameter value at end point Pn of the spline curve from Pn–1 to Pn. The range of the parameter (t) is $0 < t < Tn$.

Pn : Position vector at the specified point

Pn' : First–order differential vector at Pn

Expression (1) can be expressed as:

$$\begin{bmatrix} \overline{A} \end{bmatrix}\begin{bmatrix} P' \end{bmatrix} = \begin{bmatrix} B \end{bmatrix} \quad \cdots\cdots\cdots \quad \text{Expression (1)}$$

where

$\begin{bmatrix} \overline{A} \end{bmatrix}$: (n – 2) x n matrix

$\begin{bmatrix} P' \end{bmatrix}$: n x 1 matrix

$\begin{bmatrix} \overline{B} \end{bmatrix}$: (n – 2) x 1 matrix

As $\begin{bmatrix} \overline{A} \end{bmatrix}$ is a non–square matrix, expression (1) cannot be solved for $\begin{bmatrix} P' \end{bmatrix}$. To solve the expression for $\begin{bmatrix} P' \end{bmatrix}$, specify the boundary conditions at the ends of the curve and form square matrix $\begin{bmatrix} A \end{bmatrix}$. Then expression (2) can be expressed as:

$$\begin{bmatrix} A \end{bmatrix}\begin{bmatrix} P' \end{bmatrix} = \begin{bmatrix} B \end{bmatrix} \quad \cdots\cdots\cdots \quad \text{Expression (2)}$$

where

$\begin{bmatrix} A \end{bmatrix}$: n x n matrix

$\begin{bmatrix} B \end{bmatrix}$: n x 1 matrix

The expression above is solved for $\begin{bmatrix} P' \end{bmatrix}$ as shown below:

$$\begin{bmatrix} P' \end{bmatrix} = \begin{bmatrix} A \end{bmatrix}^{-1}\begin{bmatrix} B \end{bmatrix} \quad \cdots\cdots\cdots \quad \text{Expression (3)}$$

where $\begin{bmatrix} A \end{bmatrix}^{-1}$ is the inverse matrix of $\begin{bmatrix} A \end{bmatrix}$.

In the block containing the G06.1 command, specify P_1' obtained from expression (4) as the first–order differential vector. Calculate second–order differential vector P_1'' from the expression below, using P_1' and P_2' calculated from expression (4).

$$P_1'' = 2\ [\ \frac{3(P2-P1)}{T2^2} - \frac{2P1' + P2'}{T2}\ ] \cdots\text{ Expression (4)}$$

The following two examples shows how to determine the boundary conditions at both ends of a spline curve.

(1) When the second–order differential vectors at both ends are zero

This means that $P_1'' = Pn'' = 0$ $[\ A\ ]$ and $[\ B\ ]$ are calculated as follows:

$[\ A\ ]$

$$= \begin{bmatrix} 1 & 0.5 & 0 & \cdots & & & 0 \\ T_2 & 2(T_2+T_3) & T_2 & 0 & \cdots & & 0 \\ 0 & T_4 & 2(T_3+T_4) & T_3 & 0 & \cdots & 0 \\ 0 & 0 & T_5 & 2(T_4+T_5) & T_4 & 0\ \cdots & 0 \\ \cdots & & & & & & \\ 0 & \cdots & 0\ Tn & 2 & & (Tn-1+Tn) & Tn-1 \\ 0 & \cdots & & & 0 & 0 & 1 \end{bmatrix}$$

$[\ B\ ]$

$$= \begin{bmatrix} \frac{3}{2}\ \frac{(P_2-P_1)}{T_2} \\ \frac{3}{T_2\ T_3}[T_2^2\ (P_3-P_2)+T_3^2\ (P_2-P_1)\ ] \\ \frac{3}{T_3\ T_4}[T_3^2\ (P_4-P_3)+T_4^2\ (P_3-P_2)\ ] \\ \cdots \\ \frac{3}{T_4\ T_5}[T_4^2\ (P_5-P_4)+T_5^2\ (P_4-P_3)\ ] \\ \frac{3}{Tn-1\ Tn}[Tn-1^2\ (Pn-Pn-1)+Tn^2\ (Pn-1-Pn-2)\ ] \\ \frac{3}{Tn}(Pn-Pn-1) \end{bmatrix}$$

(2) When first–order differential vectors are specified at both ends

This means that P_1' and Pn' are specified. [A] and [B] are calculated as follows:

$[\ A\ ]$

$$= \begin{bmatrix} 1 & 0 & 0 & \cdots & & & 0 \\ T_3 & 2(T_2+T_3) & T_2 & 0 & \cdots & & 0 \\ 0 & T_4 & 2(T_3+T_4) & T_3 & 0 & \cdots & 0 \\ 0 & 0 & T_5 & 2\ (T_4+T_5) & T_4 & 0\ \cdots & 0 \\ \cdots & & & & & & \\ 0 & \cdots & 0\ \ Tn & 2 & & (Tn-1+Tn) & Tn-1 \\ 0 & \cdots & & & 0 & 0 & 1 \end{bmatrix}$$

$$[B] \begin{bmatrix} P_1 \\ \dfrac{3}{T_2\,T_3}\,[T_2{}^2\,(P_3-P_2)+T_3{}^2\,(P_2-P_1)\,] \\ \dfrac{3}{T_3\,T_4}\,[T_3{}^2\,(P_4-P_3)+T_4{}^2\,(P_3-P_2)\,] \\ \dfrac{3}{T_4\,T_5}\,[T_4{}^2\,(P_5-P_4)+T_5{}^2\,(P_4-P_3)\,] \\ \cdots\cdots\cdots\cdots \\ \dfrac{3}{Tn-1\;Tn}\,[Tn-1^2\,(Pn-Pn-1)+Tn^2\,(Pn-1-Pn-2)\,] \\ Pn' \end{bmatrix}$$

NOTE 2 The spline interpolation command (G06.1) belongs to G code group 01.

NOTE 3 The following G codes can be specified in the spline interpolation mode.

G04 : Dwell
G05.1 : Multi–buffer
G09 : Exact stop
G61 : Exact stop mode
G64 : Cutting mode
G65 : Macro calling
G66 : Continuous–state macro calling A
G66.1 : Continuous–state macro calling B
G67 : Cancellation of continuous–state macro calling A or B
G90 : Absolute command
G91 : Incremental command

NOTE 4 The spline interpolation mode can be specified in the following G–code modes:

G20 : Input in inches
G21 : Input in millimeters
G22 : Stored stroke check function on
G51 : Scaling
G51.1 : Programmable mirror image
G61 : Exact stop mode
G64 : Cutting mode
G68 : Coordinate system rotation
G90 : Absolute command
G91 : Incremental command
G94 : Feed per minute

NOTE 5 Before G06.1 is specified, the canned cycle mode must be canceled.

NOTE 6 Before G06.1 is specified, the following modes must be canceled: Tool length compensation, tool offset, cutter compensation, and three–dimensional tool compensation.

4.16 Spline Interpolation B

Spline interpolation B produces a spline curve connecting specified points. When this function is used, the tool moves along the smooth curve connecting the points. The spline interpolation command eliminates the need to approximate the smooth curve with minute straight lines or arcs. A machining program coded with this command requires less tape than that including the approximation.

4.16.1 Specifying the spline interpolation B command

When the following command is specified, the system enters the spline interpolation mode.

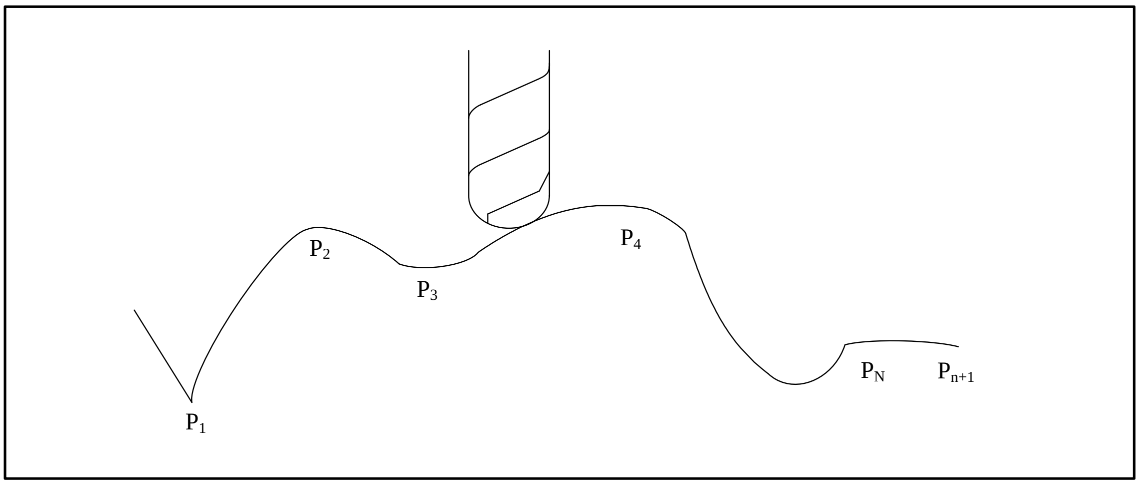
G06.1;

The spline interpolation mode is canceled when another G code of group 01 is specified.

(Sample program)

The system is in the spline interpolation mode from N120 to N500 of the program below:

```
N110G01    X___ Y___ Z___   ; P1
N120G06.1
N130       X___ Y___ Z___   ; P2
N140       X___ Y___ Z___   ; P3
N150       X___ Y___ Z___   ; P4
   :   :   :
   :   :   :
N500       X___ Y___ Z___   ; Pn
N510G01    X___ Y___ Z___   ; Pn + 1
```



In the spline interpolation mode, two curves joining at a point satisfy the following conditions at that point. This results in a smooth spline curve.

(1) The two curves have an identical joining point (specified point).

(2) The two curves have an identical tangential vector at the joining point. (The two curves have an identical first–order differential vector obtained for t.)

(3) The two curves have an identical curvature at the joining point. (The two curves have an identical second–order differential vector obtained for t.)

When a point is specified after G06.1, the CNC unit automatically produces the first spline curve, then more spline curves in succession.

In the block containing the G06.1 command, the tangential vector at the start point can be specified.

G06.1 X__ Y__ Z__ ;

where

X__ : X component of the tangential vector

Y__ : Y component of the tangential vector

Z__ : Z component of the tangential vector

In the G06.1 mode (the spline interpolation mode), specify a feedrate in the tangential direction with the F code.

The specified speed is the tangential velocity in Cartesian coordinates formed by the specified axes. In the spline interpolation mode, each axis component of the velocity changes continually. With axis components of velocity Fx, Fy, and Fz, the tangential velocity at a point is expressed as:

$F = (Fx^2 + Fy^2 + Fz^2)^{1/2}$

Spline interpolation can be executed in the three–dimensional tool compensation mode. The spline interpolation function automatically produces vectors for three–dimensional tool compensation.

Function system

A spline curve connecting n + 1 points consists of n parametric cubic curves. The general formula of the cubic curve is expressed as:

$P = At^3 + Bt^2 + Ct + D \quad (0 \leqq t < 1),$

where P is a vector at a point on the curve, t is a parameter, and A, B, C, and D are vector coefficients. The CNC unit accomplishes spline interpolation by calculating the coefficients according to the specified points and changing t.

4.16.2 Three–dimensional offset command

Spline interpolation can be executed in the three–dimensional tool compensation mode. The spline interpolation function automatically produces vectors for three–dimensional tool compensation in the spline interpolation mode. In the three–dimensional tool compensation mode, a spline curve connects the specified points which are offset by the vectors for three–dimensional tool compensation.

The spline interpolation function determines the vectors for three–dimensional tool compensation as shown below:

(1) Three–dimensional tool compensation vector at the start point

The three–dimensional tool compensation vector specified in the block of three–dimensional tool compensation command is used.

Three–dimensional tool compensation vector K specified by G41 I$\underline{i}$ J$\underline{j}$ K$\underline{k}$ D$\underline{d}$; has components Kx, Ky, and Kz, each of which is calculated as follows:

$Kx = (i \cdot r)/p$
$Ky = (j \cdot r)/p$
$Kz = (k \cdot r)/p$

where

r is the offset corresponding to the specified offset number d.

$p = (i^2 + j^2 + k^2)^{1/2}$

(2) Three–dimensional tool compensation vector at the second or subsequent point

1) Position : The vector is on the plane containing the point, previous point, and next point. It is perpendicular to the straight line connecting the previous and next points.

2) Direction : The direction of the vector is close to that of the three–dimensional tool compensation vector at the previous point. (When the direction of three–dimensional tool compensation vector V at the point is close to that of three–dimensional tool compensation vector V_0 at the previous point, the angle θ between V_0 and V satisfies the following condition : $|\theta| < 90°$)

3) Magnitude: The magnitude of the vector is the offset corresponding to the offset number specified by G41.

(3) Three–dimensional tool compensation vector at the last point

1) Position : The vector is on the plane containing the point, previous point, and next point. It is perpendicular to the straight line connecting the previous and next points.

2) Direction : The direction of the vector is close to that of the three–dimensional tool compensation vector at the previous point. (When the direction of three–dimensional tool compensation vector V at the point is close to that of three–dimensional tool compensation vector V_0 at the previous point, the angle θ between V_0 and V satisfies the following condition : $|\theta| < 90°$)

3) Magnitude: The magnitude of the vector is the offset corresponding to the offset number specified by G41.

After the system exits from the spline interpolation mode, the three–dimensional tool compensation vector produced at the last point is used.

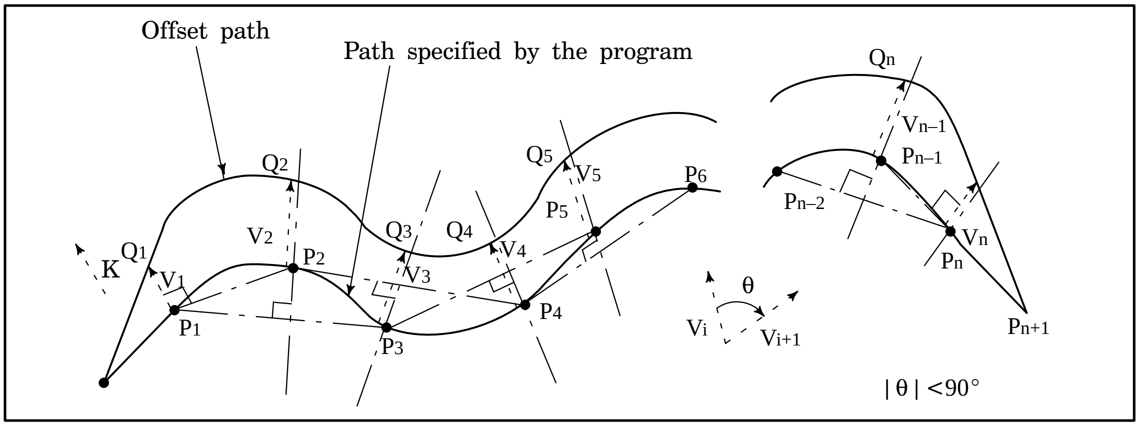
(Sample program)

The system is in the spline interpolation mode included in the three–dimensional tool compensation mode from N120 to N600 of the program below:

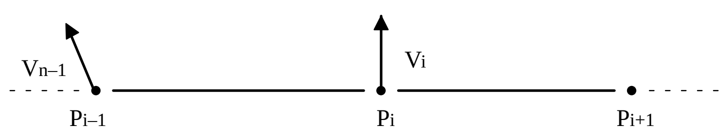
```
O1000                                        Specified point   Offset point
N110 G01 G41 X_ Y_ Z_ I_ J_ K_ D_ F_   ;      P1                Q1
N120 G06.1                             ;      P2                Q2
N130      X_ Y_ Z _                    ;      P3                Q3
N140      X_ Y_ Z _                    ;      P4                Q4
N150      X_ Y_ Z _                    ;      P5                Q5
     .    .    .                               .       .         .
N600      X_ Y_ Z _                    ;      Pn                Qn
N610 G01 X_ Y_ Z _                     ;      Pn + 1
```

K : Three–dimensional tool compensation vector specified by G41

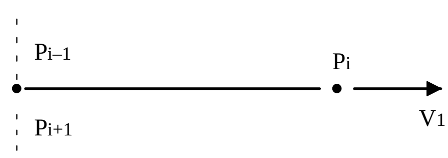
V_1 to Vn : Three–dimensional tool compensation vector used in the spline interpolation mode (V1 = K)



NOTE 1 If a point after the first point and the previous and next points are on a straight line, the vector plane cannot be determined for the three–dimensional tool compensation vector at the second or subsequent point. If this happens, the vector plane is determined by the straight line and the three–dimensional tool compensation vector at the previous point.

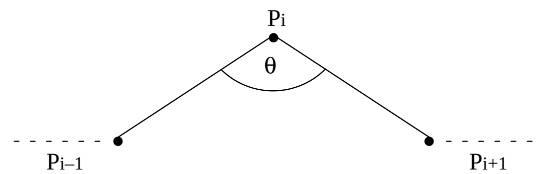


NOTE 2 If the points before and after a point other than the first point are identical, the three–dimensional tool compensation vector at the second or later point is produced in the direction of the straight line connecting the point and the previous (or next) point.



NOTE 3 If a point after the first point and the previous and next points are all identical, the three–dimensional tool compensation vector at the previous point is used as that at the point.

NOTE 4 Specify three points subjected to spline interpolation for which the three–dimensional tool compensation vectors are produced in such a manner that they make an obtuse angle. If they make an angle of 90°or less, the vector may not be produced in the correct direction.



NOTE

(1) This function requires a multi–buffer option.

(2) Specify points subjected to spline interpolation at as regular intervals as possible. The regular intervals between the specified points help a smooth spline curve to be produced.

(3) In the block containing the G06.1 command, specify a tangential vector whose magnitude is close to the distance between the first and second points subjected to spline interpolation. This helps a smooth spline curve to be produced.

(4) The spline interpolation command (G06.1) belongs to G code group 01.

(5) The axes to be specified in the spline interpolation mode must all be specified in the block containing the G06.1 command (when the tangential vector at the start point is specified) or the next block (when the tangential vector at the start point is not specified).

Example

When the tangential vector at the start point is specified		When the tangential vector at the start point is not specified	
N100 G06.1 X10. Y10. Z–10.	;	N100 G06.1	;
N110 X15.	;	N110 X15. Y0. Z0	;
N120 Y20. Z10	;	N120 Y20. Z10.	;
N130 X20. Y25. Z–10	;	N130 X20. Y25. Z–10	;

(6) In the spline interpolation mode, each block must contain a move command. The G or M codes which are not buffered must not be specified. If these rules are not observed, the block cannot produce a smooth spline curve.

(7) In the spline interpolation mode, a miscellaneous function must be specified in a block containing the corresponding move command.

(8) The following G codes can be specified in the spline interpolation mode:

G61 : Exact stop check
G64 : Cutting mode
G65 : Macro calling
G90 : Absolute command
G91 : Incremental command

The following G codes must be specified in a block containing the corresponding move command: G61, G64, G90, and G91.

(9) The spline interpolation mode can be specified in the following G–code modes:

G17	:	Selection of the XY plane
G18	:	Selection of the ZX plane
G19	:	Selection of the YZ plane
G20	:	Input in inches
G21	:	Input in millimeters
G22	:	Stored stroke check function on
G23	:	Stored stroke check function off
G40	:	Cancellation of cutter compensation or three–dimensional cutter compensation
G41	:	Three–dimensional tool compensation
G43	:	Tool length compensation mode
G49	:	Cancellation of tool length compensation
G50	:	Cancellation of scaling
G51	:	Scaling
G50.1	:	Cancellation of programmable mirror image
G51.1	:	Programmable mirror image
G54	:	Selection of workpiece coordinate system 1
G55	:	Selection of workpiece coordinate system 2
G56	:	Selection of workpiece coordinate system 3
G57	:	Selection of workpiece coordinate system 4
G58	:	Selection of workpiece coordinate system 5
G59	:	Selection of workpiece coordinate system 6
G61	:	Exact stop check
G64	:	Cutting mode
G66	:	Continuous–state macro calling A
G66.1	:	Continuous–state macro calling B
G67	:	Cancellation of continuous–state macro calling A or B
G68	:	Coordinate system rotation
G69	:	Cancellation of coordinate system rotation
G80	:	Cancellation of canned cycle
G90	:	Absolute command
G91	:	Incremental command
G94	:	Feed per minute

(10) Before G06.1 is specified, the canned cycle mode must be canceled.

(11) Before G06.1 is specified, the tool offset mode and cutter compensation mode must be canceled.

(12) Specify a move command in the first block of the subprogram to be called in the spline interpolation mode.

Example O1000; ——> O100 X_ Y_ Z_;

4.17 Spiral Interpolation and Conical Interpolation

Spiral interpolation can be carried out when the circular interpolation command is specified together with the number of circles of the helix or a radius increment or decrement per circle.

Conical interpolation can be carried out when the spiral interpolation command is specified together with commands specifying a movement along another axis and an increment or decrement along the axis per circle of the helix.

4.17.1 Spiral interpolation

· Xp Yp plane

$$\mathbf{G17}\left\{\begin{matrix}\mathbf{G02}\\\mathbf{G03}\end{matrix}\right\}\ \mathbf{X\_Y\_I\_J\_Q\_L\_F\_;}$$

· Zp Xp plane

$$\mathbf{G18}\left\{\begin{matrix}\mathbf{G02}\\\mathbf{G03}\end{matrix}\right\}\ \mathbf{Z\_X\_K\_I\_Q\_L\_F\_;}$$

· Yp Zp plane

$$\mathbf{G20}\left\{\begin{matrix}\mathbf{G02}\\\mathbf{G03}\end{matrix}\right\}\ \mathbf{Y\_Z\_J\_K\_Q\_L\_F\_;}$$

X,Y,Z	:	Position of the end point
L	:	Number of circles (positive integer)(*1)
Q	:	Radius increment or decrement per circle of the helix(*1)
I,J,K	:	Distance (positive or negative value) from the start point to the center (as specified for circular interpolation)
F	:	Feedrate

*1 Either L or Q can be omitted.

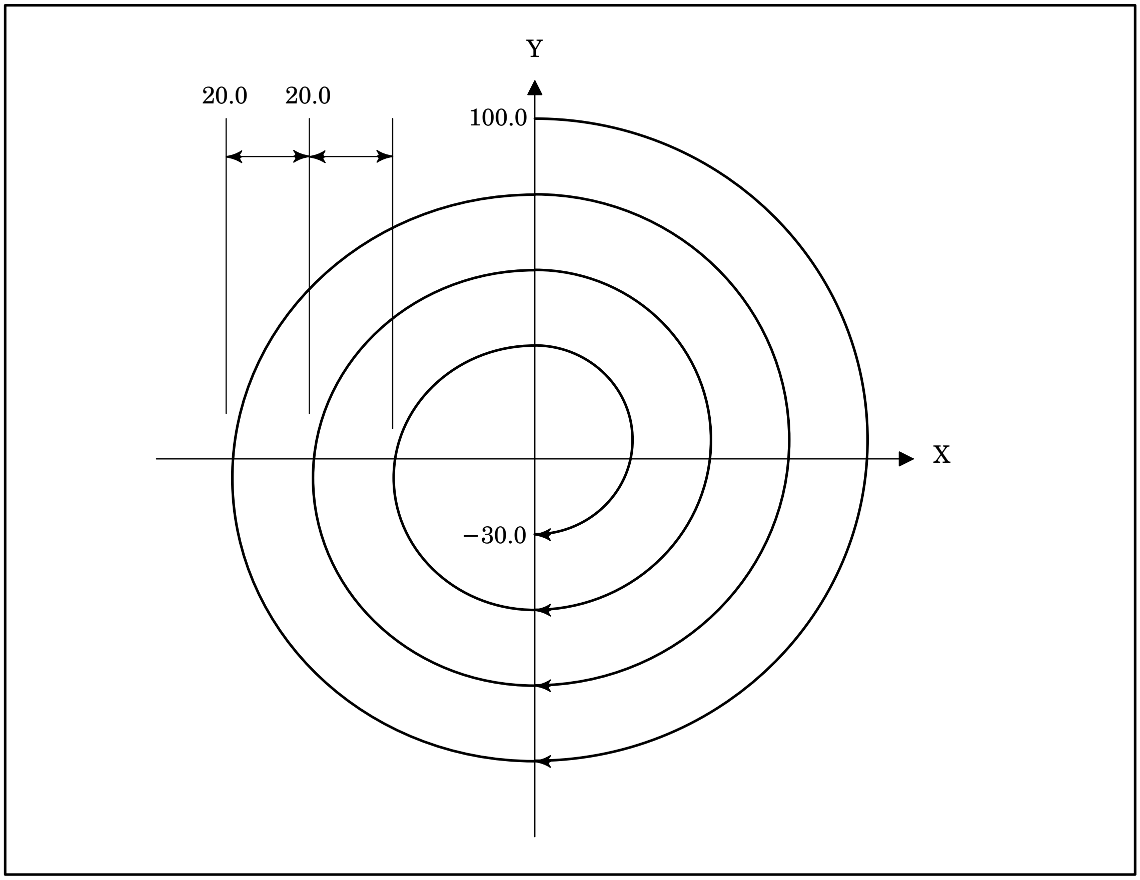
If L is omitted, the number of circles is automatically calculated from the current position, distance to the center, position of the end point, and radius increment or decrement.

If Q is omitted, the radius increment or decrement is automatically calculated from the current position, distance to the center, position of the end point, and number of circles.

If L and Q conflict when both are specified, Q has priority. Usually, specify either Q or L.

L must be a positive integer. To specify four circles and 90 degrees, round the number of circles up to 5 and specify L5.

[Example of spiral interpolation]



To specify this path in the absolute and incremental modes, program as follows:

The path can be described as follows:

- Start point : (0, 100.0)
- End point (X, Y) : (0, –30.0)
- Distance to the center (I, J) : (0, –100.0)
- Radius increment or decrement (Q) : –20.0
- Number of circles (L) : 4

(1) In the absolute mode

$$\text{G90 G02 X0 Y–30.0 I0 J–100.} \begin{Bmatrix} \text{Q–20.} \\ \text{L4} \end{Bmatrix} \text{F300;}$$

(2) In the incremental mode

$$\text{G91 G02 X0 Y–130.0 I0 J–100.} \begin{Bmatrix} \text{Q–20.} \\ \text{L4} \end{Bmatrix} \text{F300;}$$

Program as shown above. (Either Q or L can be omitted.)

[Spiral interpolation functions]

Spiral interpolation in the XY plane is defined as follows:

$(X-X_o)^2+(Y-Y_o)^2=(R+Q')^2$

X0 : X coordinate of the center

Y0 : Y coordinate of the center

R : Initial radius in spiral interpolation

Q' : Radius increment or decrement

The above equation can be rearranged using the following:

Xs : X coordinate of the start point

Ys : Y coordinate of the start point

I : Vector from the start point to the center (X coordinate)

J : Vector from the start point to the center (Y coordinate)

R : Initial radius in spiral interpolation

Q : Radius increment or decrement per circle of the helix

L' : Current number of circles minus 1

θ : Angular displacement between the start point and the current position (degrees)

$$(X-X_s-I)^2+(Y-Y_s-J)^2$$

$$=(R+(L' \frac{\theta}{360})\,Q)^2$$

The equation is transformed to:

4.17.2 Conical interpolation

Conical interpolation can be executed when the spiral interpolation command is specified with commands specifying a movement along another axis and an increment or decrement along the axis per circle of the helix.

Specify the following command to execute conical interpolation:

· Xp Yp plane

$$\mathbf{G17}\begin{Bmatrix}\mathbf{G02}\\\mathbf{G03}\end{Bmatrix}\ \mathbf{X\_Y\_I\_J\_Z\_K\_Q\_L\_F\_;}$$

· Zp Xp plane

$$\mathbf{G18}\begin{Bmatrix}\mathbf{G02}\\\mathbf{G03}\end{Bmatrix}\ \mathbf{Z\_X\_K\_I\_Y\_J\_Q\_L\_F\_;}$$

· Yp Zp plane

$$\mathbf{G19}\begin{Bmatrix}\mathbf{G02}\\\mathbf{G03}\end{Bmatrix}\ \mathbf{Y\_Z\_J\_K\_X\_I\_Q\_L\_F\_;}$$

X,Y,Z : Position of the end point

L : Number of circles (positive integer value)(*1)

Q : Radius increment or decrement per circle of the helix(*1)

I,J,K : Two of these represent a signed vector from the start point to the center. The other one indicates a height increment or decrement per circle of the helix in conical interpolation. (*1)

When the Xp Yp plane is selected,
I and J represent a signed vector from the start point to the center.
K indicates a height increment or decrement per circle of the helix.
When the Zp Xp plane is selected,
I and K represent a signed vector from the start point to the center.
J indicates a height increment or decrement per circle of the helix.
When the Yp Zp plane is selected,
J and K represent a signed vector from the start point to the center.
I indicates a height increment or decrement per circle of the helix.

F : Feedrate(*2)

*1 Only one of the height increment or decrement (I, J, or K), radius increment or decrement (Q), and number of circles (L), needs to be specified. The other two can be omitted.

·Sample command for interpolation in the Xp Yp plane

$$\mathrm{G17}\begin{Bmatrix}\mathrm{G02}\\\mathrm{G03}\end{Bmatrix}\ \mathrm{X\_Y\_I\_J\_Z}\quad\begin{Bmatrix}\mathrm{K\_}\\\mathrm{Q\_}\\\mathrm{L\_}\end{Bmatrix}\ \mathrm{F\_;}$$

If L and Q conflict when both are specified, Q has priority.

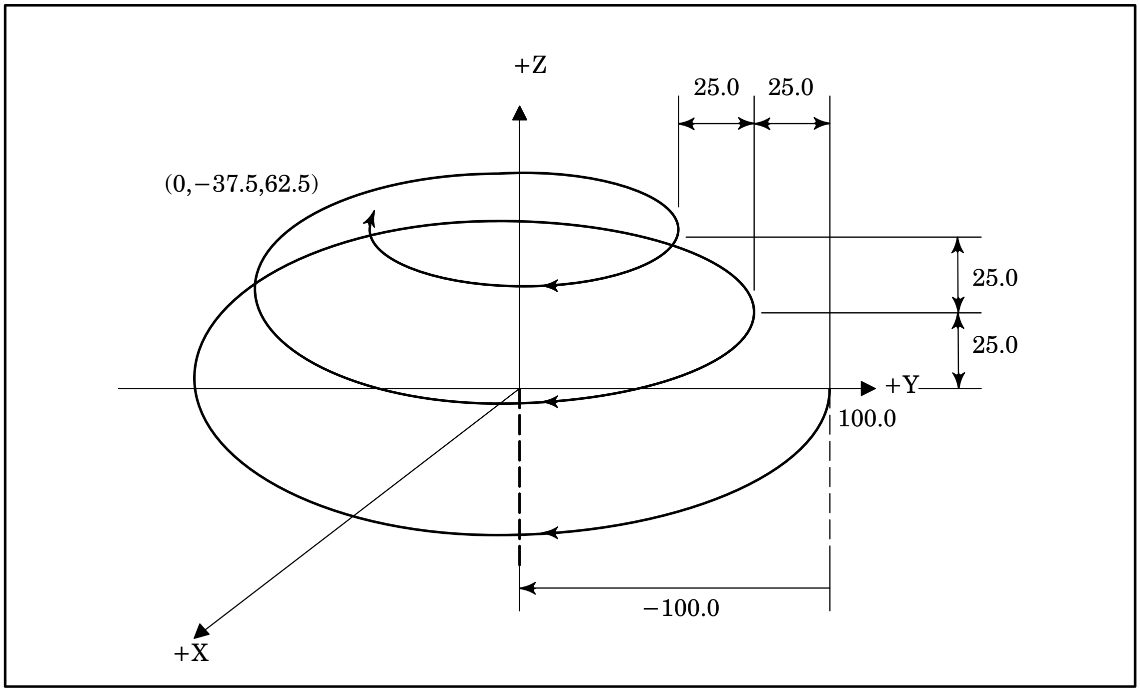
If L and the height increment or decrement (I, J, or K) conflict when both are specified, the height increment or decrement has priority.

If Q and the height increment or decrement (I, J, or K) conflict when both are specified, Q has priority.

L must be a positive integer. To specify four circles and 90 degrees, round the number of circles up to 5 and specify L5.

*2 To select whether the feedrate is specified by the speed tangential to an arc together with or without the speed along the linear axis, use the HTG bit of parameter 1401.

[Example of conical interpolation]



To specify this path in the absolute and incremental modes, program as follows:

The path can be described as follows:

·Start point	:	(0, 100.0, 0)
·End point (X, Y, Z)	:	(0, –37.5, 62.5)
·Distance to the center (I, J)	:	(0, –100.0)
·Radius increment or decrement (Q)	:	–25.0
·Height increment or decrement (K)	:	25.0
·Number of circles (L)	:	3

(1) In the absolute mode

$$\text{G90 G02 X0 Y–37.5 Z62.5 I0 J–100.} \left\{ \begin{array}{l} \text{K25.} \\ \text{Q–25.} \\ \text{L3} \end{array} \right\} \text{F300;}$$

(2) In the incremental mode

$$\text{G91 G02 X0 Y–137.5 Z62.5 I0 J–100.} \left\{ \begin{array}{l} \text{K25.} \\ \text{Q–25.} \\ \text{L3} \end{array} \right\} \text{F300;}$$

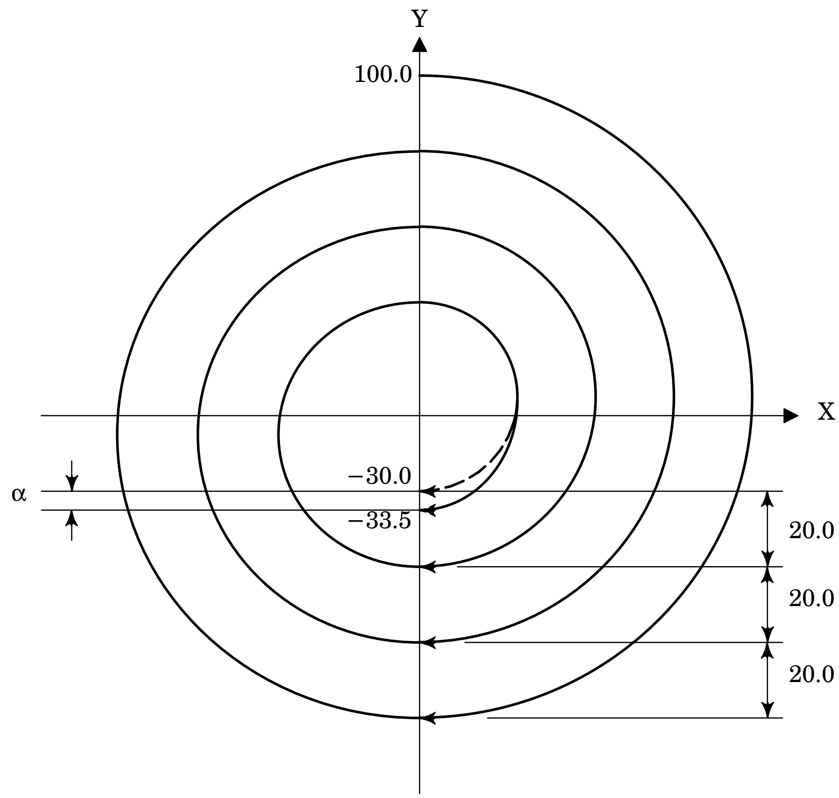
Program as shown above.

NOTE 1 An arc radius (R) for the spiral or conical interpolation function, cannot be used.

NOTE 2 For the conical interpolation function, a total of three axes, two axes in the same plane and another axis, can be specified. The axis not on the plane can be a rotation axis.

NOTE 3 If the axis name extension function specifies I, J, K, or Q as an axis name, the spiral or conical interpolation function cannot be used.

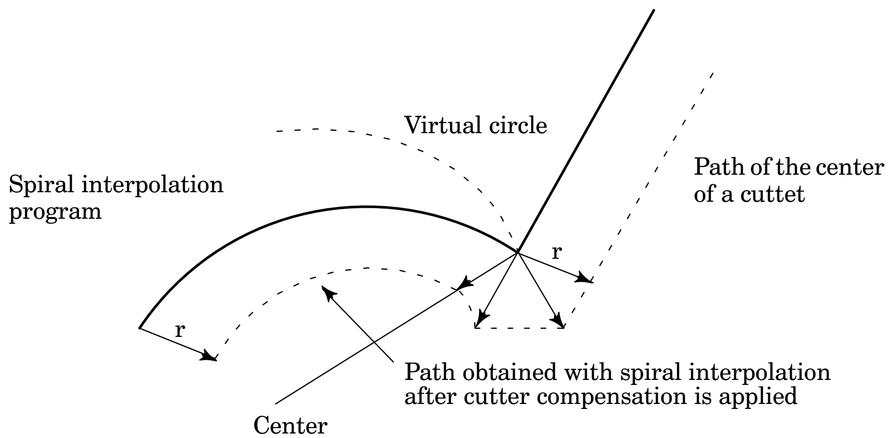
NOTE 4 If the difference between the positions of specified and calculated end points of a helix along an axis in the specified plane exceeds the value set in parameter 2511, alarm PS281 is issued. If the difference in height between the positions of specified and calculated end points of a helix exceeds the value set in parameter 2511, alarm PS281 is issued.



Command : G90 G02 X0 Y−33.5 I0 J−100. $\left\{\begin{matrix} Q-20. \\ L4 \end{matrix}\right\}$ F300;

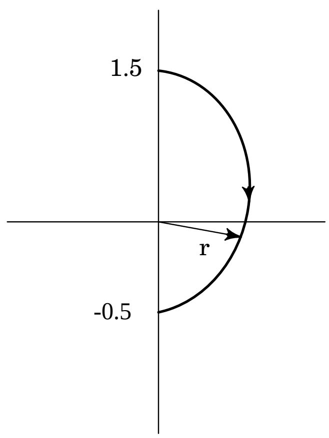
When the specified end point is (0, –33.5), the calculated end point is (0, –30.0). Specify a value greater than the difference (α : permissible error) in parameter 2511. Also set the parameter in this way, for the difference in height in conical interpolation.

NOTE 5 The spiral interpolation function and conical interpolation function can be used in cutter compensation C mode. The same compensation is applied as that described in (d) Exceptional case, (3) Offset mode, 15.3.7 Detailed explanation of cutter compensation C. Cutter compensation is applied along a virtual circle whose center is located at the center of spiral interpolation and which passes through the end point of a block. Spiral interpolation is applied to the path obtained after cutter compensation is applied.



NOTE 6 In spiral interpolation and conical interpolation, the feedrate is usually kept constant. When the radius of a spiral becomes small near the center of the spiral, however, the corresponding angular velocity may become very high. To prevent this from occurring, the system keeps the angular veiocity constant when the radius of a spiral reaches the value specified in parameter No.2440, resulting in a lower velocity than before.

An example is shown below.



Specified feedrate : F100
Parameter No.2440(r) : 1000
Increment system : IS−B

When spiral interpolation is used under the above conditions, feedrate is F100 for a redius larger then the value specified in parameter No.2440. The feedrate is reduced as the radius becomes smaller and is about F65 at the end point of the spiral.

NOTE 7 To use the conical interpolation function, the optional helical interpolation is required in addition to this option.

5. THREADING (G33)

5.1 Constant–Lead Threading (G33)

The G33 command produces a straight or tapered thread having a constant lead.

When the command is specified in the following format, a thread is produced with the lead determined by the numerical value specified after address F. An angle at which threading is started can be shifted. By changing the threading start angle, a multistart thread can be easily produced.

G33IP –– F –– Q ––;

where

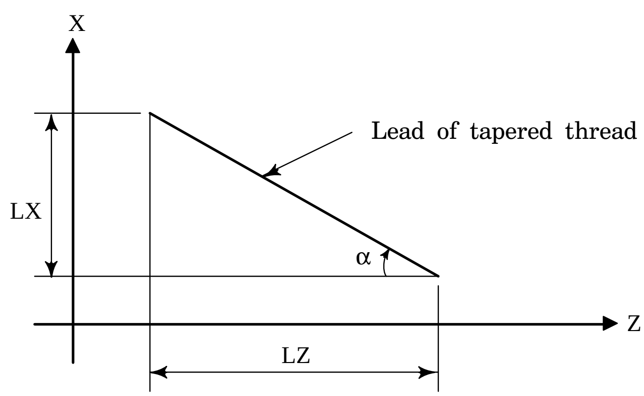
F –– : Larger component of lead

Q –– : Angle by which the threading start angle is shifted (0 to 360°)

Generally, threading includes many operations from rough machining to finish machining. A thread is produced after the different operations are done along an identical path.

At the beginning of threading, movement is started after the system detects a one–rotation signal from the position coder mounted on the spindle. The steps of threading are always started at an identical point on the circumference of the workpiece. The tools always go along the same path. The spindle speed must be kept constant during threading, that is, from rough machining to finish machining. If the spindle speed varies, the resulting thread may shift.

When a tapered thread is produced, the lead must be specified with the magnitude of a larger component. A lathe which holds and rotates a workpiece can produce a tapered thread on the workpiece.



When angle α is less than or equal to 45°, specify LZ.
When angle α is greater than or equal to 45°, specify LX.

At the beginning and end of threading, the servo system usually delays. This results in a partially incorrect lead. A longer thread than required needs to be specified so that the sections with the wrong lead can be eliminated afterward. The following leads can be specified:

	Increment		Allowable range of lead	
Input in millimeters	0.01	mm	0.0001 to 5000.0000	mm/rev
	0.001	mm	0.00001 to 500.00000	mm/rev
	0.0001	mm	0.000001 to 50.000000	mm/rev
	0.00001	mm	0.0000001 to 5.0000000	mm/rev
	0.000001	mm	0.00000001 to 0.50000000	mm/rev
Input in inches	0.001	inch	0.00001 to 500.00000	inch/rev
	0.0001	inch	0.000001 to 50.000000	inch/rev
	0.00001	inch	0.0000001 to 5.0000000	inch/rev
	0.000001	inch	0.00000001 to 0.50000000	inch/rev

The spindle speed is limited as shown below:

$$1 \leqq R \leqq \frac{\text{Maximum feedrate}}{\text{Lead}}$$

$R \leqq$ Allowable speed of position coder

R : Spindle speed (rpm)

Unit of lead : Millimeters or inches

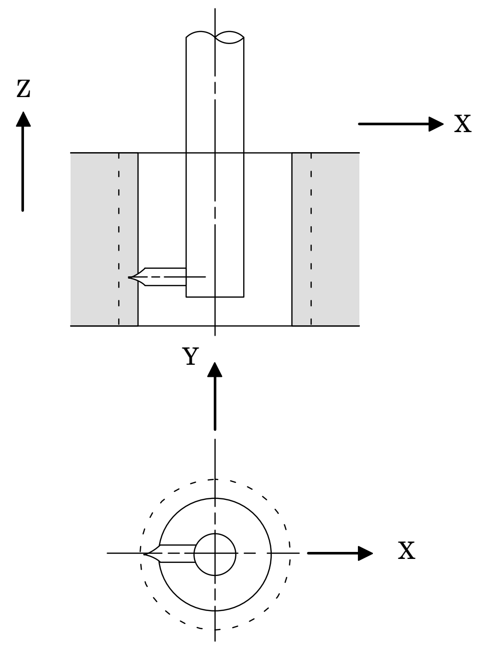
Maximum feedrate : Maximum value in the command of feed per minute (mm/min, inch/min) or maximum feedrate determined by the motor or machine, whichever is smaller

Allowable speed of position coder: 4000 rpm (position coder A), 6000 rpm (position coder B)

WARNING 1 The override for the converted cutting feedrate is always set to 100%.
WARNING 2 The feed hold function is invalidated in the threading mode. If the feed hold button is pressed in the threading mode, the feed hold function is validated at the end of the block immediately after threading (G33 mode) is terminated.

NOTE 1 The system momentarily reads the spindle speed from the position coder mounted on the spindle. It converts the detected spindle speed into a cutting feedrate per minute and moves the tool at that feedrate.
NOTE 2 The converted cutting feedrate is clamped.

Example



Program

```
N20G90G00X100.0Y...S45  M3 ;
N21Z200.0 ;
N22G33Z120.0F5.0 ;
N23M19 ;
N24G00X105.0 ;
N25Z200.0M0 ;
N26X100.0M3 ;
N27G04X2.0 ;
N28G33Z120.0F5.0 ;
 .
 .
```

Description

N20, N21	:	The center of the tool is aligned with the center of a prepared hole. The spindle rotates in the forward direction.
N22	:	The first step of threading is started. The lead is specified by address F.
N23	:	The M19 command stops the spindle at a specific position on the circumference as shown in the figure above. (M19: Spindle stop at a specific position)
N24	:	The tool is moved away along the X–axis.
N25	:	The tool is lifted to a position above the hole. The M00 command stops the program. Meanwhile, the operator can adjust a tool for the second threading operation.
N26	:	The center of the tool is aligned with the center of the prepared hole. The spindle rotates in the forward direction.
N27	:	As the move command in the N26 block is short, this block specifies the dwell command. The system waits until the spindle speed becomes constant.
N28	:	The second threading operation is started.

5.2 Inch Threading (G33)

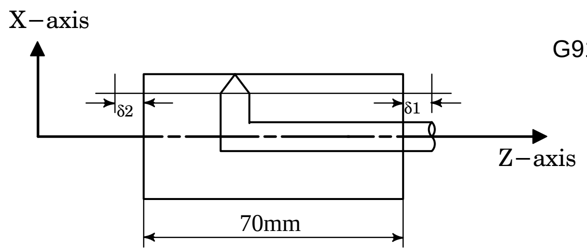
When a number of thread ridges per inch is specified with address E, an inch thread can be produced with high precision.

G33 IP– –E– –Q– –;

E– – : Number of thread ridges per inch
Q– – : Angle by which the threading start angle is shifted (0 to 360°)

(Sample inch threading)

Lead : Three ridges per inch (= 8.4666……)
δ1 = 3 mm
δ2 = 1.5 mm
Depth of cut : 1 mm (Two cuts are made.)
The program is specified with the conditions above.
(Data is input in millimeters and the diameter is specified.)



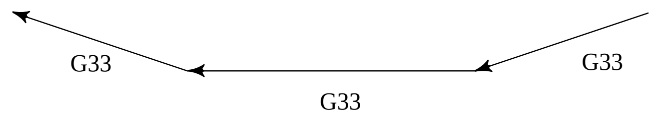

```
G91  G33  Z–74.5  E3.0 ;
              (Three ridges per inch)
     G00  X–10.0 ;
          Z–74.5 ;
          X–11.0;(A cut of another 1mm is made during
                  the second pass.)
     G33  Z–74.5  E3.0 ;
     G00  X–10.0 ;
          Z74.5
```

NOTE	Parameter SLE (data No. 2402) selects whether the number of ridges per inch or a precise lead is specified with address E.

5.3 Continuous Threading

Continuous threading can be executed when multiple blocks containing the threading command are specified in succession.

At the interface between blocks, the system keeps synchronous control of the spindle as much as possible. The lead or profile of a thread can be changed in the middle of threading.



Repeating the threading operations along an identical path with a different depth of cut enables the thread to be produced correctly.

NOTE	The threading start angle can be shifted only in the block in which the first threading operation is started.

6. FEED FUNCTIONS

6.1 Rapid Traverse

Positioning is done in rapid motion by the positioning command (G00).
There is no need to program rapid traverse rate, because the rates are set in the parameter (parameter No. 1420) (Per axis).
Rapid traverse rate can be overridden by a switch on the machine operator's panel:
F0, F1, 50, 100%.
F0: A constant speed set by parameter 1421
F1: Constant percentage set by parameter 1412

The rapid traverse feedrate can be overridden as described above.

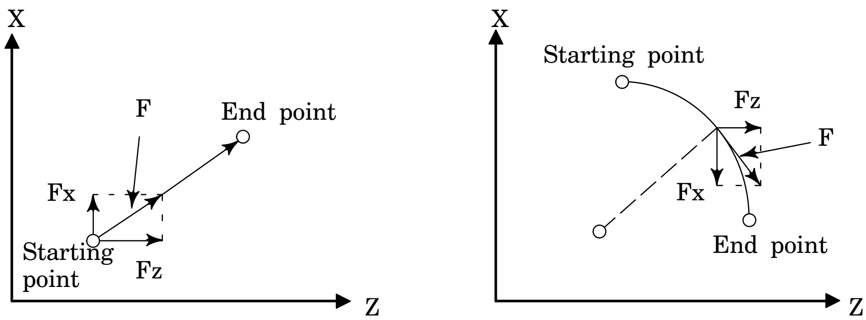
NOTE When the IS–B unit is used, an error of up to 8 mm/min may be present in the feedrate specified by the parameter.

6.2 Cutting Feed Rate

Feed speed of linear interpolation (G01), and circular interpolation (G02, G03) are commanded with numbers after the F code.

6.2.1 Tangential speed constant control

In cutting feed, it is controlled so that speed of the tangential direction is always the same commanded speed.



In case of linear   In case of circular

$$F = \sqrt{Fx^2 + Fz^2}$$

F :Feedrate along tangential direction
Fx :Feedrate component along X–axis direction
Fz :Feedrate component along Z–axis direction

6.2.2 Cutting feed rate clamp

Cutting feed rate upper limit can be set as parameter (data No.1422). If the actual cutting feed rate (feed rate with override) is commanded exceeding the upper limit it is clamped to a speed not exceeding the upper limit value.
The clamped values is set in mm/min or inch/min.

fm = fr × R

fm	:	Feed per minute	mm/min or inch/min
fr	:	Feed per revolution	mm/rev or inch/rev
R	:	Spindle speed	rpm

Except during acceleration or deceleration, the CNC arithmetic error for the command value of the feed rate is within ±2%. This error is applied to the time measured for the tool to move a distance of 500 mm or more under stationary conditions.

6.2.3 Feed per minute (G94)

With the feed per minute mode G94, tool feed rate per minute is directly commanded by numerical value after F.

G94 is a continuous–state code. It continues to be valid until G95 (feed per rotation) is specified.

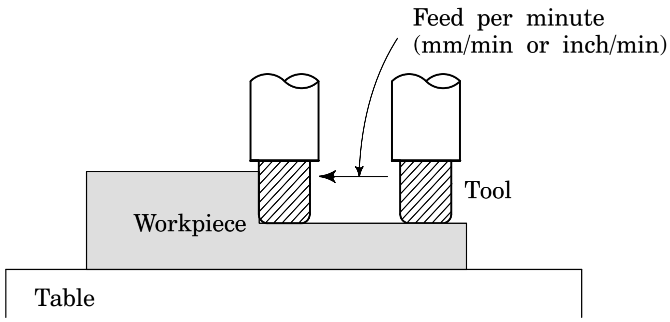
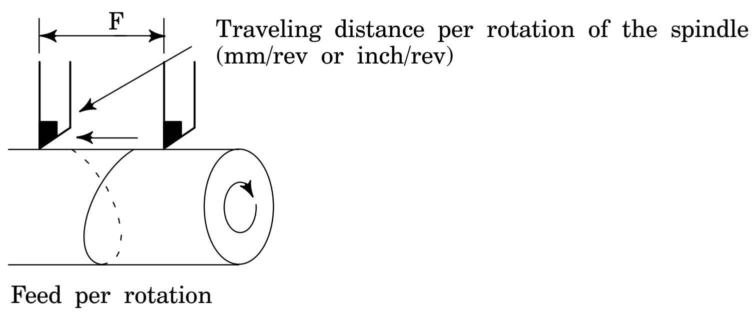


Fig. 6.2.3 (a) Feed per minute

6.2.4 Feed per rotation (G95)

When G95 is specified to select the mode of feed per rotation, specify the F code. The numerical value following the F code indicates the distance the tool travels with every rotation of the spindle.

G95 is a continuous–state code. It continues to be valid until G94 (feed per minute) is specified.



NOTE	When the position coder rotates at 1 rpm or less, the feedrate may fluctuate. The speed of 1 rpm or less can only be used if the fluctuation in feedrate does not adversely affect the machining. The degree of fluctuation depends on the spindle speed below 1 rpm. The lower the speed, the greater the fluctuation.

Unit of the F value in feed per minute and feed per rotation

(1) F value without a decimal point

Input system		Feed per minute (G94)		Feed per rotation (G95)
Millimeter	F41(*1)	0	1	
	Valid range	F1 to F24000	F1 to F240000	F1 to F50000
	Linear axis	1mm/min	0.1mm/min	0.01mm/rev
	Rotation axis	1deg/min	0.1deg/min	0.01deg/rev
Inch	Output system	Millimeter	Inch	
	Valid range	F1 to F96000	F1 to F240000	F1 to F500000
	Linear axis	0.01inch/min	0.01inch/min	0.0001inch/rev
	Rotation axis	0.01deg/min	0.01deg/min	0.0001deg/rev

*1 F41 is specified by parameter 2400, bit 1. (The setting is invalid if the IS–A unit is used.)

(2) F value with a decimal point

The numerical value after F equals the actual value in millimeters, degrees, or inches.

(3) Determining a feedrate when different units are selected for the axes

Degrees are converted to millimeters or inches.

Example 1

The input unit system is millimeters. The command of feed per minute is specified. The following units are used. F41 is set to 0.

X : IS–B (0.001mm)
Y : IS–B (0.001mm)
C : IS–C (0.0001deg)
G01 G91 G94 X 100.0 Y 200.0 C 300.0 F1000 ;

300 degrees are converted to 300 millimeters.

$$L = \sqrt{100^2 + 200^2 + 300^2} = 374.17\text{mm} :$$

Time T required for distribution is calculated as:

$$T = \frac{374.17\text{mm}}{1000\text{mm/min}} = 0.37417 \text{ min}$$

Feedrate Fc along the C–axis is calculated as:

$$Fc = \frac{300\text{deg}}{0.37417\text{min}} = 801.784 \text{ deg/min}$$

Example 2

The input unit system is inches. The command of feed per minute is specified. The following units are used:

X : IS–B (0.0001inch)
Y : IS–B (0.0001inch)
C : IS–C (0.0001deg)
G01 G91 G94 X 100.0 Y 200.0 C 300.0 F1000 ;

300 degrees are converted to 300 inches.

$$L = \sqrt{100^2 + 200^2 + 300^2} = 374.17\text{inch} :$$

Time T required for distribution is calculated as:

$$T = \frac{374.17\text{mm}}{10\text{inch/min}} = 37.417 \text{ min}$$

Feedrate Fc along the C–axis is calculated as:

$$Fc = \frac{300\text{deg}}{37.417\text{min}} = 8.018 \text{ deg/min}$$

6.2.5 Inverse time (G93)

The G93 command puts the system into the inverse time mode (G93 mode). The inverse time (FRN) is specified with the F code. The format of the F code is: F4.3 (0.001 to 9999.999).

(1) FRN in linear interpolation (G01)

$$FRN = \frac{\text{Velocity}}{\text{Distance}}$$

Unit of velocity: mm/min (input in millimeters) or inch/min (input in inches)
Unit of distance: mm (input in millimeters) or inches (input in inches)

(2) FRN in circular interpolation (G02, G03)

$$FRN = \frac{\text{Velocity}}{\text{Radius of arc}}$$

Unit of velocity : mm/min or inch/min
Unit of radius : mm or inches

NOTE 1 In the G93 mode, the F code is not handled as a continuous–state code. The F code must be specified in each block, otherwise an alarm occurs.

NOTE 2 G93 is a continuous–state G code that belongs to the same group (05) as G95 (feed per rotation) and G94 (feed per minute).

NOTE 3 If F0 is specified in the G93 mode, the feedrate equals the maximum cutting speed.

The table below shows the format of G93, G94, and G95.

	Inverse time	Feed per minute	Feed per rotation
Description	An inverse time is specified	The tool is moved by the specified distance per minute	The tool is moved by the specified distance per rotation of the spindle.
Address	F	F	F
G code	G93	G94	G95
Override	The values of G93, G94, and G95 can be overridden.		
Clamp value	The values are clamped at the maximum cutting speed. The clamp value is specified by the machine tool builder. (The overridden value is clamped.)		

NOTE 4 When the cutter compensation function is used, actual move of a programmed command is compensated by the cutter radius. Therefore, (distance) and (radius of arc) in FRN expression are different from the programmed values. And the actual velocity may be different from the commanded velocity. Therefore, do not use the cutter compensation function in the inverse time mode.

6.2.6 F1–digit feed

Specifying a one–digit number (1 to 9) following F produces a feed rate set correspondingly to that number. The feed rate is set in advance as a parameter for each number. F0 produces rapid traverse speed. Rotating the manual pulse generator with the F 1–digit feed rate change switch on the machine operator's panel ON, increases or decreases the feed rate for the currently selected number.

The increase or decrease of feed rate $\Delta F = \dfrac{\text{Fmax i}}{100X}$ **per scale of manual pulse generator,**

where, Fmax 1 : feed rate upper limit for F1–F4 set by parameter 1460
Fmax 2 : feed rate upper limit for F5–F9 set by parameter 1465
X : any value of 1–127 set by parameter 1450

The feed rate set or altered is kept even while the power is off. The current feed rate is displayed on the CRT.

When plural manual pulse generators are available, use this 1st manual pulse generator without fail.

6.2.7 Parameter input of feedrate

A parameter can be used to specify a cutting feedrate if the machine does not need to frequently change the cutting feedrate during machining. When the parameter is used, the cutting feedrate need not be specified with the F code in the NC command data.

If parameter No. 1493 is set to a value other than zero, the cutting feedrate is initialized to the value of the parameter when the power is turned on.

If the parameter is set to zero, the cutting feedrate is not specified by the parameter. The cutting feedrate is not initialized when the power is turned on.

Even when the parameter is used, the cutting feedrate specified in the NC command data continues to be valid. The cutting feedrate is retained until another cutting feedrate is specified or the power is turned off.

When a cutting feedrate is specified by the parameter, that is, when parameter No. 1493 is set to a value other than zero, the continuous–state F code is not cleared at reset, regardless of the value of parameter NCM (No. 2401, bit 7).

6.3 Override

Refer to machine tool builder’s manual for details of overrides.

6.3.1 Feedrate override

The feed per minute (G94) and the feed per revolution (G95) can be overridden using the switch on the machine operator’s panel by:
0 to 254% (per every 1%) or 0 too 200% (per every 10%)
Feed rate override cannot be applied to functions in which override is inhibited.
The feedrate cannot be overridden in threading and tapping.

6.3.2 Secondary feedrate override

Each cutting feedrate can be overridden by 0 to 254% (in 1% units). This function overrides an already overridden feedrate.
The feedrate cannot be overridden in threading and tapping.

6.3.3 Second feedrate override B

The second feedrate override B function can be applied in the range of 0 to 655.34% with an increment of 0.01%.

6.3.4 Rapid traverse override

Rapid traverse rate can be overridden using the switch on the machine operator’s panel by:

F0, 25, 50, 100%.
F0: A constant speed set in each axis by parameter (No. 1421)
F1: Constant percentage set by parameter 1412

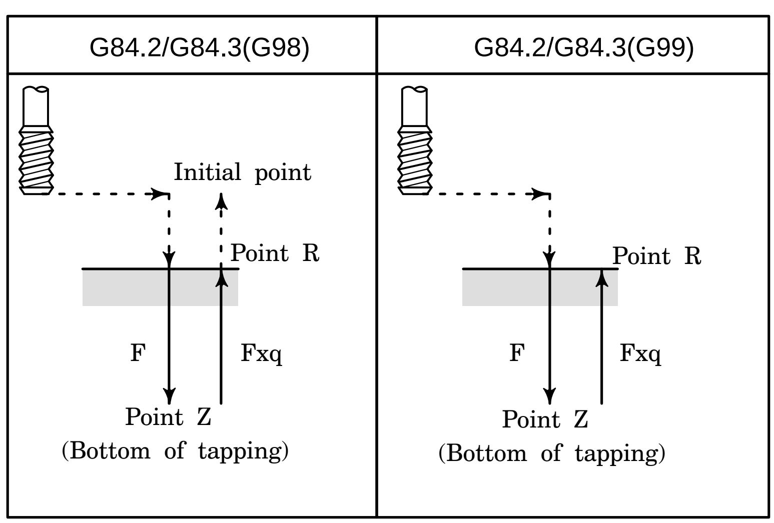
The feedrate can be overridden as described above.

6.3.5 Canceling the override function

When the machine sends a certain signal, the feedrate override and the secondary feedrate override can be clamped to 100%.

6.3.6 Function for overriding the return speed in rigid tapping

In rigid tapping, an override can be provided for the speed at which the tool is returned from the bottom of tapping to point R.
Parameter No. 5705 can specify an override in the range of 10% to 100% in 10% units.



F:Cutting speed q:Override

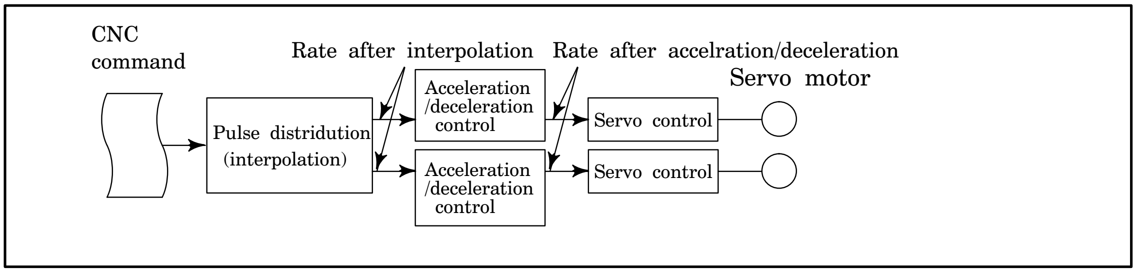
Fig. 6.3.6 Override in Rigid Tapping

6.4 Automatic Acceleration/Deceleration

6.4.1 Automatic acceleration/deceleration after interpolation

This function calculates the feedrate and travel of the tool from the specified NC statement. Once the CNC has distributed the pulses (interpolation) for each axis, the feedrate data, in pulses, for each axis is adjusted to produce smooth movement by applying acceleration/deceleration control, then passed to the servo motor.

Acceleration and deceleration is performed when starting and ending movement, resulting in smooth start and stop.
Automatic acceleration/deceleration is also performed when feed rate changes, so change in speed is also smoothly done.
It is not necessary to take acceleration/ deceleration into consideration when programming.



Rapid traverse: Linear acceleration/deceleration for rapid traverse
(when bit 0 of parameter No.1600 is set to 0 and bit 5 of parameter No.1601 is set to 0)
Bell–shaped acceleration/deceleration for rapid traverse
(when bit 0 of parameter No.1600 is set to 0 and bit 5 of parameter No. 1601 is set to 1)
Exponential acceleration/deceleration for rapid traverse
(when bit 0 of parameter No.1600 is set to 1)

Cutting feed : Linear acceleration/deceleration for cutting feed
(when bit 4 of parameter No.1600 is set to 0, without the option specifying bell-shaped acceleration/deceleration after interpolation)
Bell–shaped acceleration/deceleration for cutting feed
(when bit 4 of parameter No. 1600 is set to 0, with the option specifying bell–shaped ac celeration/deceleration after inter polation)
Exponential acceleration/deceleration for cutting feed
(when bit 4 of parameter No.1600 is set to 1)

Jog feed : Linear acceleration/deceleration for jog feed
(when bit 5 of parameter No.1600 is set ton 0, without the option specifying bell–shaped ac celeration/deceleration after interpolation)
Bell–shaped acceleration/deceleration for jog feed
(when bit 5 of parameter No.1600 is set to 0, with the option specifying bell–shaped ac celeration/deceleration after interpolation)
Exponential acceleration/deceleration for jog feed
(when bit 5 of parameter No.1600 is set to 1)

When the parameters are set to the default values, linear acceleration/deceleration is applied to each feed type.

The table below lists the parameters for the time constant and FL speed corresponding to each acceleration/ deceleration type and feed type.

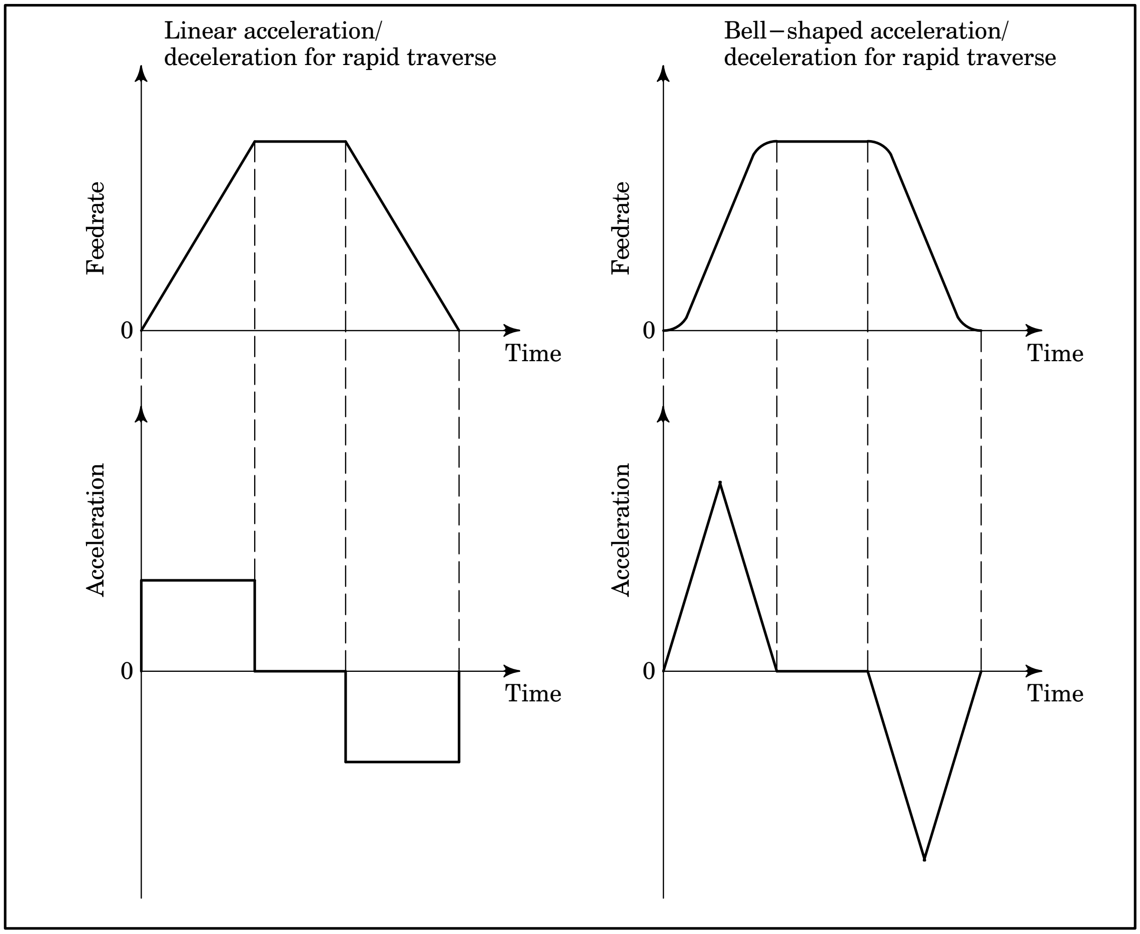
Feed type	Acceleration/deceleration type	Parameter for time constant	Parameter for FL speed
Rapid traverse	Linear Bell–shaped Exponential	1620 1620, 1636 1628	1621 1621 1629
Cutting feed	Linear Bell–shaped Exponential	1622 1622 1622	—— (*) —— (*) 1623
Jog feed	Linear Bell–shaped Exponential	1624 1624 1624	—— (*) —— (*) 1623

* 0 is assumed regardless of the specified value.

(1) Bell–shaped acceleration/deceleration after rapid traverse interpolation

The function for bell–shaped acceleration/deceleration after rapid traverse interpolation increases or decreases the rapid traverse feedrate smoothly. This reduces the shock to the machine system due to changing acceleration when the feedrate is changed.

As compared with linear acceleration/deceleration, bell–shaped acceleration/deceleration allows smaller time constants to be set, reducing the time required for acceleration/deceleration.

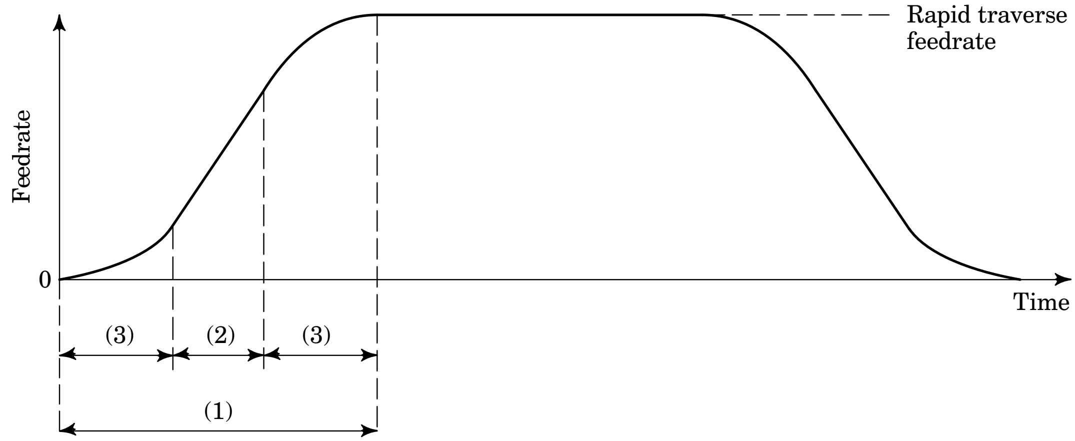


For bell–shaped acceleration/deceleration after rapid traverse interpolation, set time constants in parameters 1620 and 1636. Acceleration/deceleration is then performed according to these time constants.

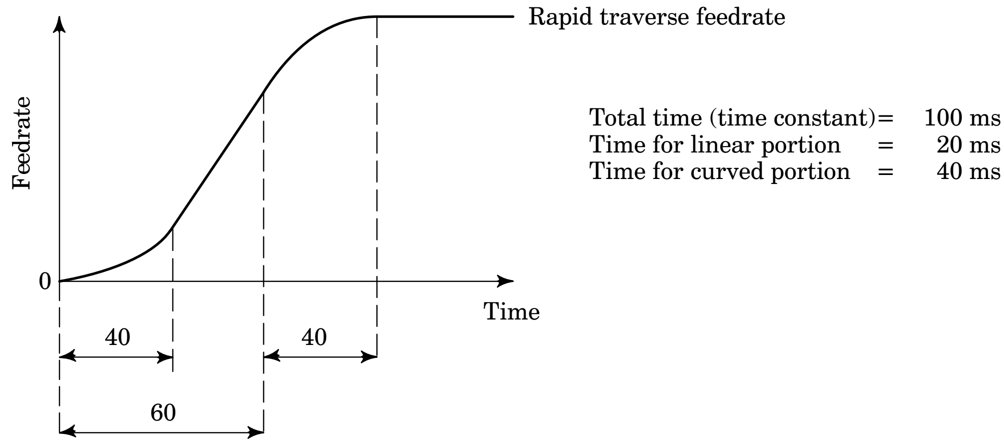
T1 : Value set in parameter 1620

T2 : Value set in parameter 1636 (Set parameters so that T1 > T2.)

Total time (time constant) = T1 + T 2 ...... (1)
Time for linear portion = $T_1 - T_2$ .... (2)
Time for curved portion = T_2 .... (3)



<**Example**> T_1 : 60msec
T_2 : 40msec

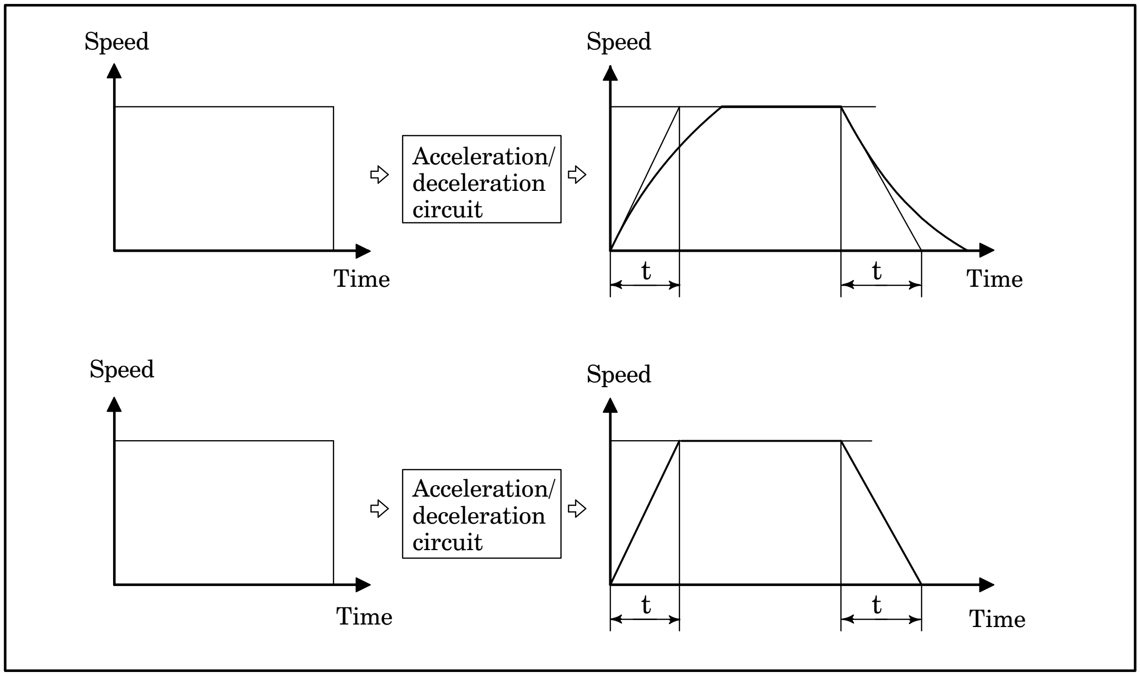


(2) Linear acceleration/deceleration of cutting feed after interpolation

The feedrate is increased or decreased as follows:

Rapid traverseq	:	Linear acceleration/deceleration (constant acceleration). The acceleration/deceleration time constant of each axis is specified in parameter 1620.
Cutting feed	:	Linear acceleration/deceleration (constant acceleration period). The acceleration/deceleration time constant of each axis is specified in parameter 1622.
Jog feed	:	Linear acceleration/deceleration (constant acceleration period). The acceleration/deceleration time constant of each axis is specified in parameter 1624.

Linear acceleration/deceleration (with a constant acceleration time) is applied to cutting feed and jog feed. If an identical time constant is used, the delay for command caused by linear acceleration/deceleration is a half of that caused by exponential acceleration/deceleration. The time required for acceleration/deceleration can be greatly reduced. In circular interpolation, especially when circular cutting is executed at a high speed, the actual path of the accelerated or decelerated tool deviates from the specified arc in the direction of the radius. The error caused by linear acceleration/deceleration is significantly smaller than that by exponential acceleration/deceleration.



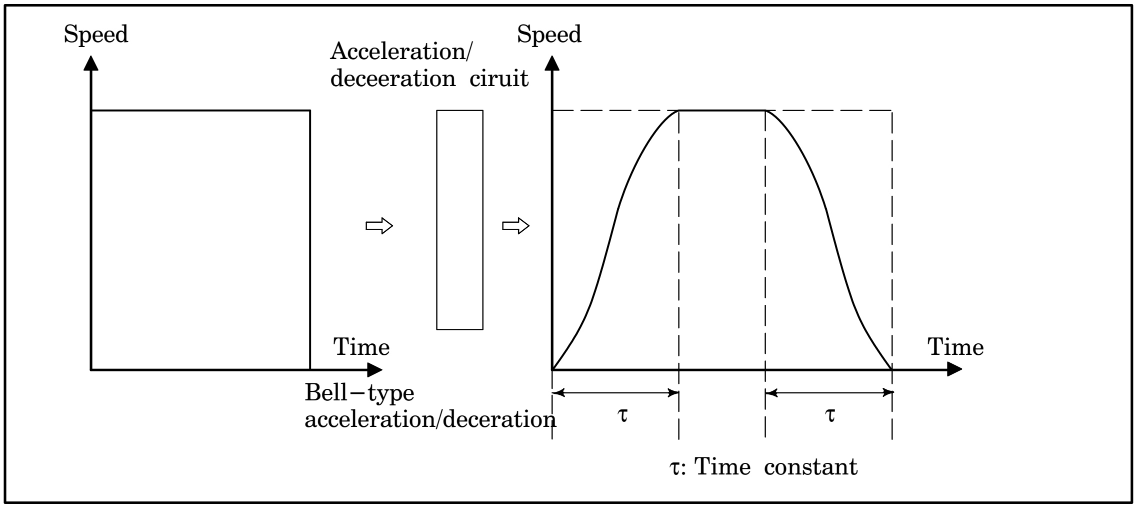
The time constants of cutting feed and jog feed along each axis must be specified in parameters 1622 and 1624 respectively. The parameters are the same as those used for exponential acceleration/deceleration. The FL speed for cutting feed (parameter 1623) and the FL speed for jog feed (parameter 1625) become invalid. The system assumes that the parameters are set to zero.

NOTE	By default, linear acceleration/deceleration is applied to cutting feed, jog feed, and dry run. When the JGEx bit of parameter No. 1600 is set to 1, exponential acceleration/deceleration is applied to jog feed. When the CTEx bit of parameter No. 1600 is set to 1, exponential acceleration/deceleration is applied to cutting feed and dry run.

(3) Bell–type acceleration/deceleration of cutting feed after interpolation

The function of bell–type acceleration/deceleration of cutting feed after interpolation enables smooth acceleration or deceleration and reduces impact on the machine. This function increases or decreases the feedrate as described below:

Rapid traverse : Linear acceleration/deceleration (constant acceleration)
Cutting feed : Bell–shaped acceleration/deceleration (constant acceleration period)
Jog feed : Bell–shaped acceleration/deceleration (constant acceleration period)
Dry run : Bell–shaped acceleration/deceleration (constant acceleration period)



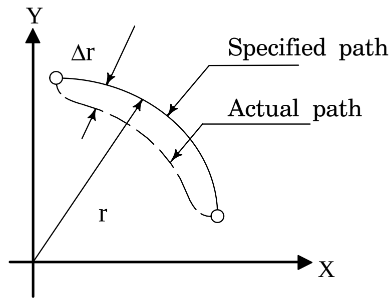
The time constant of cutting feed along each axis is specified in parameter 1622. The time constant of jog feed is specified in parameter 1624. Acceleration/deceleration is executed according to the specified time constants.

Parameter 1600 selects whether exponential or bell–type acceleration/deceleration is executed for each type of feed.

When the cutting feed is subjected to bell–type acceleration/deceleration, the FL speed of cutting feed (parameter 1623) and the FL speed of jog feed (parameter 1625) become invalid. (The system always assumes that the parameters are set to zero.)

(4) Radial error in tool path for each acceleration/deceleration type

In circular interpolation, especially when circular cutting is executed at high speed, the actual path of the accelerated or decelerated tool deviates from the specified arc in the direction of the radius.



Δr : Maximum radius error (mm)
v : Feedrate (mm/sec)
r : Radius of arc (mm)
T_1 : ccelration/deceleration time constant (sec)
T_2 : Time constant of servo loop (sec)

The maximum error in the radial direction can be approximated by the following expressions:

$$\Delta r = \left(\frac{1}{2} T_1^2 + \frac{1}{2} T_2^2 \right) \frac{v^2}{r}$$ .... Exponential acceleration/deceleration

$$\Delta r = \left(\frac{1}{24} T_1^2 + \frac{1}{2} T_2^2 \right) \frac{v^2}{r}$$ .... Linear acceleration/deceleration after interpolation

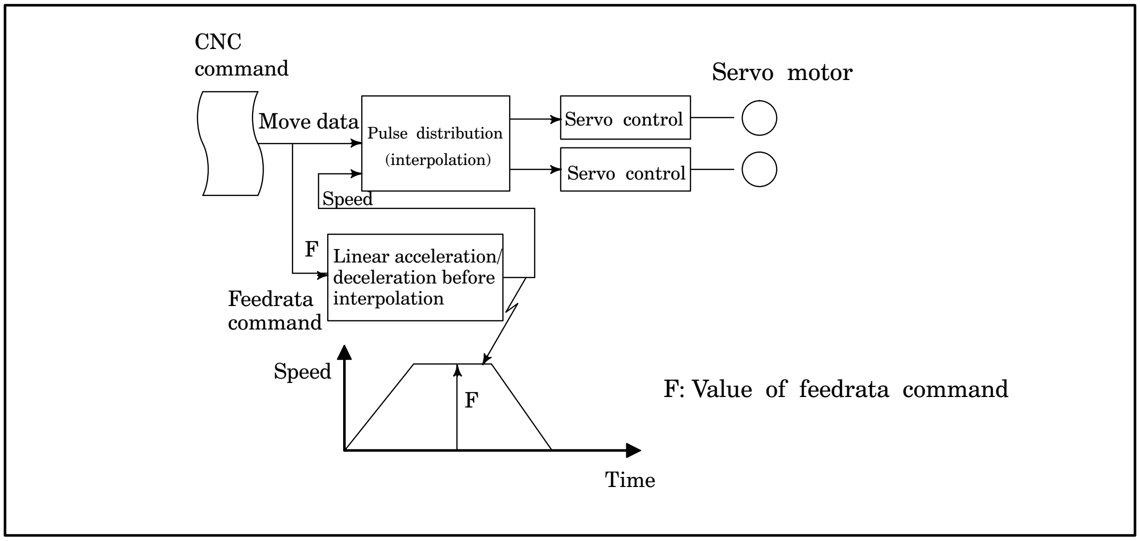
$$\Delta r = \left(\frac{1}{48} T_1^2 + \frac{1}{2} T_2^2 \right) \frac{v^2}{r}$$ .... For bell–shaped acceleration/deceleration after interpolation

The error for linear acceleration/deceleration is 1/12 of that for exponential acceleration/deceleration, while the error for bell–shaped acceleration/deceleration is 1/24 of that for exponential acceleration/deceleration, excluding the error incurred by the time constant for the servo loop.

6.4.2 Acceleration/deceleration befor interpolation

(1) Linear acceleration/deceleration of cutting feed before interpolation

A specified cutting feedrate can be linearly increased or decreased before interpolation. This function eliminates machining profile errors caused by the delay occurring in acceleration or deceleration. The time required for acceleration or deceleration by this function is significantly shorter than that by the function of exponential acceleration/deceleration.

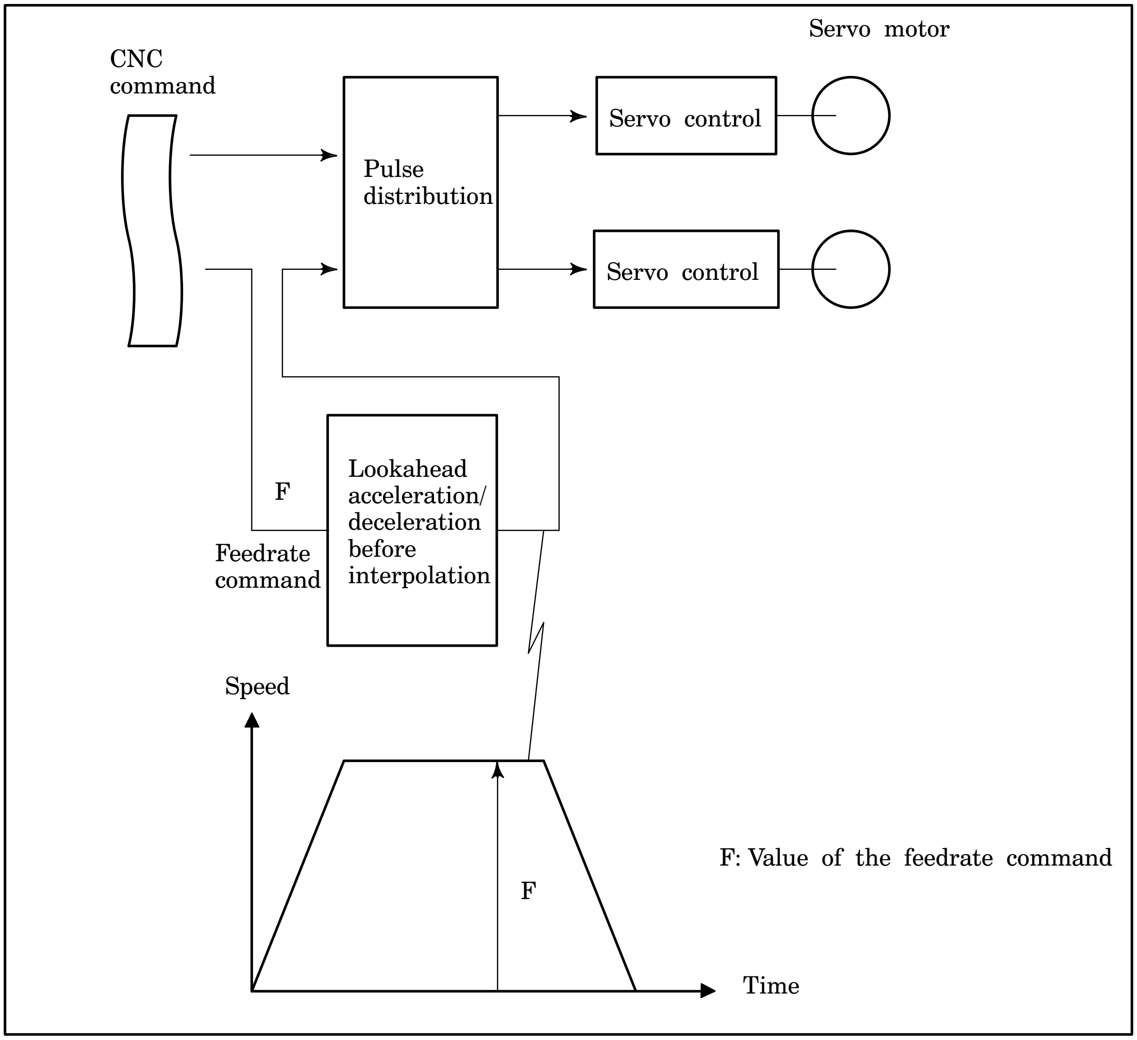


The function of linear acceleration/deceleration before interpolation increases or decreases the feedrate specified in the tangential direction.

(2) Lookahead acceleration/deceleration before interpolation

The function of acceleration/deceleration before interpolation increases or decreases the tangential velocity before interpolation (pulse distribution). It can eliminate profile errors caused by interpolation. In a block in which the direction of movement changes, a staircase variation occurs in the speed elements along the axes even if the tangential velocity is constant. Deceleration needs to be applied in a block in which the speed greatly varies. The function of lookahead acceleration/deceleration before interpolation reads blocks beforehand. The function checks whether the speed difference at a corner exceeds the specified limit when the tool is moved at a specified speed. If the estimated speed difference is larger than the specified limit, the function calculates the feedrate at a corner so that the limit is not exceeded. It gradually reduces the speed so that the calculated feedrate is attained at the end of the block. (See Example 1.) As acceleration/deceleration is executed at a constant acceleration, the time required for acceleration/deceleration is reduced.

If blocks having a shorter distance than required for deceleration are specified in succession, the target speed in a block can be reached by reducing the feedrate beforehand as required. The function of lookahead acceleration/deceleration before interpolation enables this. (See Example 2.)



(a) Specifying the function

To turn on and off the mode of lookahead acceleration/deceleration before interpolation, specify the following commands:

G05.1 Q1 ; Validates lookahead acceleration/deceleration before interpolation.

G05.1 Q0 ; Invalidates lookahead acceleration/deceleration before interpolation.

Parameter MBF (No. 2401, bit 6) can determine whether the mode of lookahead acceleration/deceleration before interpolation is set on or off at the time the power is turned on or in the reset state.

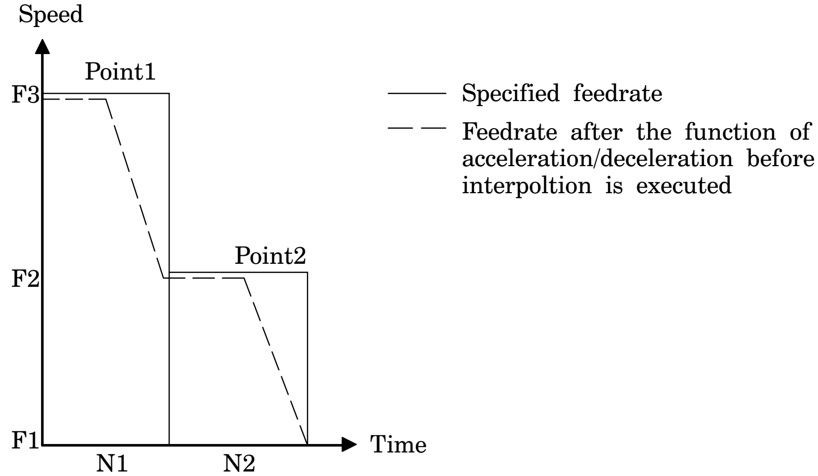
(b) Changing the feedrate command

When the feedrate command is changed, either of the following is executed:

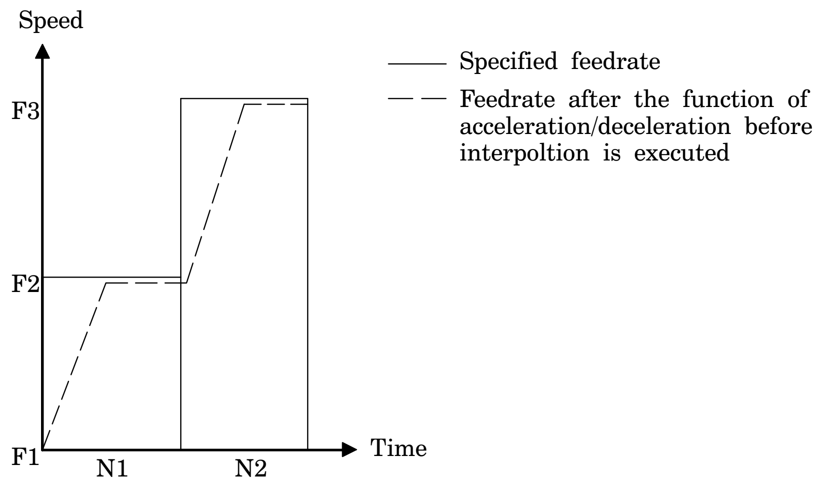
Deceleration is started in advance so that it is completed before the block in which the feedrate command is changed is executed.

Acceleration is started after the block in which the feedrate command is changed is started.

Sample deceleration



Sample acceleration



(c) Determining an allowable speed difference at a corner

The feedrate is reduced so that the speed difference at a corner does not exceed the specified limit. The smaller the limit, the smaller the impact at a corner. When a sufficiently small limit is specified in parameter No. 1478, acceleration or deceleration need not be executed after interpolation and the machining precision is improved. However, when the specified limit is small, the feedrate at a corner is reduced and the time required for the entire machining increases.

The allowable speed difference and the time constant of acceleration/deceleration after interpolation must be adjusted, depending on the desired machining period and precision.

Allowable speed difference (No. 1478)	Small	↔	Large
Time constant after interpolation (No. 1622)	Can be reduced	↔	Must be enlarged
Error in machining	Small	↔	Large
Machining period	Long	↔	Short

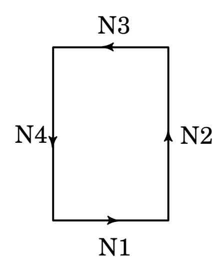
(d) Determining the time constant of acceleration/deceleration before interpolation

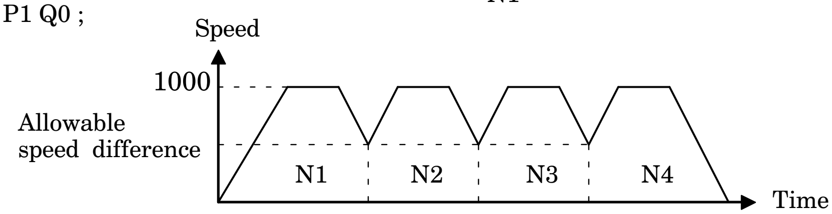
In theory, the time constant of acceleration/deceleration before interpolation does not change a machining error. If the time constant is too small, the machine cannot follow it, which causes a large machining error. If the specified time constant is too large, the period of machining is increased accordingly. The smallest time constant that the machine can follow is best.

Execute a simple movement such as reciprocation along an axis and check whether the impact on the machine is too great and whether the machine overshoots VCMD. Then determine the time constant.

Example 1

```
O0001 ;

G05.1 Q1 ;
N1 G01 G91 X10. F1000 ;
N2 Y10. ;
N3 X–10. ;

N4 Y–10. ;
G05.1 P1 Q0 ;
M30 ;
```

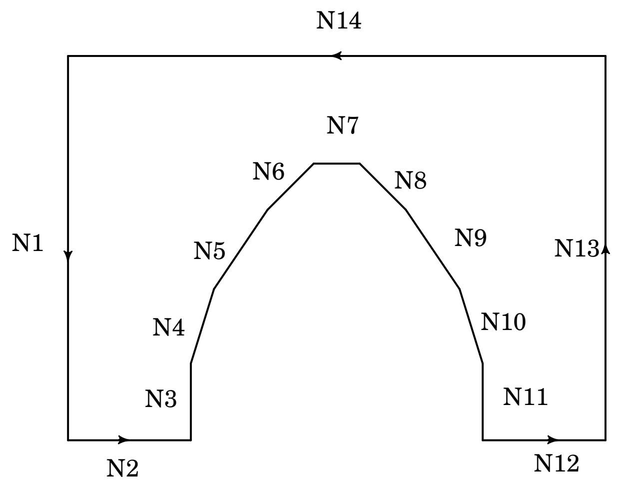


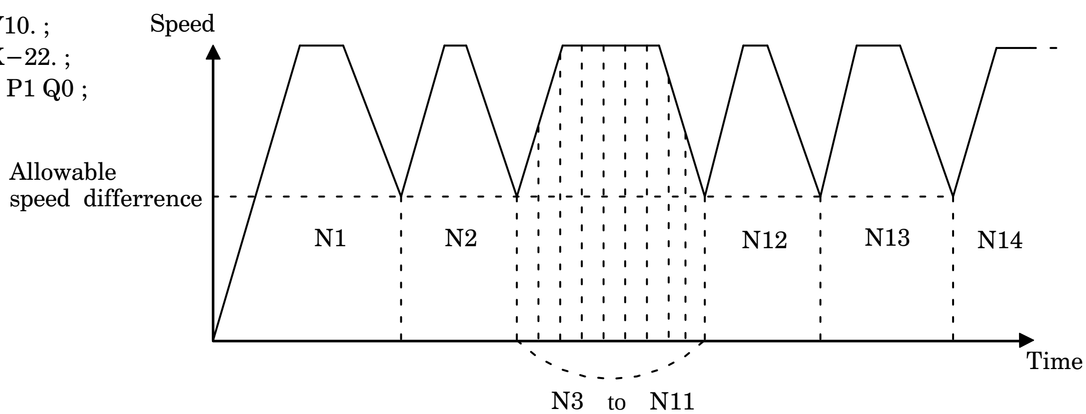


NOTE	Note that the remaining travel distance is not displayed on the screen for movement with an override of 0% in the look–ahead acceleration/deceleration before interpolation.

Example 2

```
O0002 ;
G05.1 Q1 ;

N01 G01 G91 Y–10. F1000 ;
N02 X5. ;
N03 Y2. ;

N04 X1. Y2. ;

N05 X2. Y2. ;
N06 X2. Y1. ;
N07 X2. ;
N08 X2. Y–1. ;

N09 X2. Y–2. ;
N10 X1. Y–2. ;

N11 Y–2. ;
N12 X5. ;

N13 Y10. ;
N14 X–22. ;
G05.1 P1 Q0 ;
M30 ;
```





NOTE 1 Do not attempt the following operations while the function is being executed. Otherwise, the block may terminate before deceleration is completed. Alternatively, only the deceleration is completed and the movement specified in the block is not completed.

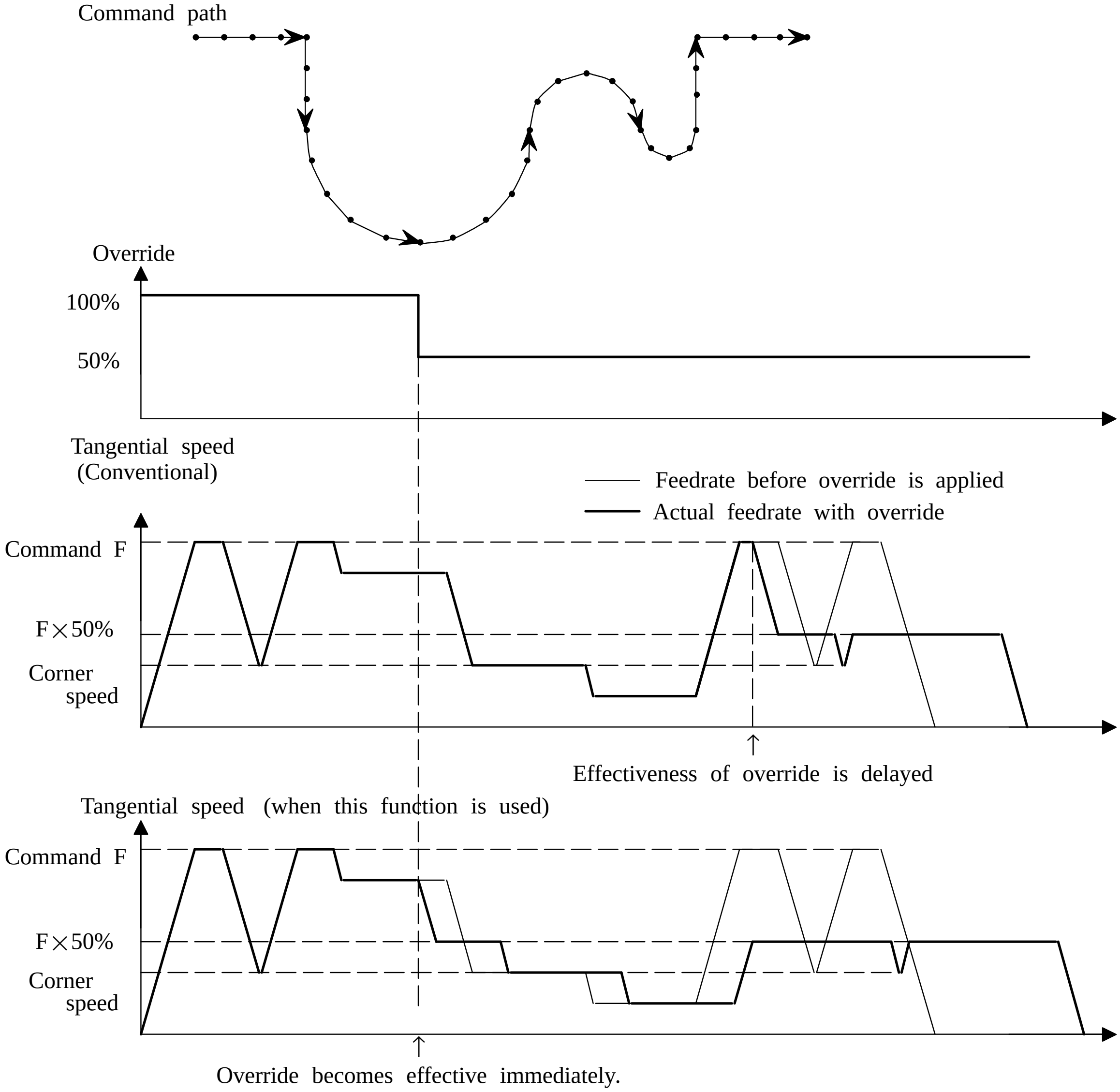
(1) Turning on or off the dry run

(2) Starting external deceleration

(3) Starting the operation in a feed mode whose feedrate is not provided by a CNC command program

NOTE 2 In the mode of lookahead acceleration/deceleration before interpolation, an F command with one digit cannot be specified.

NOTE 3 A specified override is taken in and calculated when the block is read in advance. If the override is changed in the mode of lookahead acceleration/deceleration before interpolation, the new override is not immediately validated. It is validated after the remaining 15 blocks are processed.
The override or dry run signal immediately becomes effective if bit 5 of parameter No. 1403 is set to 1. Setting this parameter can also enable inverse–time feed.



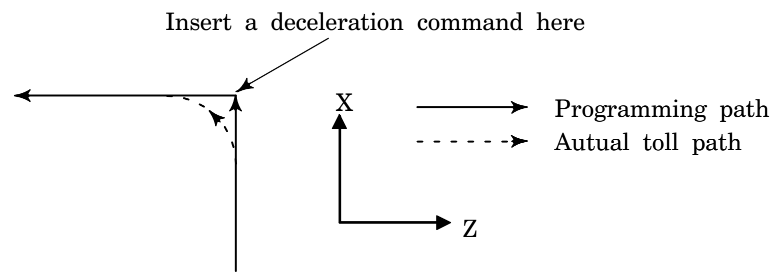
NOTE 4 In the mode of lookahead acceleration/deceleration before interpolation, the following commands must not be specified:

G02.3, G03.3	:	Exponential interpolation
G06.1	:	Spline interpolation
G07	:	Hypothetical axis interpolation
G07.1	:	Cylindrical interpolation
G12.1, G13.1	:	Polar coordinate interpolation

NOTE 5 Parameters 1630 and 1631 determine the acceleration. Parameter 1478 determines the allowable speed difference at a corner.

6.5 Speed Control at Corners of Blocks

For linear acceleration/deceleration after interpolation, the acceleration or deceleration is applied in feed start and feed stop, with a time constant automatically set by a parameter so that the machine tool system is not jarred. Therefore, this need not be considered when programming.
Because of automatic acceleration and deceleration, corners are not cut sharply. In this case, the blocks (G09 or G61) of the deceleration command must be commanded at the corner to cut sharp.
For example, if the tool moves along the X axis only in one block and along the Y axis in the next block, the feed rate for the X axis decelerates while motion along the Y axis accelerates and the actual tool path is as follows.



If the deceleration command G09 is inserted, the actual tool path matches the programmed path. The faster the feed rate and the larger the acceleration/deceleration time constant, the larger the error at the corner.
In circular interpolation, the actual arc radius is smaller than that of the programmed arc. (See the Appendix. 3) This error can be minimized by making the acceleration/deceleration time constant of feed rate small.
In this case, a deceleration command, G09 or G61, must be specified at the corner, in the block.

NOTE The following chart shows feed rate changes between blocks of information specifying different types of movement.

New block \ Previous block	Positioning	Feed	Not moving
Positioning	×	×	×
Feed	×	○	×
Not moving	×	×	×

× : The next block is executed after commanded rate has decelerated to zero.
○: The next block is executed sequentially so that the feed rate is not changed by very much. However in the block G09 is specified or G61 mode, the next block is not executed until the feed rate is decelerated and reached to zero.

6.5.1 Exact stop (G09)

Move command in blocks commanded with the command (G09) decelerates, and in–position check (Note 1) is performed. This command (G09) is not necessary for deceleration at the end point for positioning and in–position check is also done automatically. This function is used when sharp edges are required for workpiece corners in cutting feed.
Format G09 IP;

NOTE The position check refers to the following. Bit 0 (CIP) of parameter 1000 is used to select one of the position check methods:
(1) Checking that the specified speed is decelerated to 0 (the delay in acceleration / deceleration is eliminated).
(2) Checking that the specified speed is decelerated to 0 and that the machine reaches the specified position (the error falls within the specified range).

6.5.2 Exact stop mode (G61)

When G61 is commanded, deceleration is applied to the end point of cutting block and in–position check is performed every block thereafter. This G61 is valid till G64 (cutting mode) or G62 (automatic corner override), is commanded.

6.5.3 Cutting mode (G64)

When G64 is commanded, deceleration at the end point of each block is not performed thereafter, and cutting goes on to the next block. This command is valid till G61 (exact stop mode) or G62 (automatic corner override), is commanded.

6.5.4 Tapping mode (G63)

When G63 is commanded, feed rate override is ignored (always regarded as 100%), and feed hold also becomes invalid. Cutting feed does not decelerate at the end of block to transfer to the next block. This command is valid till G61 (exact stop mode), G62 (automatic corner override), or G64 (cutting mode) is commanded.

6.5.5 Automatic corner override (G62)

When G62 is commanded during cutter compensation, cutting feed rate is automatically overridden at corner. The cutting quantity per unit time of the corner is thus controlled so as not to increase. This command is valid till G61 (exact stop mode) or G64 (cutting mode) is commanded.
If cutting is made at a programmed feedrate at inner circular region during the execution of the cutter compensation, the cutter may be overloaded to rough cut surfaces. This function automatically decelerates the tool movement to lighten the cutter load at these region so as to obtain clean cut surfaces.

(1) Automatic override at inner corners

(a) Operating conditions
The feedrate is automatically overridden when all the following conditions are satisfied in both blocks before and after a corner.

(i) G code of group 01 is G01, G02, or G03.

(ii) The offset quantity is not 0 in the offset mode.

(iii) The offset is made inside at a corner to be machined.

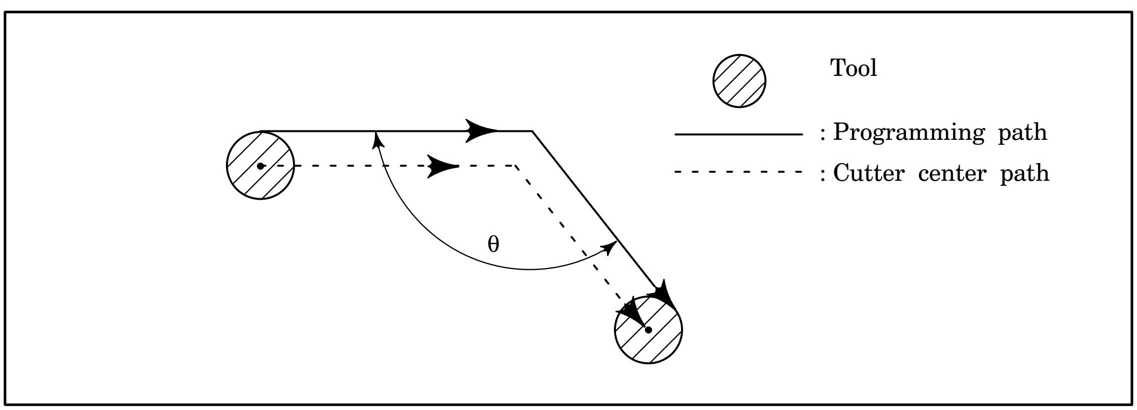
(iv) An axis moves along the offset plane.

(v) Neither G41 nor G42 command is included in the subsequent block.

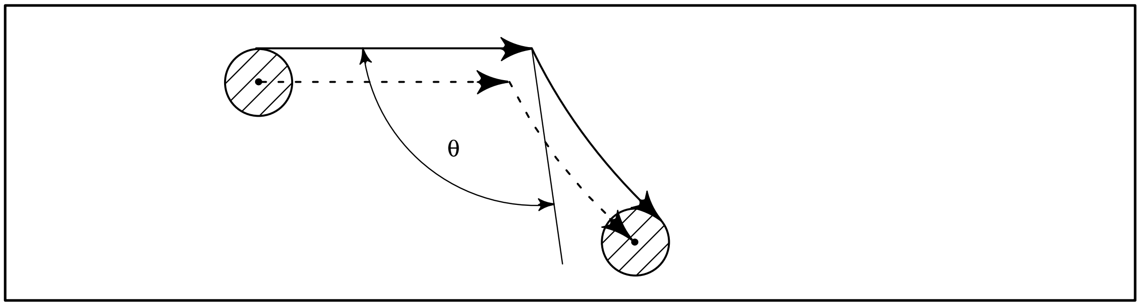
(vi) Neither G41 nor G42 command is included in the previous block, or the block is not started up, if either command is included.

(vii) An inner corner is smaller than θpreset by a parameter.
The angle is judged about the programmed path.

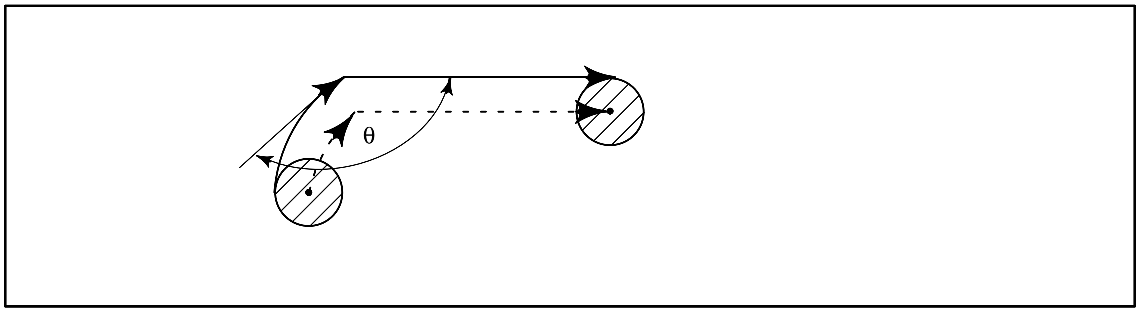
(i) Straight line – straight line



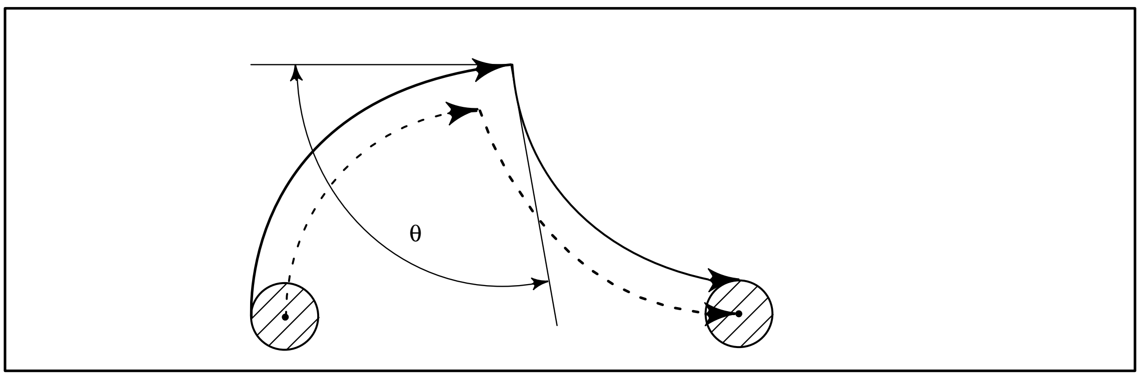
(ii) Straight line – arc



(iii) Arc – straight line



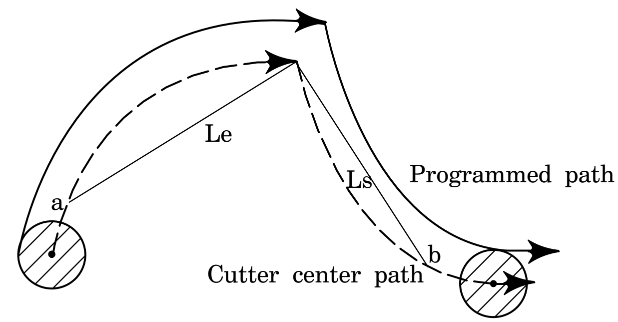
(iv) Arc – arc



When θ≦θp, the corner is regarded as inside.
θp value (2 ≦ θp ≦ 178, unit :degree) is set by parameter (No. 6611).
If θ is almost equal to θp, the decision may include an error within 0.001°.

(b) Motion area

When a corner is judged as inside, the feedrate is overridden from the distance range within Le of the block on this side from the intersection of the corner to the distance range within Ls of the next block from the intersection of the corner. Distances Ls and Le are linear distances from a point on the cutter center path to the intersection of the corner. Le and Ls are set by parameters (No. 6613 and 6614), respectively.



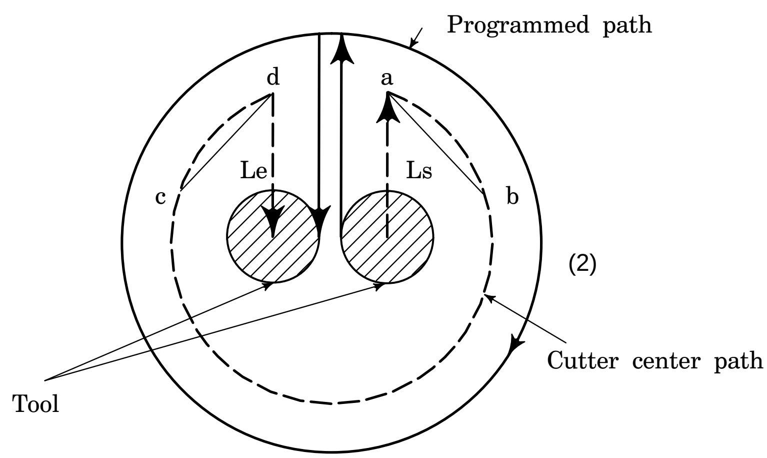
In case of an arc, this override function to the end point of a block is effective when the following conditions are satisfied.

(1) The distance is within Le.

The override function to the end point of a block is effective when the following conditions are satisfied.

(2) The distance is within Ls.

Example In case of disc



Regarding program (2) of an arc, the feedrate is overridden from point a to point b and from point c to point d.

NOTE Ls and Le are the distances between point a and b and between c and d along the straight line.

(c) Override quantity

The override quantity is set by parameter 6612.

1 ≦ Override quantity (per every 1%) ≦ 100(%)

It is also effective for the dry run and F1 digit command.

Actual feedrate becomes

F × (internal corner override) × (feedrate override)

(d) Whether internal corner override is effective or not

Whether the internal corner is overridden or not is selectable by G codes. G62 is added to G61 and G64 in group 15 as shown in the following table. These G codes are also related to whether the exact stop check mode is effective or not.

	Exact stop check mode	Internal corner override
G61	Effective	Ineffective
G62	Ineffective	Effective
G64	Ineffective	Ineffective

NOTE 1 G64 functions when power is turned on or under the clear condition.

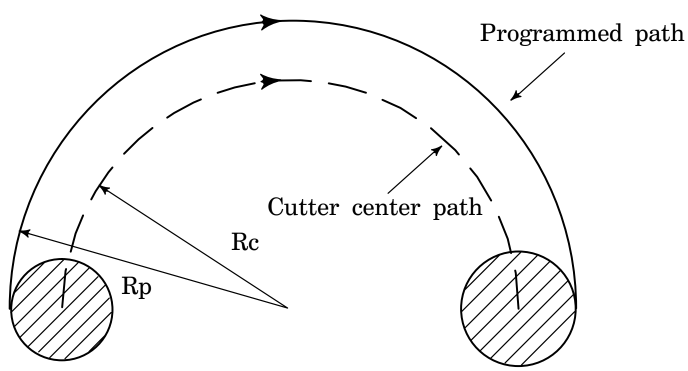
NOTE 2 Specify G09, if you want to check the exact stop in G62 mode.

NOTE 3 The internal circular cutting feedrate change described in (2) is always effective without being affected by these G codes.

(2) Internal circular cutting feedrate change

In case of the internal offset circular cutting, the feedrate in the programmed path is set to the specified F by setting actual feedrate to F $\times$ Rc/Rp (where, Rc: cutter center path radius Rp: programmed radius) with reference to the specified feedrate (F).
This change is also effective for dry run and F1 digit command.

Example 1



However, if Rc is very small as compared with Rp, $\frac{Rc}{Rp} \fallingdotseq 0$, causing the cutter to stop. Accordingly, the minimum reduction ratio (MDR) is set to make the actual feedrate F $\times$ (MDR) when $\frac{Rc}{Rp} \leqq$ MDR.

MDR is set by parameter No. 6610.
$1 \leqq$ MDR (1% step) $\leqq 100$
This is also applicable to F1 digit and dry run.
The reduction ratio of the automatic override at internal corners is not affected by MDR.

NOTE If the automatic override at internal corner overlaps the internal circular cutting, actual feedrate becomes as follows.

$$F \times \frac{Rc}{Rp} \times \text{(override at corner)} \times \text{(feedrate override)}$$

6.6 Velocity Control Command

6.6.1 Automatic velocity control during involute interpolation

(1) Outline

To enhance the machining precision, the function for automatic velocity control during involute interpolation automatically overrides the specified feedrate as follows:

– Override applied when an internal offset is applied for cutter compensation

– Override applied near the basic circle

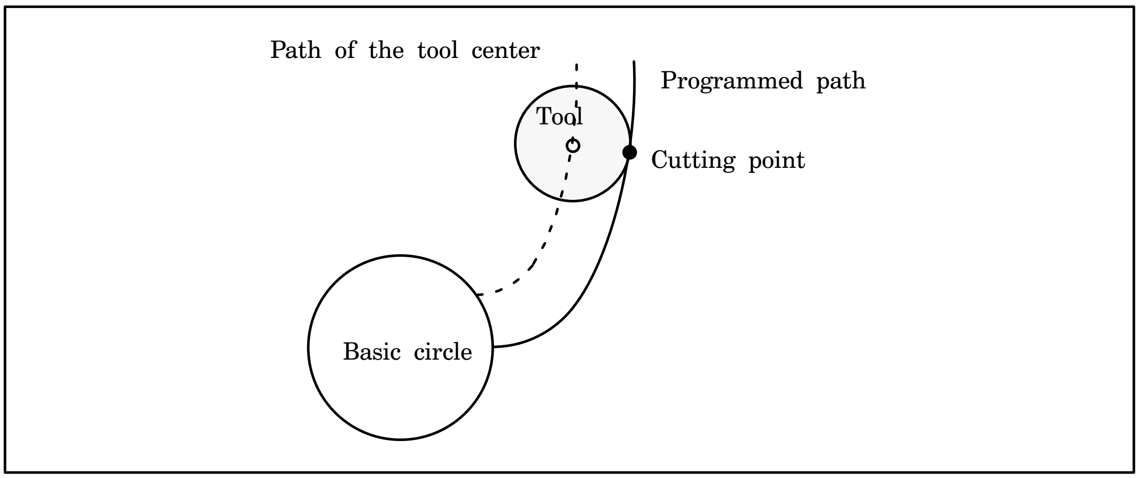
(a) Override applied when an internal offset is applied for cutter compensation

When cutter compensation is applied to involute interpolation, the tangential velocity along the path of the tool center is continuously limited to the specified velocity during usual involute interpolation.

In this case, the actual cutting speed at the tool periphery (the cutting point) on the programmed path, changes because the curvature of the involute curve sometimes changes.

In particular when the tool is offset inside the involute curve, the closer to the basic circle the tool is, the greater than the specified feedrate the actual cutting speed is.

To ensure smooth machining, it is desirable that the actual cutting speed is limited to the specified feedrate. This function calculates the override value according to the curvature of the involute curve, which is changeable. The function continually limits the actual cutting speed, which is the tangential velocity at the cutting point, to the specified feedrate. This function is particularly effective during involute interpolation with internal offset.



(b) Override applied near the basic circle

If the tool cuts a workpiece at the programmed feedrate in the proximity of the basic circle where the change in the curvature of the involute curve relatively severe, the surface of the workpiece may be rough because of the excusive load applied to the cutter.

This function automatically decelerates the tool in the proximity of the basic circle where the change in the curvature of the involute curve is relatively severe according to the setting of the parameter. In this way, the function reduces the load on the cutter, enabling the tool to cut the surface of the workpiece finely.

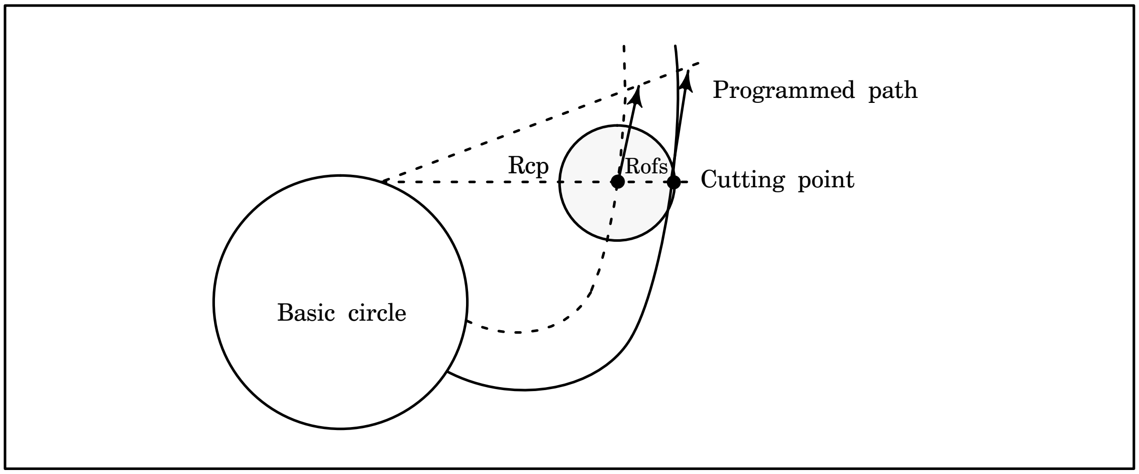
(2) Override applied during internal offset for cutter compensation (OVRa)

When cutter compensation is applied inside the involute curve, an override (OVRa) is automatically applied to the specified feedrate so that the velocity not at the center of the tool, but at the cutting point conforms to the programmed velocity. The tangential velocity at the actual cutting point is therefore made constant. See the figure below.

$$\text{OVRa} = \frac{\text{Rcp}}{\text{Rcp} + \text{Rofs}} \times 100$$

Rcp : Radius of the involute curve at the center of the tool (The involute curve passes through the center of the tool.)

Rofs : Radius of the tool



(3) Override applied near the basic circle (OVRb)

An override can be applied to the specified feedrate in the proximity of the basic circle where the change in the curvature of the involute curve is extreme. The override is determined by the radius of curvature. When the radius of curvature at the cutting point reaches the value in the range specified in parameters Rlmt1 to Rlmt5, an override (OVRb) is applied to the specified feedrate as follows. See the graph below.

When Rlmt1 > Rcp ± Rofs ≧ Rlmt2

$$\text{OVRb} = \frac{100 - \text{OVR2}}{\text{Rlmt} - \text{Rlmt2}} \times (\text{Rcp} \pm \text{Rofs} - \text{Rlmt2}) + \text{OVR2}$$

When Rlmt2 > Rcp ± Rofs ≧ Rlmt3

$$\text{OVRb} = \frac{\text{OVR2} - \text{OVR3}}{\text{Rlmt2} - \text{Rlmt3}} \times (\text{Rcp} \pm \text{Rofs} - \text{Rlmt3}) + \text{OVR3}$$

When Rlmt3 > Rcp ± Rofs ≧ Rlmt4

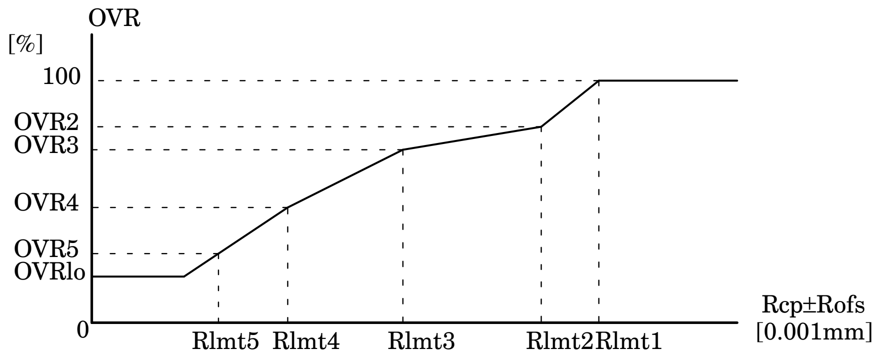
$$\text{OVRb} = \frac{\text{OVR3} - \text{OVR4}}{\text{Rlmt3} - \text{Rlmt4}} \times (\text{Rcp} \pm \text{Rofs} - \text{Rlmt4}) + \text{OVR4}$$

When Rlmt4 > Rcp ± Rofs ≧ Rlmt5

$$\text{OVRb} = \frac{\text{OVR4} - \text{OVR5}}{\text{Rlmt4} - \text{Rlmt5}} \times (\text{Rcp} \pm \text{Rofs} - \text{Rlmt5}) + \text{OVR5}$$

Rlmt1 to Rlmt5 and OVR2 to OVR5 are specified by parameters 6620 to 6624 and parameters 6631 to 6634, respectively.

Rcp ± Rofs means that Rcp + Rofs is used for the internal offset and that Rcp – Rofs is used for the external offset



(4) Lower override limit (OVR1o)

When an override is applied to the specified feedrate either when an internal offset is applied for cutter compensation or when near the basic circle, the velocity at the center of the tool may be 0 in the proximity of the basic circle. To prevent this, the lower override limit (OVR1o) must be specified using its parameter.

When the override lower than the setting of parameter 6630 (OVR1o) is applied, the velocity is clamped at the velocity to which override (OVR1o) is applied.

When OVRa < OVR1o , OVRb < OVR1o , or OVRa*OVRb < OVR1o , let the specified feedrate be F and the actual feedrate in tangent direction be F'. Then, F' = F*OVR1O.

6.6.2 Automatic velocity control during polar coordinate interpolation

When the velocity for the rotation axis exceeds the maximum cutting feedrate specified with parameter 1422 in the polar coordinate interpolation mode, alarm 0T512 (excessive velocity alarm) occurs. However, this function automatically controls the velocity so that the excessive velocity alarm does not occur.

The following methods are used to control the velocity:

(1) Automatic override

When the velocity for the rotation axis exceeds the permissible velocity(*1), an override is automatically applied to the cutting feedrate (*2).

NOTE 1 The permissible velocity is obtained by multiplying the maximum cutting feedrate for the rotation axis by the permissible rate (parameter 1056).

NOTE 2 The override is obtained as follows:
Permissible velocity –: velocity for the rotation axis x 100 (%)

(2) Automatic velocity clamp

When the velocity for the rotation axis exceeds the maximum cutting feedrate even after the method in (1) is used, the velocity is clamped so that it does not exceed the maximum cutting feedrate.

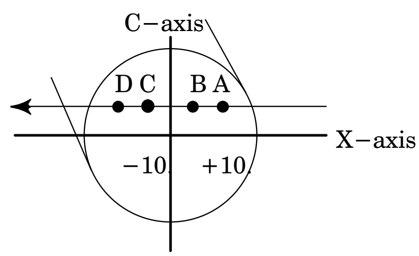
This function is used only when the center of the tool is very close to that of the rotation axis.

WARNING 1 The machine lock or interlock function sometimes does not work as soon as the corresponding switch is turned on while the automatic clamp function is being executed.

WARNING 2 If the feed hold switch is turned on while the automatic clamp function is being executed, the feed hold is accepted, and the SPL signal (automatic operation hold) is output. In this case, however, the tool sometimes does not stop immediately.

[Example]

```
G90 G00 X10.0 C0. :
G12.1 ;
G01 C0.1 F1000 ;
X–10.0 :
G13.1 ;
```



For example, the above program is executed after the maximum cutting feedrate for the rotation axis (parameter 1422) is specified as 360 (3600 deg/min), and the permissible ratio of automatic override during polar coordinate interpolation (parameter 1056) is specified as 0 (90%). In this case, the automatic override starts at point A (X = 2.273), and automatically clamping the velocity starts at point B (X = 0.524). The minimum value of the automatic override is 3%. The velocity remains clamped up to point C (X = –0.524), and the automatic override is applied up to point D (X = –2.273).

NOTE The above coordinates are Cartesian coordinate.

6.6.3 Cutting point speed control function

The cutting point speed control function is used when circular interpolation is performed in the cutter compensation C mode. This function allows a programmed feedrate to be used as the feedrate at the cutting point rather than the feedrate at the center of the tool.

This function is enabled or disabled by setting the CAFC bit of parameter 1402.

The following explanation takes into consideration the actual feedrate of the center of the tool and the tool–tip center when an offset is applied to the outside or inside of an arc while this function is being used.

(1) When an offset is applied to the outside of an arc

When an offset is applied to the outside of an arc while this function is being used, the relationship between the programmed feedrate F and the actual feedrate ACT–F of the tool center or the tool–tip center is as illustrated in Fig. 6.6.3(a).

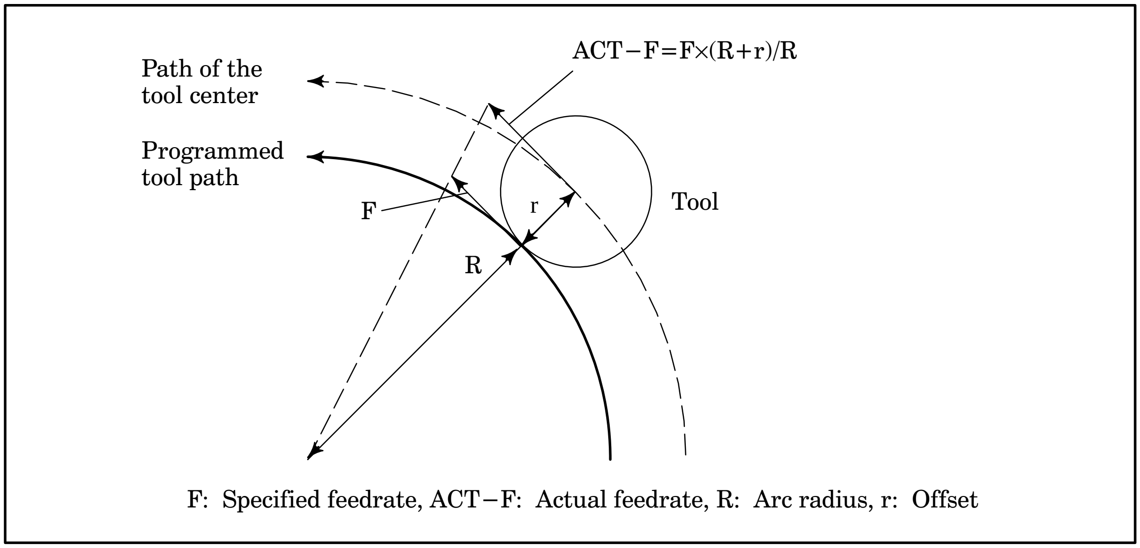


Fig. 6.6.3(a) Applying an Offset to the Outside of an Arc

(2) When an offset is applied to the inside of an arc

When an offset is applied to the inside of an arc while this function is being used, the relationship between the programmed feedrate F and the actual feedrate ACT–F of the tool center or the tool–tip center is as illustrated in Fig. 6.6.3(b).

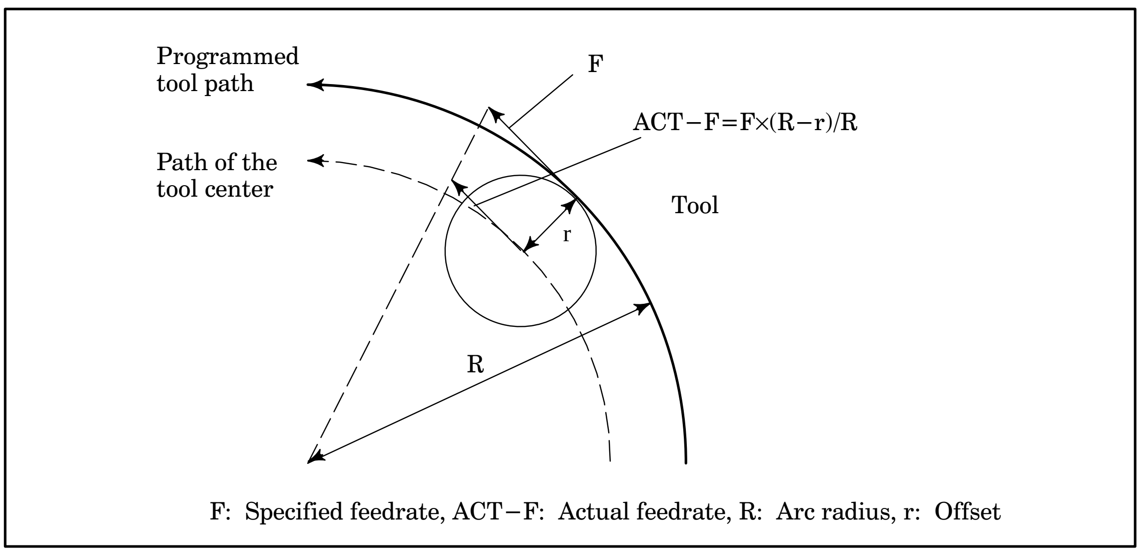


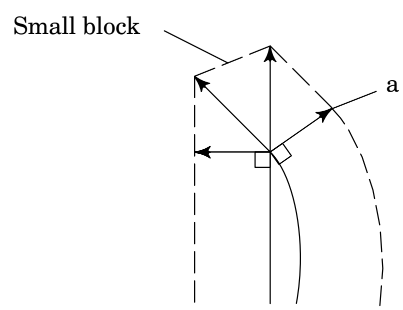
Fig. 6.6.3(b) Applying an Offset to the Inside of an Arc

NOTE 1 If the actual feedrate ACT–F reaches the maximum cutting feedrate (set in parameter 1422) when an offset is applied to the outside by using this function, the actual feedrate is clamped to the maximum cutting feedrate.

NOTE 2 If the arc radius R and offset r are very close to each other when an offset is applied to the inside of an arc using this function, actual feedrate ACT–F can approximate to 0. To prevent this, specify a minimum deceleration ratio with the MDR bit of parameter 6610.

If (R – r)/R is smaller than or equal to MDR/100, the actual feedrate ACT–F is obtained from F x MDR/100.

NOTE 3 When more than two tool center offset vectors are created at the end of a block as shown below, a small block is created between these vectors. As a result, cutting point speed control is performed from point a.



6.7 Dwell (G04)

The G04 command delays the transfer of control to the next block. The G04 command used in conjunction with the G94 command (mode of feed per minute) delays the transfer of control to the next block by the specified time.

The G04 command used in conjunction with the G95 command (mode of feed per rotation) delays the transfer of control to the next block until the spindle rotates by the specified amount.

The time can always be specified by the dwell command regardless of the G94 and G95 modes according to the setting of the parameter.

(DWL of parameter 2400)

Format

Dwell with seconds specified

$$\text{G94G04} \left\{ \begin{array}{c} \text{P ––} \\ \\ \text{X ––} \end{array} \right\} ;$$

P–– or X––: Specifies the dwell time in seconds (0.001 to 9999.999 s)

Dwell with the speed specified

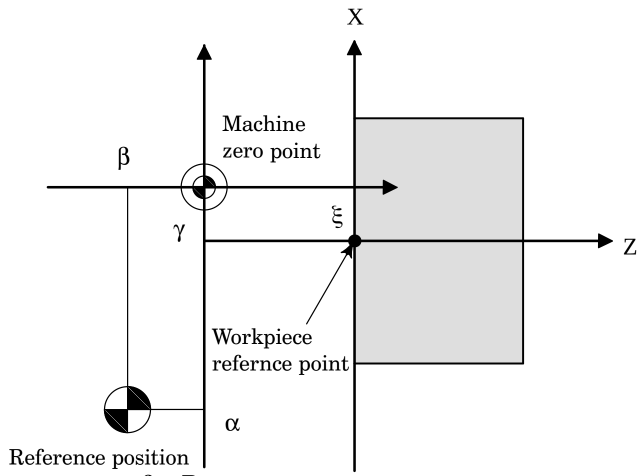
$$\text{G95G04} \left\{ \begin{array}{c} \text{P ––} \\ \\ \text{X ––} \end{array} \right\} ;$$

P–– or X––: Specifies the spindle rotation angle in rev (0.001 to 99999.999 rev)

7. REFERENCE POSITION

The reference position is a fixed machine position to which the tool can easily be moved.

The positional relationship between the reference position and the machine zero point is specified using parameters 1240 to 1243. Because the positional relationship between the machine zero point and the workpiece coordinate system (G54 to G59) is specified using parameters 1221 to 1226, the machine coordinate system and the workpiece coordinate system can automatically be set by returning to the reference position after the power is turned on.



α, β : Parameters
γ, ξ : Parameters
(offset from the workpiece reference point)

7.1 Automatic Reference Position Return (G28 and G29)

7.1.1 Automatic return to reference position (G28)

G28 IP____;

This command specifies automatic return to the reference position for the specified axes. IP___ is an intermediate coordinate and is commanded by absolute or incremental value. The intermediate coordinate specified in this block are stored in memory.

The G28 block operations

1 The commanded axis is positioned to the intermediate point (from point A to point B).

2 The axis is positioned from the intermediate point to the reference point (from point B to point R).

3 When the machine lock has not been set, the Reference Return lamp goes on. Positioning to the inter mediate or reference points are performed at the rapid traverse rate of each axis.

In general, this command is used for automatic tool changing (ATC). Therefore, for safety, the cutter compensation, and tool length compensation should be cancelled before executing this command.

NOTE 1 When the G28 command is specified when manual return to the reference position has not been performed after the power has been turned on, the movement from the intermediate point is the same as in manual return to the reference position. In this case, the direction from the intermediate point is equal to that for return to reference position selected by parameter ZMI (No. 1006 #5).

NOTE 2 The coordinate value specified in G28 command is memorized for all axes as is the coordinate value of the intermediate point. In other words, for axes not specified in the G28 block, the coordinate value memorized for intermediate point earlier is taken as the coordinate value of the intermediate point for the axis.

Example

N1X10.0Z20.0;
N2G28X40.0; Intermediate point (X40.0,)
N3G28Z60.0; Intermediate point (X40.0, Y60.0)

7.1.2 Automatic return from reference position (G29)

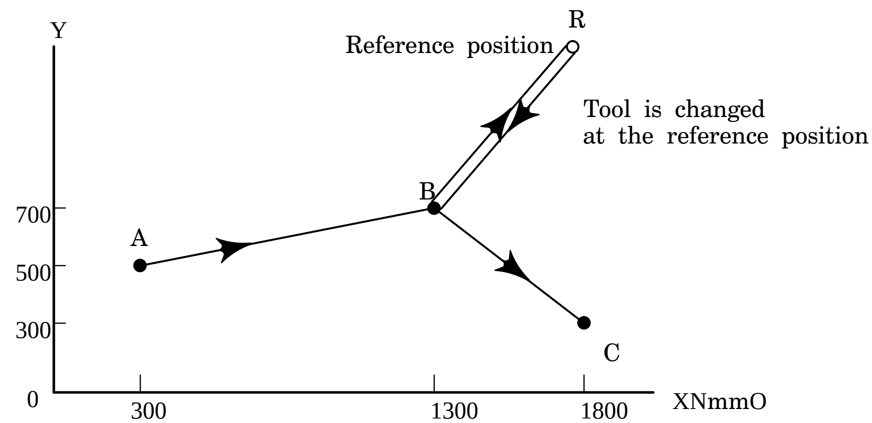
G29 IP ____;

This command positions the tool at the commanded position via the intermediate point of a commanded axis. In general, it is commanded immediately following the G28 command or G30. (2nd return to reference position) For incremental programming, the command value specifies the incremental value from the intermediate point.

The G29 block operations (See Fig. 7.1.1)

1 The commanded axis is positioned to the intermediate point defined by G28 or G30 (from point R to point B).

2 The axis is positioned from the intermediate point to the commanded point (from point B to point C). Positioning to the intermediate or reference points are performed at the rapid traverse rate of each axis.



G28X1300.0Y700.0 (Program A to B)
T1111; (Tool is changed at the reference point)
G29X1800.0Y300.0 (Program B to C)

As shown in the example, the programmer is not required to calculate the actual move distance from the intermediate point to the reference position.

NOTE When the workpiece coordinate system is changed after the tool reaches the reference position through the intermediate point by the G28 or G30 command, the intermediate point also shifts to a new coordinate system. If G29 is then commanded, the tool moves to the commanded position through the intermediate point which has been shifted to the new coordinate system.

7.2 Reference Position Return Check (G27)

A reference position is fixed on the machine tool and the cutter can be moved to that point by the reference position return function.
The G27 reference position return check function checks whether or not the program returns to the reference position normally, as programmed.

G27 IP ____;

This command positions the tool at rapid traverse rate. If the tool reaches the reference position, the reference position return lamp lights up and the next block is executed. However, if the position reached by the tool is not the reference position, an alarm is displayed.

NOTE 1 In an offset mode, the position to be reached by the tool with the G27 command is the position obtained by adding the offset value. Therefore, if the position with the offset value added is not the reference position, the lamp does not light up, but an alarm is displayed instead. Usually, cancel offsets before G27 is commanded.

NOTE 2 When the machine tool system is an inch system with metric input, the reference position return lamp may also light up even if the programmed position is shifted from the reference position by 1μ. This is because the least input increment of the machine tool system is smaller than its least command increment.

NOTE 3 If G27 is commanded with machine lock on, the reference position return check is not done.

7.3 2nd, 3rd, 4th Reference Position Return (G30)

$$\mathbf{G30}\quad \left\{ \begin{array}{c} \mathbf{P2} \\ \mathbf{P3} \\ \mathbf{P4} \end{array} \right\}\quad \mathbf{IP\_\,;}$$

P2, P3, P4:
Command for selecting of the 2nd, 3rd or 4th reference position.
If omitted, the 2nd reference position is selected.
This command returns the specified axis to the 2nd, 3rd or 4th reference position automatically through the specified position. Each reference position return LED lights after completion of the return.
The positions of the 2nd, 3rd or 4th reference position are determined by presetting the distance from the reference position as a parameter (Data Nos. 1241, 1242, 1243) during field adjustment. The 2nd, 3rd, or 4th reference position return function can be used after the first reference position return has been completed. This function is the same as G28 reference position return, except that this function returns the tool to the 2nd, 3rd or 4th reference position instead of the primary reference position. If G29 is commanded immediately following G30, the tool is positioned to the point commanded by G29 through the intermediate point commanded by G30. This motion is the same as that by the G29 command following the G28 command.The G30 command is generally used when the automatic tool changer (ATC) position differs from the reference position.

WARNING Like the G28 command, cancel the cutter compensation, and tool length offset before the G30 command is executed.

NOTE At least one manual reference position return or automatic reference position return must be performed by the G28 command after the power supply is turned on, prior to issuing G30 command.

7.4 Floating Reference Position Return (G30.1)

G30.1 IP___ ;

This command enables return to a floating reference position.

A floating reference Position is a position on a machine tool, and serves as a reference position for machine tool operation.

A floating reference position need not always be fixed, but can be moved as required.

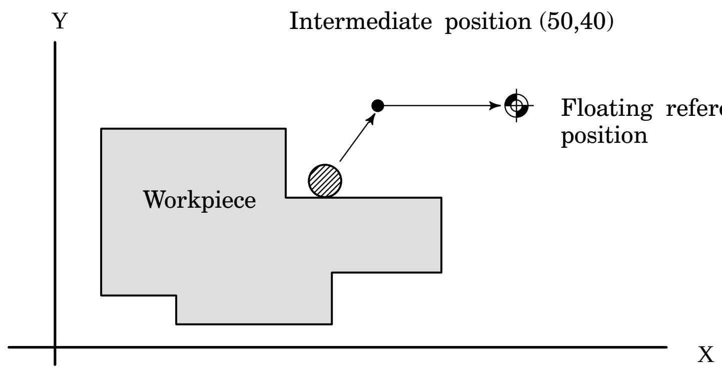
Generally speaking, on a machining center or milling machine, cutting tools can be replaced only at specific positions. A position where tools can be replaced is defined as the second or third reference position. Using G30 can easily move the cutting tools back to these positions. On some machine tools, the cutting tools can be replaced at any position unless they do not interfere with the workpiece.

With these machines, the cutting tools should be replaced at a position as close to the workpiece as possible so as to minimize the machine cycle time. For this purpose, the tool replacement position is to be changed, depending on the figure of the workpiece. This operation can easily be performed using this function. That is, a tool replacement position suitable for the workpiece is memorized as a floating reference Position. Then command G30. 1 can easily cause return to the tool replacement position.

IP___ specifies movement to an intermediate position for a floating reference position with an absolute or incremental command. A floating reference position becomes a machine coordinate position memorized by pressing the soft key [MEM–FRP] on the current positions display screen.

The operation in the G30.1 block is as follows. The tool is first positioned along the specified axis at an intermediate position in the rapid traverse mode. Then, the tool is positioned from the intermediate position at the floating reference position in the rapid traverse mode. The tool is positioned at each of the intermediate and floating reference positions along each axis at the rapid traverse feedrate. This is called non–linear positioning.

Example of using G30.1



G30.1 G90 X50.0 Y40.0;

WARNING Before using G30.1, cancel cutter compensation and tool length compensation.

NOTE A floating reference position is not lost even if power is turned off.

8. COORDINATE SYSTEM

When the position to be reached by the tool is taught, the CNC moves the tool to that position. The position to be reached by the tool is given as a coordinate value in a coordinate system.

The following three types of coordinate systems are available.

(i) Machine coordinate system

(ii) Workpiece coordinate system

(iii) Local coordinate system

The coordinate value consists of one component for each program axis.

The coordinates are represented in format (a) for three program axes: X, Y, and Z or in format (b) for the X–, Y–, and Z–axes, and C–axis about which the machine rotates.

(a) X _____ Y _____ Z _____

(b) X _____ Y _____ Z _____ C _____

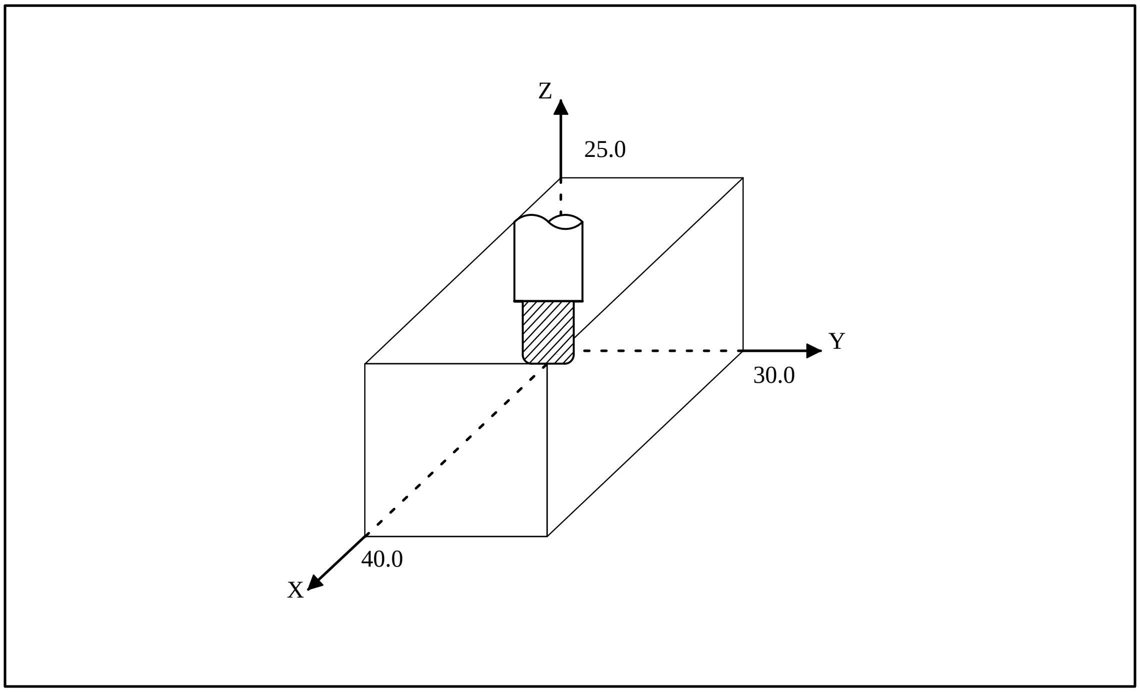


Fig. 8 Position of tool when X40.0Y30.0Z25.0 is commanded

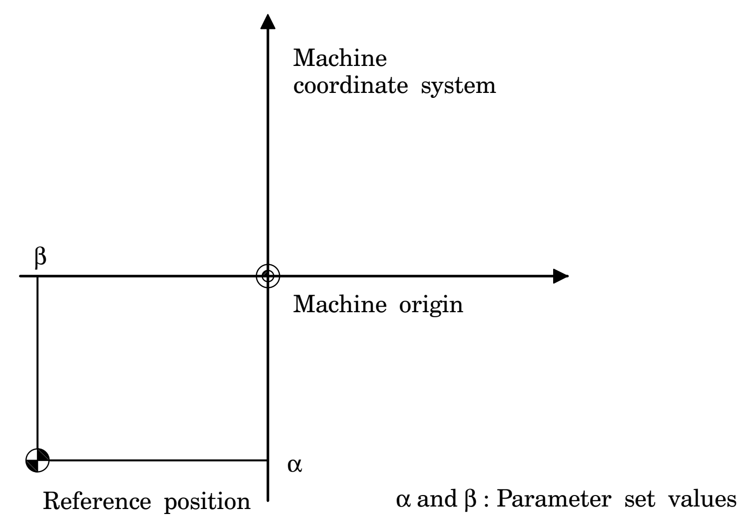
Since the number of program axes, that is, the number of components for the coordinate value, varies with the machine tools, the coordinate value is referred to as IP _____ in this manual.

8.1 Machine Coordinate System

The machine zero point is a standard point on the machine. The machine zero point is decided in accordance with the machine by the machine tool builder.
A coordinate system having the zero point at a machine zero point is called the machine coordinate system. The machine coordinate system is established when the reference position return is first executed after the power is on.
Once the machine coordinate system is established, it is not changed by reset, change of workpiece coordinate system (G92), local coordinate system setting (G52) or other operations unless the power is turned off.
The stored stroke limit (G22 or G23), which specifies the stroke of the machine, can be set by using the coordinates of the machine coordinate system.

8.1.1 Setting machine coordinate system

The machine coordinate system is inherent to each machine. It can be set through manual return to the reference position, so that the reference position is the coordinate value set by parameter 1240.



8.1.2 Selection of machine coordinate system (G53)

(G90) G53 IP ___;
When this command is specified, the tool moves to the position of the IP coordinate value in the machine coordinate system at rapid traverse. Since G53 is a one–shot G code, it is effective only in the block in which G53 is specified. It is effective only in absolute mode (G90) and is ignored in incremental mode (G91).
When the tool moves to a position specially determined for the machine tool, such as tool change position, programming is performed in the machine coordinate system by using the G53 command.

WARNING 1 When the G53 command is specified, cancel the cutter compensation, tool length offset, and tool offset.

WARNING 2 Since the machine coordinate system must be set before the G53 command is specified, at least one manual reference position return or automatic reference position return by the G28 command must be performed after the power is turned on.

NOTE 1 The machine zero point is generally different from the reference position for reference position return.

NOTE 2 Note that G50/G51, G50.1/G51.1, and G68/G69 codes are ignored in a block where a G53 code is specified.

8.2 Programming of Workpiece Coordinate System (G92, G54 – G59)

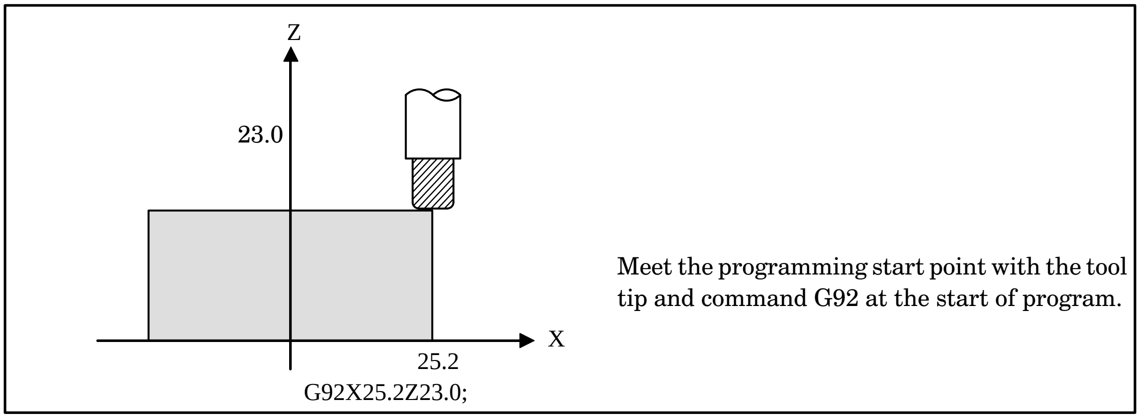
A coordinate system used for workpiece machining is called the workpiece coordinate system. The workpiece coordinate system is set by either of the following method:

1) Using G92 command.
2) Using G54 to G59 command.
   In case of 1), a workpiece coordinate system is established by specifying coordinate values after G92.
   In case of 2), six workpiece coordinate systems are preset from CRT/MDI panel as setting data and any of them can be selected by G54 to G59 command.

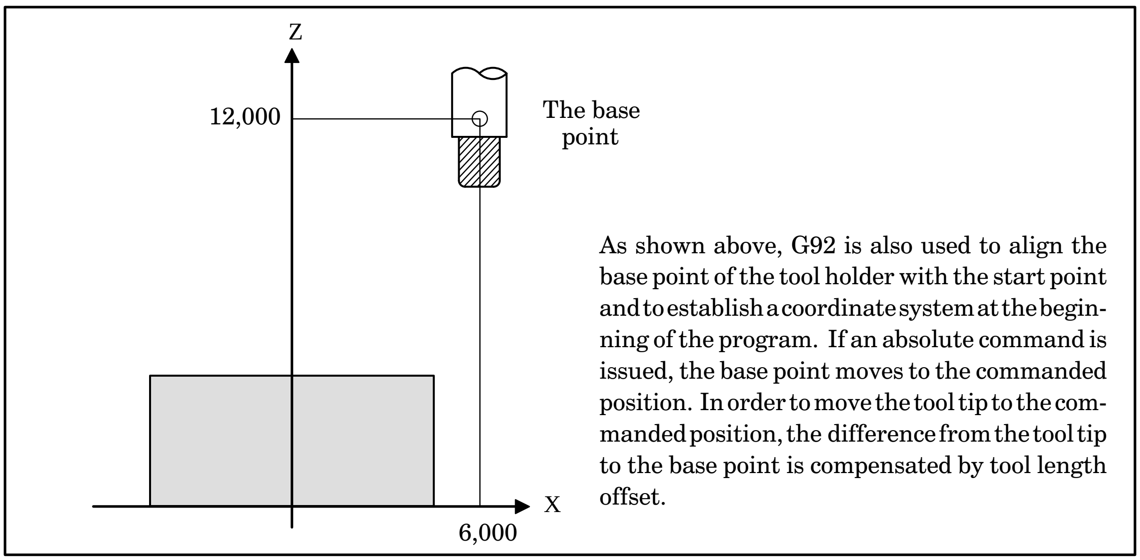
8.2.1 Setting workpiece coordinate system – method by G92

(G90) G92 IP _____;
This command establishes the workpiece coordinate system so that a certain point of the tool, for example the tool tip, becomes IP in the established workpiece coordinate system.
Any subsequent absolute commands use the position in this workpiece coordinate system.

Example 1 Setting the coordinate system by the G92X25.2Z23.0; command



Example 2 Setting the coordinate system by the G92X600.0Z1200.0; command when a start point is based on the base point of the tool holder



NOTE 1 If a coordinate system is set with the G92 command in offset mode, the coordinate system is set so that the position before tool length compensation is applied becomes that specified by G92.

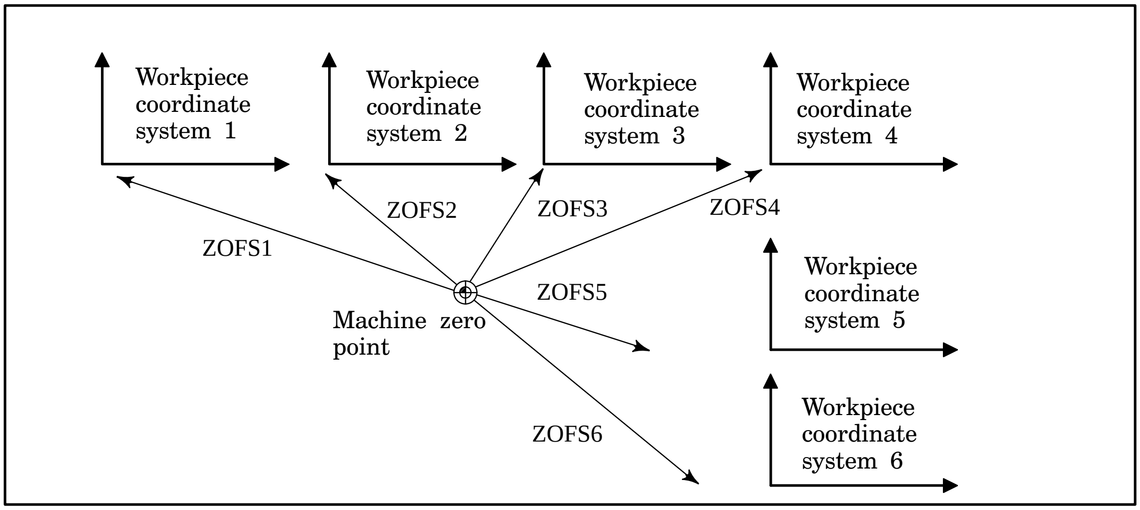
NOTE 2 In the cutter compensation, offset is temporarily cancelled by a G92 command.

NOTE 3 If the machine is manually returned to the reference position in the reset state (when the OP signal is off), the offset from the workpiece reference point specified by G92 is canceled, and the original workpiece coordinate system is established.

The corresponding parameter specifies whether the original workpiece coordinate system is established if the machine is manually returned to reference position in the automatic mode (when the OP signal is on).

8.2.2 Setting workpiece coordinate system – method by G54 – G59

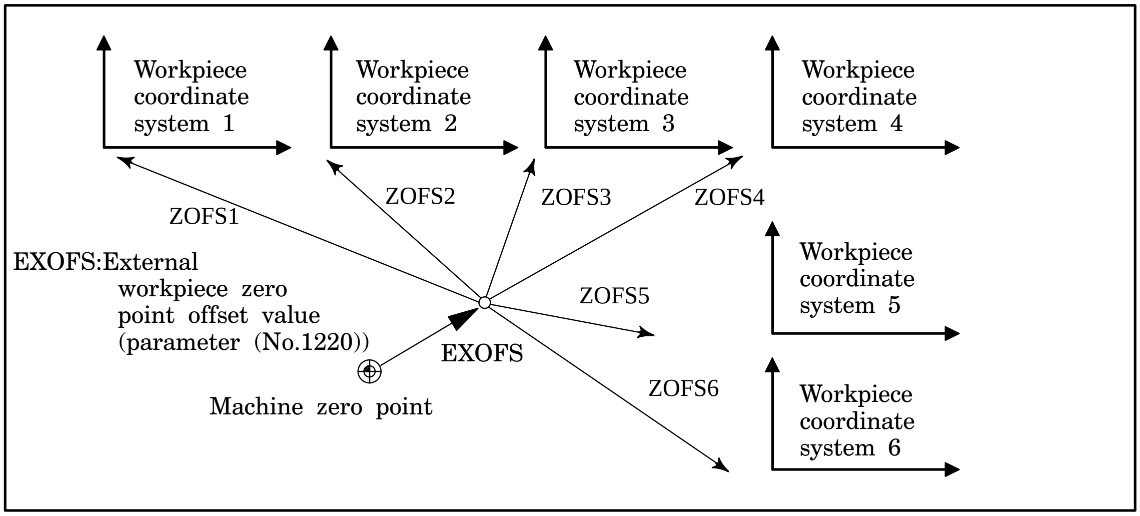
1) Setting workpiece coordinate system
   Six workpiece coordinate systems can be set.
   These six systems are decided by setting the distances of each axis from the machine zero point to the zero points of the coordinate systems, (i.e. the workpiece zero point offset value, by using the CRT/MDI panel).



ZOFS1: Workpiece zero point offset value of workpiece coordinate system 1 (parameter (No.1221))
ZOFS2: Workpiece zero point offset value of workpiece coordinate system 2 (parameter (No.1222))
ZOFS3: Workpiece zero point offset value of workpiece coordinate system 3 (parameter (No.1223))
ZOFS4: Workpiece zero point offset value of workpiece coordinate system 4 (parameter (No.1224))
ZOFS5: Workpiece zero point offset value of workpiece coordinate system 5 (parameter (No.1225))
ZOFS6: Workpiece zero point offset value of workpiece coordinate system 6 (parameter (No.1226))

2) Shifting workpiece coordinate system
Six workpiece coordinate systems can be shifted by a specified value (external workpiece zero point offset value)

Workpiece zero point offset value can be changed in three ways.



a) Using CRT/MDI panel
See Chpter III–10.

b) Using program (G10)
See Section 8.2.4, "Changing workpiece coordinate system by program command."

c) Using external workpiece coordinate system shift function
An input signal to the CNC from the PMC enables you to change the external workpiece zero point offset amount.
Refer to machine tool builder's manual for details.

NOTE 1 The external workpiece zero point offset amount can be set within ±7.999 mm or ±0.79999 inch for every axis.

NOTE 2 When G92 is specified after a shift value has been specified, the shift value is ignored.

Example When G92X100.0Z80.0; is commanded, a workpiece coordinate system in which present tool position is X = 100.0, Z = 80.0 is established, irrespective of set shift value.

NOTE 3 The diameter or radius for movement along the X–axis conforms to the diameter or radius specified in the part program.

8.2.3 Selecting workpiece coordinate system (G54 – G59)

Six coordinate systems proper to the machine tool are set in advance, permitting the selection of any of them by G54 to G59.

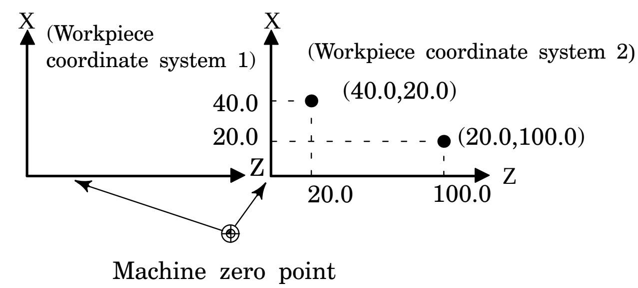
G54 ...... Workpiece coordinate system 1
G55 ...... Workpiece coordinate system 2
G56 ...... Workpiece coordinate system 3
G57 ...... Workpiece coordinate system 4
G58 ...... Workpiece coordinate system 5
G59 ...... Workpiece coordinate system 6

The six coordinate systems are determined by setting distances (workpiece zero offset values) in each axis from the machine zero point to their respective zero points using the CRT/MDI panel.

Example (G90) G55 G00 X20.0 Y100.0 ;
X40.0 Z20 .0 ;

In the above example, positioning is made to positions (X=20.0, Z=100.0) and (X=40.0, Z=20.0) in workpiece coordinate system 2.

Where the tool is positioned on the machine depends on workpiece zero point offset values.



Workpiece coordinate system 1 to 6 are established after reference position return after the power is turned on. When the power is turned on, G54 coordinate system is selected.

8.2.4 Changing workpiece coordinate systems by program command

When they are to be moved for each program, a programmed command can be used to shift them.

1) Using G10

G10 L2 Pp IP ____;

p = 0 : External workpiece zero point offset value.

p=1 to 6 : Correspond to workpiece coordinate system 1 to 6.

When the G10 command is executed in the G90 mode, the value specified by IP becomes the offset from the workpiece reference point. When the command is executed in the G91 mode, the value obtained by adding the value specified by IP to the current offset from the workpiece reference point becomes the offset from the workpiece reference point.

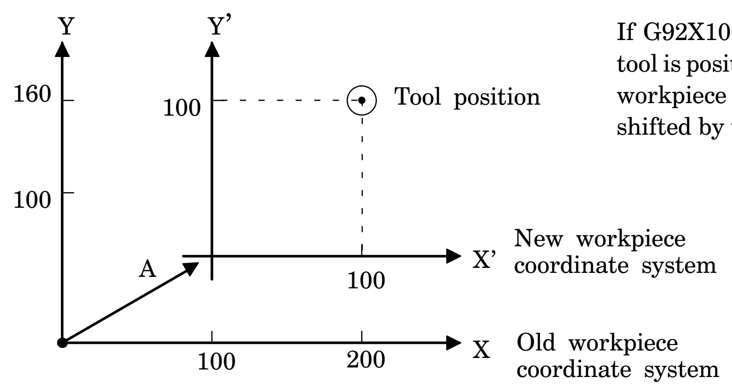
IP : Workpiece zero point offset value of each axis.

2) Using G92 IP

G92 IP ___ ;

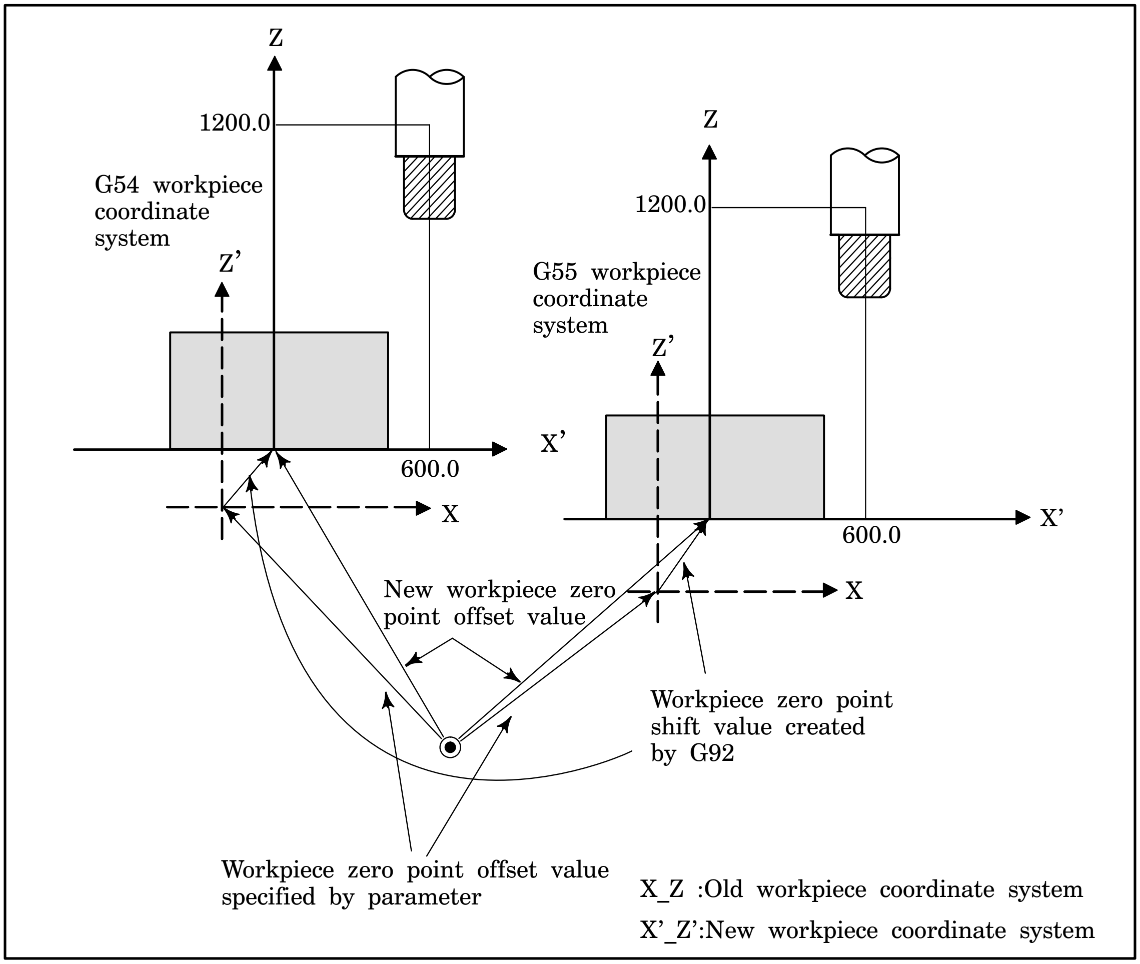
By this command, six workpiece coordinate systems move to create a new workpiece coordinate system so that the tool tip becomes the coordinate value IP commanded by G92.

Since the distance shifted at this time is added to all subsequent workpiece zero point offset values, all workpiece coordinate systems move by the same distance.



If G92X100Y100; is commanded when the tool is positioned at (200, 160) in G54 mode, workpiece coordinate system 1 (X' – Y') shifted by vector A is created.

When creating a new workpiece coordinate system with the G92 command, a certain point of the tool becomes a certain coordinate value; therefore, the new workpiece coordinate system can be determined irrespective of the old workpiece coordinate system. If the G92 command is used to determine a start point for machining based on workpiecepieces, a new coordinate system can be created even if there is an error in the old workpiece coordinate system. If the relative relationship among the G54 to G59 workpiece coordinate systems are correctly set at the beginning, all workpiece coordinate systems become new coordinate systems as desired.



If G92X600.0Z1200.0; is commanded,(shown in the diagram above) when the G54 workpiece coordinate system is specified, the G55 workpiece coordinate system is set in so that the black point on the tool shown in the diagram above becomes position (600.0, 1200.0); provided that the relative relationship between the G54 and G55 workpiece coordinate systems is set correctly.

Therefore, in loading pallets at two different positions, the two pallet coordinate systems are set as G54 and G55 workpiece coordinate systems. When the coordinate system of one pallet is moved by the G92 command, the coordinate system of the other pallet is also moved in the same manner, provided that the relative relationship of the two workpiece coordinate systems is set correctly. Accordingly, a workpiecepiece placed on two pallets can be machined with the same program by merely specifying either G54 or G55.

8.2.5 Additional workpiece coordinate systems (G54.1)

In addition to six workpiece coordinate systems (G54 through G59), forty–eight workpiece coordinate systems can be used.

An additional workpiece coordinate system can be selected with the following command:

G54.1 Pn ; (n = 1 to 48)

The coordinate systems selected with G54 through G59 are called the standard workpiece coordinate systems.

The coordinate systems selected with G54.1Pn are called the additional workpiece coordinate systems.

When G54.1 is specified together with a P code, an additional workpiece coordinate system from 1 to 48 is selected according to the P code.

A workpiece coordinate system once selected remains valid until another workpiece coordinate system is selected.

When power is turned on, G54 is selected.

> **NOTE 1** A P code must be specified after G54.1.
> If G54.1 is not followed by a P code in the same block, additional workpiece coordinate system 1 (G54.1P1) is selected.
>
> **NOTE 2** If a P code beyond the allowable range is specified, an alarm (PS305) is raised. As with the standard workpiece coordinate systems, the following operations can be performed for a workpiece reference position offset in an additional workpiece coordinate system:

(1) The OFFSET function key can be used to display and set a workpiece reference position offset.

(2) The G10 function enables a workpiece reference position offset to be set by programming.

(3) A custom macro allows a workpiece reference position offset to be handled as a system variable.

(4) Workpiece reference position offset data can be entered or output as external data.

(5) The PMC window function enables a workpiece reference position offset to be read as program command modal data.

G54: Standard workpiece coordinate system 1
G55: Standard workpiece coordinate system 2
G56: Standard workpiece coordinate system 3
G57: Standard workpiece coordinate system 4
G58: Standard workpiece coordinate system 5
G59: Standard workpiece coordinate system 6
G54.1P1: Additional workpiece coordinate system 1
G54.1P2: Additional workpiece coordinate system 2
. . .
. . .
. . .
G54.1P47: Additional workpiece coordinate system 47
G54.1P48: Additional workpiece coordinate system 48
Standard workpiece coordinate system 1 is the workpiece coordinate system selected with G54.
Additional workpiece coordinate system 1 is the workpiece coordinate system selected with G54.1P1 (or G54P1).

With the following command, a workpiece reference position offset in an additional workpiece coordinate system can be set:

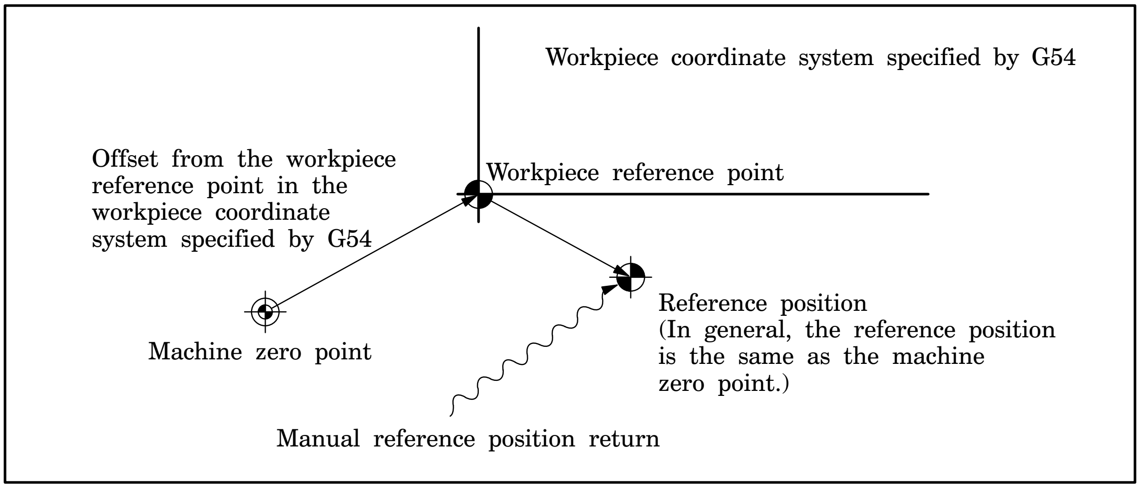
G10L20 Pn IP_ ; (n = 1 to 48)

With Pn, specify the number of a desired workpiece coordinate system. With, specify a setting value together with the address of each axis.

In the case of an absolute value, a specified value becomes a new offset. In the case of an incremental value, a specified value added to the currently set offset becomes a new offset

8.2.6 Presetting the workpiece coordinate system

When the machine is manually returned to the reference position in the reset state, the workpiece coordinate system is preset by the offset from the machine zero point to the workpiece reference point. For example, when the machine is manually returned to the reference position in the workpiece piece coordinate system selected by G54, the workpiece coordinate system, whose zero point is the position whose distance from the machine zero point is the offset from the workpiece reference position in the workpiece piece coordinate system selected by G54, is automatically specified (preset). Thus, the distance from the zero point in the workpiece coordinate system to the reference position indicates the current position in the workpiece coordinate system.



When the power to a machine equipped with an absolute–position detector is turned on, the workpiece coordinate system, whose zero point is the position whose distance from the machine zero point is the offset from the workpiece reference point in the workpiece coordinate system specified by G54, is automatically specified (preset). The coordinates of the machine after power–on are read from the absolute–position detector. The values obtained by subtracting the offset from the workpiece reference point specified by G54 from the read coordinates indicate the current position in the workpiece coordinate system. The preset workpiece coordinate system is, however, shifted from the machine coordinate system by the following command or operations:

(a) Manual intervention while the manual absolute signal is off

(b) Move command in the machine lock mode

(c) Movement by the interrupt with a handle, or operations performed manually and automatically at the same time

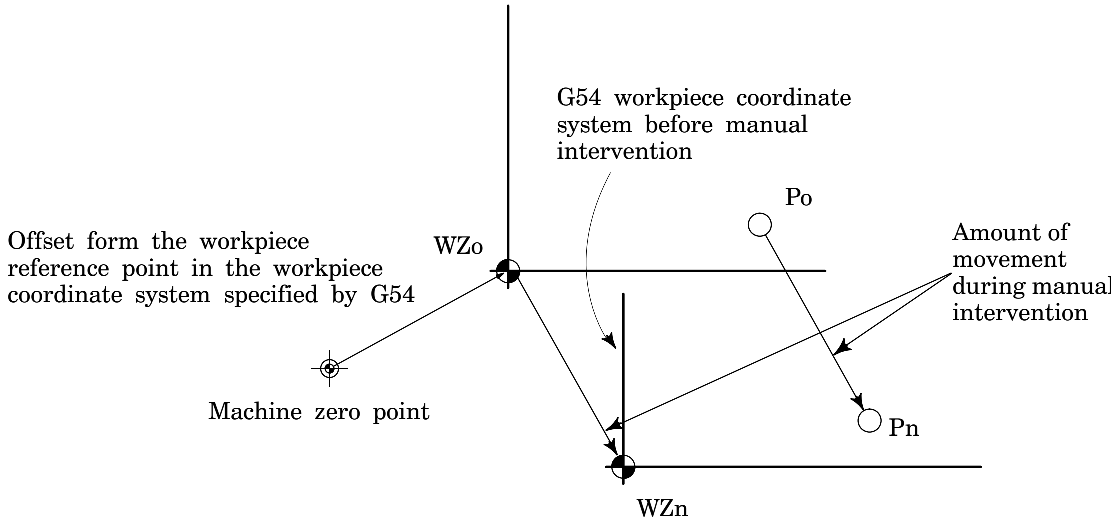
(d) Operation in the mirror image mode

(e) Setting of a local coordinate system with G52 or shifting of the workpiece coordinate system with G92

(f) Operation on the MDI to set the origin of the workpiece coordinate system

To take (a) for example, the workpiece coordinate system is shifted by the amount of movement during manual intervention.

Example Manual intervention white the manual absolute signal is off



The shifted workpiece coordinate system can be preset by the offset from the machine zero point to the workpiece reference point using a G command or by operations on the MDI in the same way as when the machine is manually returned to the reference position. Namely, in the above figure, the zero point WZn in the shifted workpiece coordinate system is returned to the original WZo, and the distance from WZo and to Pn indicates the current position in the workpiece coordinate system.

(1) Presetting the workpiece coordinate system using a G command

Issuing G92.1 enables the workpiece coordinate system to be preset. The command also cancels cutter compensation, tool length compensation, and tool offset.

G92.1 IP 0;

IP 0: Specifies the axis address for presetting the workpiece coordinate system. The workpiece coordinate system is not preset along any axis that is not specified.

(2) Presetting the workpiece coordinate system by operations on the MDI

The operations using soft keys on the MDI enable the workpiece coordinate system to be preset. For details, see Subsection 11.3.3 in Part III.

8.2.7 Automatically presetting the workpiece coordinate system

This function automatically presets the workpiece coordinate system to the position where machine lock is applied, after the machine is operated with machine lock set on and machine lock is released. This means that when this function is used, the current workpiece coordinate system can be restored to the original system, which has the specified offset from the machin zero point. This function is useful in machining after performing a test run with machine lock set on.

This function is automatically activated when the follwing conditions are satisfled:

(1) ACP, bit 7 of parameter No.1200, is set on. (This setting enables this function.)

(2) Automatic operation (MDI, memory, or tape) was performed with the all–axis machine lock signal turned on (the each–axis machine lock signals are ignored).

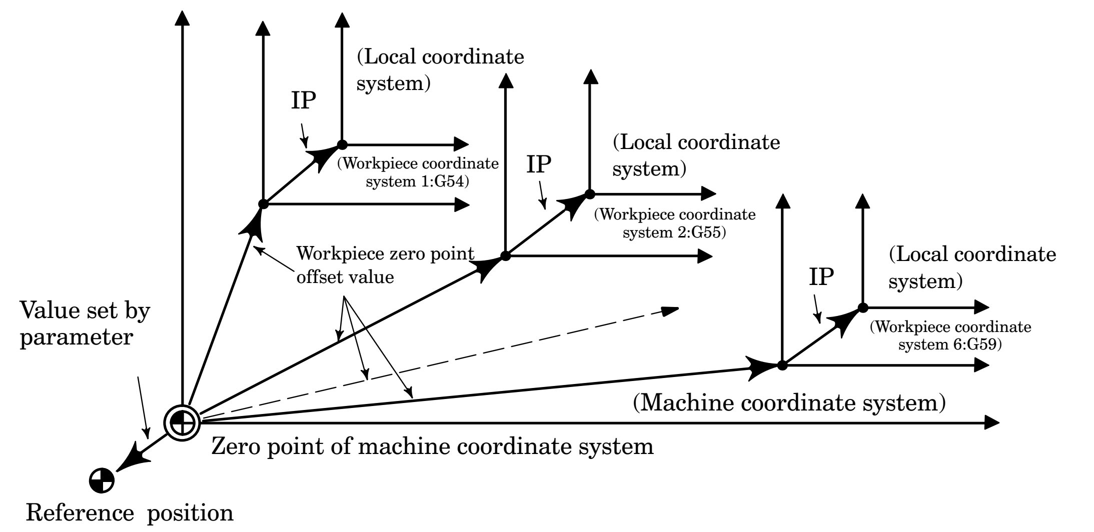
(3) When the system is reset (the STP, SPL, and OP signals are all off) after operation, the machine lock signal is off or turned off.

8.3 Local Coordinate System (G52)

While programming in a workpiece coordinate system, it is sometimes more convenient to have a child coordinate system within the workpiece coordinate system. Such a child coordinate system is called a local coordinate system.

G52 IP ___;

This command specifies child coordinate systems, (i.e., local coordinate systems), for all workpiece coordinate systems (G54 to G59). The zero point of each local coordinate system is equal to the IP position of each workpiece coordinate system.



When a local coordinate system is set, the move commands in absolute mode (G90), which is subsequently commanded as are the coordinate values in the local coordinate system. The local coordinate system can be changed by specifying the G52 command with the zero point of a new local coordinate system in the workpiece coordinate system. To cancel the local coordinate system and specify the coordinate value in the workpiece coordinate system, match the zero point of the local coordinate system with that of the workpiece coordinate system.

That is, **G52 IP 0;** should be specified.

WARNING The absolute move command must be commanded immediately after the G52 block.

NOTE 1 When the machine is manually returned to the reference position, the zero point in the local coordinate system whose axis is returned to the reference position matches that in the workpiece coordinate system. The following command produces the same effect as returning the machine manually to the reference position:

G52α 0; (α : Axis along which the machine is returned to the reference position)

NOTE 2 If local coordinate systems are established, the workpiece coordinate system and machine coordinate system are not changed.

NOTE 3 The reset operation cancels the local coordinate systems.

NOTE 4 Unless the coordinates along all axes are specified to establish the workpiece coordinate system using the G92 command, the local coordinate systems with the axes for which the coordinates are not specified are not canceled or changed.

NOTE 5 G52 temporarily cancels the offset for cutter compensation. If G52 is specified even to an axis while a cutter compensation vector remains (no movement command is specitied with G40), the local coordinate system is cancelled.

8.4 Plane Selection (G17, G18, G19)

Select the planes for circular interpolation, cutter compensation, and drilling by G–code.
The following table lists G–codes and the planes selected by them.

G–code	Selected plane	Xp	Yp	Zp
G17	Xp Yp plane	X–axis or an axis parallel to it	Y–axis or an axis parallel to it	Z–axis or an axis parallel to it
G18	Zp Xp plane			
G19	Yp Zp plane			

Which axis, of several parallel axes is actually assumed, is determined by the axis address appearing in the block in which G17, G18, or G19 was specified.

Example When X and U, Y and V, and Z and W are parallel axes :

G17X____ Y____ XY plane
G17U____ Y____ YU plane
G18X____ W____ WX plane
G18U____ W____ WU plane
G19Y____ Z____ YZ plane
G19V____ Z____ VZ plane

If G17, G18, or G19 is not specified in a block, the plane remains unchanged.

Example

G17X____ Y____XY plane
U____ Y____ The plane is not changed (XY plane)

When the axis address is omitted in blocks G17, G18, and G19, three basic axis addresses are assumed to be omitted

Example

G17 XY plane
G17X____ XY plane
G17Y____ XY plane
G17U____ UY plane
G17V____ XU plane
G18 ____ ZX plane
G18W___ WX plane

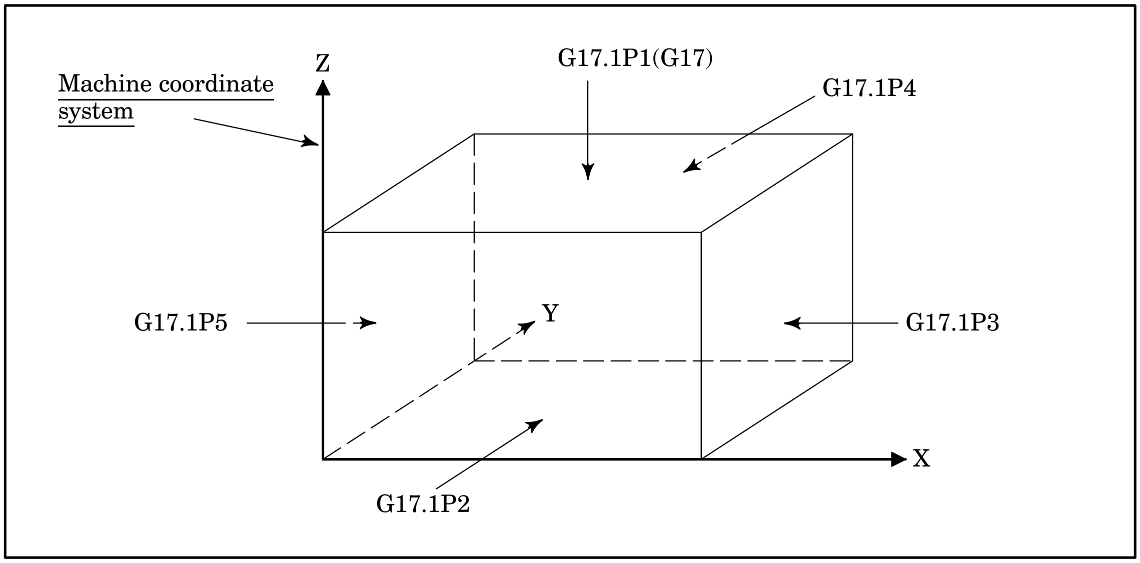
NOTE 1 An additional axis can be made a parallel axis of the X, Y, or Z axes by parameter setting (parameter numbers : 1022).

The movement command is not related to the plane selection.
For example, when G17Z____; is specified, the Z axis does not exist in the XpYp plane. Only the XY plane is selected and Z moves independently of the plane.

NOTE 2 G17 (XY plane) or G18 (ZX plane) is selected when powered on or reset according to parameter G18 (Data No. 2401).

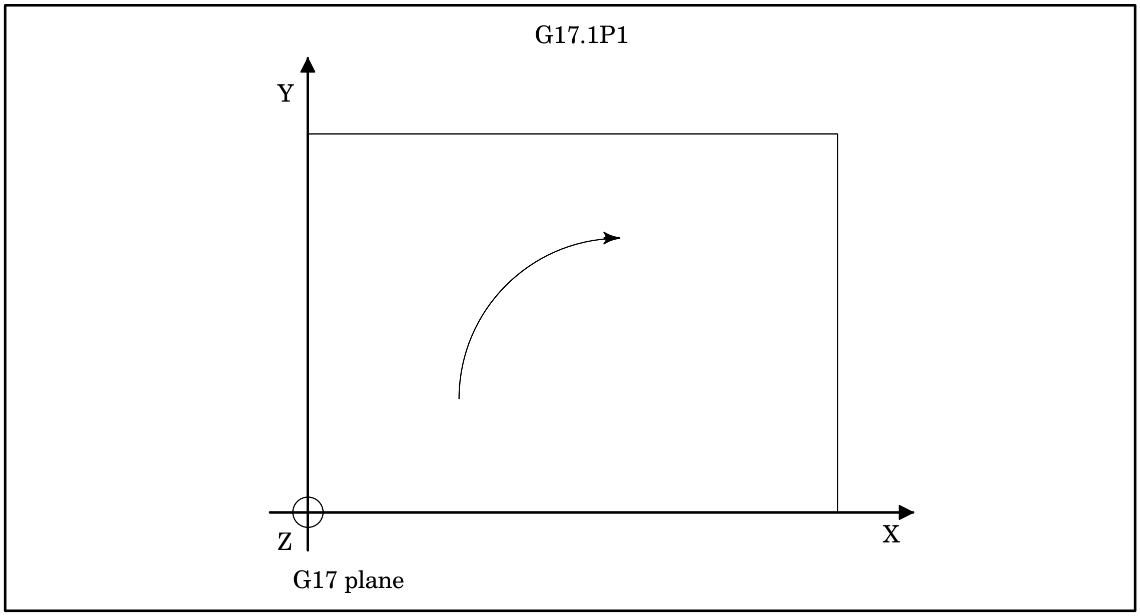
8.5 Plane Conversion Function

This function converts a machining program created on the G17 plane in the right–hand Cartesian coordinate system to programs for other planes specified by G17.1Px commands, so that the same figure appears on each plane when viewed from the directions indicated by arrows.



The plane conversion for a machining figure on the G17 plane is performed as follows:

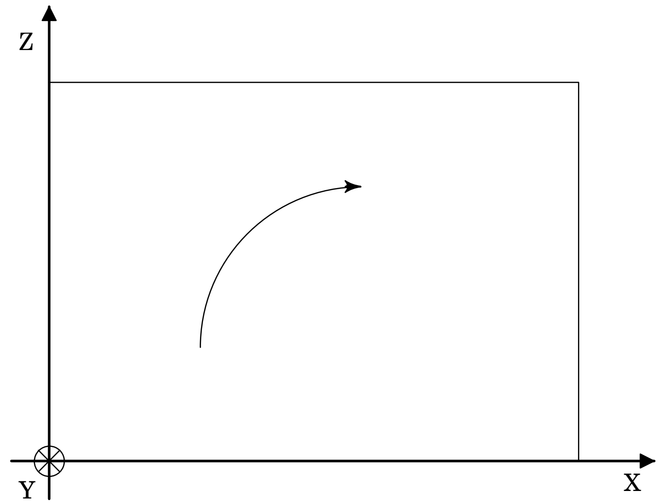
(1)



NOTE ○ indicates that the positive direction of the axis perpendicular to the page is the direction coming out of the page (in this case, the Z–axis perpendicular to the XY plane).

(2)

G17.1P2

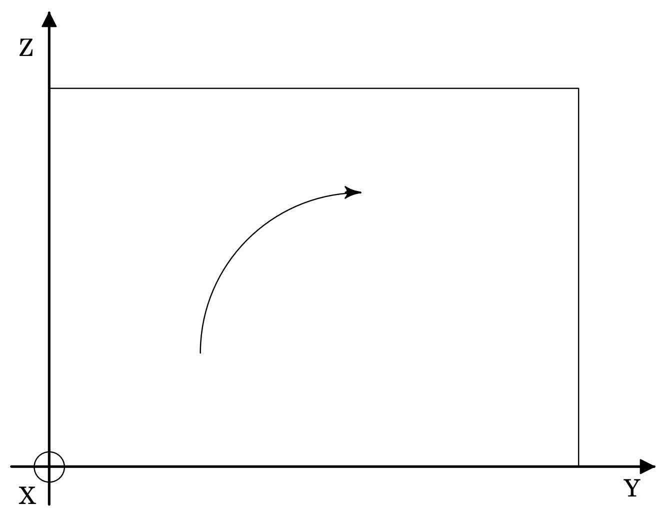


G18 plane

NOTE	⊗ indicates that the negative direction of the axis perpendicular to the page is the direction coming out the page (in this case, the Y–axis perpendicular to the XZ plane).

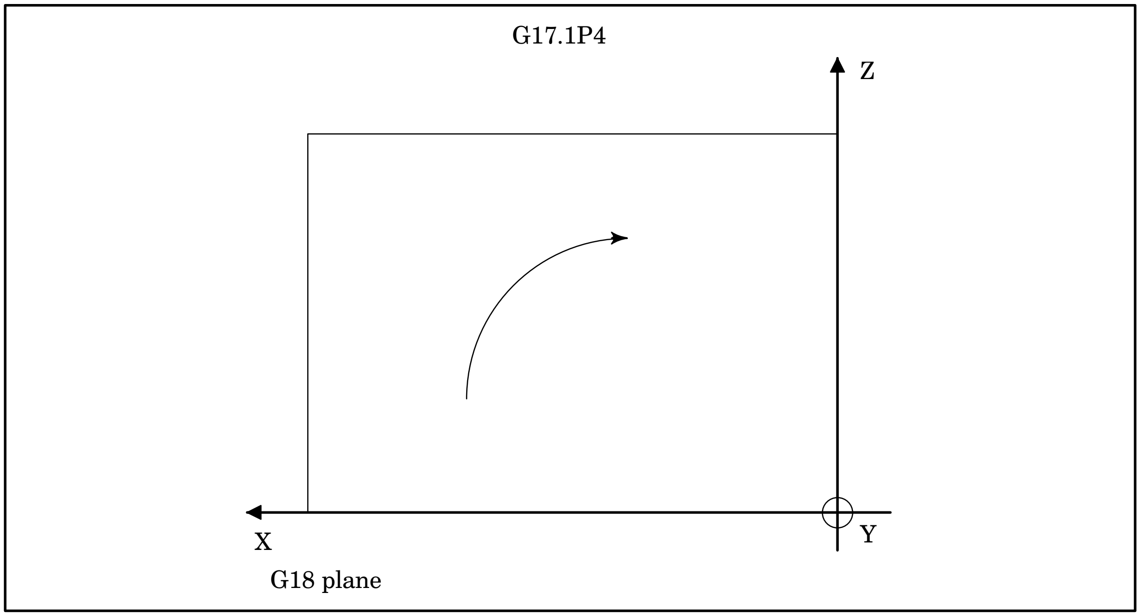
(3)

G17.1P3

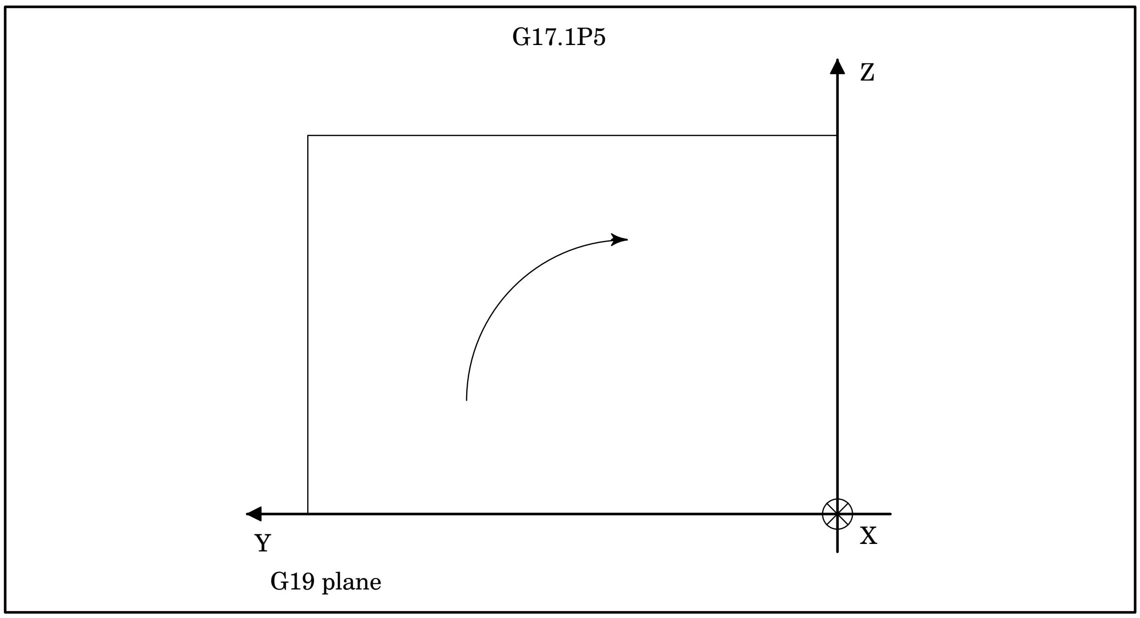


G19 plane

(4)



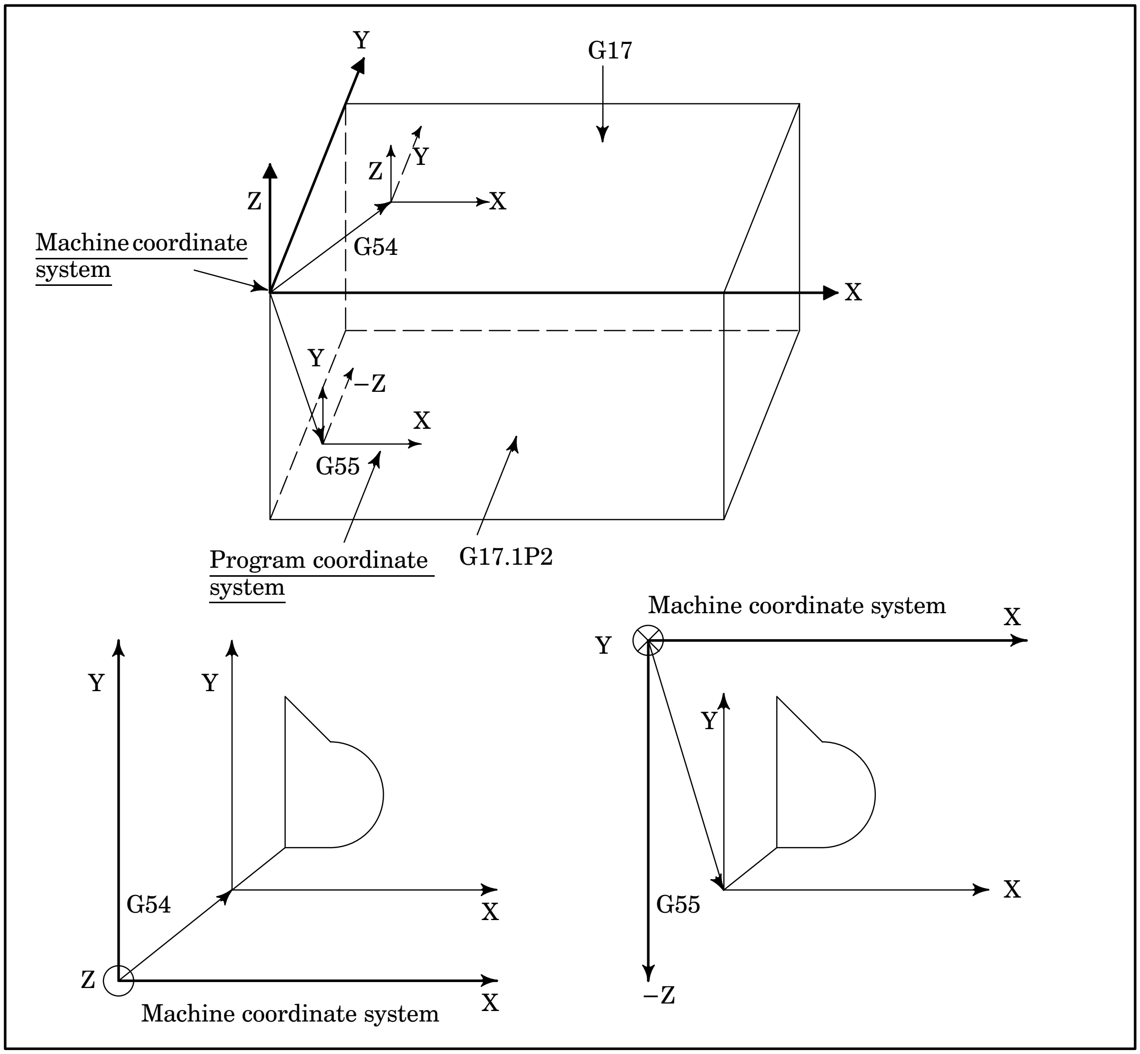
(5)



Program commands on the G17 plane are converted to the following commands by plane conversion:

Command	G17.1P1	G17.1P2	G17.1P3	G17.1P4	G17.1P5
X	X	X	Y	–X	–Y
Y	Y	Z	Z	Z	Z
Z	Z	–Y	X	Y	–X
G02	G02	G03	G02	G02	G03
G03	G03	G02	G03	G03	G02
I	I	I	J	–I	–J
J	J	K	K	K	K
K	K	–J	I	J	–I
G41	G41	G42	G41	G41	G42
G42	G42	G41	G42	G42	G41
Tool length compensation	+	–	+	+	–
Direction of coordinate rotation	+	–	+	+	–
Direction of the drilling axis	+	–	+	+	–
Plane	G17	G18	G19	G18	G19

Program example:



O1000 (MAIN PROGRAM) ;	O2000 (SUB PROGRAM) ;
N10 G91 G28 X0 Y0 Z0 ;	N2010 G90 G0 Z0 ;
N20 G54 ;	N2020 G0 X0 Y0 ;
N30 G17 ;	N2030 G0 X30. Y20. ;
N40 M98 P2000 ;	N2040 G01 Z–50. F200 ;
N50 G55 ;	N2050 Y90. F500 ;
N60 G17.1 P2 ;	N2060 Y60. Y70.
N70 M98 P2000 ;	N2070 G02 Y20. J–25. ;
N80 G91 G28 X0 Y0 Z0 ;	N2080 G01 X30. ;
N90 M02 ;	N2090 G0 Z0 ;
	M2100 M99 ;

WARNING Plane conversion cannot be performed for the following commands for moving the tool to a specified position, commands related to the machine coordinate system, and commands for setting a coordinate system:

(1) Automatic reference position return (G28 and G30)

(2) Floating reference position return (G30.1)

(3) Return from the reference position (G29)

(4) Selecting the machine coordinate system (G53)

(5) Stored stroke limit (G22)

(6) Setting the coordinate system (G54 to G59 and G92)

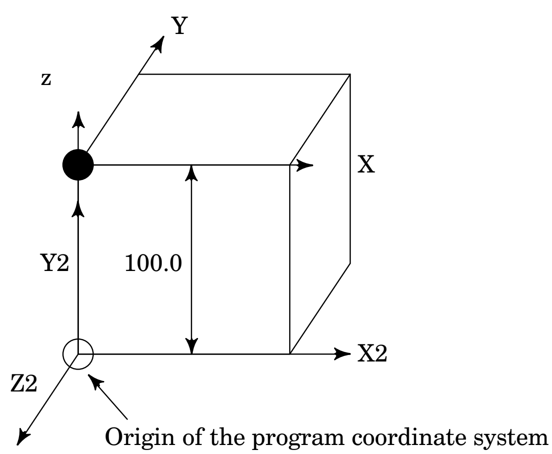
(7) Presetting the workpiece coordinate system (G92.1)

(8) Setting the offset (G10)

Setting bit 0 of parameter No. 2407 performs plane conversion for a G92 or G92.1 command.

Example When bit 0 of parameter No. 2407 is 1

```
:
:
N100 G00 X0 Y0 Z0 ;
N110 G17.1 P2 ;
N120 G92 Y100. ;
```



Command bloc	Absolute coordinate value		
	X	Y	Z
N100	0	0	0
N110	0	0	0
N120	0	0	100.

NOTE 1 Plane conversion can be performed only for commands for the X–, Y–, or Z–axis.
NOTE 2 Plane conversion cannot be performed for manual operation.
NOTE 3 The current position display on the CRT shows the coordinates after plane conversion.

Program command	Absolute coordinate value		
	X	Y	Z
G90 G00 X0 Y0 Z0;	0	0	0
G17.1 P3;	0	0	0
G00 X10. Y20.;	0	10.0	20.0
G01 Z–50. F200;	–50.0	10.0	20.0
G02 X50. Y60. I40.;	–50.0	50.0	60.0

NOTE 4 Plane conversion cannot be performed together with the axis switching function.
NOTE 5 Specify plane conversion commands after canceling the following modes.
(1) Cutter compensation
(2) Tool length compensation
(3) Canned cycle
(4) Three–dimensional coordinate conversion
(5) Coordinate rotation
(6) Scaling
(7) Programmable mirror image
NOTE 6 Plane conversion cannot be performed for the following commands which control a rotation axis together with the X–, Y–, or Z–axis:
(1) Polar coordinate interpolation
(2) Cylindrical interpolation
(3) Control in normal directions
(4) Exponential interpolation
(5) Circular threading B
NOTE 7 If a G17, G18, or G19 command is executed during plane conversion, the conversion is disabled and the plane specified by the command is selected.
NOTE 8 When 1 is set in NCM (bit 7 of parameter No. 2401), resetting the system in the plane conversion mode does not change the mode.

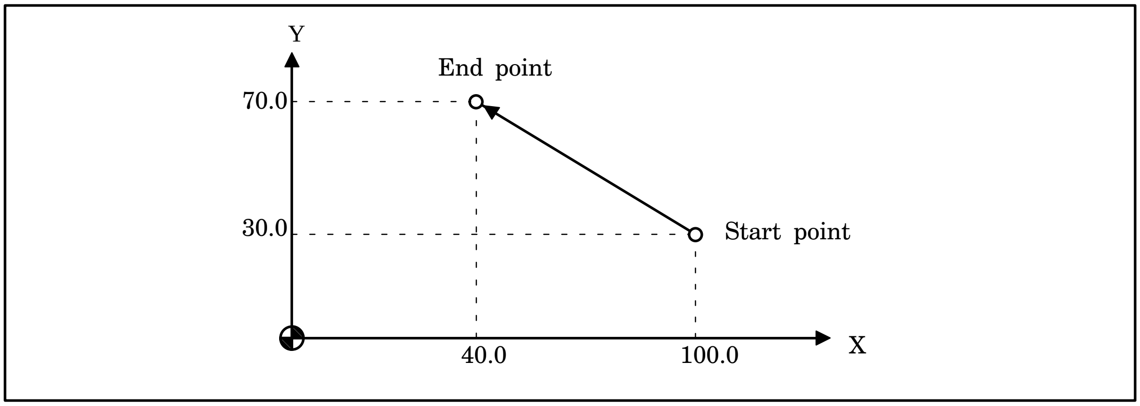
9. COORDINATE VALUE AND DIMENSION

9.1 Absolute and Incremental Programming (G90, G91)

There are two ways to command travels of the axes; the absolute command, and the incremental command. In the absolute command, coordinate value of the end point is programmed; in the incremental command, move distance of the axis itself is programmed.

G90 and G91 are used to command absolute or incremental command, respectively.

G90 : Absolute command
G91 : Incremental command



For the above figure, absolute command and incremental command programming results in:

G90 X40.0 Y70.0;————Absolute command
G91 X–60.0 Y40.0;————Incremental command

9.2 Polar Coordinate System Command (G15, G16)

The end point coordinate value can be input in polar coordinates (radius and angle). Use G15, G16 for polar coordinates command.

G15: Polar coordinate system command cancel
G16: Polar coordinate system command

Plane selection of the polar coordinates is done using G17, G18, G19.
Command radius in the first axis of the selected plane, and angle in the second axis. For example, when the X–Y plane is selected, command radius with addres X, and angle with address Y. The plus direction of the angle is counterclockwise of the selected plane first axis + direction, and the minus direction is clockwise.
Both radius and angle can be commanded in either absolute or incremental command (G90, G91).

(1) When the radius is specified with absolute command

The zero point of the local coordinate system becomes the center of the polar coordinate system.

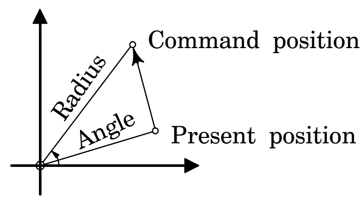


Fig. 9.2 (a) When the angle is specified with absolute value

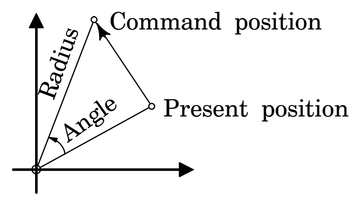


Fig. 9.2 (b) When the angle is specified with incremental value

(2) When the radius is specified with incrmenal command

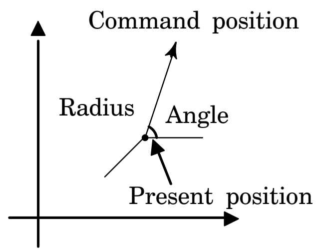


Fig. 9.2 (c) When the angle is specified with absolute value

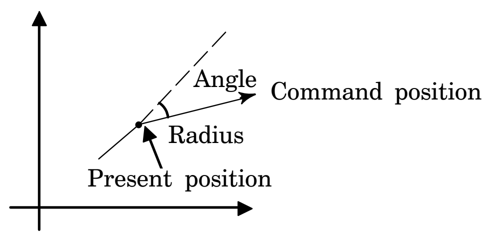


Fig. 9.2 (d) When the angle is specified with incremental value

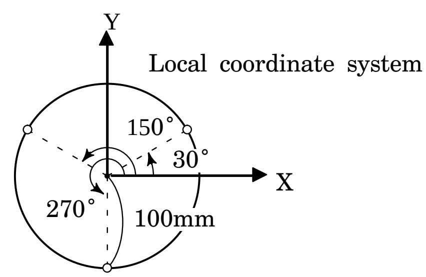
Example Bolt hole cycle

(Both radius and angle are absolute.)

N1 G17 G90 G16; Polar coordinates command, X–Y plane
N2 G81 X100.0Y30.0Z–20.0R–5.0F200.0; 100mm radius, 30° angle
N3 Y150; 100mm radius, 150° angle
N4 Y270; 100mm radius, 270° angle
N5 G15 G80; Polar coordinates cancel

(Radius is absolute and angle is incremental.)

N1 G17 G16; Polar coordinates command, X–Y plane
N2 G81 G90 X100.0Y30.0Z–20.0R–5.0F200.0; 100mm radius, 30° angle
N3 G91 Y120.; 100mm radius, 120° angle
N4 Y120.; 100mm radius, 120° angle
N5 G15 G80; Polar coordinates cancel



NOTE 1 The following commands are not regarded as polar coordinate commands even if specified with an axis address.

1) Dwell (G04)
2) Change of offset value by programming (G10)
3) PMC data setting (G10.1)
4) Setting of local coordinate system (G52)
5) Change of work coordinate system (G92)
6) Selection of machine coordinate system (G53)
7) Stored stroke check (G22)
8) Coordinate system rotation (G68)
9) Scaling (G51)

NOTE 2 Specify a radius value in circular interpolation or in helical cutting (G02, G03) when polar coordinate system is selected.

9.3 Inch/Metric Conversion (G20, G21)

Either inch or metric input can be selected by G code.

Unit systems	G code	Least input increment		
		IS–A	IS–B	IS–C
Inch	G20	0.001 inch	0.0001 inch	0.00001 inch
Millimeter	G21	0.01 mm	0.001 mm	0.0001 mm

This G code must be specified in an independent block before setting the coordi–nate system at the beginning of the program. The unit systems of the following items can be changed with G codes:

(1) Feedrate commanded by F code

(2) Positional command

(3) Workpiece reference position offset value

(4) Unit of scale for manual pulse generator

(5) Movement distance in step feed

(6) Some parameters

NOTE 1 When the power is turned on, the G code is the same as that held before the power was turned off.

NOTE 2 G20 and G21 must not be switched during a program.

NOTE 3 When the machine unit and the input unit systems are different, the maximum error is half of the least command increment. This error is not accumulated.

NOTE 4 When switching inch input (G20) to metric input (G21) and vice versa, the workpiece reference position offset value must be reset according to the input unit. The tool offset value is automatically converted into inches or millimeters.

9.4 Decimal Point Programming/Pocket Calculator Type Decimal Point Programming

This control can input numerical values with a decimal point. However, some addresses cannot use a decimal point. A decimal point may be used with mm, inches or second values. The location of decimal point is mm, inch or second.

Z15.0 Z15 millimeters or Z15 inches.
F10.0 10mm/rev, 10 mm/min, 10 inch/rev, or 10 inch/min.
G04X1.0 Dwell for one second

The following addresses can be used with a decimal point: X, Y, Z, U, V, W, A, B, C, E, F, G, I(*), J, K, Q(*), R, S.

(*) : Invalid in G26 block (Spindle speed fluctuation detection mode on)
Use parameter DPI (No. 2400#0) to select input method; whether to input by pocket calculator type input, or by the former type decimal point input.

Example

Program command	Pocket calculator type decimal point input	Former type decimal point input
X1000 X1000.	1000 mm 1000 mm	1 mm 1000 mm

WARNING There is a great difference in values with and without the decimal point, when programmed by conventional decimal point programming.

G21; (millimeter dimensions)
X1. ……… X1 mm
X1 ……… X0.001 mm
G20; (inch dimensions)
X1. ……… X1 inch
X1 ……… X0.0001 inch

NOTE 1 In the dwell command, decimal point can be used with address X but not with address P. (This is because P is also used for a sequence number.)

NOTE 2 The appropriate G code should be specified before the numerical values are specified in one block.

1) G20; (inch dimension)
X1.0G04;
Because the value X1.0 is not regarded as the number of seconds, but the distance of motion (in inches), X10000G04 is assumed resulting in dwelling for 10 seconds. Indication also changes from 1.0 to 10.0 when G04 is input.
G04X1.0 ;
This is regarded as G04X1000 and dwell is performed for a second.

2) G94; (mm/min specification)
F1.G95; …………………F1G95 is assumed, resulting in 0.01 mm/rev.
(G95; mm/rev specification)
G94; mm/min specification
G95F1.; …………………G95F100 is assumed, resulting in 1 mm/ rev.

NOTE 3 Values with and without a decimal point can be specified together.
X1000Z23.7;
X10.0Y22359;

NOTE 4 Values less than the least input increment are rounded off.
When X1.23456 is specified, X1.234 is assumed in millimeter input and X1.2345 is assumed in inch input.
Also, the number of digits must not exceed the maximum number of digits allowed.
X1.23456789 …… This is an error because it exceeds 8 digits.
X1.2345678 …… This is not an error because it is within 8 digits.

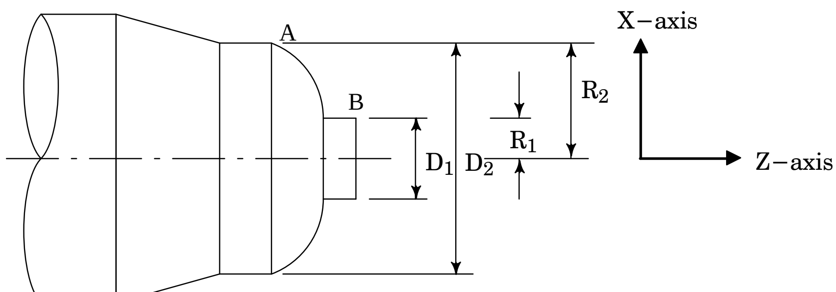
NOTE 5 When a number with a decimal point has been input, the number is converted into an integer of the least input increment

Example X12.34 → X12340 (input in millimeters)
This converted integer is checked for its number of digits.

Example X1234567.8 → X123456800(input in millimeters)
An alarm occurs because the converted integer exceeds 8 digits.

9.5 Diameter and Radius Programming

Since the section of a workpiece to be machined in a lathe is usually circular, the sectional dimensions can be programmed with diameters or radiuses in an NC unit.



Specifying the dimensions of a workpiece using diameters is called diameter programming. Specifying the dimensions using radiuses is called radius programming. DIA of parameter 1006 is used to specify whether the dimensions for each axis are specified using diameters or radiuses.

Conform to the conditions in the table below when specifying the dimensions of a workpiece using diameters for the X–axis.

Item	Notes
Commands for the Z–axis	Irrespective of diameters and radiuses
Commands for the X–axis	Use diameters.
Incremental commands for the X–axis	Use diameters. B → A in the above figure: D2–D1
Coordinate system specification (G92)	Specify the X coordinates using diameters.
Radius commands (R, I, K) for circular interpolation	Use radiuses.
Feedrate along the X–axis	Radius change/rev Radius change/min
Displaying X coordinates	Use diameters.

NOTE The following does not describe the details of diameter and radius programming. In short, the gradations for the X–axis are defined in diameters during diameter programming, and defined in radiuses during radius programming.

9.6 Function for Switching Between Diameter and Radius Programming

In previous systems, the DIAx bit (bit 3 of parameter 1006) specifies whether to use diameter or radius programming for each controlled axis. With this function, the G code can switch between diameter and radius programming for axis commands.

If the PGDM bit (bit 5 of parameter 1001) is set, the G code for selecting diameter or radius programming is valid for all axis commands specified in the program.

(Command format)

G10.9 IP_;

IP is the address of an axis for which diameter or radius programming can be selected. The following values can be specified after IP:

0: Radius programming

1: Diameter programming

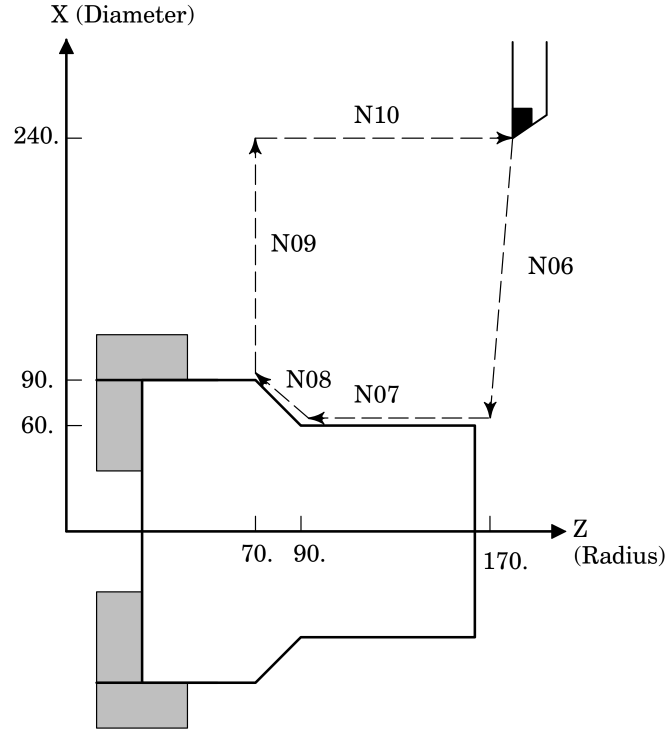
Specify a value without a decimal point after IP.

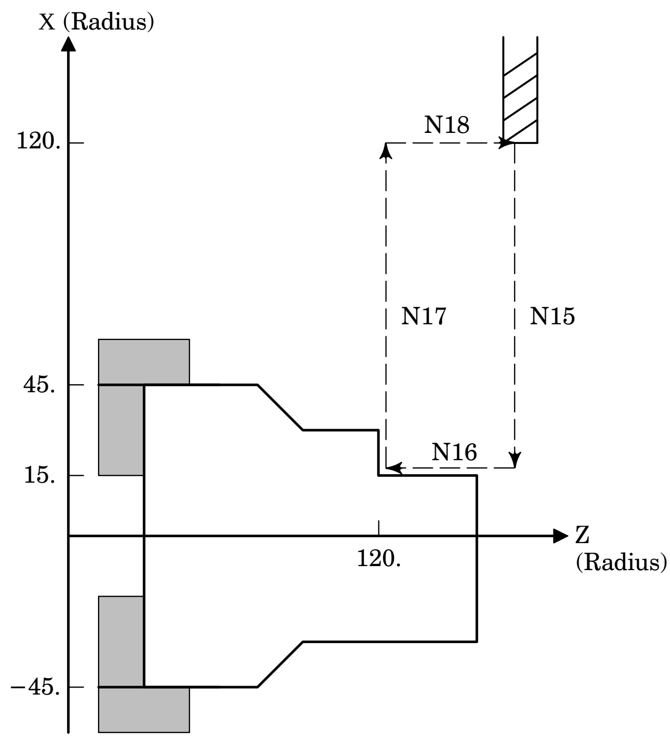
Do not specify any other codes in the block with G10.9.

When G10.9 is specified, the PDIAx bit (bit 3 of parameter 1009) is rewritten.

Example

```
O0010 ;
N01 T1001;
N02 G10.9 X1 Z0;
N03 S200;
N04 M03;
N05 G0 G90 X240. Z180.;
N06 X600 Z170.;
N07 G1 Z90.;
N08 X90. Z70.;
N09 G0 X240.;
N10 Z180.;
N11 M05;
N12 T2002;
N13 G10.9 X0 Z0;
N14 G0 G90 X120. Z175.;
N15 X15.;
N16 G1 Z120.;
N17 G0 X120.;
N18 Z175.;
N19 M30;
%
```





NOTE 1 Specify parameters and offset values according to the value of the DIAx bit (bit 3 of parameter 1006).

NOTE 2 The position is displayed according to the value of the PDIAx bit (bit 3 of parameter 1009).

NOTE 3 If G10.9 is specified during parallel axis control, the PDIAx bit (bit 3 of parameter 1009) of the master and slave axes is changed regardless of whether the tool is parked along a parallel axis.

NOTE 4 G10.9 is ignored during background graphics display.

NOTE 5 On the graphic display screen, the scale is indicated according to the value of the DIAx bit (bit 3 of parameter 1006).

NOTE 6 When a manual numerical command is specified with G00 or G01, the tool moves along the axis according to the value of the PDIAx bit (bit 3 of parameter 1009).

NOTE 7 The value of the PDIAx bit (bit 3 of parameter 1009) determines how the current position is displayed and how the workpiece coordinate system is preset.

10.SPINDLE SPEED FUNCTION (S FUNCTION)

10.1 Spindle Speed Command

By specifying a numerical value following address S, a binary code signal and a strobe signal are transmitted to the machine tool. This is mainly used to control the spindle speed.

An S code can be commanded in a block.Refer to the machine tool builder's manual for the number of digits commandable with address S and the correspondence between the S codes and machine operations.

When a move command and an S code are specified in the same block, the commands are executed in one of the following two ways:

i) Simultaneous execution of the move command and S function commands

ii) Executing S function commands upon completion of move command execution. The selection of either sequence depends on the machine tool builder's specifications. In certain cases, these sequences may be provided together in a CNC machine. Refer to the manual issued by the machine tool builder for details.

10.2 Constant Surface Speed Control (G96, G97)

The surface speed (relative speed of tool and a workpiece) is specified after S. The spindle speed is calculated so the surface speed is kept as specified with a tool position change. It then supplies a voltage, corresponding to the calculated spindle speed, to the spindle control to rotate the spindle at the correct surface speed. This procedure is the constant surface speed control.

The surface speed unit is a shown below.

Input unit	Surface speed unit
Millimeter	m/min
Inch	feet/min

This surface speed unit may change according to machine tool builders.

10.2.1 Specification method

For the constant surface speed control, specify the following G codes. Command constant surface speed as follows:

G96S ○○○○○;

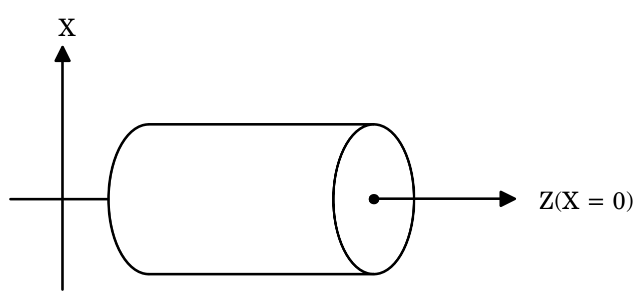
Surface speed (m/min or feet/min)

Cancel constant surface speed as follows:

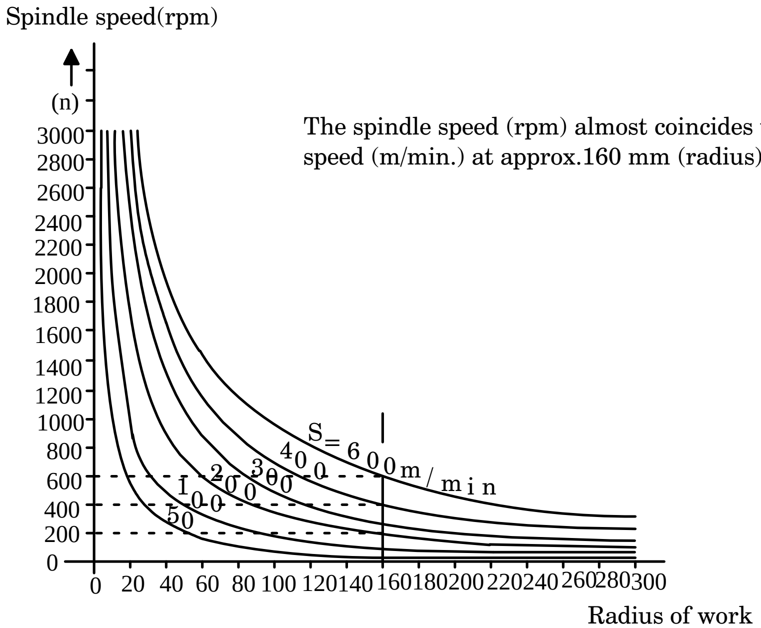
G97S ○○○○○;

Spindle speed(rpm)

To execute the constant surface speed control, it is necessary to set the work coordinate system so the coordinate value at the center of the rotary axis, for example, Z axis, (axis to which the constant surface speed control applies) becomes zero.



The surface speed (S) in G96 mode is regarded as S = 0 until either M03 (clockwise spindle rotation) or M04 (counterclockwise spindle rotation) is specified; S is not effectuated until M03 or M04 is specified.



Ralation between work radius, spindle speed and surface speed

NOTE 1 The S value specified in G96 mode is stored even after G97 mode has turned to G97 mode, and is recovered when G97 mode has returned to G96 mode.

G96S50:	(50 m/min or 50 feet/min)
G97S1000:	(1000 rpm)
G96X3000:	(50 m/min or 50 feet/min)

NOTE 2 With machine lock, the constant surface speed is calculated according to a change of the coordinate values of the axis to which the constant surface speed control is applied, even if the machine tool does not move.

NOTE 3 The constant surface speed control is also effective during threading. Accordingly, it is recommended that the constant surface speed control with G97 command be invalidated before starting the face threading and taper threading, because the response problem in the servo system may not be considered when the spindle speed changes.

NOTE 4 The constant surface speed control mode (G96) is allowable in G94 (feed per minute).

NOTE 5 If S (rpm) is not specified in advance in G97 block when G96 mode has turned to G97 mode, the last spindle speed in G96 mode is employed as S in G97 mode.

```
N111G97S800;   800 rpm
     .
     .
     .
N222G96S100;   100 m/min
     .
     .
     .
N333G97; X rpm
```

X shows spindle speed X rpm in the block before N333. In other words, the spindle speed remains unchanged when the mode has changed from G96 to G97.

The S value specified last in G96 mode becomes effective when G97 turns to G96.

S = 0 m/min (feet/min), if S is not specified.

10.2.2 Clamp of maximum spindle speed (G92)

The maximum spindle speed can be specified by rpm with a numeric value following G92S during the constant surface speed control.

G92S _____ ;

If the spindle speed becomes higher than the programmed value during the constant surface speed control, it is clamped to the maximum spindle speed.

NOTE 1 When power is turned on, spindle speed is not clamped.

NOTE 2 The spindle speed is clamped to the maximum speed in the G96 mode only. It is not clamped in G97 mode.

NOTE 3 G92S0; means the spindle speed is clamped to 0 rpm.

NOTE 4 The setting of SMX in parameter 2402 depends on whether the S code specified in the same block as G92 is regarded as a maximum spindle speed command.

10.2.3 Rapid traverse (G00) in constant surface speed control

In a rapid traverse block specified by G00, the constant surface speed control is not made by calculating the surface speed to a transient change of the tool position, but is made by calculating the surface speed based on the position at the end point of the rapid traverse block, on the condition that cutting is not executed at rapid traverse.

NOTE When RSC of parameter 5602 is 0, even when G00 is specified the surface speed is calculated according to the tool position that changes every moment.

10.2.4 Constant surface speed control axis

The constant surface speed controlled axis is selectable by a parameter (No. 5640) a programmed command.

When it is specified by the program, specify the axis for the constant surface speed control by address P of the block of G96. When address P is omitted, or if P0 is specified, an axis set by parameter (No. 5640) becomes effective.

Command : G96Px ;
P1 : X axis, P2 : Y axis, P3 : Z axis, P4 : U axis,
P5 : V axis, P6 : W axis, P7 : A axis, P8 : B axis, P9 : C axis

10.3 Spindle Positioning (Indexing)

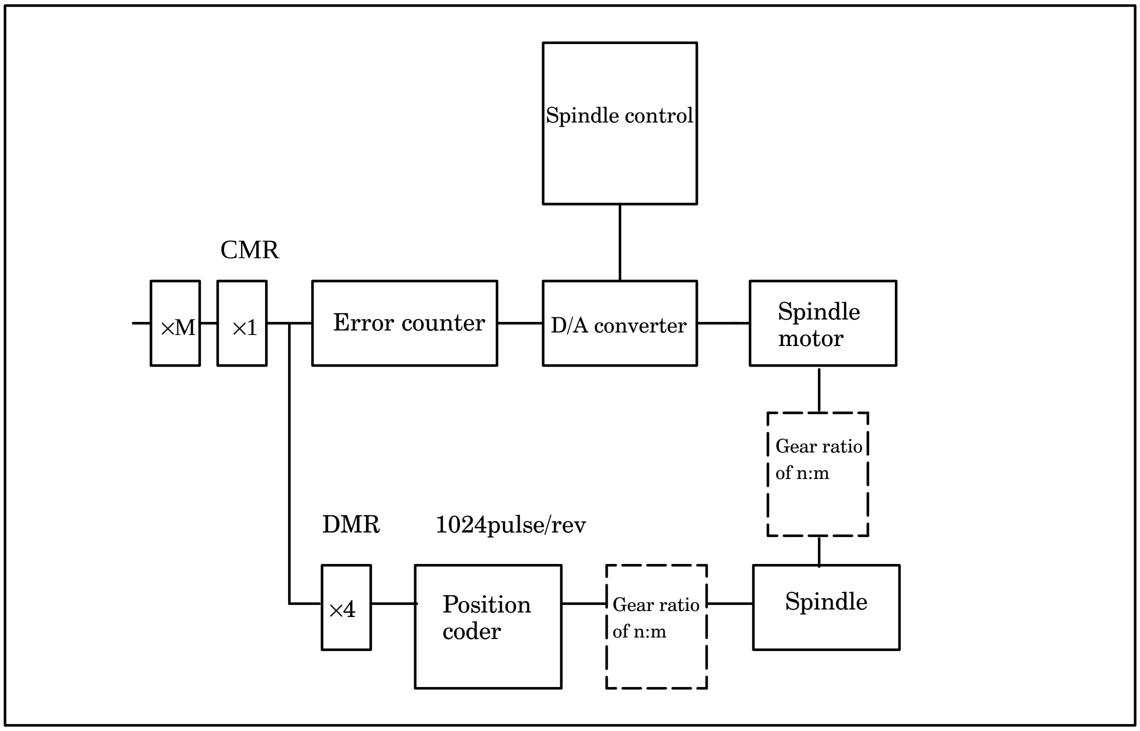
Turning is described as follows: The spindle connected to the spindle motor is rotated at a certain speed. As a result, the workpiece fixed to the spindle is rotated, and turning is performed.

The spindle positioning function is described as follows: The spindle connected to the spindle motor is rotated up to a certain angle to position the workpiece fixed to the spindle at a certain angle.

The use of this function enables a workpiece to be drilled at any position.

The following figure shows the spindle control system.

When the spindle is rotated for turning (referred to hereinafter as spindle rotation mode), the value corresponding to the spindle speed is input from the spindle control unit to the D/A converter. When the spindle is positioned (referred to hereinafter as spindle positioning mode), the specified travel distance is input to an error counter and converted into the velocity command for the spindle motor via the D/A converter. The position of the spindle is then detected by the position coder installed in the spindle.



Gear ratio (1:p) of spindle to position coder	Least command increment (detection unit: deg)
1:1	0.088 (1×360 / 4096)
1:2	0.176 (2×360 / 4096)
1:4	0.352 (4×360 / 4096)
1:8	0.703 (8×360 / 4096)

The operations performed by the spindle positioning function are classified as follows:

(1) Operation from canceling spindle rotation mode to changing to spindle positioning mode

(2) Positioning the spindle in spindle positioning mode

(3) Operation from canceling spindle positioning mode to changing to spindle rotation mode

10.3.1 Programming methods

(1) Least command increment

$$\frac{360}{4096} \fallingdotseq 0.088\text{deg}$$

(2) Programming methods

There are two programming methods: indexing at an arbitrary angle, and indexing at a semi–fixed angle.

(a) Indexing at a semi–fixed angle with an M code

This is specified with a two–digit numeric value following address M. Up to six codes of Mα to M (α+5) can be specified. α is preset with parameter 5682. The following table shows the correspondence between Mα to M(α+5) and indexing angles. β is preset with parameter 5683.

M code	Indexing angle	**Example** When β = 30x
Mα	β	30_
M (α + 1)	2β	60_
M (α + 2)	3β	90_
M (α + 3)	4β	120_
M (α + 4)	5β	150_
M (α + 5)	6β	180_

(b) Indexing at an arbitrary angle with address C

This is specified with a several–digit numeric value following address C. A plus (+) or minus (–) sign may precede the numeric value. Absolute and incremental programming can be performed.

Specify address C in the G00 mode.

Example C–1000;

(i) Least input increment

0.001 deg

(ii) Maximum value

±99999.999 deg

(iii) Input of a numeric value with a decimal point

A numeric value with a decimal point can be input. The decimal point should be specified at the Degree–Unit position.

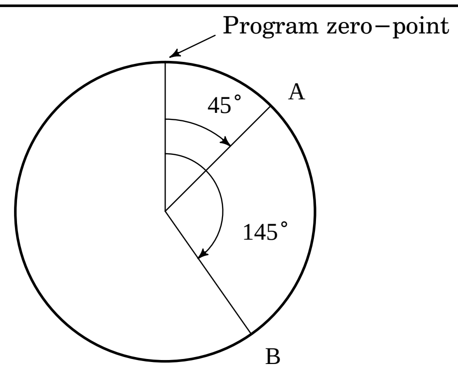
Example C35.0 = C35 degrees

(3) Program zero–point

The position after orientation (see Subsection 10.3.2) is assumed to be a program zero–point. The program zero–point can, however, be changed by setting a coordinate system (G92).

(4) Absolute and incremental programming

Incremental programming is always performed in programming method (a) using an M code. Absolute and incremental programming is enabled in programming method (b).



(5) Direction of rotation

Whether the direction of rotation is clockwise or counterclockwise is specified with IDM, a bit of parameter 5605.

10.3.2 Orientation

Orientation must be performed before:

– the spindle is positioned (indexed) for the first time after the spindle is used in normal machining.

– the positioning of the spindle is suspended.

The orientation is performed by automatic reference position return (G28) of low–speed type in a grid system.

With the reference position return for spindle positioning (indexing), the grid is automatically created by the position coder connected to the spindle. Therefore, the deceleration signal for reference position return is not required.

A grid shift function enables the orientation position to be shifted within the range of +180 deg.

The orientation is performed as follows: The machine moves at the rapid traverse feedrate specified with parameter 1420. Next a single rotation signal is detected from the position coder. Then the machine is automatically decelerated to the FL speed specified with parameter 1425. Thus, the machine returns to the reference position.

The orientation command is issued using the M code specified by parameter 5680.

When the orientation is completed, the point is assumed to be the program zero–point for the spindle (C–axis), and the coordinate of the point is set to 0.

WARNING Set PLZx, bit 3 of parameter 1005, to 1. The setting of PLZx specifies whether the workpiece coordinate system is always preset when the machine is manually returned to the reference position.

10.3.3 Detection unit for spindle positioning

The detection unit varies according to the spindle to position coder gear ratio.

Spindle to position coder gear ratio	Detection unit (deg/pulse)
1 :1	0.088 ($= 1 \times \frac{360}{4096}$)
1 :2	0.176 ($= 2 \times \frac{360}{4096}$)
1 :4	0.352 ($= 4 \times \frac{360}{4096}$)
1 :8	0.703 ($= 8 \times \frac{360}{4096}$)

NOTE The spindle to position coder gear ratio is specified with parameter 5610.

10.3.4 Feedrate during spindle positioning

(1) Feedrate during spindle positioning

The feedrate during spindle positioning is set to the rapid traverse feedrate specified in parameter 1420. Linear acceleration/deceleration is applied to the feedrate.

And an override of 100%, 50%, F1%, or F0% can be applied. F1 and F0 are specified with parameters 1412 and 1421, respectively.

(2) Feedrate during orientation

The orientation is performed as follows: The machine moves at the rapid traverse feedrate specified with parameter 1420. Next a single rotation signal is detected from the position coder. Then the machine is automatically decelerated to the FL speed specified with parameter 1425. Thus, the machine returns to the reference position.

If the rapid traverse feedrate during orientation does not satisfy expression 1, the single rotation signal cannot be detected from the position coder. The orientation is therefore performed abnormally.

$$\left\{ \left[\frac{F}{60 \cdot G} \right] / 0.088 \times P \right\} \geqq \underset{\text{(pulse)}}{128} \qquad \textbf{Expression 1}$$

F : Feedrate (deg/min)
G : Loop gain (–sec)
P : Spindle to position coder gear ratio

When the machine moves at the rapid traverse feedrate, the positional deviation for the servo (diagnosis 3000) must be at least 128 pulses.

The rapid traverse feedrate specified with parameter 1420 and the FL speed specified with parameter 1425 must not exceed the speeds that make the positional deviation for the servo exceed 1024 pulses under the condition of expression 1.

10.3.5 Canceling the spindle positioning mode

To change spindle indexing to normal spindle rotation, a specific M code must be specified with parameter 5681.

Specifying RESET, when bit 4 of parameter 5605 is 0, enables the spindle positioning mode to be canceled.

When a servo alarm occurs during spindle positioning (indexing), the spindle positioning mode is canceled.

WARNING 1 A feed hold function is ineffective during spindle positioning.

WARNING 2 The dry run function, machine lock function, and auxiliary function lock function are ineffective during spindle positioning.

NOTE 1 Use a single block when specifying spindle positioning. The move commands for X– and Z–axes cannot be specified in the same block.

NOTE 2 The spindle cannot be positioned manually.

NOTE 3 The program or block cannot be restarted for spindle indexing. To restart it, use MDI commands.

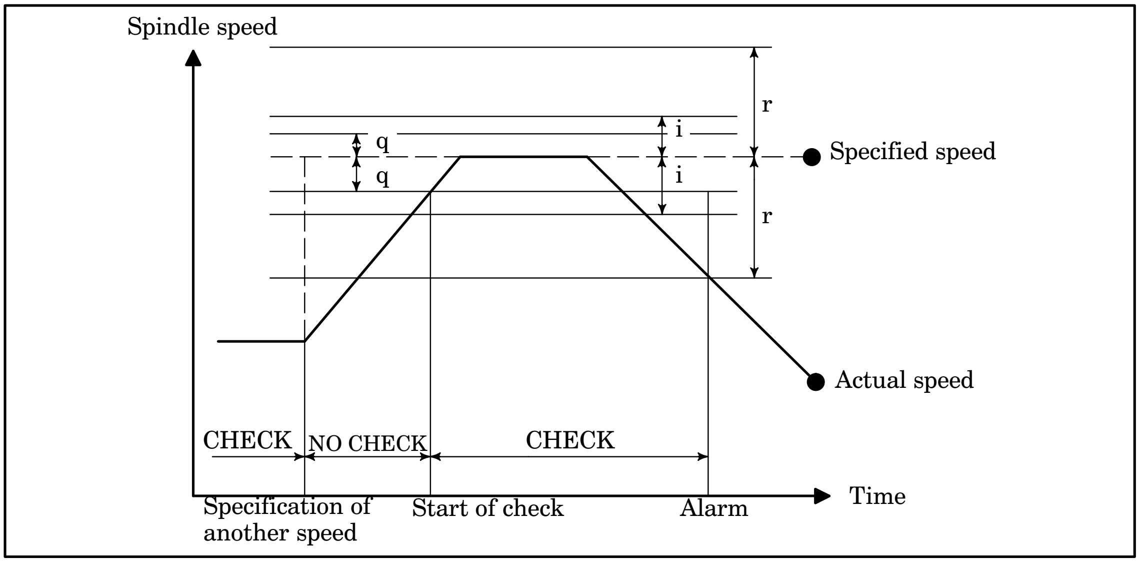
NOTE 4 If an emergency stop is applied during spindle positioning, spindle positioning is stopped. Retry spindle positioning from orientation.

10.4 Detection of fluctuation in spindle speed (G25 and G26)

Burning of the spindle can be prevented by reporting an alarm to the machine when the detected fluctuation is spindle speed is out of the allowable range when the spindle speed is checked.

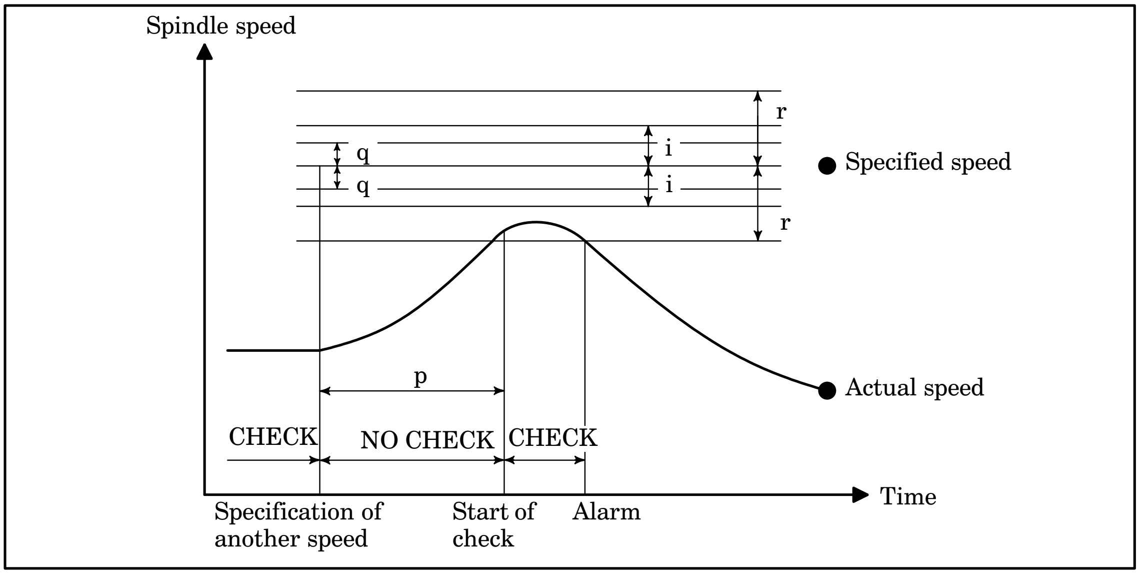
(1) Detection method

(a) When an alarm occurs after a specified spindle speed is reached



Specified spindle speed: (Spindle speed specified with S and four digits) x (spindle override)
Actual speed: Speed detected by the position coder
q : Allowable ratio for start the checking of a spindle speed
r : Variable ratio causing an alarm
i : Variable range causing an alarm

(b) When an alarm occurs before a specified spindle speed is reached

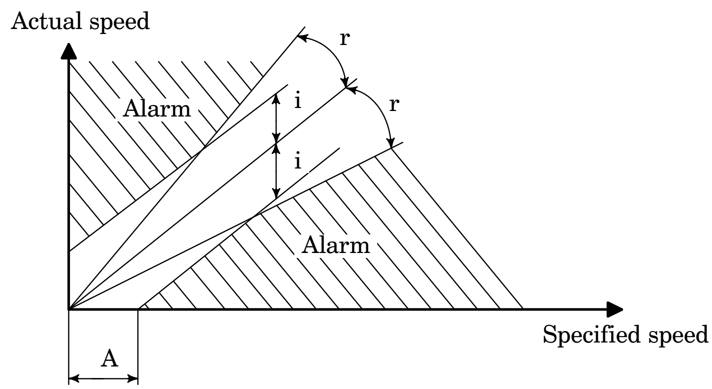


p: Time from fluctuation in specified speed occurs to start of check

NOTE The speed is not checked after the specified speed fluctuates until one of the following conditions is satisfied:
(i) The actual speed reaches the range of the specified error for the specified speed. (Allowable ratio enabling start of the check: q)
(ii) The specified time or longer elapses after the specified speed fluctuates. (Timer specifying start of the check: p)

(2) Detecting an alarm

An alarm occurs if the fluctuation between the specified speed and the actual speed exceeds the specified value. The following data is obtained when the fluctuation is converted into the speed of the spindle motor:



r : Variable ratio causing an alarm (BNLMT)

To be specified as a percentage of the specified speed

i : Variable range causing an alarm (BNWDT)

To be specified with the speed variable from the specified speed

When both r and i are exceeded, an alarm occurs. However, an alarm also occurs when the actual speed of 0 rpm continues for at least one second in area A.

(3) Specifying whether the spindle speed is checked

Whether the spindle speed is checked can be specified by the following G codes.

(G code group 18)
G25: Not checked
G26: Checked

When the power is turned on and in the reset state, G26 is issued. Once G26 is issued, it is effective till G25 is issued.

The timer specifying start of the check (p), allowable ratio enabling start of the check (q), variable ratio causing an alarm (r), and variable range causing an alarm (i) can be specified as parameters shown in item 3, and can also be changed by a program.

Once the values are changed, they are effective till the command is changed. They are not changed when the power is removed.

Specify the commands for the parameters in the same block as the G26 command as follows:

G26 P$\underline{p}$ Q$\underline{q}$ R$\underline{r}$ I$\underline{i}$;

NOTE 1 The block must begin with G26.
NOTE 2 Specify either G25 or G26 alone in a block.

(4) Actual speed

The actual speed is calculated using the number of pulses from the position coder that is counted every 65 ms.

(5) Specified speed

The specified speed for the Series 3/6 interface refers to the speed specified with S and four digits in the G97 mode (constant surface speed control off) or the spindle speed to be calculated in the G96 mode (constant surface speed control on). The following conditions must be also considered:

i) Spindle override

ii) Maximum speed specified with G92S__ in the G96 mode

iii) Upper limit of the output to the spindle motor specified in parameter 5619

iv) Minimum spindle speed for each gear specified in parameter 5641

v) Minimum clamp speed of the spindle speed (0 to 4095) (parameter 5618)

Maximum clamp speed of the spindle speed (0 to 4095) (parameter 5619)

The speed calculated according to the voltage signals (RI0 to RI12) specifying the spindle motor is specified for the BMI interface.

(6) Conditions for no check of the spindle speed

The spindle speed is not checked under the following conditions:

i) G25 mode

ii) A spindle stop (*SSTP) signal is low in the 3/6 interface. Or, the voltage signals specifying the spindle motor (RI0 to RI12) are all low in the BMI interface.

iii) Searching to restart the program

iv) An S indirect signal (SIND) is high in the 3/6 interface.

v) A gear shift signal (GST) is high in the 3/6 interface.

(7) Detecting an alarm

When an alarm is detected, a spindle alarm signal (SPALM) is sent to the machine.

An overtravel alarm signal (OTALM) is also sent to the machine. In this case, the machine stops at a single block during automatic operation.

When the alarm occurs, ”OT117 SPINDL OVERHEAT” is displayed.

(8) Canceling an alarm

To cancel an alarm, correct the cause of the alarm, and reset the machine.

(9) Change in the specified speed

As described in the note in item (1), the spindle speed is not checked after the specified speed fluctuates until one of two conditions is satisfied. The specified speed frequently fluctuates while the machine moves along the X–axis in constant surface speed control mode. In this case, care must be taken.

(10) Setting parameters in the program

G26 I$\underline{i}$ P$\underline{p}$ Q$\underline{q}$ R$\underline{r}$;

The following parameters can be set or changed using the above command:

Data in the above code	Parameter	Description	Unit	Valid range
q	5701	Allowable error range in which the actual speed is assumed to reach the specified speed	%	1 to 50
r	5702	Variable ratio causing a spindle fluctuation alarm	%	1 to 50
i	5721	Variable range causing a spindle fluctuation alarm	rpm	0 to 32767
p	5722	Delay timer for the time from fluctuation is specified speed occurs to start of check	msec	0 to 32767

NOTE Use a multiple of 64 ms.

11. TOOL FUNCTION (T FUNCTION)

11.1 Tool Selection Command

By specifying an up to 8–digit numerical value following address T, tools are selected on the machine.

One T code can be commanded in a block. Refer to the machine tool builder's manual for the number of digits commandable with address T and the correspondence between the T codes and machine operations.

When a move command and a T code are specified in the same block, the commands are executed in one of the following two ways:

(i) Simultaneous execution of the move command and T function commands.

(ii) Executing T function commands upon completion of move command execution.

The selection of either (i) or (ii) depends on the machine tool builder's specifications. Refer to the manual issued by the machine tool builder for details.

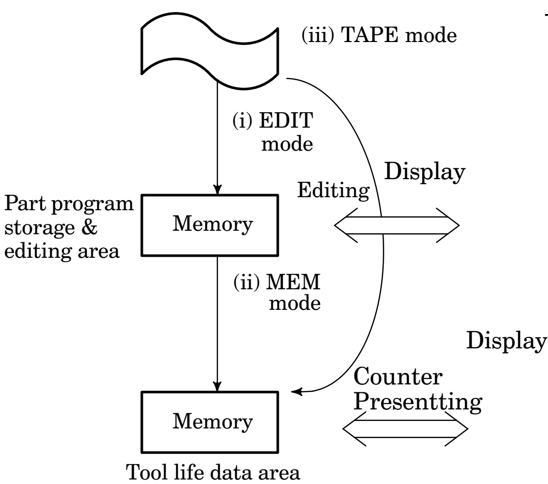
11.2 Tool Life Management

Tools are classified into various groups, with the tool life (time or frequency of use) for each group being specified. The function of accumulating the tool life of each group in use and selecting and using the next tool previously sequenced in the same group, is called the tool life management function.

1) Setting the tool life data

Tools used sequentially in each group and their respective tool life are previously set by the following format tape in the CNC equipment.

Tape format	Meaning
O–––– ;	Program number
G10L3 ;	Start of setting tool life data
P––– L–––– ;	P followed by group No. (1 to 128) L followed by tool life (1 to 9999) (Note 1)
T–––– H–– D–– ; T–––– H–– D–– ; : : :	(1) (2) (n) T followed by tool No. (Note 2) H followed by tool length offset No. (Note 3) D followed by cutter compensation No. (Note 3) Tools are selected in the sequence of (1) through (2) to (N).
P––– L–––– T–––– H–– D–– ; T–––– H–– D–– ; : : :	Data for the next group
G11 ;	End of setting tool life data
M02 (M30) ;	End of program



The setting operation is as follows.

(i) Insert the above tape into the tape reader and load the tape in the EDIT mode, as with an ordinary CNC tape.

The program will be registered in the part program memory and will be made ready for display and editing.

(ii) Perform a cycle start operation in the MEM mode to run the program. The data will be stored in the tool life data area of the memory; at the same time, the already existing tool life data of all groups will be cancelled and the life counters will be cleared. Data once stored is not erased by turning the power off.

(iii) Executing a cycle start operation in the TAPE mode instead of the operation of (i), stores the program contents directly from the tape onto the tool life data area. In this case, however, display and editing cannot be done as in (i).

NOTE 1 Whether tool life is to be indicated by time (minutes) or by frequency, it is set by a parameter LTM (No. 7400 #3) .

Maximum value of tool life is either 4,300 minutes or 9,999 times.

NOTE 2 Tool number is up to 4 digits.

NOTE 3 H99 and D99 are not registered.

NOTE 4 Using a parameter, specify one of the following four combinations of the number of groups that can be registered and the number of the tools that can be registered per group.

	Number of groups	Number of tools
(1)	16	16
(2)	32	8
(3)	64	4
(4)	128	2

When the optional tool life management for 512 sets of tools is not specified

	Number of groups	Number of tools
(1)	64	32
(2)	128	16
(3)	256	8
(4)	512	4

When the optional tool life management for 512 sets of tools is specified

	Number of groups	Number of tools
(1)	128	32
(2)	256	16
(3)	512	8
(4)	1024	4

When the optional tool life management for 1024 sets of tools is specified

After this parameter is changed, specify the tool life management data again with the G10 L3; command.

NOTE 5 When the optional tool life management for 512 or 1024 sets of tools is not specified, the D and H codes that can be specified as tool life management data do not exceed 254 if the number of the offset areas in memory is greater than 254. When the optional tool life management for 512 or 1024 sets of tools is specified, all D and H codes (offset numbers) can be used.

NOTE 6 Codes H and D, when not in use, can be omitted.

NOTE 7 The same tool No. can appear at any frequency and at any place in the set data.

The following is a concrete example of the tape format.

```
O0001 ;
G10L3 ;
P001L0150 ;          }
T0011H02D13 ;        }
T0132H05D08 ;        }  Data of group 1
T0068H14D16 ;        }
P002L1400 ;          }
T006H15D07 ;         }
T02425D04 ;          }  Data of group 2
T0134H17D03 ;        }
T0074H08D21 ;        }
P003L0700 ;          }
T0012H14D08 ;        }  Data of group 3
T0202H22D02 ;        }
G11 ;
M02 ;
```

NOTE 8 Group No. specified by P need not be consecutive. All registerable groups need not be set, either.

NOTE 9 When this function is effective, the usable length of the storage tape is reduced by 6 m. When the number of the sets for tool life management is 512, the usable length of the storage tape is reduced by 45 m. When the number of sets for tool life management is 1024, the usable length of the storage tape is reduced by 90 m.

NOTE 10 The tool compensation number that can be registered as tool life management data must be within the range of the number of the tools which are to be compensated and which can be used in the CNC unit.

NOTE 11 When the optional tool life management for 1024 sets is specified, the valid ranges for the tool number and H and D codes are as follows:

– Tool number : 0 to 9999 (up to four digits)

– H and D : 0 to 999

When the option for tool offset selection by a T code is used, the following ranges are valid:

– Tool number: 0 to 99999999 (up to eight digits)
– H and D: Not used

NOTE 12 When the optional tool life management for 1024 sets is specified, up to 1024 sets are also supported for the PMC window read/write function. To use 1024 sets with the PMC window read/write function, however, the new format must be specified (by setting parameter No.7401#4=1).

2) Specifying tool life data, etc. in a machining program

In a machining program, tool groups, etc. are specified by using the T codes as follows.

Tape format	Meaning
: : T∇∇∇∇ ;	Group No. of tools which are to be used after the next M06 command + tool life management ignore No. (Note 1)
: : T06T □□□□ ;	Terminates the tool specified by ¢¢¢¢ (Note 2)and begins to use the tool specified by ∇∇∇∇
: : H□□ ;	99:Makes the tool offset specified by group No.effective. 00:Cancels tool length offset.
: : D○○ ;	99:Makes cutter compensation, specified by group No. effective. 00:Cancels cutter compensation.
: : T ΔΔΔΔ ;	Uses a tool specified by ΔΔΔΔ after the next M06 command.
: : M06T∇∇∇∇ ;	Terminates the tools of ∇∇∇∇ and begins to use the tool of ΔΔΔΔ
: : M02 (M30)	End of the machining program.

NOTE 1 From T0000 to T ΔΔΔΔ, stipulated by the tool life management ignore No.ΔΔΔΔ , are handled as ordinary T code commands, and no tool life management is performed.

When the T code of ΔΔΔΔ plus the group No. is designated, tool life management is executed for the related group. The value of the tool life management ignore No. is set by a parameter.

When the value is 100, for example, from T0000 to T0100 are output as ordinary T codes; and when the T0101 is designated, a T code of a tool, which has not reached its life end among the tools of group 1, is output.

NOTE 2 The above tape format uses the tool return No. command method (TYPE A), which requires a tool return command at the time of tool change. (TYPE B) and (TYPE C) are also available.

Concrete examples of the tape format when the tool life management ignore No. is 100 are described on each type.

Type A

Tape format	Meaning
T101 ;	Set the tool of group 01 to the waiting position
M06 (T □□□□);	Set the tool of group 01 to the spindle. (Note 1)
T102 ;	Set the tool of group 02 to the waiting position.
⋮	
⋮	Machining using the tool of group 01.
⋮	
M06T101 ;	Set the tool of group 02 to the spindle.
	Load the tool of group 01 to the magazine.
T103 ;	Set the tool of group 03 to the waiting position.
⋮	
⋮	Machining using the tool of group 02.
⋮	
M06T102 ;	Bring the tool of group 03 to the spindle.
⋮	Load the tool of group 02 to the magazine.
⋮	
⋮	Machining using the tool of group 03.
G43H99 ;	Uses the tool length compensation value of the tool of group 03.
⋮	
⋮	
⋮	
G41H99 ;	Uses the cutter compensation value of the tool of group 03.
⋮	
⋮	
⋮	
D00 ;	Cutter compensation cancel.
⋮	
⋮	
H00 ;	Tool length compensation cancel.

NOTE The return tool may not be specified. If a T code is specified in the same block as M06, the T code shall be the tool number of the tool to be loaded in the magazine. The tool number is of the tool attached to the spindle at present. If the tool is under the tool life management, specify the tool group of the tool. When the tool group must be specified, if the tool group specified here (group of return tool) and the tool group currently under management are different each other, an alarm is generated. (No alarm is issued if parameter ABT (No. 7400#6) is set).

Type B

Tape format	Meaning
T01 ;	Set the tool of group 01 to the waiting position.
M06T102 ;	Set the tool of group 01 to the spindle.
⋮	Set the tool of group 02 to the waiting position.
⋮	
⋮	Machining using the tool of group 01.
⋮	
M06T103 ;	Set the tool of group 02 to the spindle.
⋮	Load the tool of group 01 to the magazine.
⋮	Load the tool of group 03 to the waiting position.
⋮	
⋮	Machining using the tool of group 02.
⋮	
G43H99 ;	Uses the tool length compensation value of the tool of group 02.
⋮	
⋮	
⋮	
G41D99 ;	Uses the cutter compensation value of the tool of group 02.
⋮	
⋮	
⋮	
D00 ;	Cutter compensation cancel.
⋮	
⋮	
⋮	
H00 ;	Tool length compensation cancel.
⋮	
⋮	
⋮	
M06T104 ;	Bring the tool of group 03 to the spindle.
⋮	Load the tool of group 02 to the magazine.
⋮	Bring the tool of group 04 to the waiting position.
⋮	
⋮	Machining using the tool of group 03.
⋮	

Type C

Tape format	Meaning
M06T101 ;	Bring the tool of group 01 to the waiting position.
M06T102 ;	Bring the tool of group 01 to the spindle.
⋮	Bring the tool of group 02 to the waiting position.
⋮	
⋮	Machining using the tool of group 01.
⋮	
M06T103 ;	Bring the tool of group 02 to the spindle.
⋮	Bring the tool of group 01 to the magazine.
⋮	Bring the tool of group 03 to the waiting position.
⋮	
⋮	Machining using the tool of group 02.
⋮	
G43H99 ;	Uses the length compensation value of the tool of group 02.
⋮	
⋮	
⋮	
G41D99 ;	Uses the tool cutter compensation value of the tool of group 02.
⋮	
⋮	
⋮	
D00 ;	Cutter compensation cancel.
⋮	
⋮	
⋮	
H00 ;	Tool length compensation cancel.
⋮	
⋮	

3) Counting tool life

Tool life is counted by time or by frequency. The life count is executed every group and the contents of life counter is not cleared by turning off power.

a) When tool life is specified by time (minutes)

In this case, a T code in the machining program specifies tool life management group and the time during which the tool is used in the cutting mode from the M06 specification to the next M06 specification is counted at certain intervals (four seconds).

The time for single block stop, feed hold, rapid traverse, dwell, machine lock, time to wait for FIN signal etc., is disregarded.

b) When tool life is specified by frequency

The counter for groups of tools that were used is increased by one, every process from the time that a cycle start operation is performed until the time that M02 or M30 is commanded and the CNC is reset. Even if the command for the same group is given a number of times in one process, the counter increase stays at one.

NOTE When having executed M02 or M30 with life specified by the frequency of use, input the External Reset (ERS) signal, Reset & Rewind (RRW) or Reset to the CNC.

11.3 Enhanced Tool Life Management

The following functions are added to the ordinary tool life management functions (including those 512 sets), making it much easier to use the machining center.

(1) Setting the tool life management data by the program for each tool group

(2) Setting the life count type by the program for each tool group

(3) Displaying the tool life management data in detail

(4) Editing the life management data

(5) Overriding the life count

See Section 10.11.10 of Part III for (3), (4), and (5).

11.3.1 Setting the tool life management data for each tool group

Tool life management data for a specified tool group can be added, changed, and deleted without affecting the tool life management data for the other tool groups.

(1) Addition and change (G10L3 P1)

P1 is specified for the same block as for G10L3.

```
G10L3P1 ;
P––L–– ;
T––H––D–– ;
T––H––D–– ;
P––L–– ;
T––H––D–– ;
T––H––D–– ;
G11 ;
```

G10L3P1	:	The beginning of an addition or change
P	:	Group number (1 to MAX (*))
L	:	Life value (1 to 4300 minutes or 1 to 9999 times)
T	:	Tool number
H	:	H–code
D	:	D–code
G11	:	The end of an addition or change for the group

If a specified group already exists, a change occurs.

Otherwise, an addition occurs.

NOTE 1 MAX stands for the highest group number specified by parameters (GS1, GS2, No. 7400).

NOTE 2 If P1 is omitted from G10L3P1, the conventional tool life management data is set up (after all groups are cleared).

(2) Deletion (G10L3P2)

P2 is specified for the same block as for G10L3.

```
G10L3P2 ;
P––
P––
P––
P––
G11 ;
```

G10L3P2	:	The beginning of deletion of groups
P	:	Group number
G11	:	The end of deletion

Groups specified (P––) between G10L3P2 and G11 are deleted.

11.3.2 Setting the life count type for each group (P––L–––Qn)

Qn (life count type) is specified for the same block as for P (group number) and L (life value).

```
G10L3 or G10L3P1;
P––L––Qn ;
T––H––D–– ;
T––H––D–– ;
P––L––Qn ;
T––H––D–– ;
T––H––D–– ;
G11 ;
```

Qn : Life count type (Q1—life count in number of times the tool was used, Q2—life count in time)

NOTE	If Q is omitted, a life count type is set up by a parameter (LTM, No. 7400#3)

11.3.3 Selecting tools

Tool life management is performed in such a way that when the expected life of a tool expires (”dead” tool), another tool is selected. A ”living” tool is selected by searching forward through the tools in use. When all tools up to the last one are found dead, a search returns to the first tool. If there is no living tool among all tools in use, the tool used most recently is selected. With the tools described below, for example, tool 0700 is selected when the expected life of tool 6279 expires.

```
GROUP 001:    LIFE 3000   COUNT 0200

*1186         *3601       0700      @6279
```

11.3.4 Life count

The life of a tool is measured in time or number of times the tool was used according to the life count type specified for each group. If the life count is in time, an external signal can be used to override the life count.

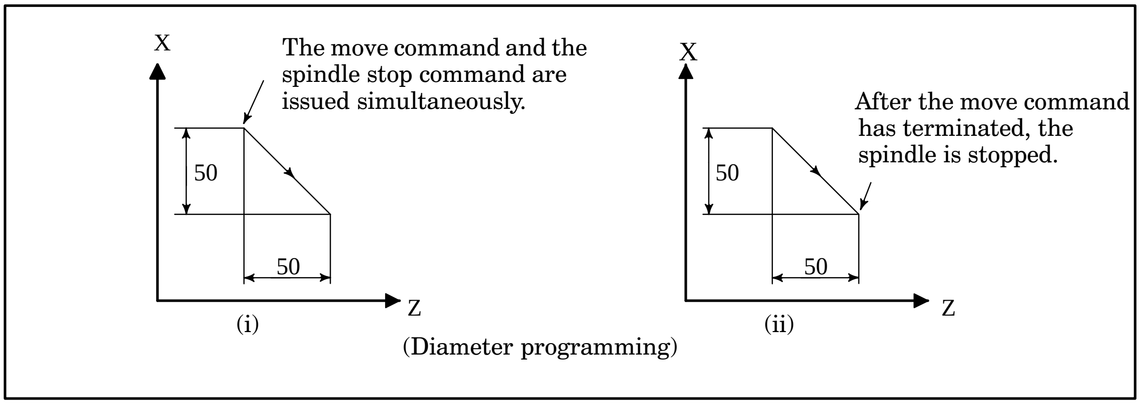
12.MISCELLANEOUS FUNCTION (M, B FUNCTIONS)

When a move command and M or B codes are specified in the same block, the commands are executed in one of the following two ways:

i) Simultaneous execution of the move command and M or B function commands.

ii) Executing M and B function commands upon completion of move command execution.

Example N1G91G01X–100.0Z50.0M05; (The spindle is stopped.)



The selection of either sequence depends on the machine tool builder's specification. Refer to the manual issued by the machine tool builder for details.

12.1 Miscellaneous Function (M Function)

When a numeral is specified following address M, a code signal and a strobe signal are transmitted. These signals are used for ON/OFF control of a machine function. One M code can be specified in one block. Selection of M codes for functions varies with the machine tool builder.

The following M codes indicate special meaning.

1) M02, M30: End of program

   i) This indicates the end of the main program and is necessary for registration of CNC commands from tape to memory.

   ii) Cycle operation is stopped and the CNC unit is reset. (This differs with the machine tool builder.)

   iii) Only M30

   The CNC type is rewound to the start of the program in both memory and tape operation. However, when using a tape reader without reels, the tape is not rewound. When using a tape reader with reels, the tape returns to the ER(%) code at the start of the tape even if several programs exist.
   (This differs with the machine tool builder. Some machines indicate tape rewind with M02.)

2) M00: Program stop

   Cycle operation is stopped after a block containing M00 is executed. When the program is stopped, all existing modal information remains unchanged as in single block operation. The cycle operation can be restarted by actuating the CNC. (This differs with the machine tool builder.)

3) M01: Optional stop

   Similarly to M00, cycle operation is stopped after a block containing M01 is executed. This code is only effective when the Optional Stop switch on the machine control panel has been pressed.

4) M98: Calling of subprogram

   This code is used to call a subprogram. See section 13.4.1 (2) for details.

5) M99: End of subprogram

   This code indicates the end of a subprogram. M99 execution returns control to the main program. See the subprogram control section for details. (Section 13.4.2 (2))

NOTE 1 The block following M00, M01, M02 and M30, is not read into the input buffer register, if present. Similarly, ten M codes which do not buffer the next block can be set by parameters (Nos. 2411 to 2418). Refer to the machine tool builder’s instruction manual for these M codes.
NOTE 2 The code and strobe signals are not sent to the machine tool side for the M98, M99.
NOTE 3 Except for M98, M99, all M codes are processed by the machine tool. Refer to the machine tool builder’s instruction manual for details.

12.2 Auxiliary Functions

When a number after address B is commanded, a code is output. This code is kept until the next B code is commanded. This function is used for indexing a rotary table on the machine side. A single B code can be commanded in one block. By parameter setting (Data No. 1030), A, C, U, V or W can also be used in place of address B; however, the same address as the control axes cannot be used. Refer to machine tool builder's manual for details.

12.3 Multiple M Commands in a Single Block

So far, one block has been able to contain only one M code.

However, this function allows up to five M codes to be contained in one block.

Up to five M codes specified in a block are simultaneously output to the machine. This means that compared with the conventional method of a single M command in a single block, a shorter cycle time can be realized in machining.

Example

(i) Single M command in a single block

```
M40;
M50;
M60;
G28G91X0Y0Z0;
    :
    :
    :
```

(ii) Multiple M commands in a single block

```
M40M50M60;
G28G91X0Y0Z0;
    :
    :
    :
```

NOTE 1 CNC allows up to five M codes to be specified in one block. However, some M codes cannot be specified at the same time due to mechanical operation restrictions. For example, M42 can be specified only after the mechanical operation of M41 is completed. For detailed information about the mechanical operation restrictions on simultaneous specification of multiple M codes in one block, refer to the handbook of each machine tool builder.

NOTE 2 M00, M01, M02, M30, M98, or M99 must not be specified together with another M code.

NOTE 3 Some M codes other than M00, M01, M02, M30, M98, and M99 cannot be specified together with other M codes; each of those M codes must be specified in a single block. For example, no other M codes can be specified in a block which already contains an M code that causes CNC to perform internal processing in addition to just sending an M code signal; such an M code may be an M code for calling a program number (9001 to 9009) or an M code for stopping the pre–read of the next block (for performing no buffering). Only those M codes that causes CNC to just send an M code signal can be specified in the same block.

NOTE 4 When multiple M codes are specified in the same block, only the M code specified first is displayed on the screen displaying the ACTIVE or LAST command values for program check.

NOTE 5 This function is ineffective for an M code of Manual Numevic Command function.

13.PROGRAM CONFIGURATION

A program consists of the following sections :

(1) Tape start . . . . See Section 13.1.
(2) Leader section . . . . See Section 13.2.
(3) Program start . . . . See Section 13.3.
(4) Program section . . . . See Section 13.4.
(5) Comment section . . . . See Section 13.5.
(6) Program end . . . . See Section 13.6.
(7) Tape end . . . . See Section 13.7.

13.1 Tape Start

The start file mark (tape start) indicates the start of a file that contains NC programs.

The mark is not required when programs are entered using SYSTEM P or ordinary personal computers. The mark is not displayed on the CRT display screen. However, if the file is output, the mark is automatically output at the start of the file.

13.2 Leader Section and Label Skip

Data entered before the programs in a file constitutes a leader section. When a file is read into the CNC system from an I/O device after setting the label skip state by turning on power or reset operation to start machining, the leader section is usually skipped by the label skip function. A leader section generally contains information such as a file header. When a leader section is skipped, even a TV parity check is not made. So a leader section can contain any codes except the EOB code.

NOTE	The label skip function ignores all information until the first EOB (end of block) code is read.

13.3 Program Start

The program start code is to be entered immediately after a leader section, that is, immediately before a program section. This code indicates the start of a program, and is always required to disable the label skip function.

With SYSTEM P or ordinary personal computers, this code can be entered by pressing the return key.

13.4 Program Section

The data entered between a program start code and program end code constitutes a program section. (Note, however, that comment sections described in Section 13.5 are not part of a program section.) A program section contains information for specifying actual machining such as move commands and ON/OFF commands. This part is called a significant information section. On the other hand, a leader section or a comment section is called a nonsignificant section.

A program section consists of several blocks. (See section 13.4.3 for details of blocks.) A block consists of words, and is distinguished from another block by an EOB code.

NOTE TV check (vertical parity check along tape)
A parity check is made for a block on input tape horizontally. If the number of characters in one block (starting with the code immediately after an EOB and ending with the next EOB) is odd, an alarm (No. 002) is output. No TV check is made only for those parts that are skipped by the label skip function. A comment section enclosed in parentheses is also subject to TV check to count the number of characters. The TV check function can be enabled or disabled by setting on the MDI unit. (See Section 10.3 in Part III.)

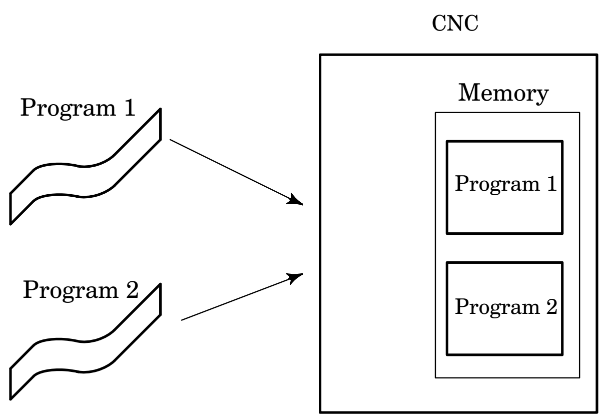
13.4.1 Main program and subprogram

(1) Main program

A program is divided into a main program and subprogram. Normally, the CNC operates according to the main program. However, when a command calling a subprogram is encountered in the main program, control is passed to the subprogram. When a command specifying a return to the main program is encountered in a subprogram, control is returned to the main program.

The CNC memory can hold up to 100 main programs and subprograms in total. One main program can be selected from these main programs to operate the CNC machine tool.

NOTE The number of programs to be registered can be specified as 400 or 1000 with an optional function.



See Section 9.2 in Part III for the methods of registering and selecting programs.

(2) Subprogram

If a program contains a fixed sequence or frequently repeated pattern, such a sequence or pattern can be stored as a subprogram in memory to simplify the program.

A subprogram can be called in tape mode or memory mode. A called subprogram can also call another subprogram.

When the main program calls a subprogram, it is regarded as a one–level subprogram call. Thus, subprogram calls can be nested up to eight levels as shown below. (II–17.1.10)

A single call command can call a subprogram repeatedly. One call command can call a subprogram up to 9999 times.

(a) Subprogram creation

A subprogram must be created in the following format :

```
O□□□□ ; subprogram–number
_ _ _ _ _ _ _ _ _ _ _ _ _ _   ;  ┐
_ _ _ _ _ _ _ _ _ _ _ _ _ _   ;  │
            .                    │  Subprogram
            .                    │
            .                    │
_ _ _ _ _ _ _ _ _ _ _ _ _ _   ;  ┘
 M99;
```

A subprogram must start with a subprogram number that follows O (or : optionally in the case of ISO). A subprogram must end with M99. M99 need not constitute a separate block as indicated below.

Example X100.0Y100.0M99;

See Section 9.2 in Part III for the method of subprogram registration.

NOTE For tape compatibility with I/O devices such as SYSTEM P, Nxxxx can be used instead of a subprogram number O (:) in the first block. in this case, a sequence number after N is registered as a subprogram number.

(b) Subprogram execution

A subprogram is called from a main program or parent subprogram for execution. A subprogram is called as shown below.

M98 P ……………… L …………… ;

Number of times the subprogram is called in succession

Subprogram number

When L is omitted, the subprogram is called once.

Example M98P1002L5;
This command specifies calling subprogram 1002 five times in succession.
The M98P__L__command can be specified in the same block as the move command.

Example X1000. 0M98P1200 ;
This example calls the subprogram (number 1200) after an X movement.

Example Execution sequence of subprograms called from a main program

A subprogram can call another subprogram in the same way as a main program calls a subprogram.

NOTE 1 The M98 and M99 signals are not output to the machine tool.
NOTE 2 If the subprogram number specified by address P cannot be found, an alarm (PS76) is output.
NOTE 3 The program is not stopped in the block containing M98P ;M99; in the single block stop mode. If the block for M98 and M99 contains addresses other than O, N, P, and L, the program is stopped in the block.
NOTE 4 When M98P__ is specified in the canned cycle mode or in the G04 block, P__ can be used to specify a dwell. In this case, the M98P__ block must be specified independently as shown in the example below:

Example G82X_Y_Z_R_M98P_;
↓
G82X_Y_Z_R_;
M98P_;

(c) Special usage

Special usage is possible as described below.

(i) If P is used to specify a sequence number when a subprogram is terminated, control does not return to the block after the calling block, but returns to the block with the sequence number specified by P. Note, however, that P is ignored if the main program is operating in a mode other than memory operation mode.

This method consumes a much longer time than the normal return method to return to the main program.

```
Main program                       Subprogram
N0010________ ;                    O1010          ;
N0020________ ;                    N1020________ ;
N0030________ ;                    N1030________ ;
N0040 M98 P1010 ;                  N1040________ ;
N0050________ ;                    N1050________ ;
N0060________ ;                    N1060________ ;
N0070________ ;                    N1070 M99 P0070 ;
```

(ii) If M99 is executed in a main program, control returns to the start of the main program. For example, M99 can be executed by placing /M99 ; at an appropriate location of the main program and setting off the optional block skip function when executing the main program. When M99 is executed, control returns to the start of the main program, then execution is repeated starting at the head of the main program.

Execution is repeated while the optional block skip function is set off. If the optional block skip function is set on, the /M99 ; block is skipped ; control is passed to the next block for continued execution.

If/M99Pn ; is specified, control returns not to the start of the main program, but to sequence number n. In this case, a longer time is required to return to sequence number n.

```
                   N0010__________ ;
Optional           N0020__________ ;
block skip         N0030__________ ;
off                N0040__________ ;
                   N0050__________ ;
                   N0060__________ ;
                   /N0070M99P0030  ;      Optional
                   N0080__________ ;      block skip
                   N0090M02;       ;      on
```

(iii) A subprogram can be executed just like a main program by searching for the start of the subprogram with the MDI.

(See Section 5.2.1 and 5.2.2 in Part III for information about search operation.)
In this case, if a block containing M99 is executed, control returns to the start of the subprogram for repeated execution. If a block containing M99Pn is executed, control returns to the block with sequence number n in the subprogram for repeated execution. To terminate this program, a block containing /M02 ; or /M30 ; must be placed at an appropriate location, and the optional block switch must be set to off ; this switch is to be set to on first.

```
N1040__________ ;
N1050__________ ;
N1060__________;
/N1070M02;
N1080M99P1050   ;
```

(iv) M99Lα ;

This command forcibly sets the value of L to α while the subprogram is called. L specifies the number of times the subprogram is called.

If the optional block skip function is made ineffective, the number of times the subprogram is called is set to 0, and control is returned to the main program.

13.4.2 Program number

With Series 15–B, multiple programs can be stored in memory. To distinguish among these programs, a program number consisting of address O followed by a four–digit number is assigned to each program.

A program starts with a program number and ends with M02 ;, M03 ;, or M99 ;,M02 ; or M30 ; indicates the end of a main program. M99 ; indicates the end of a subprogram.

NOTE 1 In ISO code, the colon (:) can be used instead of O.
NOTE 2 A block containing an optional block skip code such as /M02 ; , /M30 ; , or /M99 ; is not regarded as the end of a program.
NOTE 3 When no program number is specified at the start of a program, the sequence number (N....) at the start of the program is regarded as its program number. Note, however, that N0 cannot be used for a program number.
NOTE 4 If there is no program number or sequence number at the start of a program, a program number must be specified using the CRT/MDI panel when the program is stored in memory. (See 9.2.1 in Part III.)
NOTE 5 If one file contains multiple programs, the EOB code for label skip operation must not appear before a second or subsequent program number. However, an EOB code is required at the start of a program if the preceding program ends with %.
NOTE 6 A program on the tape can be executed without its program number being specified. However, the number of a subprogram must always be specified.
NOTE 7 Program numbers 8000 to 9999 may be used by machine tool builders, and the user may not be able to use these numbers.
NOTE 8 When an optional robot is provided, program numbers 9000 to 9999 are used as the data for the robot.
NOTE 9 The setting of NPE in parameter 2200 determines whether ER (EIA), % (ISO), or the next program number (O), but not M02, M30, or M99, is assumed to be the end of a program.
NOTE 10 Up to 16 characters can be used to specify a program name. When a 48–character program name option is provided, however, up to 48 characters can be used.
NOTE 11 The space between programs on the paper tape must be at least 2 m.

13.4.3 Sequence number and block

A program consists of several commands. One command unit is called a block. One block is separated from another with an EOB code.

At the head of a block, a sequence number consisting of address N followed by a number not longer than five digits (1 to 99999) can be placed. Sequence numbers can be specified in a random order, and any numbers can be skipped. Sequence numbers may be specified for all blocks or only for desired blocks of the program. In general, however, it is convenient to assign sequence numbers in ascending order in phase with the machining steps (for example, when a new tool is used by tool replacement, and machining proceeds to a new surface with table indexing.)

NOTE 1 N0 must not be used for the reason of file compatibility with other CNC systems.
NOTE 2 Program number 0 cannot be used. So 0 must not be used for a sequence number regarded as a program number.

13.4.4 Optional block skip

When a slash followed by a number (/n (n=1 to 9)) is specified at the head of a block, and optional block skip switch n on the machine operator panel is set to on, the information contained in the block for which /n corresponding to switch number n is specified is ignored in tape operation or memory operation.

When optional block skip switch n is set to off, the information contained in the block for which /n is specified is valid. This means that the operator can determine whether to skip the block containing /n.

Number 1 for /1 can be omitted. However, when two or more optional block skip switches are used for one block, number 1 for /1 cannot be omitted.

When an optional block skip switch is set to on, the following range is ignored :

```
N100 X10.0 ;
/2N101 Z10.0 ;      ⎫
/2/3N102 Z20.0 ;    ⎬  Range ignored
/3N103 Z20.0 ;      ⎭
N104 Z30.0 ;
      :
```

The program above skips the following blocks :

N101 and N102 when switch No. 2 is on

N102 and N103 when switch No. 3 is on

NOTE 1 A slash (/) must be specified at the head of a block. If a slash is placed elsewhere, the information from the slash to immediately before the EOB code is ignored.

NOTE 2 When an optional block skip switch is on. TH and TV checks are made for the skipped portions in the same way as when the optional block skip switch is off.

NOTE 3 Optional block skip operation is processed when blocks are read from memory or tape into a buffer. Even if a switch is set to on after blocks are read into a buffer, the blocks already read are not ignored.

NOTE 4 This function is effective even during sequence number search operation.

NOTE 5 This function is ignored when programs are loaded into memory. Blocks containing /n are also stored in memory, regardless of how the optional block skip switch is set.

NOTE 6 Programs held in memory can be output, regardless of how the optional block skip switches are set.

NOTE 7 Depending on the machine tool, all optional block skip switches (1 to 9) may not be usable. The user should check with the machine tool builder to find which switches are usable.

NOTE 8 If more than one optional block skip code is specified in one block when the additional optional block skip function is used, number 1 of /1 cannot be omitted. In such a case, /1 must always be specified without any omission.

Example	(Incorrect)	(Correct)
	//3 G00X10.0;	/1/3 G00X10.0;

13.4.5 Word and addresses

A block consists of one or more words. A word consists of an address followed by a number some digits long as indicated below. (The plus sign (+) or minus sign (–) may be prefixed to a number.)

X – 1000
Address+ number = Word

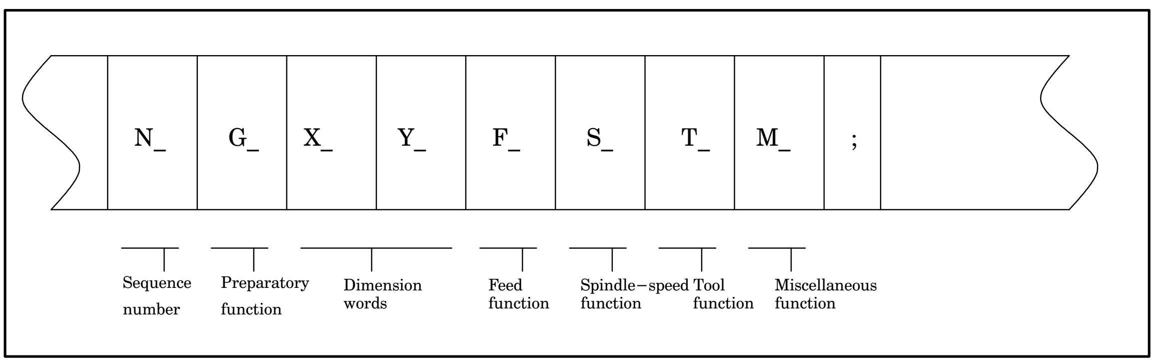
For an address,one of the letters (A to Z) is used ; an address defines the meaning of a number that follows the address. Table 13.4.5 indicates the usable addresses and their meanings.
The same address may have different meanings, depending on the preparatory function specification.

Table 13.4.5 Functions and Addresses

Function	Address	Meaning
Program number	O(*1)	Program number
Sequence number	N	Sequence number
Preparatory–function	G	Specifies motion mode (linear, arc, etc.)
Dimension word	X,Y,Z	Coordinate axis move command
	A,B,C,U,V,W	Additional axis move command
	R	Arc radius
	I,J,K	Coordinate values of arc center
Feed function	F	Specifies feedrate.
Spindle speed function	S	Specifies spindle speed.
Tool function	T	Specifies tool number.
Miscellaneous function	M	Specifies on/off control on machine tool.
	B	Table indexing, etc.
Offset number	D,H	Specifies offset number.
Dwell	P,X	Specifies dwell time.
Program number designation	P	Specifies subprogram number.
Number of repetitions	L	Number of subprogram repetitions, Number of canned cycle repetition
Parameter	P,Q,R	Conned cycle parameter

*1 In ISO code, the colon (:) can also be used as the address of a program number.

For example, a block can consist of these words as follows :



In the next process sheet example, one line represents a block. And one frame in a block is a world.

Program name Test Program 2	Note	Date	Page /
Program number 0 (;) 2002		Programmer	

/	N	G	Z	Y	Z	A/B/C	U/V/W	R/1	I	K	F	S	T	M	B	H/D	L	P	Q	:
	N20	G92	X100.0	Y200.0	Z300.0															;
	N21	G00	X196.0	Y315.0								S400	T15	M03						;
	N22	G01			Z500.0						F10.0									;

Fig. 13.4.5 Example of Process Sheet

NOTE End of block code (;) is CR in EIA code or LF in ISO code.

13.4.6 Basic addresses and command value range

Table 13.4.6 shows the basic addresses and command value range. Note that these figures represent limits on the CNC side, which are totally different from limits on the machine tool side. For example, the CNC allows a tool to traverse up to about 100 m (in millimeter input) along the X axis.

However, an actual stroke along the X axis may be limited to 2 m for a specific machine tool.

Similarly, the CNC may be able to control a cutting federate of up to 24 m/min, but the machine tool may not allow more than 6 m/min. When developing a program, the user should carefully read the manuals of the machine tool as well as this manual to be familiar with the restrictions on programming.

Table 13.4.6 Basic addresses and command value range

Function		Address	Input in mm	Input in inches
Program number		O (*)	1 – 9999	1 – 9999
Sequence number		N	1 – 99999	1 – 99999
Preparatory function		G	0 – 99	0 – 99
Dimension word		X,Y,Z		
Increment system	IS–A	A,B,C,I,J,K	±999999.99 mm	±99999.999 inch
	IS–B	R,U,V,W	±99999.999 mm	±9999.9999 inch
	IS–C		±9999.9999 mm	±999.99999 inch
Feedrate per minute		F		
Increment system	IS–A		0.0001–240000.0 mm/min	0.0001–24000.00 inch/min
	IS–B		0.0001–24000.00 mm/min	0.0001–2400.000 inch/min
	IS–C		0.0001–2400.000 mm/min	0.0001–240.0000 inch/min
Feedrate per rotation		F		
Increment system	IS–A		0.0001–5000.0000 mm/rev	0.00001–500.00000 inch/rev
	IS–B		0.00001–500.00000 mm/rev	0.000001–50.000000 inch/rev
	IS–C		0.000001–50.000000 mm/rev	0.0000001–5.0000000inch/rev
Screw pitch		F	Same as for feed per rotation	Same as for feed per rotation
Tool function		T	0 – 99999999	0 – 99999999
Spindle function		S	± 99999999	± 99999999
Miscellaneous function		M	0 – 99999999	0 – 99999999
Second miscellaneous function		B,A,C	±99999999	± 99999999
Sequence number specification		P,Q	1 – 99999	1 – 99999
Dwell per second		P,X	0 – 99999.999 sec	0 – 99999.999 sec
Dwell per rotation		P,X	0 – 99999.999 rev	0 – 99999.999 rev
Repetition function		L	0 – 9999	0 – 9999

NOTE 1 (*) In ISO code, the colon (:) can also be used as the address of a program number.
NOTE 2 The maximum value of the dimension word is limited to ±39370.078, ±3937.0078, or ±393.70078 inches for input in inches and output in millimeters.

13.5 Comment Section

Any information enclosed in the control–out and control–in codes is regarded as a comment, and is skipped. The user can enter a header, comment, directions to the operator, and so forth. Even the EOB code can be used.

Notation in this manual	EIA	ISO	Meaning
(	2–4–5	(	Control–out (start of comment section)
)	2–4–7	)	Control–in (end of comment section)

Example In ISO code

```
    :
N1000G00X ;
(MEASURE WORK) ;
N10010G01X ;
    :
```

When a command tape is read into memory for memory operation, comment sections, if any, are not ignored but are also read into memory. Note, however, that codes other than those listed in the code table in Appendix 8 are ignored, and thus are not read into memory. When data in memory is punched out on paper tape with the punch function, the comment sections are also punched out.

When a program is displayed on the screen, its comment sections are also displayed. However, those codes that were ignored when read into memory are not punched out or displayed.
During memory operation in memory command mode, all comment sections are ignored.

NOTE 1 Do not use a rewind stop code such as % or ER in the comment section. If the rewind stop code is read, the CNC is reset. While the tape is rewound, sections are not checked. Even if the rewind stop code is punched in the common section, rewinding the tape is stopped at the position. This code is not related to operation using memory.

NOTE 2 If a long comment section appears in the middle of a program section, a move along an axis can be suspended for a long time because of such a comment section. So a comment section should be placed where movement suspension may occur or no movement is involved.

NOTE 3 If only a control–in code is read with no matching control–out code, the read control–in code is ignored.

NOTE 4 The TV check function can not be disabled for a comment section by specifying the parameter TVC (No. 0000#0).

13.6 Program End

The end of a program is indicated by punching one of the following codes at the end of the program:

Code	Meaning
M02;	Program end
M30;	Program end
M99;	Subprogram code

If one of the program end codes above is executed in program execution, the device terminates the execution of the program, and the reset state is set. When the subprogram end code is executed, control returns to the program that called the subprogram.

13.7 End File Mark (Tape End)

An end file mark is to be placed to be placed at the end of a file containing NC programs.

If programs are entered using SYSTEM P or ordinary personal computers, the mark need not be entered. The mark is not displayed on the CRT display screen. However, when a file is output, the mark is automatically output at the end of the file.

Notation in this manual	EIA	ISO	Meaning
%	ER	%	End file mark (tape end)

NOTE If an attempt is made to execute % when M02 or M03 is not placed at the end of the program, the CNC is placed in the reset state.

13.8 Tape Format

The format of the tape conforms to that of the word addresses in the variable blocks having a decimal point.

13.9 Tape Codes

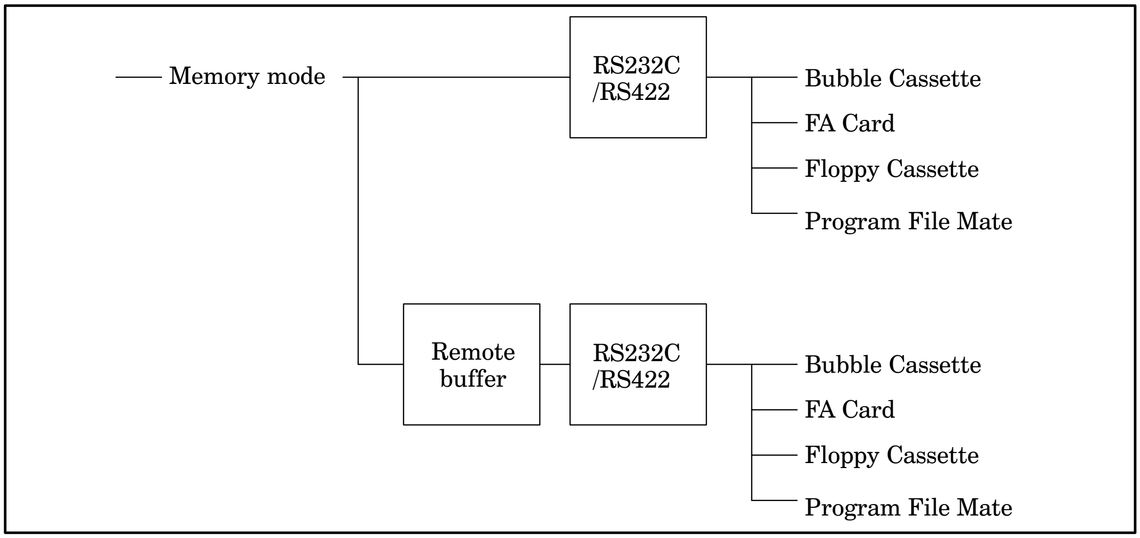
EIA and ISO codes can be used as tape codes. The first end–of–block code (CR for EIA or LF for ISO) is used to automatically check whether the codes of the input program are EIA or ISO codes.

For the tape codes that can be used, see Appendix 8.

13.10 Function for Calling a Subprogram Stored in External Memory

The use of this function enables calling and executing the programs stored in the Bubble Cassette, FA Card, Floppy Cassette, or Program File Mate during operation using memory.

Adding a remote buffer enables calling and executing the programs stored in the Bubble Cassette, FA Card, Floppy Cassette, and Program File Mate connected to the remote buffer during operation using memory.



13.10.1 Programs

When the next program is executed with the currently executed program in memory, the specified programs stored in the Bubble Cassette, FA Card, Floppy Cassette, and Program File Mate are called and executed.

Even when a remote buffer is added, the programs stored in the Bubble Cassette, FA Card, Floppy Cassette, and Program File Mate connected to the remote buffer are called and executed.

Setting bit 3 of parameter 7616 to 1 enables this function.

M198 P[program–number]L[number of times the program is called in succession]

* Caution

When the external I/O device connected to Bubble Cassette or remote buffer is used as an external memory unit, a program cannot be executed by specifying its program number. If a subprogram is called with its program number specified from the external memory unit, an alarm may occur. For details, see Chapter 3. In this case, specify a file number to execute the subprogram.

To change specifying a program number to specifying a file number, set bit 5 of parameter 2404 to 1.

A program can be called using the M code specified in parameter 2431.

(The M198 is used as a usual M code.)

Setting bit 5 of parameter 2404 changes specifying a program number to specifying a file number.

If the program not stored in the Bubble Cassette, FA Card, Floppy Cassette or Program File Mate is specified, P/S alarm No. 79 occurs.

Selecting an external memory unit using the function for calling a subprogram from the external memory unit is enabled by setting parameter 20 as follows:

Setting	External memory unit
1	External memory unit to be connected to JD5A of the main CPU board
2	External memory unit to be connected to JD5B of the main CPU board
3	External memory unit to be connected to JD5C of the subboard
10	Remote buffer
13	External memory unit to be connected to JD6A of the subboard

NOTE The function for calling a subprogram from an external memory unit can only be executed during operation using memory.

13.10.2 Calling a subprogram using the calling a subprogram stored in cxternal memory

The function for calling a subprogram stored in external memory was upgraded toenable a subprogram to be called from a program that has been called using this function. However, it is impossible to call a subprogram in external memory from a program that has been called using this function. Only a program in the CNC memory can be called.

```
Main
Program           O0001 ;           Sub
· · ·             Program           External memory
M98 P1000 ;  →    O1000 ;           Sub
· · ·             · · ·             Program
M99 ;             M198 P2000 ;→     O2000 ;           Sub
                  · · ·             · · ·             Program
                  M99 ;             M98 P3000 ;  →    O3000 ;           Sub
                                    · · ·             · · ·             Program
                                    M99 ;             M98 P4000 ;  →    O4000 ;
                                                      · · ·             · · ·
                                                      M99 ;             M99 ;
```

NOTE 1 It is impossible to call a subprogram in external memory from a program called using this function. Such an attempt will result in the PS121 alarm.

NOTE 2 This function cannot be executed in the test mode of the retrace edit function.

NOTE 3 Override records for the override playback function become ineffective in a program called using this function.

NOTE 4 An attempt to use this function together with the background graphic display function results in a warning message related to an NC statement error (B. G.).

NOTE 5 It is possible to call a program in memory using a custom macro call instruction (G/ M/ S/ T/ B/ G65/ G66/ G66.1) in a program called using this function.

NOTE 6 It is possible to call the macro executor (execution macro) in a program called using this function.

NOTE 7 It is possible to execute an interrupt type custom macro in a program called using this function. The called program must be in memory.

NOTE 8 For the nesting level of calling, an external memory subprogram call is counted as one in the same manner as before.

14.FUNCTIONS TO SIMPLIFY PROGRAMMING

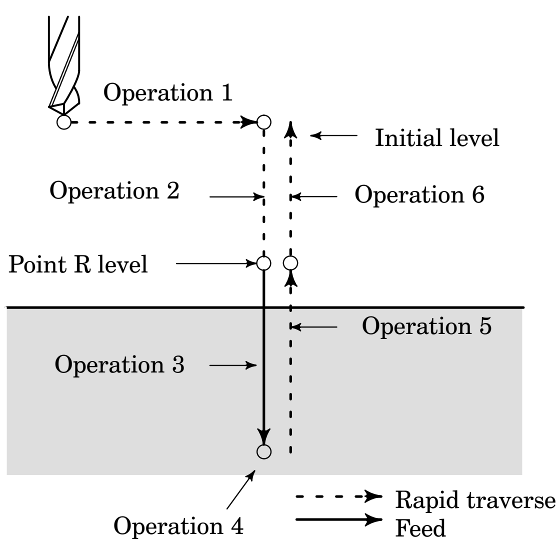
14.1 Canned Cycles (G73, G74, G76, G80 to G89)

A canned cycle simplifies the program by using a single block with a G code to specify the machining operations specified in several blocks. Table 14.1 lists canned cycles.

Table 14.1 Canned cycles

G code	Drilling (–Z direction)	Operation at the bottom of a hole	Retraction (+Z direction)	Application
G73	Intermittent feed	————	Rapid traverse	High–speed peck drilling cycle
G74	Feed	Spindle CW	Feed	Left–hand taping cycle
G76	Feed	Oriented spindle stop	Rapid traverse	Fine boring cycle (Only canned cycle II)
G80	————	————	————	Cancel
G81	Feed	————	Rapid traverse	Drilling cycle, spot drilling cycle
G82	Feed	Dwell	Rapid traverse	Drilling cycle counter boring cycle
G83	Intermittent Feed	————	Rapid traverse	Peck drilling cycle
G84	Feed	Spindle CCW	Feed	Tapping cycle
G85	Feed	————	Feed	Boring cycle
G86	Feed	Spindle stop	Rapid traverse	Boring cycle
G87	Feed	Spindle stop	Manual/Rapid traverse	Boring cycle, Back boring cycle
G88	Feed	Dwell → spindle stop	Manual/Rapid traverse	Boring cycle
G89	Feed	Dwell	Feed	Boring cycle

NOTE G87 specifies a different operation in canned cycles I and II.



A canned cycle consists of a sequence of six operations (Fig 14.1(a))

Operation 1 ----- Positioning of axes X and Y (including also another axis)
Operation 2 ----- Rapid traverse up to point R level
Operation 3 ----- Hole machining
Operation 4 ----- Operation at the bottom of a hole
Operation 5 ----- Retraction to point R level
Operation 6 ----- Rapid traverse up to the initial point

Fig. 14.1(a) Canned cycle operation sequence

A canned cycle has a positioning plane and a drilling axis. The positioning plane is determined by G17, G18, and G19.

The drilling axis conists of the basic axis (X, Y, or Z) which does not constitute the positioning plane.

G code	Plane for positioning	Axis for drilling
G17	Xp–Yp Plane	Zp
G18	Zp–Xp Plane	Yp
G19	Yp–Zp Plane	Xp

Xp: X axis or an axis parallel to the X axis
Yp: Y axis or an axis parallel to the Y axis
Zp: Z axis or an axis parallel to the Z axis

An axis address determines which basic axis or which axis parallel to a basic axis is used for drilling. The axis address is specified in a block in which a G code that is one of G73 to G89 is specified. If an axis address is not specified, the axis for drilling is assumed to be a basic axis.

Axes other than the axis for drilling are positioning axes.

Example Assume that the U, V and W axes be parallel to the X, Y, and Z axes respectively. This condition is specified by parameter No. 1022.

G17 G81.................... Z _ _ : The Z axis is used for drilling.
G17 G81.................... W_ _ : The W axis is used for drilling.
G18 G81.................... Y _ _ : The Y axis is used for drilling.
G18 G81.................... V _ _ : The V axis is used for drilling.
G19 G81.................... X _ _ : The X axis is used for drilling.
G19 G81.................... U _ _ : The U axis is used for drilling.

G17 to G19 may be specified in a block in which any of G73 to G89 is not specified.

NOTE 1 A parameter FXY (No. 6200#0) can be set so the Z axis is always used as the drilling axis. When FXY=0, the Z axis is always the drilling axis.

NOTE 2 Switch the drilling axis after canceling a canned cycle.
Canned cycle operations can be specified by three modal G–codes.

(i) Data format
G90 : Absolute command
G91 : Incremental command

(ii) Codes to define a return point level
G98 : Initial level return
G99R : Point R level return

(iii) Codes to define drilling mode
See G73 to G89 in Table 14.1

NOTE 3 The initial level means the absolute value of the Z–axis at the time of changing from the canned cycle cancel mode into the canned cycle mode.

(1) Data format

In canned cycle operations, the distance traveled along the Z–axis varies between G90 and G91 as shown in Fig. 14.1(b).

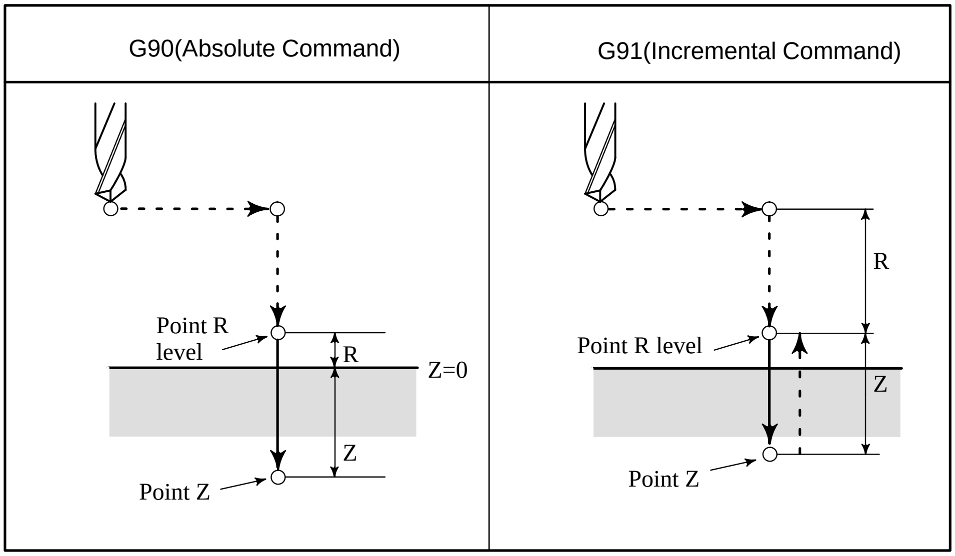


Fig. 14.1 (b) Data format for canned cycle

(2) Return point level

Whether the tool is to be returned to point R level or to the initial level is specified according to G98 or G99. This is shown in Fig. 14.1 (c).

Use G99 for the first drilling, and use G98 for the last drilling. When the canned cycle is to be repeated by K in G98 mode, tool is returned to the initial level from the first time drilling.

In general, G99 is usually used for the first drilling, and G98 is used for the last drilling. If L for G98 and G99 specifies the number of times, the tool is returned to the initial level after the first drilling when G98 is issued.

In the G99 mode, the initial level does not change even when drilling is performed.

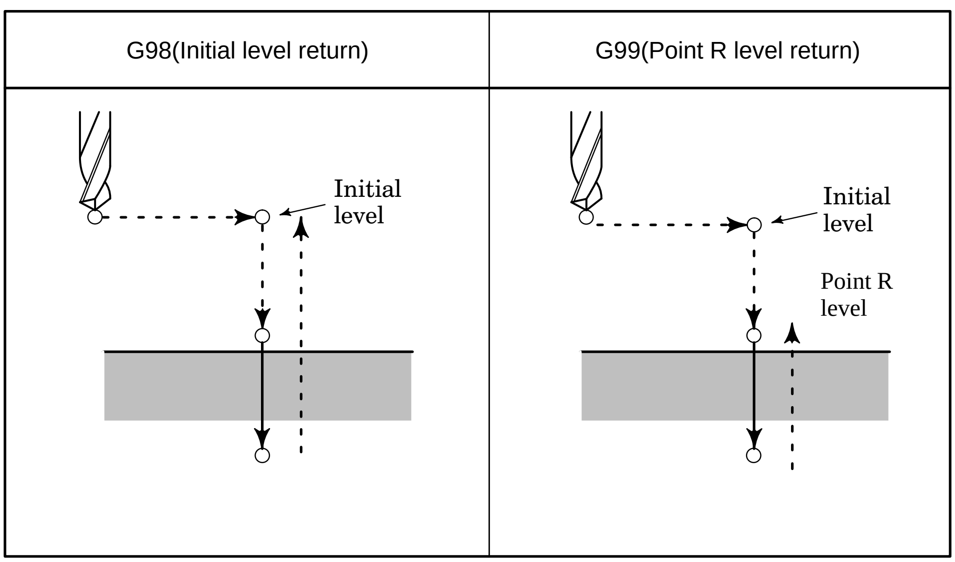


Fig. 14.1 Initial level and point R level returns

(3) Data on a canned cycle

The drilling data can be specified following

G73/G74/G76/G81 to G89 and a single block can be formed.

This command permits the data to be stored in the control unit as a modal value.

This command stores the data as continuous–state values in the control unit.

The machining data on a canned cycle is specified as shown below:

$$\underbrace{G\square\square}_{\text{Drilling mode}}\ \underbrace{X--\ Y--}_{\text{Hole position date}}\ \underbrace{Z--\ R--\ Q--\ P--\ F--}_{\text{Drilling data}}\ \underbrace{L--}_{\text{Number of repeat}}$$

Drilling mode G	See Table 14.1
Hole position data X, Y	Specifies the hole position by an incremental or absolute value. Setting FCU, a bit of parameter 6200, specifies whether the path and feedrate are determined by the G code in the current group, 01, or are unconditionally determined by G00. For the current group, 01, G01 is assumed for G02 and G03.
Drilling data Z	Specifies the distance from point R to the bottom of the hole in Fig. 14.1 (a) with an incremental value or the position of the hole bottom with an absolute value. The feed rate is specified in the F code in operation 3. In operation 5, it is the rapid traverse rate or the feed rate specified in the F code, according to the drilling mode.
Drilling data R	Specifies the distance from the initial level to point R level in Fig. 14.1 (a) with an absolute value. The feed rate is the rapid traverse rate in operation 2 and 6.
Drilling data Q	Specifies each cut–in value with G73 and G83 or the shift value with G76 and G87. (Always specified with an incremental value.)
Drilling data P	Specifies the dwell time at the bottom of the hole. The relationship between the time and the specified value is the same as for G04.
Drilling data F	Specifies the feed rate.
Number of repeats L	Specifies the number of repeats for a series of operation 1 to 6. when L is not specified, L = 1 is assumed. When L = 0 is specified, the drilling data is simply stored, and drilling is not performed.

Since the drilling mode (G □□) remains unchanged until another drilling mode is specified or the canned cycle is canceled with a G code, it need not be specified any blocks when the same drilling mode continues. G80 and the 01 group G codes are used to cancel a canned cycle. Once the drilling data has been specified in the canned cycle, it is retained until it is changed or the canned cycle is canceled. Therefore, all required drilling data shall be specified when the canned cycle is started and only data to be changed shall be specified during the cycle.

The number of repeats L shall be specified only when operation must be repeated. Data L is not retained. The feed rate specified with the F code is retained even if the canned cycle is canceled.

When the system is reset during the canned cycle, the drilling mode, drilling data, hole position data and repetition count are deleted.

An example of the above data retention and deletion is shown below.

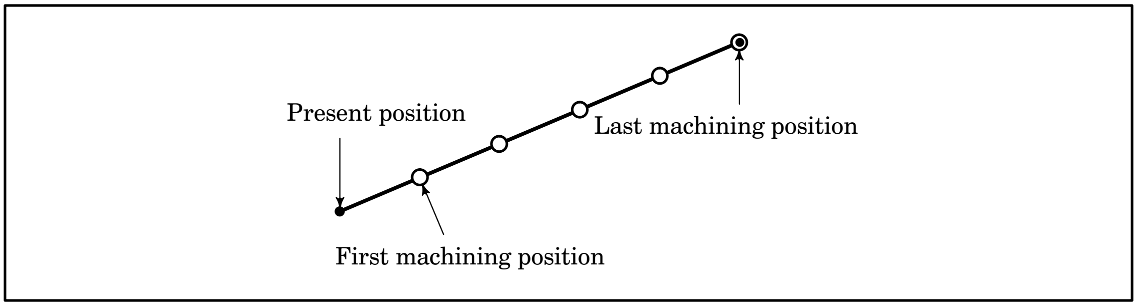
(1)	G00X ___ M3;
(2)	G81 X ___ Y ___ Z ___ R ___ F ___ L ___ ; In the beginning, Z, R, and F specify necessary values. Drilling is repeated L times according to G81.
(3)	Y ___ ; Since the drilling mode and drilling data are the same as those specified in (2), G81, Z ___ , R ___ , and F ___ can be omitted. The hole position moves only for Y ___ and drilling is performed only once according to G81.
(4)	G82 X ___ P ___ L ___ ; The hole position moves only for X ___ from (3) and the hole is drilled according to G82 with Z, R, and F specified in (2) and the drilling data P specified in (4). This is repeated L times.
(5)	G80 X ___ Y ___ M5; Drilling is not performed. All drilling data (except F) is canceled.
(6)	G85 ___ X ___ Z ___ R ___ P ___ ; Since drilling data is cancelled in (5), Z and F must be specified again. Since F is the same as F specified in (2), it can be omitted. P is not required in this block, but it is stored.
(7)	X ___ Z ___ ; Performs drilling which differs only for the Z value with (6) by moving the hole position only for X.
(8)	G89 X ___ Y ___ ; Preforms drilling according to G89 by using the Z specified in (7), R and P specified in (6) and F specified (2) as drilling data.
(9)	G01 X ___ Y ___ ; The drilling mode and all drilling data (except F) are deleted.

(4) Repeat of canned cycle

If holes are repeatedly drilled at equal intervals in the same canned cycle, specify the number of repeats using address L. The maximum value of L is 9999. L is effective only for the specified block.

Example

G81 X __ Y __ Z __ R __ L5F __;



X _____ Y _____ specifies the first drilling position for an incremental command (G91).If it is specified for an absolute command (G90), a hole is repeatedly drilled at the same position.

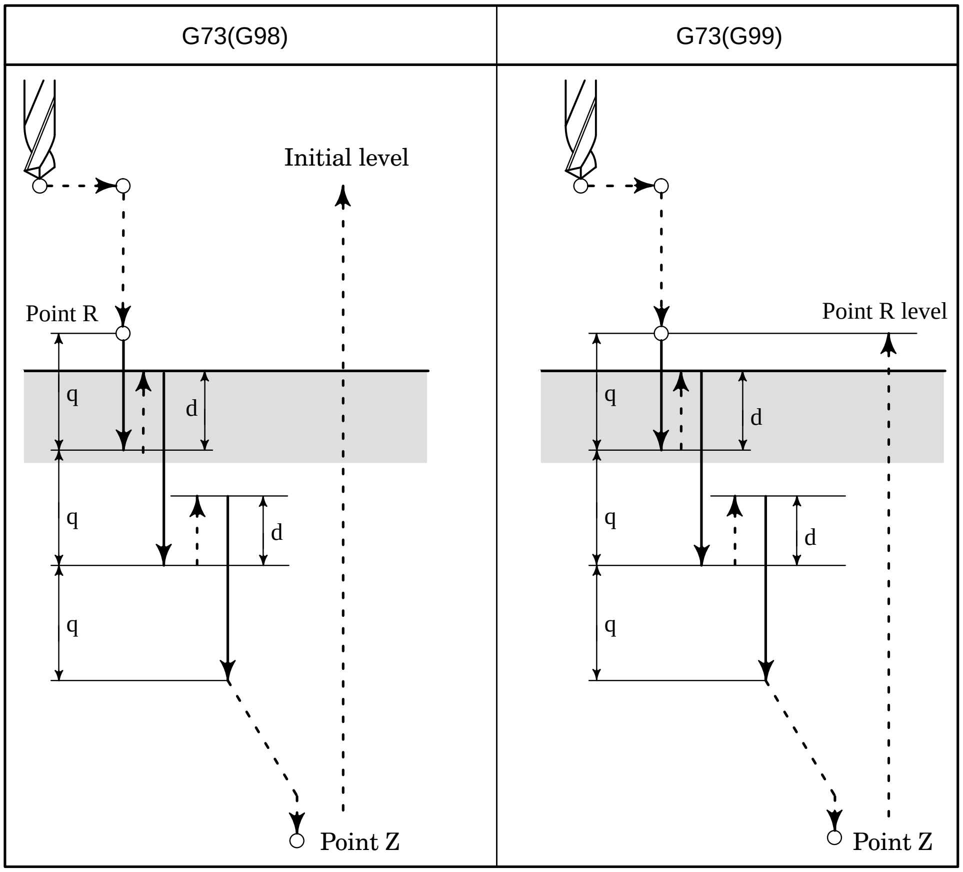
The automatic deceleration time constant in the canned cycle is automatically switched according to the rapid traverse or cutting feed used for each operation. At the end point,the tool decelerates and moves on to the next sequence. However, when the tool returns from the bottom of the hole to the R level at the rapid traverse rate with G98 (initial level return) (for example, G81), the tool does not decelerate, but immediately returns to the initial level at the rapid traverse rate.

Each hole machining operation is detailed below.

Rapid traverse and Feed are indicated as follows.

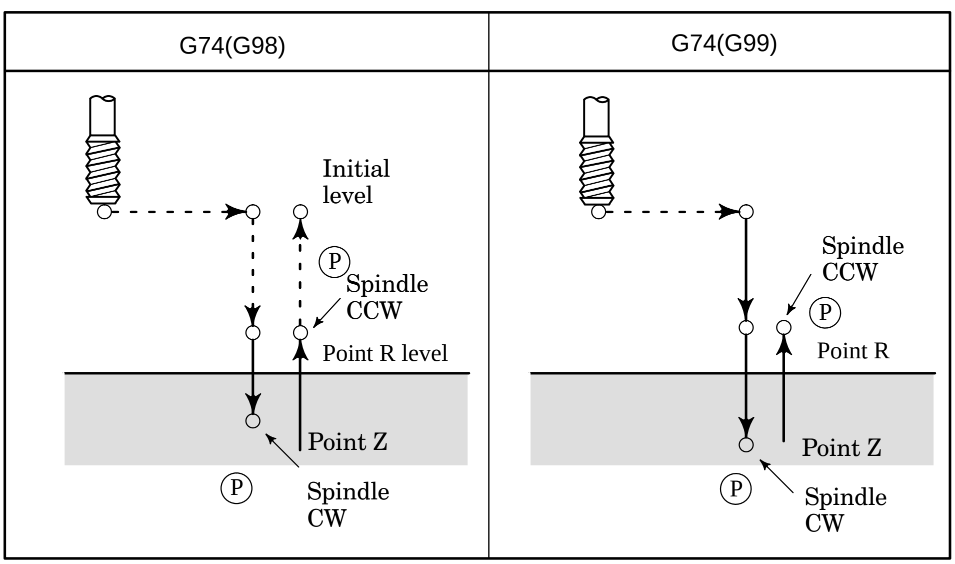
14.1.1 High–speed peck drilling cycle (G73)

The retraction d is set by a parameter (No. 6210). Since the Z–axis direction intermittent feed simplifies chip disposal and permits a very small retraction value to be set in deep hole drilling, efficient machining is performed. Retraction is performed at the rapid traverse rate.



Rapid traverse   Cutting Feed

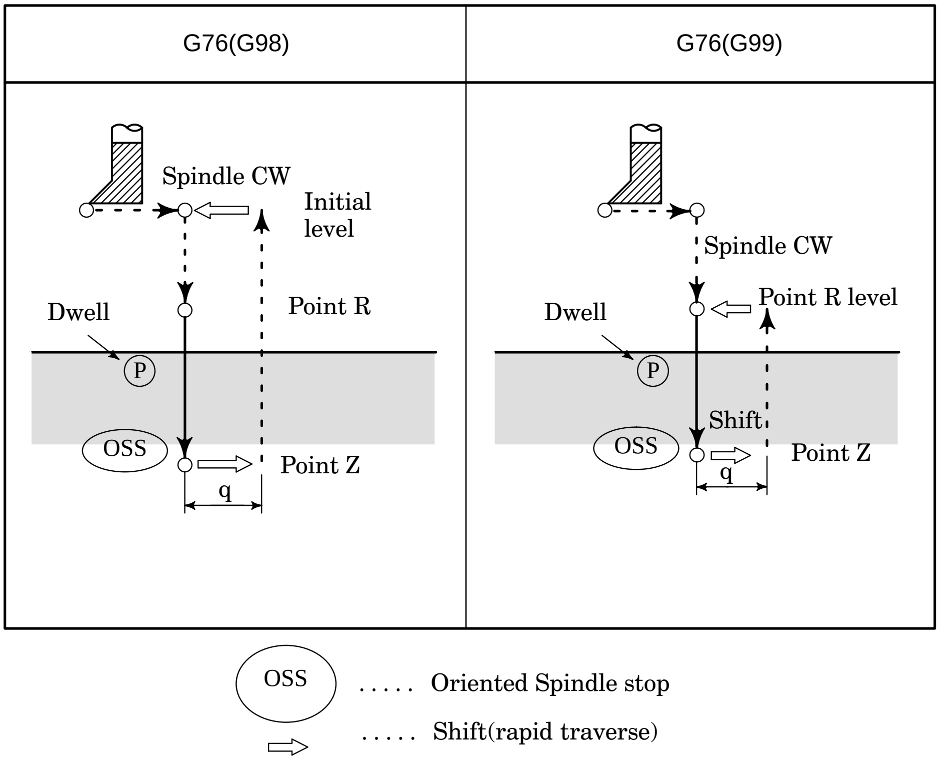
14.1.2 Left–handed tapping cycle (G74)



At the bottom of the hole, the spindle rotates clockwise and left handed tapping performed.

WARNING	During left–handed tapping specified in G74, the feed rate override is ignored and the cycle does not stop until the end of the return operation, even if a feed hold is applied.

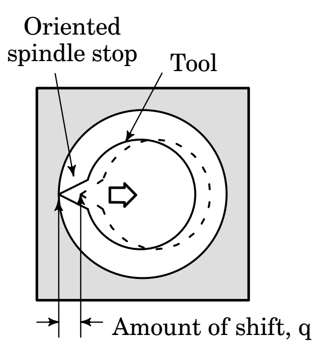
14.1.3 Fine boring cycle (G76)



WARNING	When FXY, a bit of parameter 6200, is specified so that the Z–axis is always a drilling axis, the amount of shift can also be specified using address Q, but not I or J. Q must always be set to a positive value. If address Q is specified using a negative value, the minus sign is ignored. The direction of shift must be specified in advance as either +X, –X, +Y, or –Y using parameter 6240. The value of Q is in the continuous state during the canned cycle, and can also be used as the depth of cut for G73 or G83. Care must therefore be taken when specifying the value of Q.

NOTE G76 can be used only when the parameter FXB (No–6210#1) that which specifies output of the M code (canned cycle II) as the spindle CCW, spindle CW and oriented spindle stop output signal is set.

Since a spindle oriented stop is performed at the bottom of the hole and the spindle retracts after shifting in the direction opposite to the cutter direction, high precision and efficient boring is performed without scratching the work piece surface.



The amount of shift is specified corresponding to the following positioning planes:

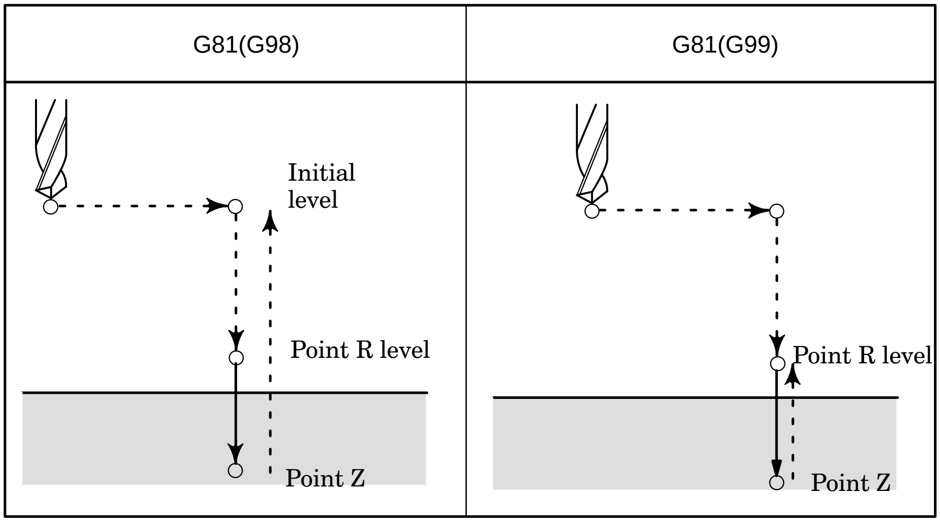
G17 (Xp–Yp plane) : Use I and J.
Xp: X–axis or the axis parallel to the X–axis
G18 (Zp–Xp plane) : Use K and I.
Yp: Y–axis or the axis parallel to the Y–axis
G19 (Yp–Zp plane) : Use J and K.
Zp: Z–axis or the axis parallel to the Z–axis

When the X–Y plane is selected, for example, the tool is shifted along the X– and Y–axes by the incremental distance specified by I and J using linear interpolation. The tool can therefore be shifted in any direction on the positioning plane. The feedrate is controlled by the F command. I, J, and K are in the continuous state during a canned cycle. Specify I, J, and K in the same block as for the hole position data.

14.1.4 Canned cycle cancel (G80)

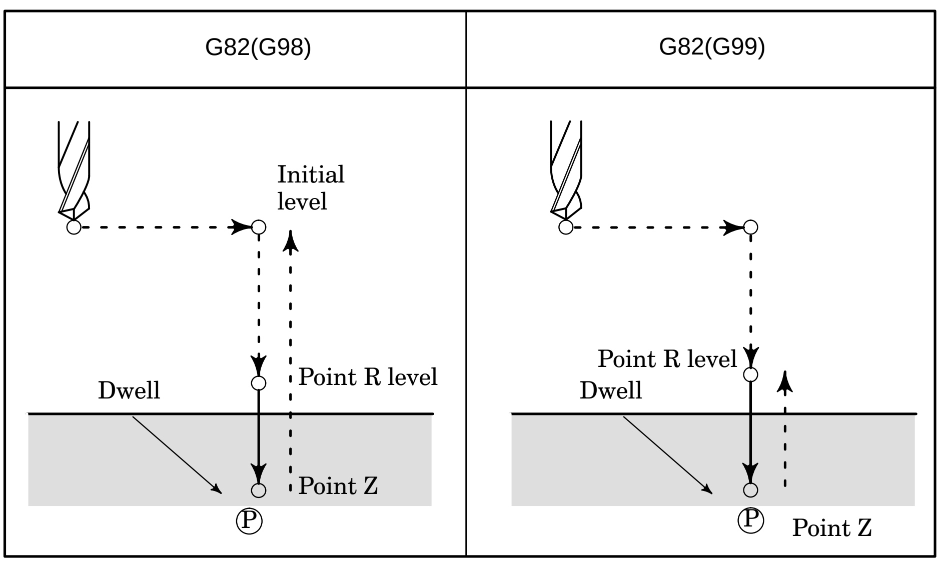
The canned cycle (G73, G74, G76, G81 to G89) is cancelled and normal operation is subsequently performed. The points R and Z are also cancelled. (That is,R=0 and Z=0 for the incremental command.) Other drilling data is also cancelled.

14.1.5 Drilling cycle, spot boring cycle (G81)



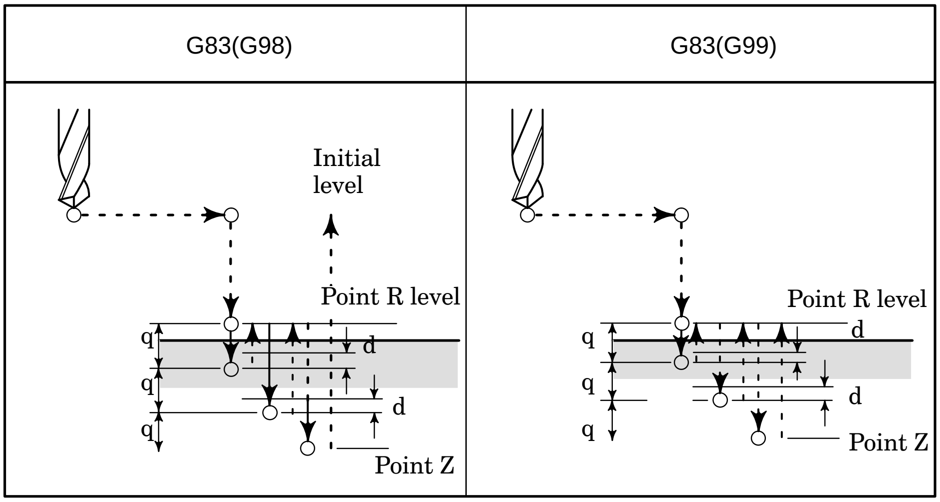
G81 performs positioning with respect to the X–and Y–axes, forwards the tool quickly to the point R level, and causes the tool to drill from the point R level to point Z. On completing drilling, the tool is retracted to the point R level and returned to the initial level quickly.

14.1.6 Drilling cycle, counter boring cycle (G82)



This is the same as G81. Dwell (specified by the P code) is performed at the bottom of the hole and the spindle is raised. Dwell at the bottom of the hole improves the hole depth precision in blind hole drilling.

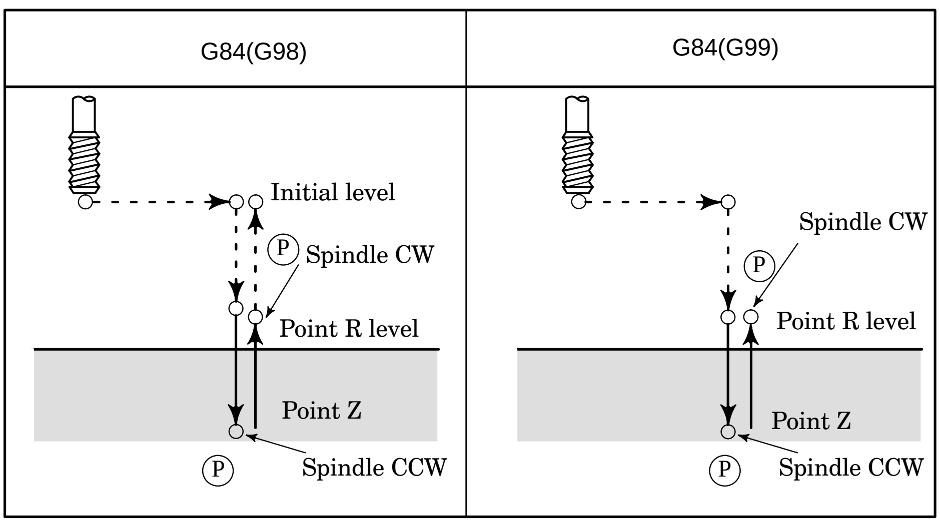
14.1.7 Peck drilling cycle (G83)



G83 X ___ Y ____ Z ____
Q ___ R ____ F ____;

The command above specifies the peck drilling cycle.Q is the every cut–in value and specify with an incremental value. When the second and subsequent cut–in is performed, rapid traverse is changed to feed by d mm (or inch) before the position where cut–in is performed. The specified Q value must always be positive. If a negative value is specified, the sign is ignored. The d value is set by parameter (data No. 6211).

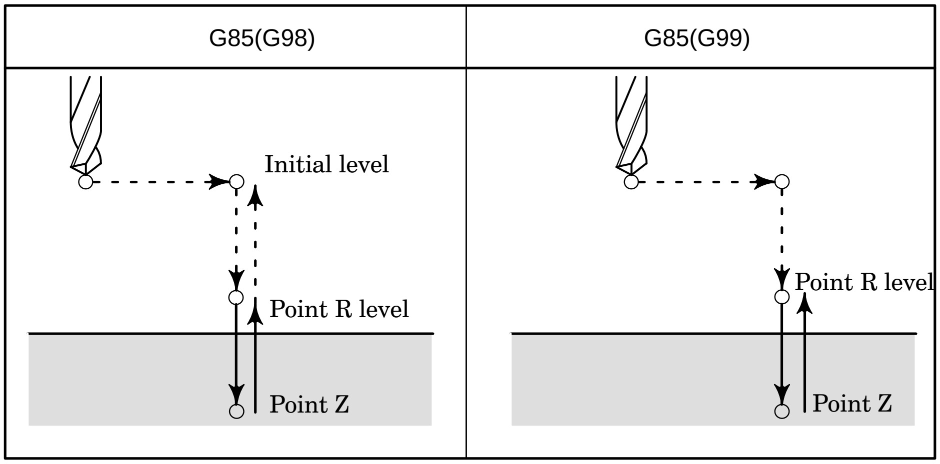
14.1.8 Tapping cycle (G84)



The spindle is reversed at the bottom of the hole and the tapping cycle is performed.

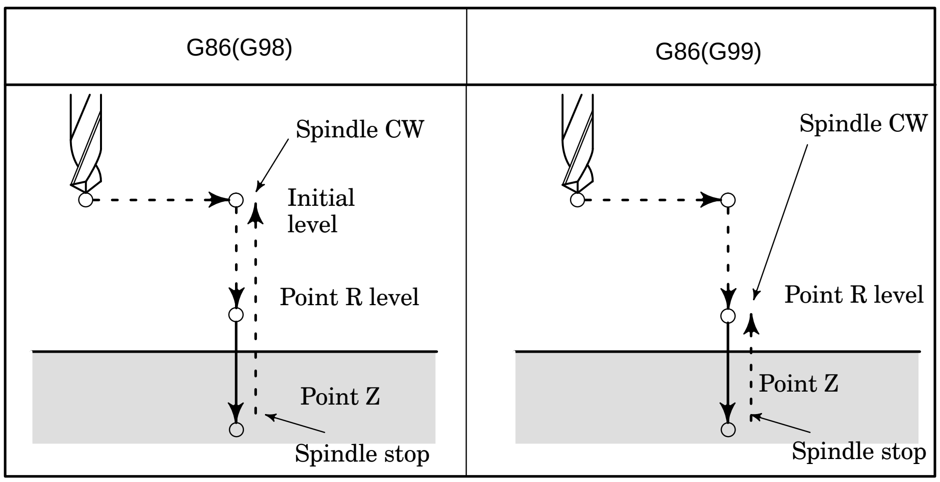
WARNING	During tapping specified in G84, the feed rate override is ignored and the cyde does not stop until the end of the return operation, even if feed hold is applied

14.1.9 Boring cycle (G85)



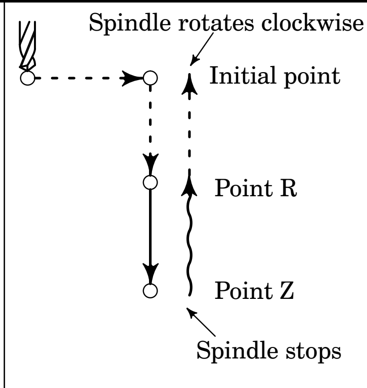
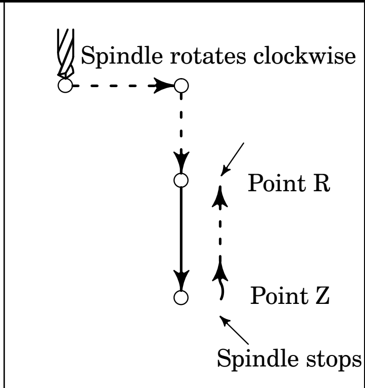
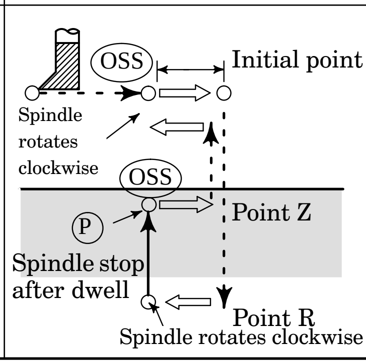
This is the same as G84, but the spindle is not reversed at the bottom of the hole.

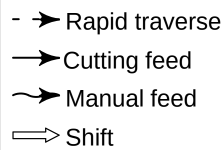
14.1.10 Boring cycle (G86)



This is the same as G81, but the spindle stops at the bottom of the hole and is retracted at the rapid traverse rate.

14.1.11 Boring cycle/back boring cycle (G87)

<table>
<tr><th></th><th>G87(G98)</th><th>G87(G99)</th></tr>
<tr><td>Canned cycle I</td><td>
</td><td>
</td></tr>
<tr><td>Canned cycle II</td><td>
</td><td>This command is not used.</td></tr>
</table>



Canned cycle I (boring cycle)

After the spindle stops at the bottom of the hole, the machine enters the stop state. At this time, changing the current mode to the manual mode enables the tool to be moved manually. Although any manual operation is possible, it is better to extract the tool from the hole for safety at the end.

Restart machining in the tape or memory mode. Then, the tool is returned to the initial level, and to the point–R level according to G98 and G99. The spindle rotates clockwise, and the machine restarts operation according to the command of the next block in the NC tape.

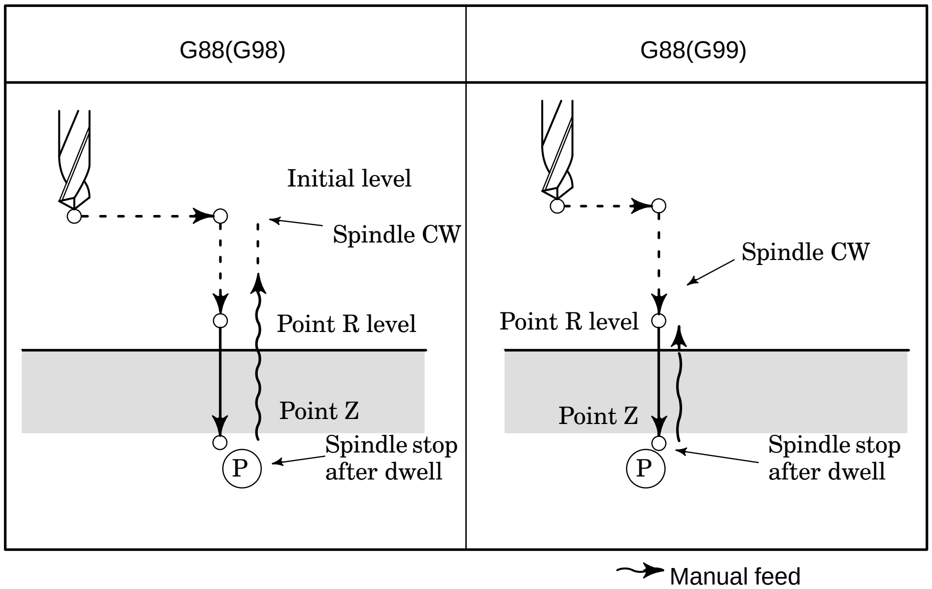
Canned cycle II (back boring cycle)

After the tool is positioned along the X– and Y–axes, the spindle stops at the fixed rotation position. The tool is shifted in the opposite direction to that of the tool tip, and is positioned at the bottom of the hole (point R) in the rapid traverse mode. The tool is moved back from the position according to the specified amount of shift, and the spindle rotates clockwise. Then, the machine performs drilling in the positive direction along the Z–axis to point Z. At point Z, the spindle is stopped again at the fixed rotation position. The tool is then shifted in the opposite direction to that of the tool tip and extracted from the hole.

After the tool is returned to the initial point, it is moved back according to the specified amount of shift, the spindle rotates clockwise, and the machine executes the machining specified in the next block. The amount of shift and direction are exactly the same as for G76. (The settings of the direction for G76 and G87 are the same.)

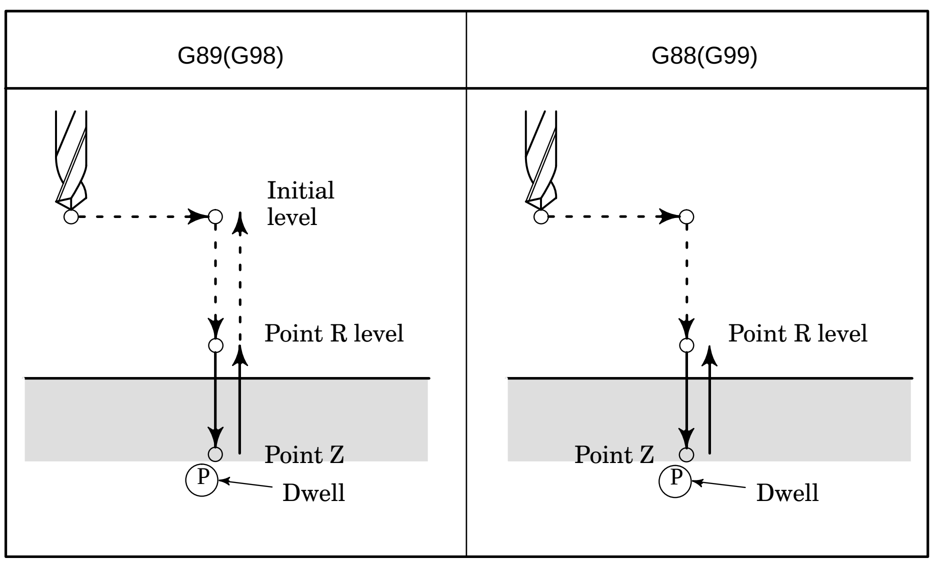
NOTE	Canned cycle I	:	This mode is specified by setting FXB, a bit of parameter 6201, to 0 so that separate signals, the SRV and SSP signals, are used as output signals for counterclockwise spindle rotation and spindle stop.
	Canned cycle II	:	This mode is specified by setting FXB, a bit of parameter 6201, to 1 so that M codes are output as output signals for counterclockwise spindle rotation, spindle stop, and oriented spindle stop.

14.1.12 Boring cycle (G88)



G88 performs positioning with respect to the X–and Y–axes, forwards the tool quickly to the point R level, and causes the tool to bore from the point R level to point Z. On completing boring, the spindle dwells allowing the operator to retract the tool from point Z to the point R level by hand. From the point R level to the initial level, the tool is retracted quickly by rotating the spindle forward.

14.1.13 Boring cycle (G89)



This is the same as G85, but dwell is performed at the bottom of the hole.

14.1.14 Notes on canned cycle specifications

WARNING Operator precautions

a) Single block

When a canned cycle is performed in the single block mode, the control stops at the end of operation 1, 2, and 6, as shown in Fig.14.1 (a). Therefore, it must be started three times to drill one hole. The feed hold lamp lights at the end of operations 1 and 2. When the number of repeats to be machined is left at the end of operation 6, the system stops with feed hold state, and when not, the system stops in stop state.

b) Feed hold

When a feed hold is applied between operations 3 to 5 in canned cycle G74 or G84, the feed hold lamp immediately lights, but the control continues to operate up to operation 6 and stops. If a feed hold is applied again during operation 6, it immediately stops.

c) Override

The feed rate override is assumed to be 100% during the operation of canned cycle G74 or G84.

d) Manual absolute function

In the G87 (canned cycle I) and G88 mode, the following conditions are satisfied during manual operation depending on whether the MANUAL ABSOLUTE switch is turned on or off.

On : Point R and initial point agree with the programmed values.
Off : Point R and initial point are shifted by the specified distance during manual operation.

e) Origin button

Do not set the origin in the canned cycle mode.

NOTE 1 The spindle must be rotated by the miscellaneous function (M code) before a canned cycle is specified.

```
M3;            Spindle CW
.
.
.
G□□ ....... ;  Correct
.
.
.
M5;            Spindle stop
.
.
.
G□□ .......;   Incorrect (M3 or M4 must be specified before )
```

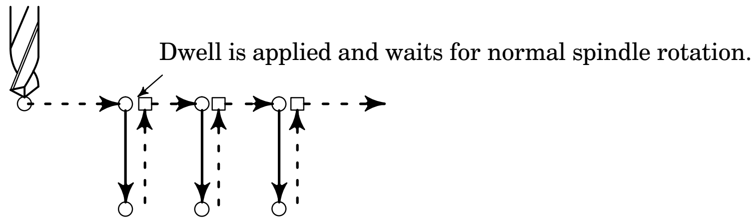
NOTE 2 If the block contains X, Y, Z, R, or additional axes data, drilling is performed in canned cycle mode. If the block does not contain the X, Y, Z, R, and additional axes data, drilling is not performed. However, when G04X __; is specified, drilling is not performed even if X is specified.

```
G00 X ___ ;
G81X _ Y _ Z _ R _ F _ P _ L _;
;        (Drilling is not performed)
F _ ;    (Drilling is not performed.  Value F is updated.)
M _ ;    (Drilling is not performed.)
         (Only the miscellaneous function is executed.)
G04P _;  (Drilling is not performed.  Drilling data P is not changed by G04P.)
```

NOTE 3 Specify drilling data Q, P, I, J, K in the block where drilling is performed. In other words, specify it in the block for which the X, Y, Z, addition axis, or R axis data is specified.This data is not stored as modal data if specified in a block where drilling is not performed.

NOTE 4 When drilling with a short distance between the hole positions (X, Y) or the initial level to the R point level are repeated using the canned cycle (G84.1, G84, G86) which controls the rotation of the spindle the spindle may not reach the normal rotation before it enters the cutting operation (–Z) of the hole.

When this occurs, a time lag must be provided by inserting a dwell command (G04) between each drilling operation. In this case, without specifying the number of repeats with K, program as shown below.




```
G00   M ___ ;
G86   X ___ Y ___ Z ___ R ___ F ___ ;
G04   P __ ;    (Dwell is performed but drilling is not performed)
      X ___ Y ___  ;
G04   P __ ;    (Dwell is performed, but drilling is not performed)
      X ___ Y ___;
G04   P __ ;    (Dwell is performed, but drilling is not performed)
.     .
.     .
.     .
```

This may not have to be considered, depending on the machine tool. Refer to the machine tool builder's instruction manual.

NOTE 5 As previously mentioned, a canned cycle may also be cancelled with G00 to G03 (group 01 G code).This cancel is performed when G00 to G03, G33 is read if it is specified in the same block as the G code of a canned cycle (# indicates 0 to 3, 33, □ □ indicates the canned cycle code):

G# G □ □ X __ Y __ Z __ R __ Q __ P __ F __ L __;
(The canned cycle is performed.)
G □ □ G# X __ Y __ Z __ R __ Q __ P __ F __ L __;
(Movement along the X and Y axes is performed as specified in G#, R, P, Q and L are ignored and F is stored.)

NOTE 6 When a canned cycle command and a miscellaneous function are specified in the same block, the M code and MF signals are sent at the first positioning operation (operation 1 in Fig.14.1 (a)). The control waits for the finish signal (FIN) at the end of positioning before starting the next drilling operation. When the number of repeats (L) is specified, the M code and MF signals are sent only the first time.

NOTE 7 The tool offset commands (G45 or G48) are ignored in canned cycle mode.

NOTE 8 When tool length offset (G43, G44, G49) is specified in canned cycle mode, offset is applied when the tool is positioned to point R (operation 2).

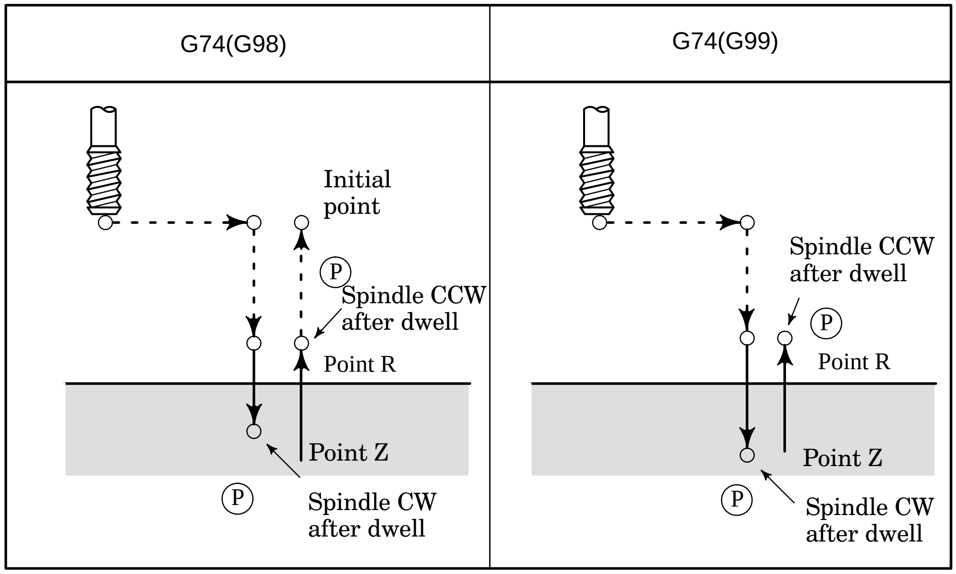
NOTE 9 Setting parameters adds the following operation to that specified by G74 and G84.

Dwelling is executed for the time specified with P before the spindle rotates clockwise or counterclockwise. This function is useful when a special tap is used.

(Tapping by moving the tap backward and forward only by the rotation force while dwelling, without the machine moving along the Z–axis)

(a) G74 (inverse tapping cycle)

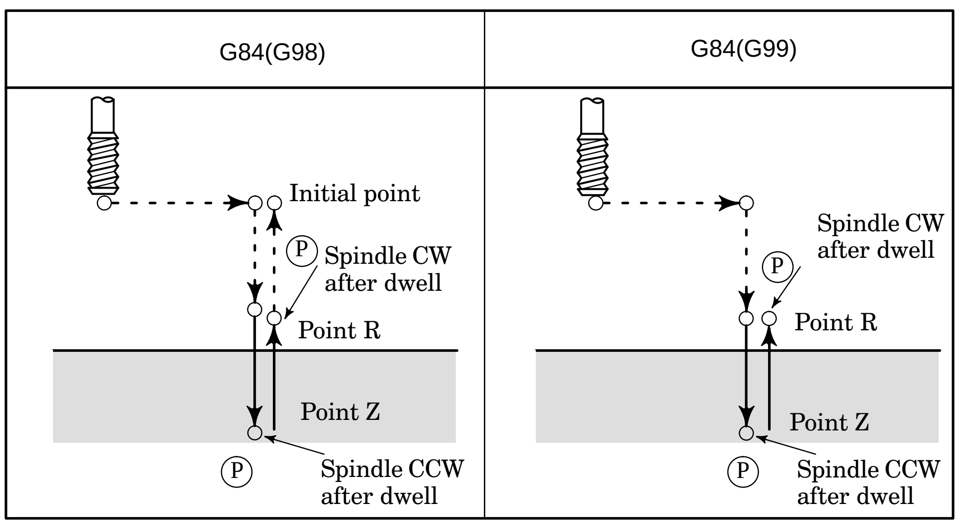
G74X__Y__Z__R__P__F ;



Rapid traverse Feed

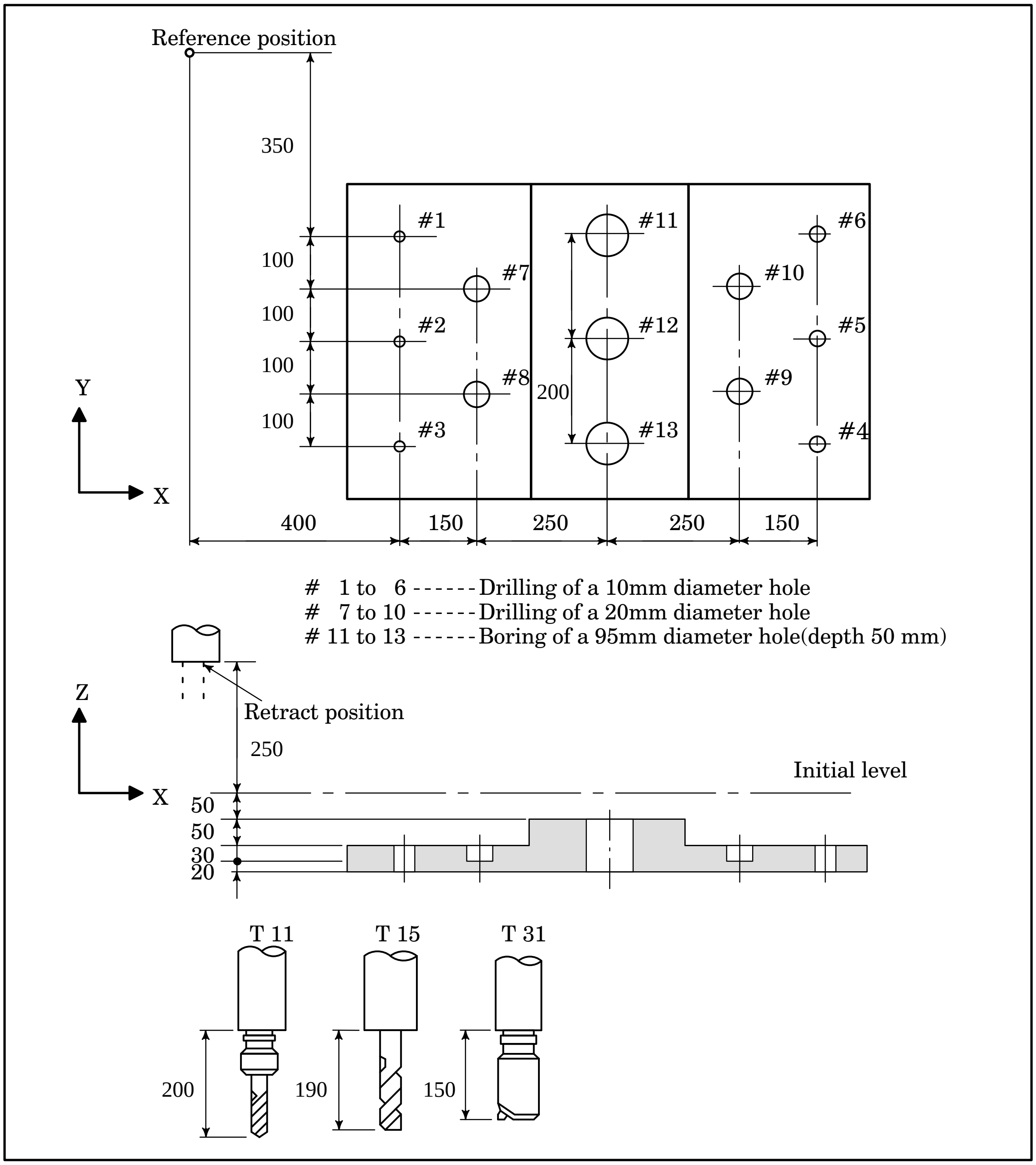
(b) G84 (tapping cycle)

G84X__Y__Z__R__P__F ;



Rapid traverse Feed

14.1.15 Program example using tool length offset and canned cycle



Offset value +200.0 is set in offset No.11, +190.0 is set in offset No.15, and +150.0 is set in offset No.31

Program example

```
;
N001   G92X0Y0Z0;                 Coordinate setting at reference position
N002   G90  G00  Z250.0  T11  M6; Tool change
N003   G43  Z0  H11;              Initial level, tool length offset
N004   S30  M3                    Spindle start
N005   G99  G81X400.0  R  Y–350.0
       Z–153,0R–97.0  F120;         Positioning, then #1 drilling
N006   Y–550.0;                   Positioning, then #2 drilling and point R level return
N007   G98Y–750.0;                Positioning, then #3 drilling and initial level return
N008   G99X1200.0;                Positioning, then #4 drilling and point R level return
N009   Y–550.0;                   Positioning, then #5 drilling and point R level return
N010   G98Y–350.0;                Positioning, then #6 drilling and initial level return
N011   G00X0Y0M5;                 Reference position return, spindle stop
N012   G49Z250.0T15M6;            Tool length offset cancel, tool change
N013   G43Z0H15;                  Initial level, tool length offset
N014   S20M3;                     Spindle start
N015   G99G82X550.0Y–450.0        Positioning, then #7 drilling, point R level return
       Z–130.0R–97.0P300F70;
N016   G98Y–650.0;                Positioning, then #8 drilling, initial level return
N017   G99X1050.0;                Positioning, then #9 drilling, point R level return
N018   G98Y–450.0;                Positioning, then #10 drilling, initial level return
N019   G00X0Y0M5;                 Reference position return, spindle stop
N020   G49Z250.0T31M6;            Tool length offset cancel, tool change
N021   G43Z0H31;                  Initial level, tool length offset
N022   S10M3;                     Spindle start
N023   G85G99X800.0Y–350.0        Positioning, then #11 drilling, point R level return
       Z–153.0R47.0F50;
N024   G91Y–200.0K2;              Positioning, then #12, 13 drilling. point R level return
N025   G28X0Y0M5;                 Reference position return, spindle stop
N026   G49Z0;                     Tool length offset cancel
N027   M0;                        Program stop
```

NOTE	When the number of repeats is specified by K in G98 and G99, the tool returns to the initial level (G98) or point R level (G98) from the first hole.

14.2 Rigid Tapping

14.2.1 Basic operations in rigid tapping

In tapping, the feed along the Z–axis per rotation of the spindle must equal the pitch of the tap. Correct tapping always satisfies the following condition:

P = F/S where P : Pitch of tap (mm)
F : Feedrate along the Z–axis (mm/min)
S : Spindle speed (rpm)

The tapping cycle (G84) and reverse tapping cycle (G74) control the rotation of the spindle and the feed along the Z–axis independently of each other. This means that the condition described above is not always satisfied in actual tapping. At the bottom of a hole, both the rotational speed of the spindle and the feedrate along the Z–axis are reduced to zero, then increased in the reverse direction. As they are increased and decreased independently of each other, the condition described above is not necessarily satisfied. Usually a spring is mounted in the holder of the tap to provide correct feed and improve the precision of tapping.

The rigid tapping cycle (G84.2) and rigid reverse tapping cycle (G84.3) control the rotation of the spindle and the feed along the Z–axis so that they are synchronized. While the conventional tapping cycles control only the spindle speed, the rigid tapping cycles control both the speed and phase of the spindle. They control the rotation of the spindle and the feed along the Z–axis by applying linear interpolation to the spindle and Z–axis. This enables the condition of P = F/S to be satisfied even during deceleration and acceleration at the bottom of the hole. Precise tapping is thereby accomplished.

G84.2 specifies the rigid tapping cycle and G84.3 specifies the rigid reverse tapping cycle.

$$\left.\begin{matrix}\text{G84.2}\\ \\ \text{G84.3}\end{matrix}\right\}\ \text{X—}\ \ \text{Y—}\ \ \text{Z—}\ \ \text{R—}\ \ \text{P—}\ \ \text{F—}\ \ \text{S—}\ \ \text{L—}\ \ ;$$

L— : Number of times tapping is repeated

S— : Spindle speed

F— : Cutting feedrate

When the cutting feedrate is specified using a decimal point, the following values are assumed, depending on the settings of bits 3 (RFDA) and 7 (RFDE) of parameter No. 6201:

Specified value	F10	F10.0	F10.5
RFDA=0 RFDE=0	F10.0	F10.0	F10.0
RFDA=0 RFDE=1	F10.0	F10.0	F10.5
RFDA=1 RFDE=0 or 1	F10.0	Alarm PS007 (invalid use of decimal point)	

P— : Dwell time at return to points Z and R (when P is validated by a parameter)

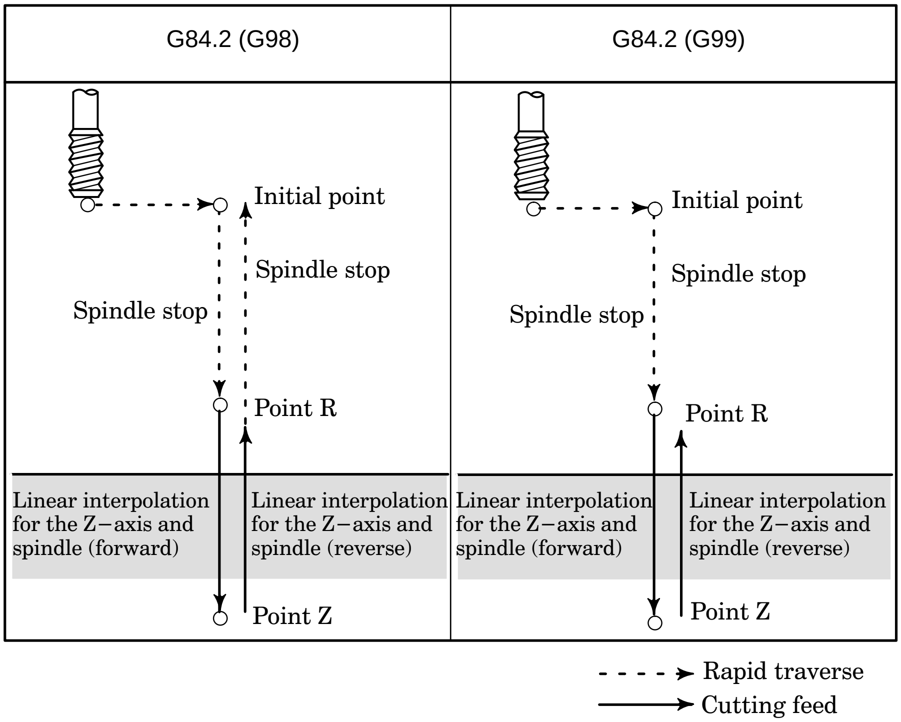
R— : Coordinate of point R

Z— : Coordinate of point Z

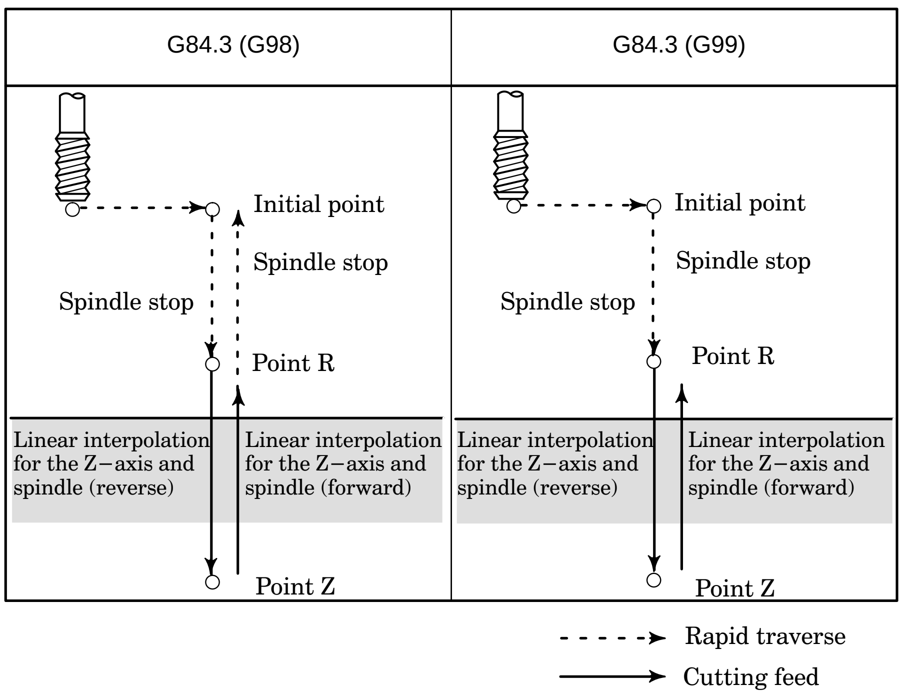
X— Y— : Point of drilling

When G84.2 or G84.3 is specified in the mode of feed per rotation (G95), the cutting feedrate after F must be specified in units of millimeters or inches per revolution. In other words, the pitch of the tap can be directly specified.

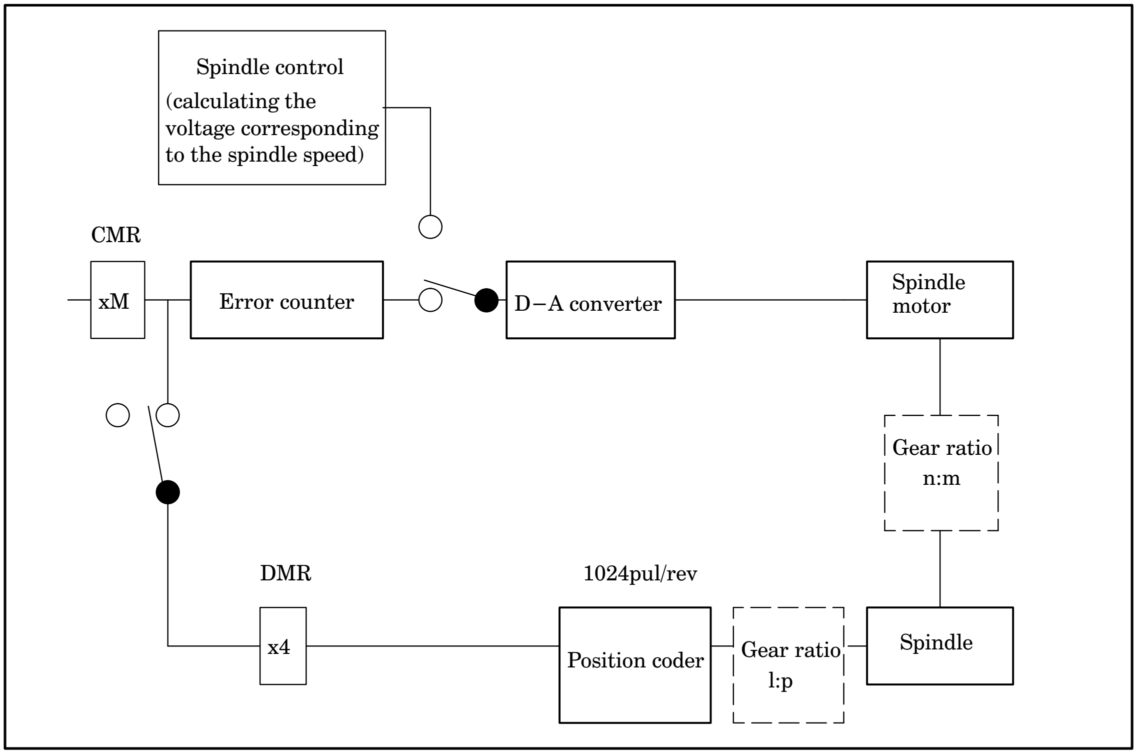
(1) G84.2 (Rigid tapping cycle)



(2) G84.3 (Rigid reverse tapping cycle)



NOTE The rigid tapping cycle (G84.2) and reverse rigid tapping cycle (G84.3) differ from the tapping cycle (G84) and reverse tapping cycle (G74) respectively in how the spindle is controlled when the tool moves between points R and Z along the Z–axis. In rigid tapping, the phase of the spindle is controlled by detecting the angular displacement of the spindle according to the number of pulses output from of the position coder as shown below. The spindle motor is controlled like a servo motor. Linear interpolation is applied to the Z–axis and spindle.



The table below shows the traveling distance and feedrate in linear interpolation of the Z–axis and spindle.

	Traveling distance	Feedrate
Z–axis	z =Distance between points R and Z (mm, inch)	Fz=Specified F value (mm/min, inch/min)
Spindle	s=zx(Specified S value/Specified F value) x360 (deg)	Fs=Specified S value (rpm)

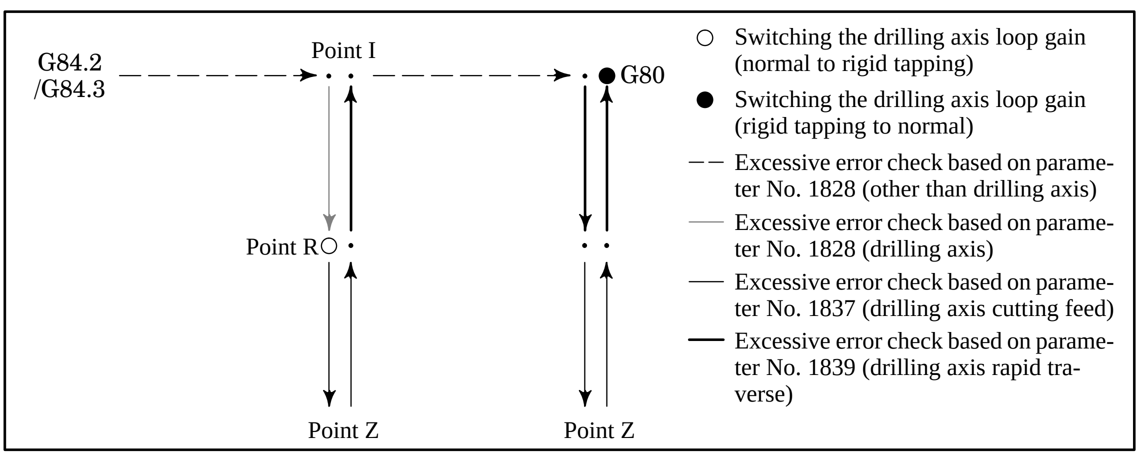
The rigid tapping cycles control the spindle in the same manner as the G84 and G74 cycles, except when the tool moves between points R and Z along the Z–axis. For details of programming that are not described here, see the section on the corresponding canned cycle.

NOTE 10 In the above explanation, the drilling axis is called the Z–axis. It can also be the X– or Y–axis.
Example G18G84.2X––Z––Y––R––F––; (The Y–axis is the drilling axis.)

NOTE 11 The feedrate override is fixed to 100% during cutting. During return, however, it can be varied within a range of 10% to 200% (in steps of 10%), using parameter No. 5705.

NOTE 12 The maximum position deviation of the drilling axis in rigid tapping mode can be set separately from the maximum position deviation during rapid traverse and that during cutting feed.

Checking for excessive error is based on the following maximum position deviation parameters:



14.2.2 Additional functions for rigid tapping

(1) Gear ratio and gear selection

– The gears between the spindle and position coder for rigid tapping can be selected from the following three types:

Type I : Only a single gear ratio can be used between the spindle and position coder. It is fixed to 1:1, 2:1, 4:1, or 8:1.

Type II : Only a single gear ratio can be used between the spindle and position coder. It can be set by the user as required.

Type III : Up to eight gear ratios can be used between the spindle and position coder when an analog spindle is used. Up to four gear ratios can be used when a serial spindle is used. The gear ratios can be set by the user as required.

Select type I, II, or III by using parameters.

Type	Parameters used to select type	Parameters used to set gear ratio	
		M/T/M multiaxis/TT first spindle	TT second spindle
I	No. 5604 bit 1 (RTG) = 0 No. 5604 bit 2 (SFGB) = 0	No. 5610	No. 5660
II	No. 5604 bit 1 (RTG) = 1 No. 5604 bit 2 (SFGB) = 0	No. 5703: Spindle gear No. 5704: Position coder gear	No. 5909: Spindle gear No. 5910: Position coder gear
III	No. 5604 bit 2 (SFGB) = 1	Analog spindle Nos. 5771 to 5778: Spindle gear Nos. 5781 to 5788: Position coder gear Serial spindle Nos. 5771 to 5774: Spindle gear Nos. 5781 to 5784: Position coder gear	Analog spindle Nos. 5929 to 5936: Spindle gear Nos. 5937 to 5944: Position coder gear Serial spindle Nos. 5929 to 5932: Spindle gear Nos. 5937 to 5940: Position coder gear

(2) Orientation

In a G84.2/G84.3 block, an I command can be specified to perform spindle orientation at the first R point. The following restrictions are, however, imposed on the use of this command:

– Orientation is valid only for the first block in rigid tapping mode. The orientation specified in the second and subsequent blocks in rigid tapping mode is ignored.
– Orientation is valid only when gear type I has been selected. If orientation is specified when type II or III has been specified, alarm PS534 is issued.
– The specification of the orientation differs slightly between analog and serial spindles.

1) Analog spindle

a) Command

G84.2/G84.3 I0 ; Orientation in positive direction
G84.2/G84.3 I–0 ; Orientation in negative direction

b) Operation and speed

When the orientation command is specified, tool movement is started at the rapid traverse rate, specified in parameter No. 5801 for the M series, T series, M series multiaxis, or TT series first spindle, or in parameter No. 5953 for the TT series second spindle. Once the one–rotation signal has been detected from the position coder, the rapid traverse rate is automatically reduced to the FL speed, specified in parameter No. 5802 for the M series, T series, M series multiaxis, or TT series first spindle, or in parameter No. 5954 for the TT series second spindle. Then, the spindle returns to the reference position.

Rapid traverse rate F must satisfy the following condition. If the condition is not satisfied, the system cannot detect the one–rotation signal from the position coder and, therefore, continues rapid traverse.

$$1024 \geqq \left(\frac{F \times P}{60 \times G \times 0.088} \right) \geqq 128 \text{ (pulse)}$$

F : Rapid traverse rate (deg/min)
G : Loop gain (1/s)
P : Gear ratio between spindle and position coder

FL speed FL must satisfy the following condition.

$$1024 \geqq \left(\frac{FL \times P}{60 \times G \times 0.088} \right) \text{ (pulse)}$$

FL : FL speed (deg/min)
G : Loop gain (1/s)
P : Gear ratio between spindle and position coder

c) Grid shift

The orientation position can be shifted within a range of ±180 degrees. Use parameter No. 5803 for the M series, T series, M series multiaxis, or TT series first spindle, or parameter No. 5955 for the TT series second spindle.

2) Serial spindle

a) Command

$$\left.\begin{array}{l}\text{G84.2/G84.3 I0 ;}\\ \text{G84.2/G84.3 I–0 ;}\end{array}\right\}\ \text{Performs orientation}$$

$$\left.\begin{array}{r}\text{M/T/M multiaxis/TT first spindle: No.3000 bit 4 (RETSV)}\\ \text{TT second spindle: No.3140 bit 4 (RETSV)}\end{array}\right\}\begin{array}{l}\text{=0 Orientation in the positive direction}\\ \text{=1 Orientation in the negative direction}\end{array}$$

b) Operation and speed

The orientation speed depends on the spindle.
For an analog spindle, executing an orientation command first moves the tool at the rapid traverse rate, then at the FL speed, and finally stops the tool at the orientation position. For a serial spindle, however, executing the orientation command immediately stops the tool at the orientation position. The maximum orientation speed, which is imposed on an analog spindle, is not, therefore, applied to a serial spindle.

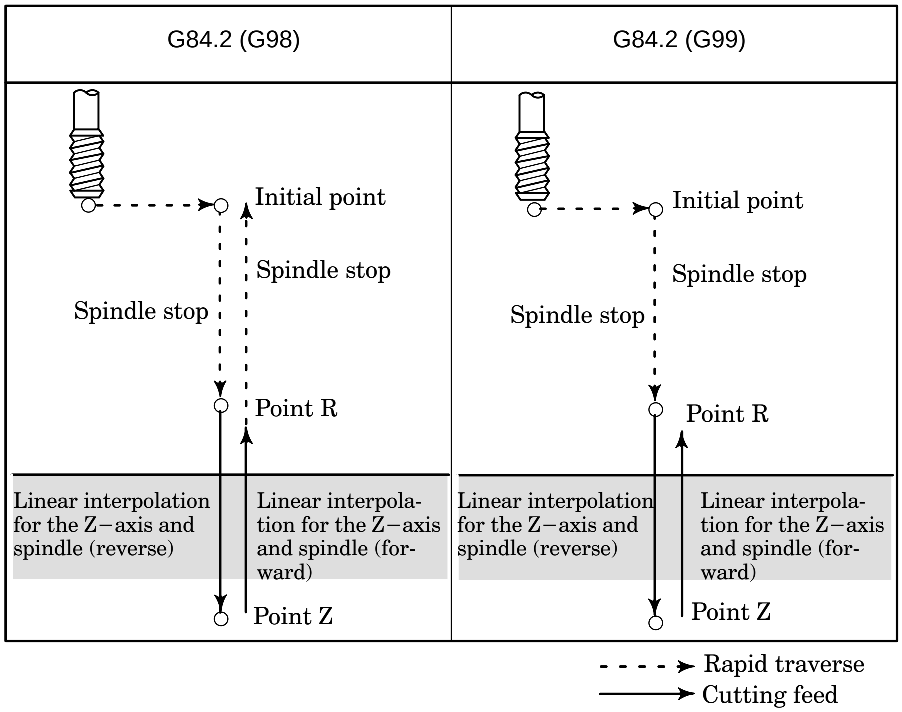
c) Grid shift

The orientation position can be shifted within a range of 0 to 360 degrees. Use parameter No. 3073 for the M series, T series, M series multiaxis, or TT series first spindle, or parameter No. 3213 for the TT series second spindle.

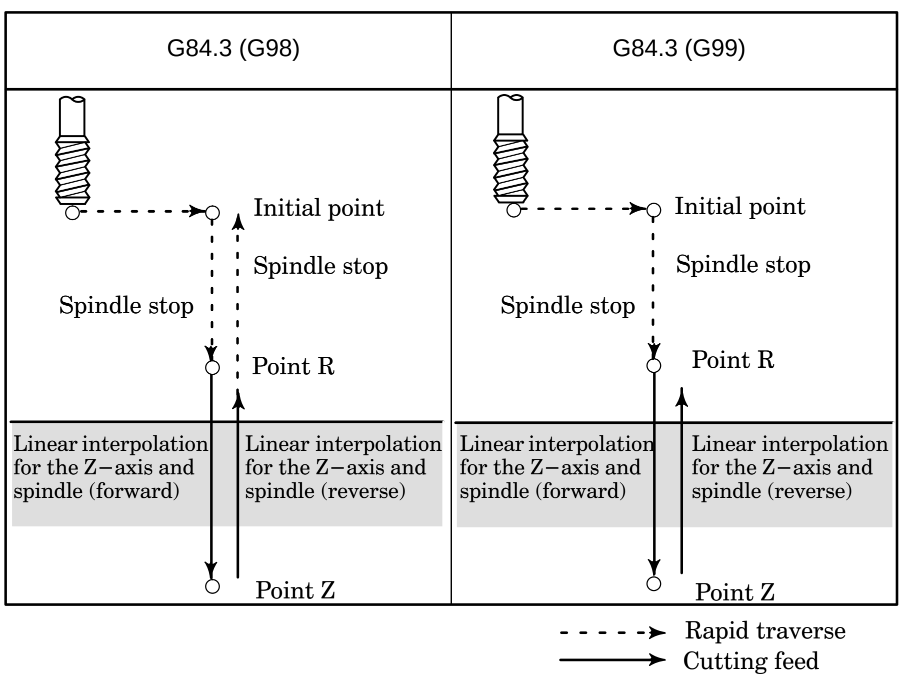
(3) Spindle rotation direction reversing function

When a rigid tapping cycle (G84.2) or rigid reverse tapping cycle (G84.3) command is executed in canned cycle cancel mode (G80), the direction of spindle rotation during rigid tapping is determined from the state of the spindle rotation direction reversing signal (RSPR).

1) G84.2 (when the spindle rotation direction reversing signal (RSPR) is high)



2) G84.3 (when the spindle rotation direction reversing signal (RSPR) is high)



NOTE Switching of the spindle rotation direction reversing signal (RSPR) is disabled in rigid tapping mode. To switch the direction of spindle rotation, execute a cancel command (G80), switch the spindle rotation direction reversing signal (RSPR), then execute a G84.2 or G84.3 command.

Example

```
.........................
N10 G91 G84.3 X10. Z10. R50. F100. ;
N11 X20. Z–50. R–10. ;
N12 X30. Z–50  R–10. ;
.........................
N20 G80 ;
N21 G84.2 X10. Z–50. R–10. F100. ;
```

During execution of the above program, the state of the spindle rotation direction reversing signal (RSPR) is detected upon the execution of N10 and N21. Switching of the RSPR signal in any other block is not detected until N10 or N21 is executed.

14.2.3 Pecking cycle function in rigid tapping

In rigid tapping, it may be difficult to form a deep hole as the chips snare the tool and the cutting resistance increases. The pecking cycle function can be used to accomplish tapping from point R to Z through repeated operations.
The pecking cycle function is executed when the Q command with the depth of each cut is specified in the conventional rigid tapping command.
Figure 14.2.3 shows the procedure for rigid tapping in the pecking cycle.

q:Depth of cut d:Amount of return

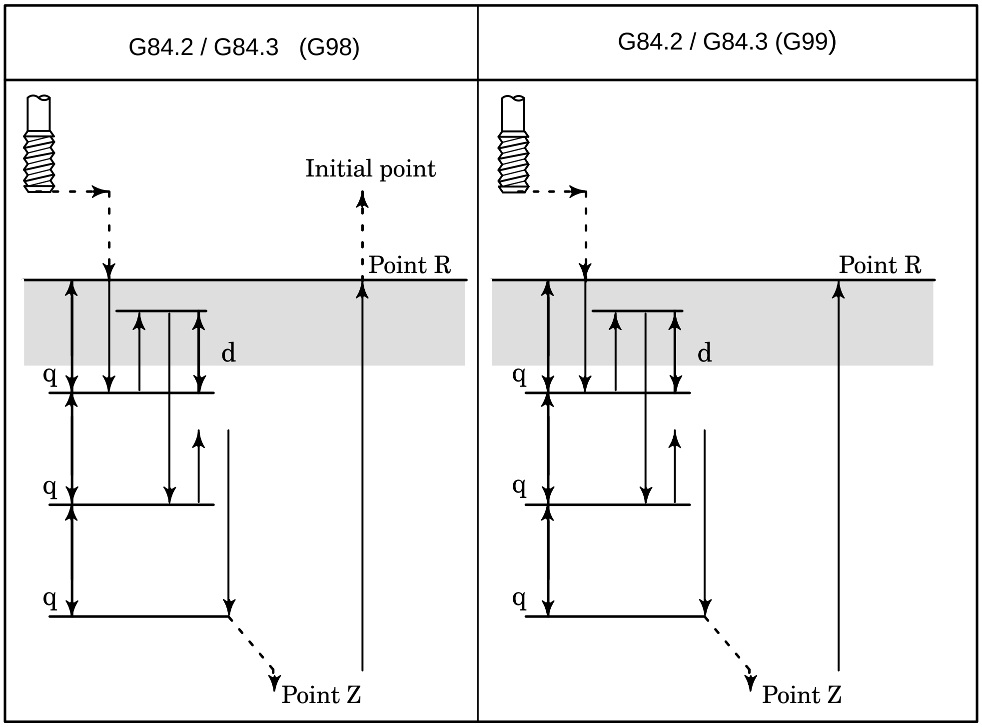


Fig. 14.2.3 Pecking Cycle in the G84.2 or G84.3 Mode

When the tap moves in the negative direction along the Z–axis during tapping as shown in the figure above, the direction in which the spindle rotates depends on the mode as follows:
G84.2: The spindle rotates forward during cutting feed from point R to Z and in reverse during movement from point Z to R.
G84.3: The spindle rotates in reverse during cutting feed from point R to Z and forward during movement from point Z to R.

Program format

G84.2 (G84.3) X––Y––Z––R––P––Q––F––S––L––;

X,Y : Position of the hole
Z : Distance from point R to the bottom of the hole, position of the bottom of the hole
R : Distance from the initial level to point R
P : Dwell time at points Z and R (when P is validated by the parameter)
Q : Depth of each cut. An incremental value must be specified. When Q is not specified or when Q0 is specified, the pecking cycle is not executed.
F : Cutting feedrate
S : Spindle speed
L : Number of times the operation is repeated

NOTE 1 The amount of each return (d) is specified by parameter No. 6221.
NOTE 2 The speed of each return can be overridden in the range of 10% to 200% in units of 10%, according to the value specified in parameter No. 5705.
NOTE 3 In rigid tapping, the overridden speed of return is also used when the tool retreats from point Z to R.
NOTE 4 The pecking cycle function is executed for rigid tapping when the Q command is specified in the G84.2 or G84.3 block or in the mode of rigid tapping cycle. If Q0 is specified, the pecking cycle is not executed.

14.2.4 Three–dimensional rigid tapping

When the machine is provided with axes for swiveling the tool, this function allows rigid tapping in the direction in which the tool is pointing after the tool is swiveled about the specified axes.

This three- dimensional rigid tapping function must be used together with the three- dimensional coordinate conversion function (G68) .

Format

The command for three- dimensional rigid tapping is specified in the same format as for usual rigid tapping. Refer to 14. 2. 1.

```
G84. 2 }
       }  G19  X- -  Y- -  Z- -  R- -  P- -  F- -  S- -  L- -  ;
G84. 3 }
```

Example

```
G68  X0  Y0  Z0  I1  J0  K0  R60.  ;     ——————————————    (1)
G68  X0  Y0  Z0  I0  J1  K0  R30.  ;     ——————————————    (1)'
G90  G00  A60.  ;                        ——————————————    (2)
G90  G00  B30.  ;                        ——————————————    (2)'
X100.  Y100.  Z100.  ;
G91  G84. 2  X10.  Y10.  Z- 50.  R- 20.  F12. 345  S1000 —————    (3)
;                                        ——————————————    (4)
X20.  ;                                  ——————————————    (5)
Y20.  ;                                  ——————————————    (6)
X- 20.  ;                                ——————————————    (7)
G80  ;
G90  G00  X0  Y0  Z0  ;                  ——————————————    (8)
G69  ;
```

(1) Rotate the coordinate system about the X-axis 60_ by using three-dimensional coordinate conversion.

(1)' Rotate the coordinate system about the Y-axis 30_ by using three- dimensional coordinate conversion.

(2) Rotate the A axis 60_.

(2)' Rotate the B-axis 30_.

(3) Enter the rigid tapping mode. Perform positioning, then tap hole #1.

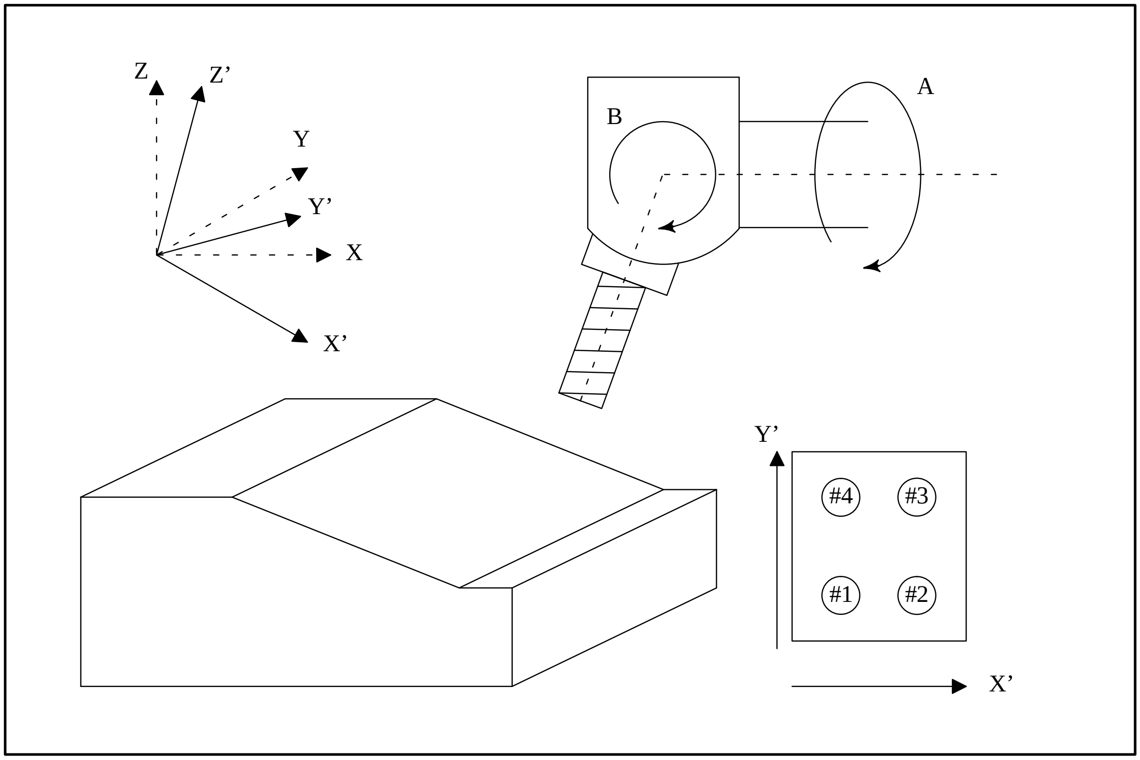
(4) Perform positioning, then tap hole #2.

(5) Perform positioning, then tap hole #3.

(6) Perform positioning, then tap hole #4.

(7) Cancel the rigid tapping mode.

(8) Cancel the three- dimensional coordinate conversion mode.



Notes

(1) To change the tapping direction when rigid apping is being performed in the specified direction, specify G80 to cancel the rigid tapping mode, then specify rigid tapping again.

(Example of a program)

```
G68  X0  Y0  Z0  I0  J1  K0  R30.   ;
 .    .    .    .    .    .    . ;
G91  G84. 2  X10.   Y10.   Z- 50.   R- 20.   F12. 345  S1000  ;

X20.   ;
G80  ;  — — — — — — — — — — — — — — —   Cancel the rigid tapping mode
G68  X0  Y0  Z0  I0  J0  K1  R60.   ;
 .    .    .    .    .    .    . ;
G91  G84. 2  X10.   Y10.   Z- 50.   R- 20.   F12. 345  S1000  ;

Y20.   ;
X- 20.   ;
G80  ;
G69  ;
```

(2) When 15–MB multi–axis system is used, three- dimensional rigid tapping cannot be performed during simplified synchronization control or twin table control.

14.3 Function for External Operations

G81IP– –L– –;

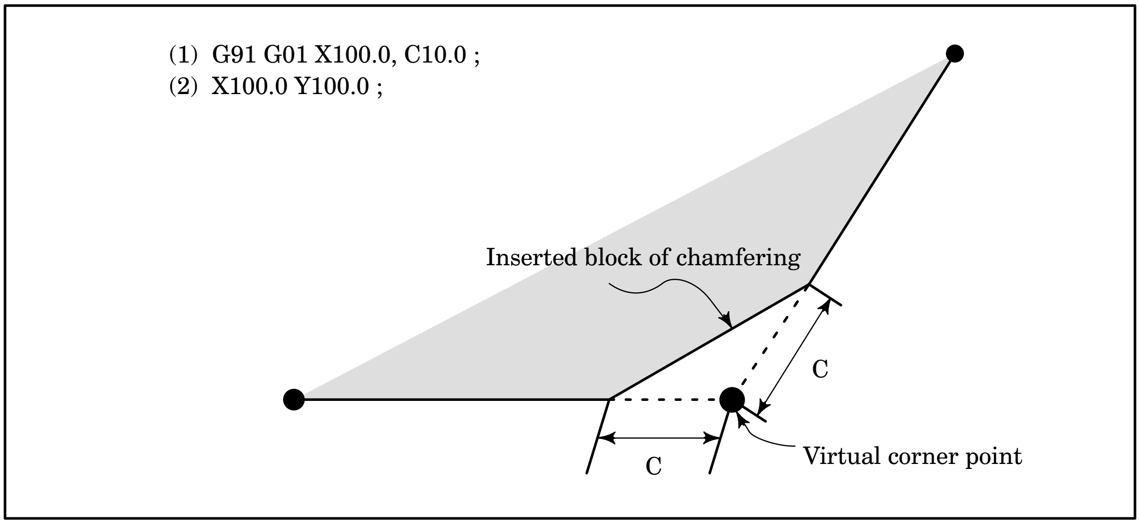
When this command is specified, the NC unit sends an external operation signal to the machine after each IP positioning is completed. The machine executes clamping, drilling, punching, or another operation or cycle specified by the signal. Until the function is canceled by the G80 command, the NC unit continues sending the signal after each positioning. At reset, the system enters the G80 mode. When the power is turned on, the system enters the G80 cancel mode.

The numeric value following address L indicates the number of times positioning is repeated. The external operation signal is sent after each positioning. In a block which does not contain a move command, the external operation signal is not sent. The user can select whether G81 is used as the external operation function or a canned cycle.

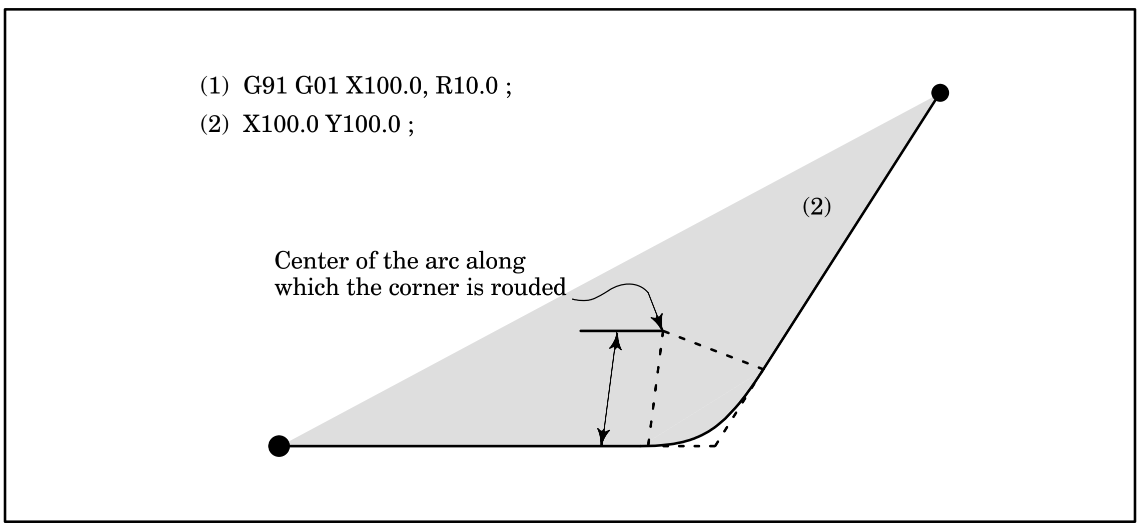
14.4 Chamfering an Edge at a Desired Chamfer Angle and Rounding a Corner

A block for chamfering or corner rounding can be automatically placed between two blocks of linear interpolation or circular interpolation or between a block of linear interpolation and a block of circular interpolation.

To place the block for chamfering, specify, **C__** at the end of the preceding block of linear interpolation (G01) or circular interpolation (G02 or G03). To place the block of corner rounding, specify , **R__**.
The numeric value following C indicates the distance from the start or end point of chamfering to the virtual corner point where the two sides of the original corner meet. (See the figure below.)



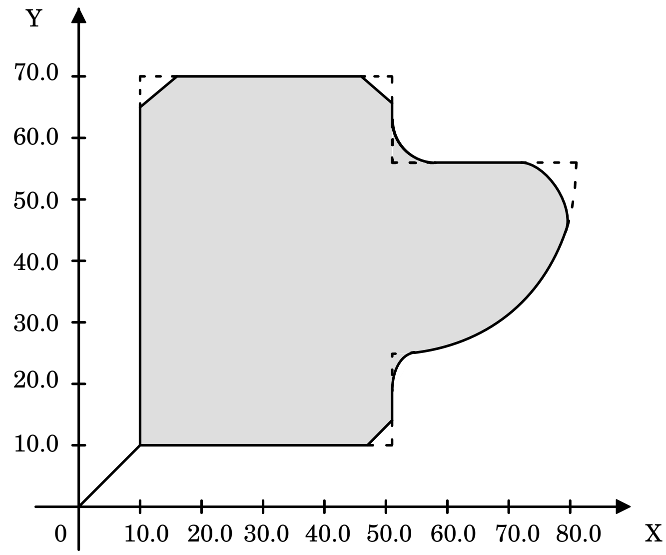
The numeric value following R indicates the radius of the arc along which the corner is rounded. (See the figure below.)



The block of chamfering or corner rounding specified as described above must be followed by a block containing a move command for linear interpolation (G01) or circular interpolation (G02 or G03). If another block is specified, P/S alarm 431 occurs.

Two or more blocks of chamfering or corner rounding can be specified in succession. (See the sample program below.)

```
N001 G92 G90 X0 Y0 ;
N002 G00 X10.0 Y10.0 ;
N003 G01 X50.0 F10.0, C5.0 ;
N004 Y25.0, R8.0 ;
N005 G03 X80.0 Y50.0 R30.0, R8.0 ;
N006 G01 X50.0, R8.0 ;
N007 Y70.0, C5.0 ;
N008 X10.0, C5.0 ;
N009 Y10.0 ;
N010 G00 X0 Y0 ;
N011 M0 ;
```



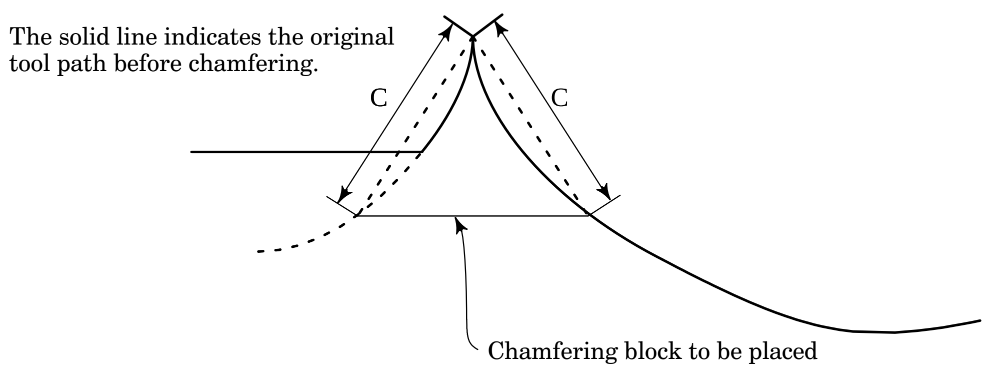
NOTE 1 The block of chamfering or corner rounding can be placed only between blocks specifying movement on an identical plane. When the XY plane (G17) is selected, for example, the block for chamfering or corner rounding causes a movement on the XY plane. The block for chamfering or corner rounding cannot be specified immediately after the plane is switched, that is, it cannot follow the block containing the G17, G18, or G19 command.

NOTE 2 If the block for chamfering or corner rounding causes movement beyond the original traveling range determined by linear interpolation, P/S alarm 429 occurs. (See the figure below.)

(1) G91 G01 X30.0 ;

(2) G03 X7.5 Y16.0 R37.0, C28.0 ;

(3) G03 67.0 Y–27.0 R55.0 ;



NOTE 3 When blocks containing the commands for chamfering or corner rounding are executed in the single–block mode, the operation is continued up to the end of the inserted block of chamfering or corner rounding, then the machine stops.

NOTE 4 The system ignores the command of chamfering

(, C__) or corner rounding (, R__) if it is specified between blocks other than linear interpolation (G01) or circular interpolation (G02 or G03).

NOTE 5 The command of chamfering or corner rounding cannot be specified in the block following the command for changing the coordinate system (G92 or G52 to G59) or return to the reference position (G28 to G30).

NOTE 6 When the following conditions are satisfied, the block of chamfering or corner rounding is processed as a block whose traveling distance is zero: The block is placed between two blocks of linear interpolation. The angular difference between the two lines is within ±1°.

When the following conditions are satisfied, the block for corner rounding is processed as a block whose traveling distance is zero: When the block is placed between a block for linear interpolation and a block for circular interpolation. The angular difference between the straight line and the tangent of the arc at the intersection of the straight line and the arc is within ±1°.

When the following conditions are satisfied, the block for corner rounding is processed as a block whose traveling distance is zero: When the block is placed between two blocks for circular interpolation. The angular difference between the tangents of the arcs at the intersection is within ±1°.

14.5 Programmable Mirror Image (G50.1 and G51.1)

A program command can produce the mirror image of a figure on the other side of an axis. The conventional mirror image (specified by an external switch or a parameter) is formed after this programmable mirror image is formed.

(1) Specifying the programmable mirror image function

G51.1X––Y––Z––––;

This command produces a mirror image as if a mirror is placed on the specified axis.

(2) Canceling the programmable mirror image function

G50.1X––Y––Z––––;

This command cancels the programmable mirror image function specified for the axis. The values speci fied in this command do not matter.

WARNING 1 When a mirror image is produced on the other side of one axis on a specified plane, other commands are processed as follows:

(1) Circular command: The direction of rotation is reversed.

(2) Cutter compensation: The direction of offset is reversed.

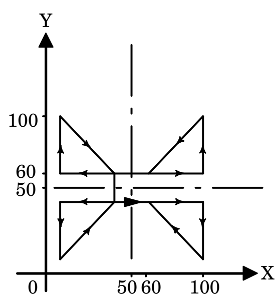
(3) Coordinate system rotation: The direction of rotation is reversed.

WARNING 2 Relation between move command and machine travel/coordinate value is as follows in the programmable mirror image.

Move command	Machine travel	Machine coordinate	Relative coordinate	Absolute cordinate
+	–	–	–	–
–	+	+	+	+

WARNING 3 The first move command coming after G50.1 or G51.1 must be specified with absolute values.

If the contour of the workpiece to be machined is symmetrical about an axis, use the programmable mirror image function and subprograms. The entire contour can be produced by programming only a part of it.



○Subprogram
O9000 ;
G00 G90 X60.0 Y60.0 ;
G01 X100.0 Y60.0 F100 ;
G01 X100.0 Y100.0 ;
G01 X60.0 Y60.0 ;
M99

○Main program
N10 G00 G90 ;
N20 M98 P9000 ;
N30 G51.1 X50.0 ;
N40 m98 P9000 ;
N50 G51.1 Y50.0
N60 M98 P9000 ;
N70 G50.1 X0 ; (only cancels operations for the X–axis.)
N80 M98 P9000 ;
N90 G50.1 Y0 ;

NOTE	G50.1 and G51.1 must not be specified in the G68 or G51 mode.

14.6 Indexing Function

The index table of the machining center can be indexed by specifying an axis of indexing. The index command only requires an address, which is the programmed name of the indexing axis (X, Y, Z, A, B, C, U, V, or W), and the angle of indexing. No special M codes are required for clamping or unclamping the table. Indexing can be programmed easily.

(1) Specifying the indexing command

(a) Minimum angle of indexing

The table can be moved in increments of the minimum angle of indexing specified in the parameter. If the specified angle of movement is not a multiple of the minimum indexing angle, P/S alarm No. 198 occurs.

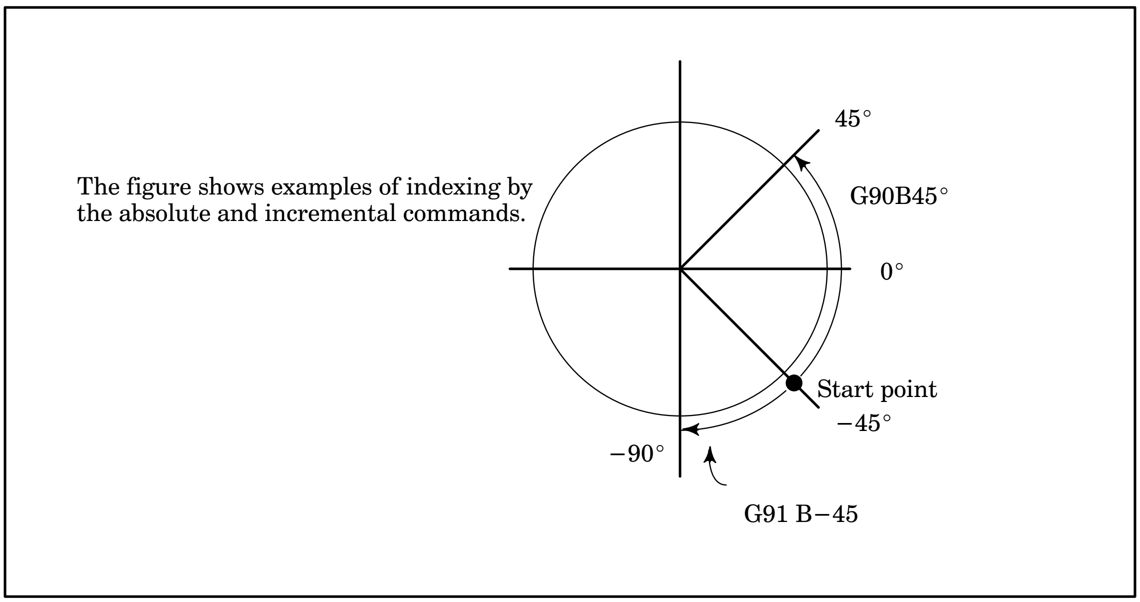
(b) Absolute or incremental value

When G90 is specified, an absolute value is specified. When G91 is specified, an incremental value is specified.

Example When the current value is –45°

Absolute command G90 B45°; provides the position of 45°.

Incremental command G91 B–45°; provides a position after allowing rotation of 45° in the negative direction.



(c) Number of axes to be controlled simultaneously

The indexing function must always be specified for only one axis. If the indexing axis and another control axis are specified at one time, PS alarm No. 197 occurs.

(2) Minimum indexing angle

0.01 degrees (IS–A)
0.001 degrees (IS–B)

(3) Feedrate

The indexing axis is positioned at a rapid traverse feedrate, irrespective of the specified G code of group 01 (G00, G01, G02, or G03).

If the indexing axis is specified in the G01, G02, or G03 mode, the mode becomes valid for another axis in subsequent blocks. G01, G02, or G03 need not be specified again.

G01 X10. F5;	....	The tool moves along the X–axis at the cutting feedrate.
B45.;	....	The B–axis is used for indexing at the rapid traverse feedrate.
X20.;	....	The tool moves along the X–axis at the cutting feedrate. (G01 becomes effective.)

The dry run mode is invalid for the indexing axis.

(4) Clamping or unclamping the index table

At the beginning and end of movement of the indexing table, the index table is automatically clamped and unclamped.

NOTE 1 If the system is reset while the table is moving, a return to the reference position must be executed before starting the next indexing.
NOTE 2 If the system is reset while the CNC unit is waiting for the table to be clamped or unclamped, the clamp or unclamp signal is cleared. The CNC unit exits the waiting status.
NOTE 3 No movements can be made by the following functions: automatic return from the reference position (G29), return to the second reference position (G30), and selection of the machine coordinate system (G53).

(5) Jog/step/handle

The indexing axis cannot be operated in the jog, step, or handle mode. However, manual return to the reference position is possible.

When the axis selection signal is set to 0 in manual return to the reference position, the table immediately stops and the clamp command is not issued. If this situation is inconvenient, set the axis selection signal to 1. Then adjust the machine so that the signal is not set to 0 before return to the reference position is completed.

(6) Parameters

The following parameters are used for indexing:

Parameter Nos.: 7602, 7631, 7632, and 7682

For details, see Appendix 7.

WARNING 1 The feed hold, interlock, and emergency stop functions are valid while the table is moving. If the machine does not allow a stop at an arbitrary position, the necessary actions must be taken by the machine.
WARNING 2 The machine lock function is valid. If the function is set on during indexing, it continues to be ignored until indexing completes.

NOTE 1 The indexing axis must be a rotation axis.
NOTE 2 Secondary miscellaneous functions can be executed. The address specified for the function must not agree with the address of the indexing axis.
NOTE 3 Specify 1 for the following bits in parameter No. 1006:
ROTx : Specifies whether the inch or metric mode need not be switched for the axis (rotation axis).
ROSx : Specifies whether the machine coordinate system for stroke check and automatic return to the reference position uses a linear axis or rotation axis.
ROPx : Specifies whether the machine coordinate system for stored pitch error compensation uses a linear axis or rotation axis.
NOTE 4 Values such as the current position on the CRT screen and the indication on the external position indicator are all indicated with a decimal point.
Example B180.000
NOTE 5 The servo off signal must be invalidated for the indexing axis.
Parameter SVFx (No. 1820)
The indexing axis is usually in the servo–off state.
NOTE 6 When the incremental command is used for indexing (when parameter No. 7603, bit #3, in G90 is set to 0), the offset from the workpiece reference point of the index table axis must always be zero. That is, the machine coordinate system of the index table axis must always agree with the workpiece coordinate system.
NOTE 7 If the traveling distance is zero, the table is not clamped or unclamped. In automatic return to the reference position (G28), the table is clamped or unclamped even if the traveling distance is zero.
NOTE 8 The unidirectional positioning command (G60) cannot be specified.

14.7 Figure Copy Function

A figure specified by a subprogram can be repeatedly produced with rotation or parallel movement. G17, G18, or G19 selects the plane on which the figure is copied.

(1) Copying a figure with rotation

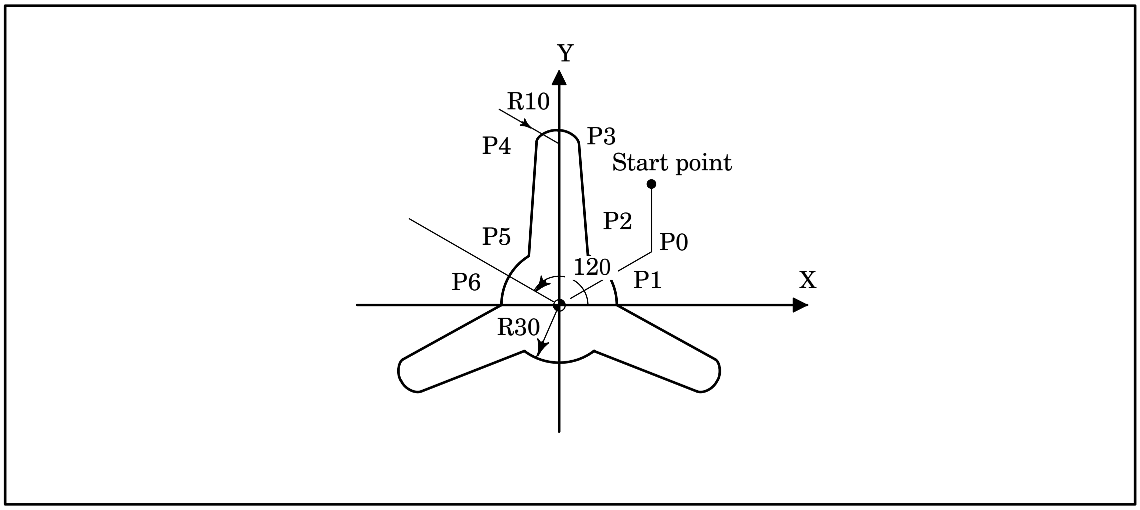
The following commands repeatedly rotate and copy the figure specified by the subprogram.

Formats

G72.1 P_ L_ Xp_ Yp_ R_; Xp–Yp plane
G72.1 P_ L_ Zp_ Xp_ R_; Zp–Xp plane
G72.1 P_ L_ Yp_ Zp_ R_; Yp–Zp plane

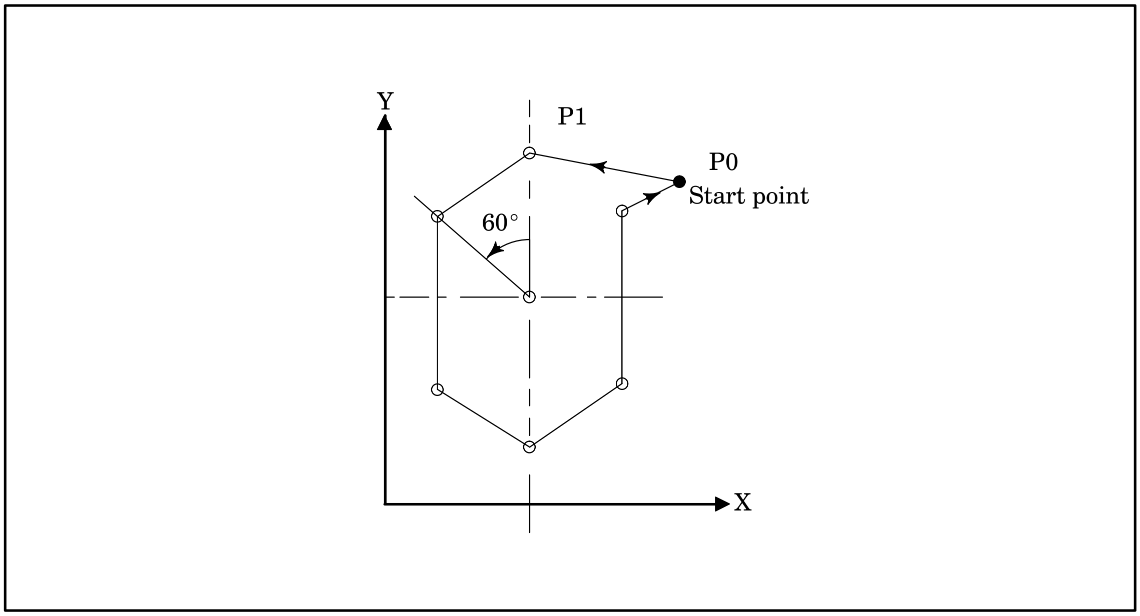
P : Subprogram number
L : Number of times the figure is copied
Xp : Xp coordinate of the center of rotation (Xp: X–axis or an axis parallel to the X–axis)
Yp : Yp coordinate of the center of rotation (Yp: Y–axis or an axis parallel to the Y–axis)
Zp : Zp coordinate of the center of rotation (Zp: Z–axis or an axis parallel to the Z–axis)
R : Angular displacement (A positive value indicates counterclockwise rotation. Specify it with an absolute value.)

Sample program 1 for copying a figure with rotation



J Main program		Subprogram	
O1000 ;		O2000 G03 X__Y__ R30.0 ;	(P2)
N10 G92 X40.0 Y50.0 ;		N100 G01 X__Y__ ;	(P3)
N20 G00 G90 X__Y__;	(P0)	N200 G03 X__Y__ R10.0 ;	(P4)
N30 G01 G17 G41 X__Y__ D01 F10 ;	(P1)	N300 G01 X__Y__;	(P5)
N40 G72.1 P2000 L3 X0 Y0 R120.0 ;		N400 G03 X__Y__ R30.0 ;	(P6)
N50 G40 G01 X__Y__I __J__;	(P0)	N500 M99 ;	
N60 G00 X40.0 Y50.0 ;			
N70 M30 ;			

Sample program 2 for copying a figure with rotation (bolt hole circle)



Main program		Subprogram
O3000 ;		O4000 N100 G90 G81 X__Y__R__
N10 G92 G17 X80.0 Y50.0 ;	(P0)	Z__ F__ ; (P1)
N20 G72.1 P4000 L6 X0 Y0 R60.0 ;		N200 M99 ;
N30 G80 G00 X80.0 Y50.0 ;	(P0)	
N40 M30 ;		

(2) Copying a figure with parallel movement

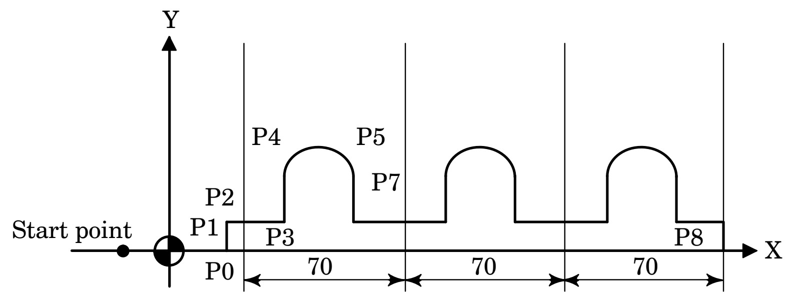
The following commands repeatedly copy the figure specified by the subprogram with parallel movement. G17, G18, and G19 select the plane on which the figure is copied.

Formats

G17 G72.2 P_ L_ I_ J_; Xp–Yp plane
G18 G72.2 P_ L_ K_ I_; Zp–Xp plane
G19 G72.2 P_ L_ J_ K_; Yp–Zp plane

P : Subprogram number
L : Number of times the figure is copied
I : Shift in the Xp direction
J : Shift in the Yp direction
K : Shift in the Zp direction

Sample program for copying a figure with parallel movement



Main program		Subprogram	
O1000 ;		O2000 G90 G01 X__ ;	(P3)
N10 G92 X–20.0 Y0 ;		N100 Y__ ;	(P4)
N20 G00 G90 X0 Y0 ;		N200 G02 X__ I__ ;	(P5)
N30 G01 G17 G41 X__ Y__ D01 F10 ;	(P0)	N300 G01 Y__ ;	(P6)
N40 Y__ ;	(P1)	N400 X__ ;	(P7)
N50 X__ ;	(P2)	N500 M99 ;	
N60 G72.2 P2000 L3 I70.0 J0 ;			
N70 X__ Y__ ;	(P8)		
N80 X0 ;			
N90 G00 G40 X–20.0 Y0 ;			
N100 M30 ;			

Notes on the figure copy function

WARNING 1 When a subprogram is used to copy a figure with rotation or parallel movement, the first block of the subprogram must contain a move command. If the first block contains only a program number (for example, O1234;), the movement may stop at the start point of the nth figure copied (n = 1, 2, 3, ...). Specify the first move command in the absolute mode.

(Wrong subprogram)	(Correct subprogram)
O1000 ;	O1000 G00 G90 X100. Y200.0 ;
G00 G90 X100.0 Y200.0 ;	··· ;
··· ;	M99 ;
M99	

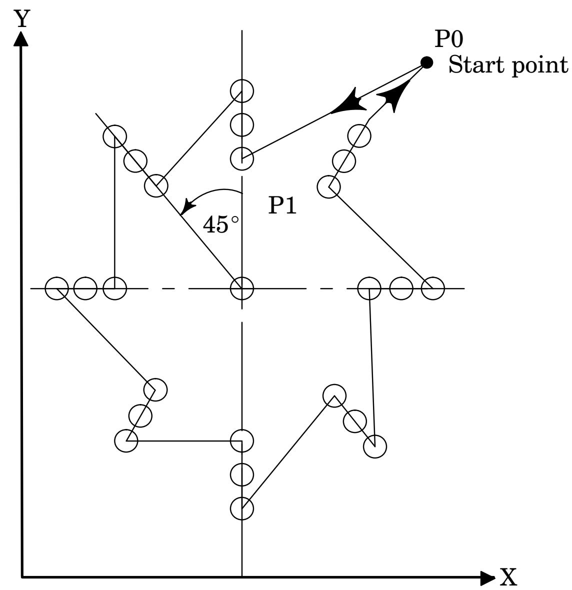
WARNING 2 The coordinate of the center of rotation specified for G72.1 is handled as an absolute value even if it is specified in the incremental mode.

WARNING 3 Two axes on the plane on which the figure is copied must use an identical unit system.

NOTE 1 G72.1 must not be specified in the subprogram in which G72.1 is specified to rotate and copy a figure. G72.2 must not be specified in the subprogram in which G72.2 is specified to copy a figure with parallel movement. The command for copying a figure with parallel movement can be specified in the subprogram that copies a figure with rotation, and vice versa. M98 that calls a subprogram and G65 that calls a macro can be specified in the subprogram that copies a figure with rotation or parallel movement.

Sample program that copies a figure with rotation and parallel movement: Bolt hole circle

Main program		Subprogram (that copies a figure with rotation)
O1000 ;		O2000 N100 G72.2 P3000 I0 J· L3 ;
N10 G92 G17 X100.0 Y80.0 ;	(P0)	N200 M99 ;
N20 G72.1 P2000 X0 Y0 L8 R45.0 ;		
N30 G80 G00 X100.0 Y80.0 ;	(P0)	Subprogram
		(that copies a figure with parallel movement)
N40 M30 ;		O3000 N110 G90 G81 X_Y_ R_ Z_ F_ ; (P1)
		N210 M99 ;



NOTE 2 In the G72.1 block, addresses other than P, L, Xp, Yp, Zp, and R are ignored. A subprogram number (P), coordinate of the center of rotation (Xp, Yp, Zp), and angular displacement (R) must be specified. In the G72.2 block, addresses other than P, L, I, J, and K are ignored. A subprogram number (P), and shift (I, J, K) must be specified.

NOTE 3 If a subprogram having the number specified at address P is not detected, P/S alarm No. 076 occurs.

NOTE 4 If L is not specified, the figure is copied once. In other words, the subprogram is called only once.

NOTE 5 In the G72.1 block, specify an increment in the angular displacement at address R. The angular displacement (degree) for the Nth figure is calculated as follows: R×(N–1). In the G72.2 block, specify an increment for the shift (I, J, K). The shift for the Nth figure is calculated as follows: (specified shift)×(N–1).

NOTE 6 Like M98, calling a subprogram by G72.1 or G72.2 is assumed to have a nesting level of 1.

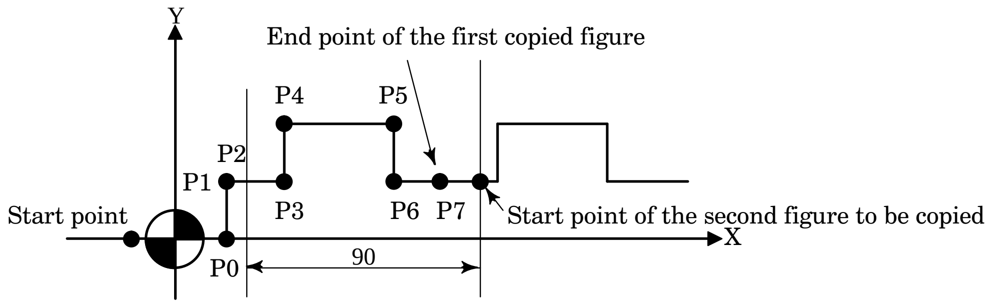
NOTE 7 The following commands must not be specified while the system is copying a figure with rotation or parallel movement:

- Polar coordinates command
- Return to the reference position
- Coordinate system rotation, scaling, programmable mirror image. (The command for copying a figure with rotation or parallel movement can be specified after the function of coordinate system rotation, scaling, or programmable mirror image is started.)
- Axis switching

NOTE 8 A single–block stop is not made in the G72.1 or G72.2 block.

NOTE 9 A move command immediately following the G72.1 or G72.2 block must be specified in the absolute mode. If it is specified in the incremental mode, the value indicates an incremental traveling distance from the point at which G72.1 or G72.2 is terminated.

NOTE 10 If the end point of a copied figure (n) does not agree with the start point of the next figure to be copied (n + 1), a shift from the end point to the start point is automatically made before starting copying the next figure. (This happens when the specified angular displacement or shift is wrong.)



Main program		Subprogram	
O1000 ;		O2000 G90 G01 X40. ;	(P3)
N10 G92 X–20.0 Y0 ;		N100 Y40. ;	(P4)
N20 G00 G90 X0 Y0 ;		N200 G01 X80. ;	(P5)
N30 G01 G17 G41 X20. Y0 D01 F10 ;	(P0)	N300 G01 Y20. ;	(P6)
N40 Y20. ;	(P1)	N400 X100. ;	(P7)
N50 X30. ;	(P2)	N500 M99 ;	
N60 G72.2 P2000 L3 I90.0 J0 ;			

The actual shift is 70 mm. Although I70.0 must be specified, I90.0 is specified instead. The end point of a copied figure (n) does not agree with the start point of the next figure to be copied (n + 1).

NOTE 11 The following variable can be used to read the coordinates of the original figure: #5001 or a later system variable (block end point) of a custom macro called while the system is copying a figure with rotation or parallel movement.

14.8 Normal–Direction Control (G40.1, G41.1, and G42.1)

G41.1 or G42.1 controls the rotation axis (C–axis) so that a tool is always perpendicular to the direction of movement during cutting.

G40.1: Cancels normal–direction control. (Normal–direction control is not exercised.)

G41.1: Exercises normal–direction control on the left side. (Keeps the tool perpendicular to the direction of movement when the tool faces the left relative to the direction of movement.)

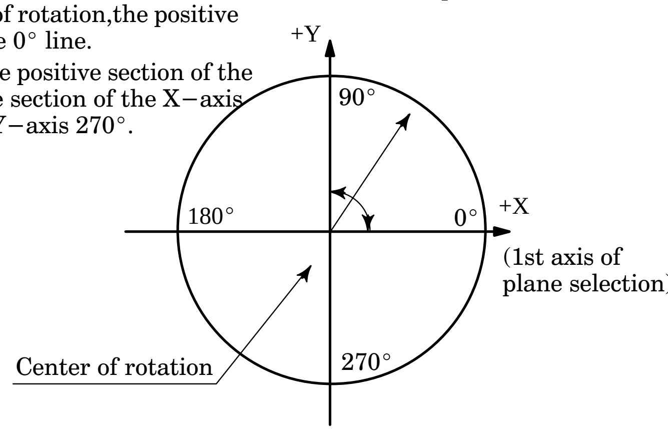
G42.1: Exercises normal–direction control on the right side. (Keeps the tool perpendicular to the direction of movement when the tool faces the right relative to the direction of movement.)

The normal–direction control function keeps the tool perpendicular to the direction of movement on the selected plane.

This section describes the function, assuming that the X–Y plane has been selected.

The angular displacement of the C–axis is defined as shown on the right: Viewed from the center of rotation,the positive section of the X–axis is defined as the 0° line.

Counterclockwise from the 0° line ,the positive section of the Y–axis is defined as 90° ,the negative section of the X–axis 180°,and the negative section of the Y–axis 270°.

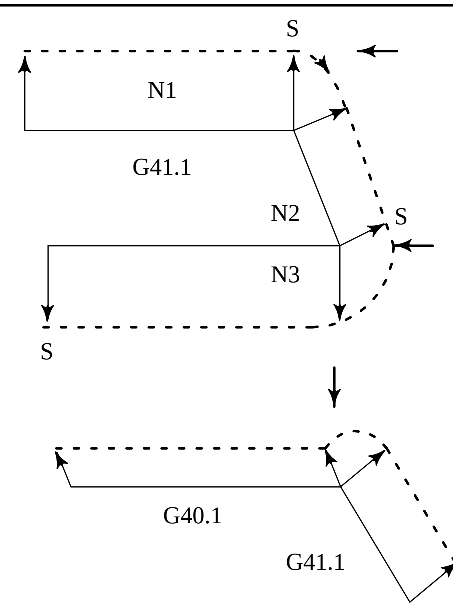


When the cancel mode is switched to the normal–direction control mode, the C–axis becomes perpendicular to the direction of movement at the beginning of the block in which G41.1 or G42.1 is specified.

If the direction of movement in a block is different from that in the next block in the normal–direction control mode, the direction of the C–axis is automatically adjusted between the blocks. The C–axis is adjusted so that it becomes perpendicular to the direction of movement at the beginning of each block. After the C–axis is rotated so that it becomes perpendicular to the specified direction of movement, movement along the X–axis or Y–axis is started.

In the cutter compensation mode, C–axis is adjusted so that the tool becomes perpendicular to the direction of movement along the tool path determined after the compensation.

In the single–block operation,there is no stop between the command for rotating the C–axis and the command for causing movement along the X–or Y–axis. The single–block stop occur after the movement along the X–or Y–axis is terminated.



At the beginning of a block, the tool moves along the C–axis at the feedrate specified in parameter 1472. If the dry run mode is valid at that time, the tool moves along the C–axis at the dry run speed. If the movement along the X– or Y–axis is to be made in the rapid traverse mode (G00), the tool moves along the C–axis at the rapid traverse feedrate.

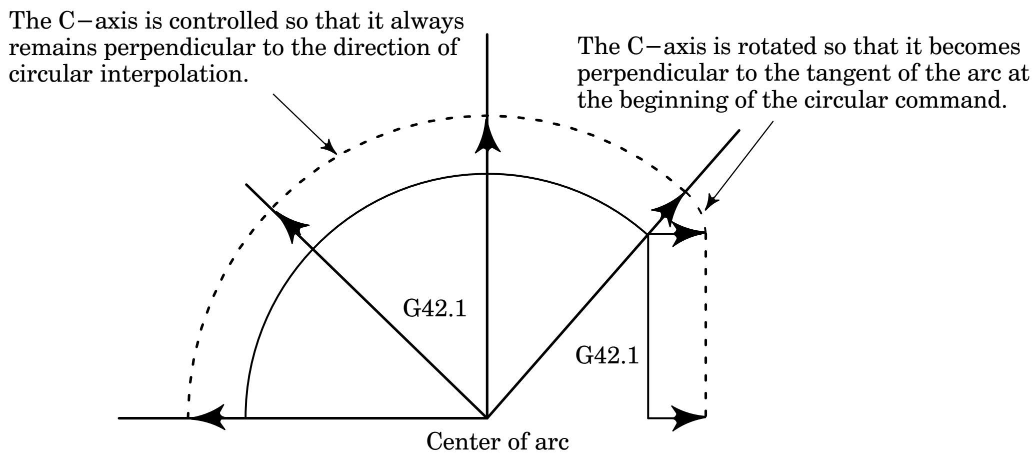
Circular interpolation is started after the C–axis is rotated so that it becomes perpendicular to the tangent of the arc at the start point. The C–axis is kept perpendicular to the direction of circular interpolation.

While the circular command is executed, the tool moves along the C–axis at the following feedrate:

$$F \times \frac{\text{Angular displacement of the C–axis (degree)}}{\text{Length of arc (mm or inch)}} \text{ [deg/min]}$$

F: Feedrate specified in the block of circular command (mm/min or inch/min)

The angular displacement for the C–axis is the difference between the angle at the start point and that at the end point of a block.



WARNING	In the normal–direction control mode, a command for the C–axis must not be specified. If it is specified, the command is ignored.

NOTE 1 If the feedrate along the C–axis is greater than the maximum cutting feedrate of the C–axis specified in parameter No. 1422, the feedrate of another axis is clamped to the maximum cutting feedrate of the C–axis.

NOTE 2 In the normal–direction control mode, the C–axis is rotated in such a direction that the angular displacement does not exceed 180 . In other words, the shortest path is always selected.

NOTE 3 Before starting machining, the coordinate of the C–axis on the workpiece coordinate system must be correctly related to the actual position of the C–axis of the machine by pressing the origin key or setting the coordinate system (G92).

NOTE 4 The C–axis must be a rotation axis.

NOTE 5 In the normal–direction control mode, helical cutting using two linear axes cannot be executed. Only helical cutting with one linear axis can be executed.

NOTE 6 Normal–direction control cannot be executed for movement in the G53 mode.

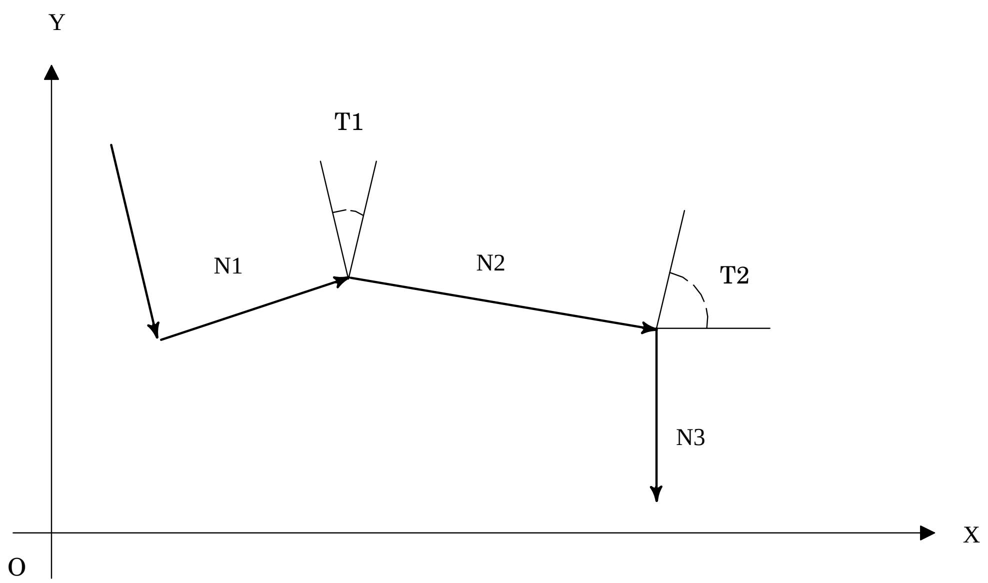
NOTE 7 If an error is detected in parameters for normal–direction control, P/S alarm No. 470 occurs.

NOTE 8 The feedrate of the C–axis for the circular command can be calculated as described above when bit HTG of parameter No. 1401 is set to 0. (The speed of helical interpolation is specified by the tangential velocity in circular movement.)

14.9 Gentle Curve Normal Direction Control

In typical direction control, rotation of the rotation axis (C–axis) is inserted ahead of the block responsible for moving the tool along the linear axes (X– and Y–axes). Rotation is performed independently of the movement along the X– and Y–axes such that, during rotation, movement along the X– and Y–axes is stopped. If the rotation involves only a small angular displacement, however, the movement along the X– and Y–axes need not always be stopped. With the gentle curve normal direction control function, when the angular displacement of the C–axis does not exceed a predetermined angle (specified with parameter No. 7793), the C–axis rotation is included in the block controlling movement along the X– and Y–axes and is performed simultaneously. Therefore, movement along the X– and Y–axes can be performed continuously without interruption. Hence, good machining profiles can be obtained.

Example



T1 : Angular displacement between N1 and N2 (T1 < x)
T2 : Angular displacement between N2 and N3 (T2 >= x)
x : Angle set with parameter (parameter. 7793)

When N1, N2, and N3 are specified as shown above, the tool movement is as follows:

1) Since T1 between N1 and N2 is smaller than x, the C–axis is rotated by T1 when the tool is moved along the X– and Y–axes in N2.
2) Since T2 between N2 and N3 is greater than or equal to x, the tool is moved as normal. That is, the C–axis is rotated by T2 before the execution of N3 moves the tool along the X– and Y–axes.

NOTE 1 Even when the angle between a straight line and arc, or between two arcs, is smaller than a parameter–set value, C–axis rotation is performed at the same time as X– and Y–axes movement, as when the angle between straight lines is smaller than a parameter–set value.

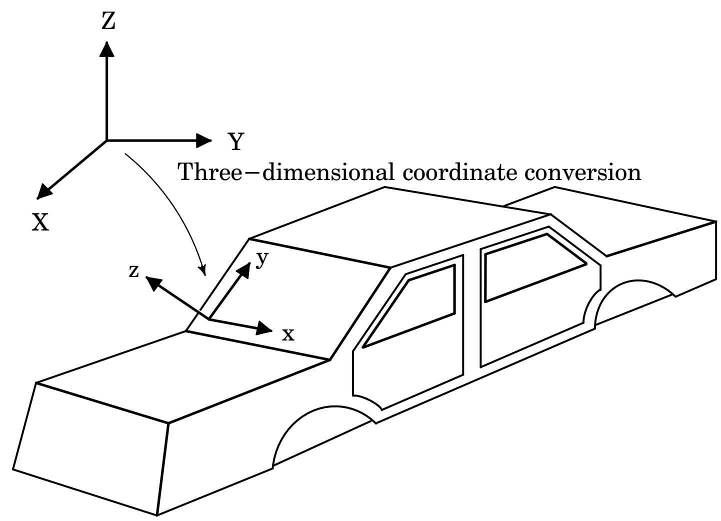
NOTE 2 To use the gentle curve normal direction control function, the simultaneously controllable axes expansion function and helical interpolation function are required.

NOTE 3 When the gentle curve normal direction control function is to be used, set bit 0 of parameter No. 1008 to 0.

NOTE 4 When C–axis rotation is included in the block that moves the tool along the X– and Y–axes, the composite feedrate that includes the C–axis rotation speed can be adjusted to the feedrate specified in F if bit 0 of parameter No. 7620 is set to 1.

14.10 Three–Dimensional Coordinate Conversion

The coordinate system can be rotated about an axis by specifying the center of rotation, direction of the axis of rotation, and angular displacement. This coordinate conversion function is quite useful for three–dimensional machining using a diesinking machine. By applying three–dimensional coordinate conversion to a program generated for machining on the XY plane, identical machining can be executed on a desired plane.



G68 selects the three–dimensional coordinate conversion mode and G69 cancels it.

N1 G68 X x1 Y y1 Z z1 I i1 J j1 K k1 Rα ;
N2 G68 X x2 Y y2 Z z2 I i2 J j2 K k2 Rβ ;

X,Y,Z : Center of rotation (absolute)
I,J,K : Direction of the axis of rotation (All of I, J, and K must be specified.)
R : Angular displacement

The center, axis, and angle of the first rotation are specified in the N1 block. The N1 block produces a new coordinate system, X', Y', Z'. Viewed from the original workpiece coordinate system, the new coordinate system is created by shifting the origin of the original coordinate system by (X1, Y1, Z1) and rotating the original coordinate system about vector (i1, j1, k1) by an angle α . In the N2 block, the center, axis, and angle of the second rotation are specified. The X, Y, Z, I, J, K, and R values specified in the N2 block indicate the values and angle on the coordinate system produced after coordinate conversion of the N1 block. The N2 block produces coordinate system X", Y", Z". Viewed from X' Y' Z', new coordinate system X", Y",Z" is created by shifting the center of X', Y', Z' by (X2, Y2, Z2) and rotating X', Y', Z' about vector (i2, j2, k2) by an angle β . The X, Y, and Z values specified in the N3 block are coordinates on X", Y", Z". X'', Y'', Z'' is called the program coordinate system.

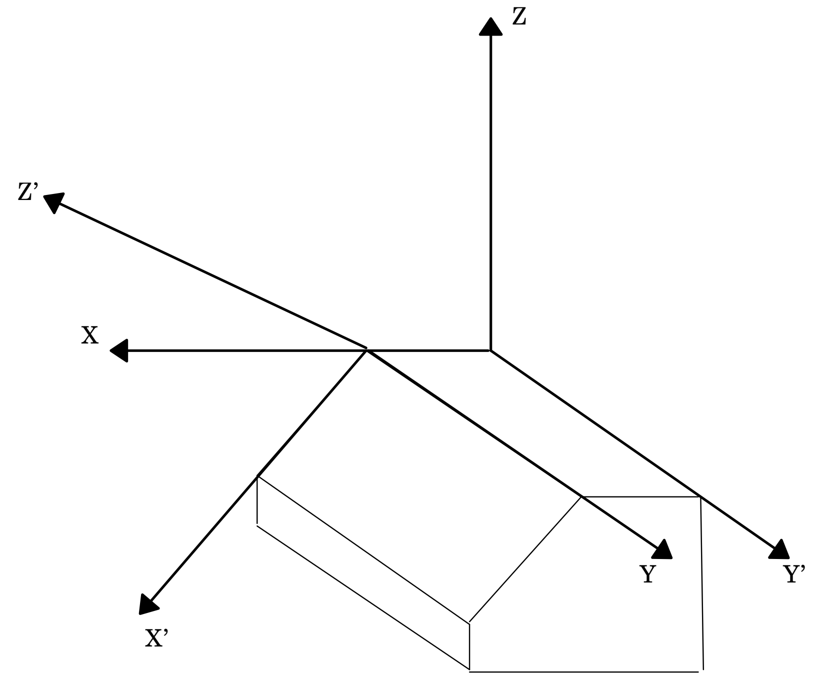
If the X, Y, and Z values are not specified in the N2 block, the X, Y, and Z values specified in the N1 block are used as the center of the second rotation. This means that the N1 and N2 blocks have a common center of rotation. When only one rotation is required, the N2 block need not be specified. In the G68 block, specify X, Y, and Z using absolute values.

Angular displacement R is positive when the coordinate system is rotated clockwise like a right–hand screw advancing in the direction of the axis of rotation. Bit RTR of parameter No. 6400 determines the unit of R.

– Bit 4 of parameter No. 6400 can specify that only the G69 command cancels the three-dimensional coordinate conversion mode (G68). With such a specification, a system reset, the ERS, ESP, or RRW input signal from the PMC does not cancel the three-dimensional coordinate conversion mode (G68).

– In the three- dimensional coordinate conversion mode (G68), making the M3R input signal from the PMC (address G031.3) high moves the tool in the direction of an axis selected in the coordinate system submitted to three-dimensional conversion (program coordinate system) during manual jog feed, manual incremental feed, or manual handle feed.

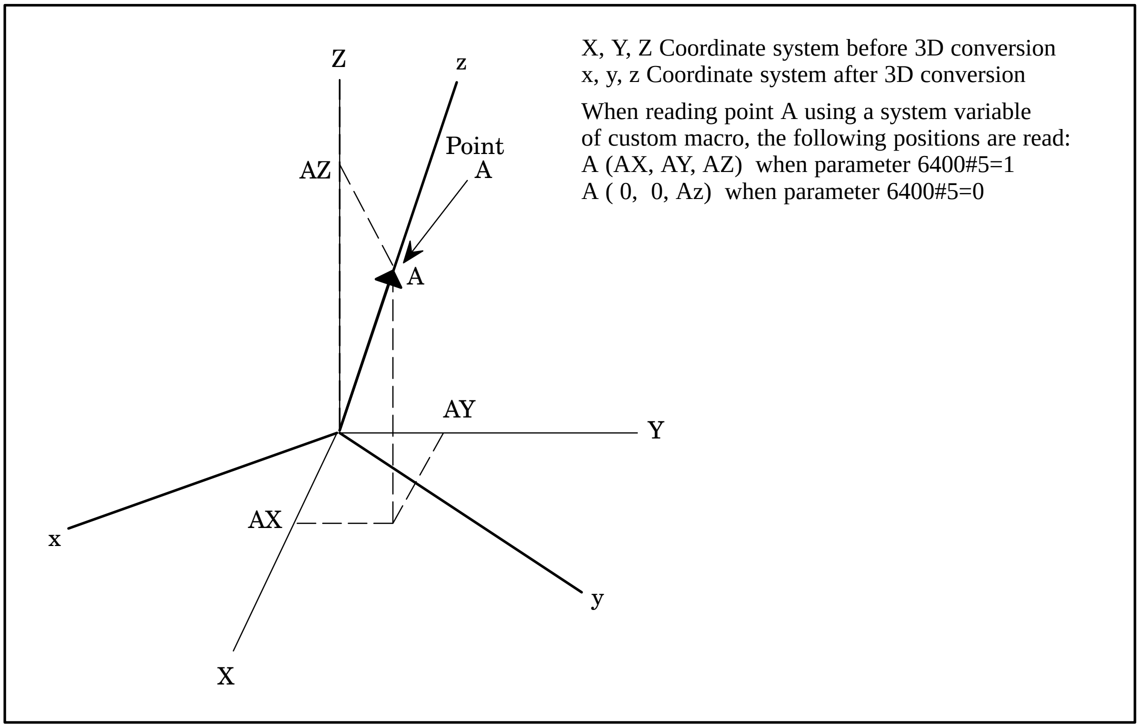
  When the M3R signal is low, three- dimensional conversion is not effective for the above three manual operations even in the three- dimensional coordinate conversion mode (G68).



X –Y –Z : Coordinate system before conversion (Workpiece coordinate system)
X' –Y' –Z' : Coordinate system after conversion (Program coordinate system)

Example When the M3R signal is made high during the three-dimensional coordinate conversion mode, manual feed with the Z-axis selected causes a movement in the Z-axis direction shown above.

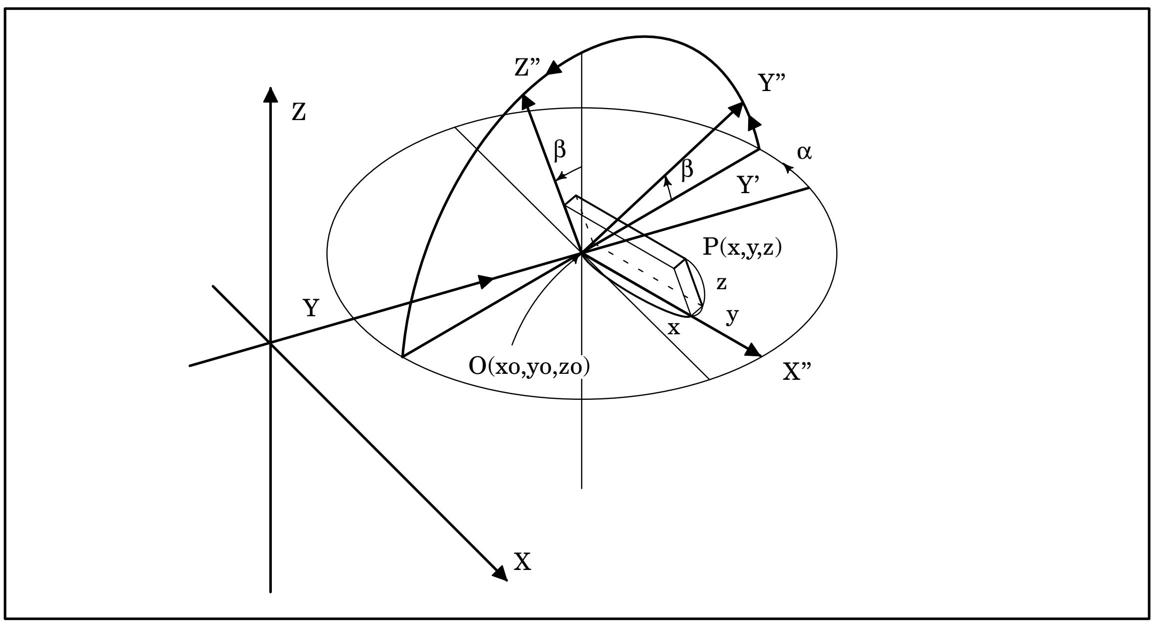
– When the current tool position in the workpiece coordinate system is read using the custom macro system variables #5041 to #5055 (ABSOT), conventionally, the coordinates that are read are those in the coordinate system that has not be converted by coordinate conversion even in the three- dimensional coordinate conversion mode (G68). However, bit 5 of parameter No.6400 can specify that the coordinates that are read be those in the workpiece coordinate system that has been converted by three-dimensional coordinate conversion.



– The 3DROT output signal (address F159.3) informs the PMC that the system is in the three- dimensional coordinate conversion mode (G68). The 3DROT output signal is high during the three- dimensional coordinate conversion mode (G68).

– A status display on the CRT screen indicates that the system is in the three-dimensional conversion mode (G68).

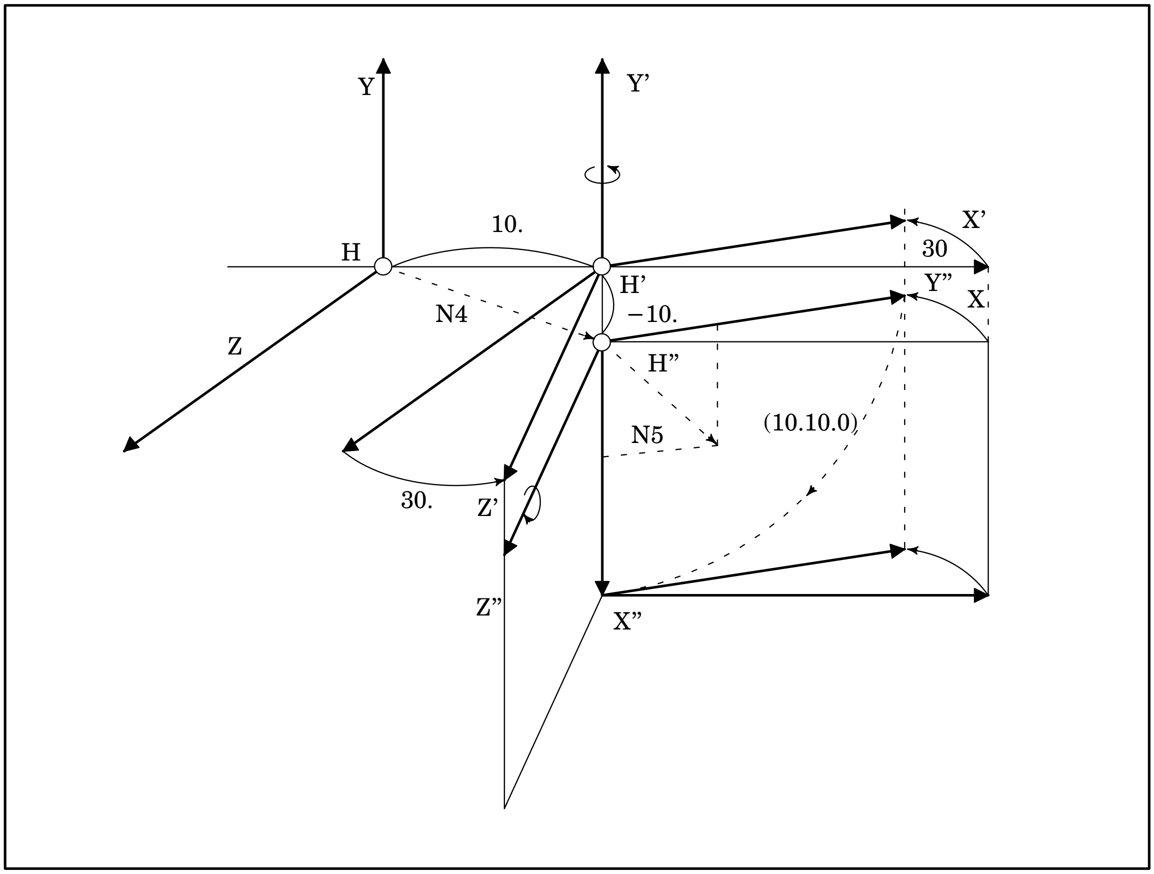
Example G68 X xo Y yo Z zo I 0 J 0 K 1 Rα ;
G68 I 1 J 0 K 0 Rβ ;



X Y Z	:	Workpiece coordinate system
X’ Y’ Z’	:	Coordinate system produced after the first conversion
X” Y” Z”	:	Coordinate system produced after the second conversion
α	:	Angular displacement in the first conversion
β	:	Angular displacement in the second conversion
O(xo, yo,zo)	:	Center of rotation
P(x, y, z)	:	Coordinates in coordinate system X”, Y”, Z” (program coordinate system)

Sample program for three–dimensional coordinate conversion

N1 G90X0Y0Z0 ;	Positioning at origin H
N2 G68X10.Y0Z0 I0J1K0 R30.;	New coordinate system X', Y', Z' is produced having origin is H'.
N3 G68X0Y–10.Z0I0J0K1R–90.;	New coordinate system X'', Y'', Z'' is produced. Its origin H'' is (0, –10, 0) on previous coordinate system X' Y' Z'.
N4 G90X0Y0Z0;	Positioning at H'' on coordinate system X'', Y'', Z''
N5 X10.Y10.Z0;	Positioning at (10, 10, 0) on coordinate system X'', Y'', Z''



WARNING 1 Positioning in the machine coordinate system such as G28, G30, or G53 is not subjected to coordinate conversion.

WARNING 2 When the three–dimensional coordinate conversion function is used, linear rapid traverse must be specified by setting parameter LRP of data 1400 to 1.

WARNING 3 Three basic axes must use a common unit system (bits ISUFx, ISRx, ISFx, IPRx of parameter No. 1004).

WARNING 4 The canned cycle specified by G41, G42, or G51.1 must be embedded in the G68 cycle. The entire canned cycle must be included in the routine determined by G68 and G69. See the example below:

```
G68X100.Y100.Z100.I0J0K1.R45. ;
 - - - - - - - - - -
G41D01;
 - - - - - - - - - -
G40 ;
G90G00X100.Y100.Z0. ;   <------  An absolute move command must be specified.
G69
```

G40, G50.1 or G80 must not be followed by G69. The G40 block must be followed by move command G00 or G01 with an absolute value.

NOTE 1 Equation for three–dimensional coordinate conversion

The following equation expresses the relationship between (x, y, z) on the program coordinate system and (X, Y, Z) on the original coordinate system (workpiece coordinate system).

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} & & \\ & M1 & \\ & & \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} x1 \\ y1 \\ z1 \end{bmatrix}$$

If the coordinate system is converted twice, the relationship is expressed as:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} & & \\ & M1 & \\ & & \end{bmatrix} \begin{bmatrix} & & \\ & M2 & \\ & & \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} & & \\ & M1 & \\ & & \end{bmatrix} \begin{bmatrix} x2 \\ y2 \\ z2 \end{bmatrix} + \begin{bmatrix} x1 \\ y1 \\ z1 \end{bmatrix}$$

X, Y, Z : Coordinates in the original coordinate system (workpiece coordinate system or machine coordinate system)

x, y, z : Coordinates specified by a program (coordinates in the program coordinate system)

x1, y1, z1 : Center of rotation in the first conversion

x2, y2, z2 : Center of rotation in the second conversion (value in the coordinate system produced after the first conversion)

M2 : Matrix for the second conversion

M1 : Matrix for the first conversion

M1 and M2 are conversion matrices determined by the angular displacement and axis of rotation. They are generally expressed as:

$$\begin{bmatrix} n1^2 + (1 - n1^2)\cos\theta & n1n2\,(1-\cos\theta) - n3\sin\theta & n1n3\,(1-\cos\theta) + n2\sin\theta \\ n1n2\,(1-\cos\theta) + n3\sin\theta & n2^2 + (1 - n2)\cos\theta & n2n3\,(1-\cos\theta) - n1\sin\theta \\ n1n3\,(1-\cos\theta) - n2\sin\theta & n2n3\,(1-\cos\theta) + n1\sin\theta & n3^2 + (1 - n3^2)\cos\theta \end{bmatrix}$$

n1 : Cosine of the angle between the X axis and the axis of rotation $\dfrac{i}{\sqrt{i^2 + j^2 + k^2}}$

n2 : Cosine of the angle between the Y axis and the axis of rotation $\dfrac{j}{\sqrt{i^2 + j^2 + k^2}}$

n3 : Cosine of the angle between the Z axis and the axis of rotation $\dfrac{k}{\sqrt{i^2 + j^2 + k^2}}$

θ : Angular displacement

The conversion matrix for rotation on a plane is expressed as follows:

(i) Coordinate conversion on the XY plane

$$M = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(ii) Coordinate conversion on the YZ plane

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix}$$

(iii) Coordinate conversion on the ZX plane

$$M = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}$$

NOTE 2 In the mode of three–dimensional coordinate conversion, the following G codes can be specified:

- G00: Positioning
- G01: Linear interpolation
- G02: Circular interpolation (clockwise)
- G03: Circular interpolation (counterclockwise)
- G04: Dwell
- G10: Data setting (Writing coordinate system data by G10L2 cannot be specitied)
- G17: Selection of the XY plane
- G18: Selection of the ZX plane
- G19: Selection of the YZ plane
- G28: Return to the reference position
- G29: Return from the reference position
- G30: Return to the second, third, or fourth reference position
- G40: Cancellation of cutter compensation
- G41: Cutter compensation, left
- G42: Cutter compensation, right
- G43: Tool length compensation (plus direction)
- G43.1: Tool length compensation in tool axis direction
- G44: Tool length compensation (minus direction)
- G45: Tool offset (increase)
- G46: Tool offset (decrease)
- G47: Tool offset (double increase)
- G48: Tool offset (double decrease)
- G49: Tool length compensation cancel
- G50.1: Cancellation of programmable mirror image
- G51.1: Programmable mirror image
- G53: Selection of machine coordinate system
- G65: Custom macro (single shot)
- G66: Custom macro mode
- G67: Cancellation of custom macro
- G73: Canned cycle (Peck drilling cycle
- G74: Canned cycle (Reverse tapping cycle)
- G76: Canned cycle (Fine boring cycle)
- G80: Cancellation of canned cycle
- G81 to G89: Canned cycle
- G90: Absolute command
- G91: Incremental command
- G94: Command of feed per minute
- G95: Command of feed per rotation
- G98: Canned cycle (return to the initial level)
- G99: Canned cycle (return to the R–point level)

NOTE 3 In the mode of three–dimensional coordinate conversion, rapid traverse in the canned cycle of drilling is executed at the cutting feedrate specified in parameter No. 2052. When the parameter is set to 0, the rapid traverse is executed at the maximum cutting feedrate.

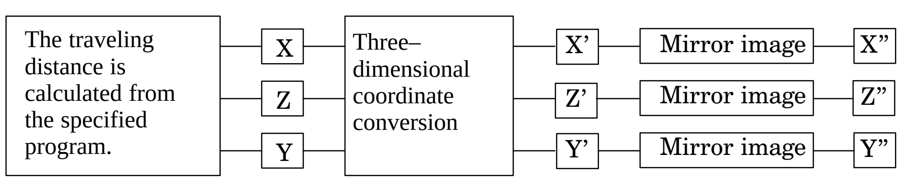
NOTE 4 The amount of manual intervention or manual interruption is not subjected to coordinate conversion.

NOTE 5 If tool length compensation, cutter compensation, or tool offset compensation is required, it is executed before three–dimensional coordinate conversion.

NOTE 6 In the G68 or G69 block, another G code must not be specified. When specifying G68, specify I, J, and K all together.

NOTE 7 Three–dimensional coordinate conversion and two–dimensional coordinate rotation use the same G code, G68. The G68 command is assumed to be for three–dimensional coordinate conversion when all of I, J, and K are specified. Otherwise, it is assumed to be for two–dimensional coordinate rotation.

NOTE 8 The external mirror image function (executed by DI or a parameter) produces a mirror image on the other side of each control axis when the three–dimensional coordinate conversion is applied



NOTE 9 When the absolute position is indicated in the mode of three–dimensional coordinate conversion, bits DTL and DCR of parameter No. 2202 must be set to 0.

NOTE 10 The unit of angular displacement, R, is determined by bit RTR of parameter No. 6400, which is also used for coordinate system rotation.

NOTE 11 If the parking signal is applied to an axis and the move command is specified for another axis, the axis for which the move command is specified is subjected to three–dimensional coordinate conversion. The axis to which the parking signal is applied may be affected.

NOTE 12 In the mode of three–dimensional coordinate conversion, the absolute value in the program coordinate system or the workpiece coordinate system can be indicated by specifying bit DAK of parameter No. 2204.

NOTE 13 In the mode of three–dimensional coordinate conversion, the remaining traveling distance in the program coordinate system or the workpiece coordinate system can be indicated by specifying bit DP3D of parameter No. 2208.

NOTE 14 In the mode of three–dimensional coordinete conversion, unidirectional positioning (G60) cannot be specitied.

NOTE 15 The three- dimensional coordinate conversion mode begins when a G68 block is executed and ends when a G69 block is executed or when a reset is performed (if bit 4 of parameter No.6400 is 0).

NOTE 16 For three axes selected for three- dimensional conversion, the same time constant, FL rate, and acceleration/deceleration method as in the cutting mode apply to manual feed.

NOTE 17 The feedrate, time constant, and acceleration/ deceleration method in manual rapid traverse are the same as in programmed rapid traverse (positioning by G00).

NOTE 18 If manual feed is performed during the three- dimensional coordinate conversion mode, the tangential feedrate in the converted coordinate system (program coordinate system) is set to the lowest of the feedrates of the selected axes.

NOTE 19 Even if a parameter (bit 4 of parameter No.6400) has specified that only the G69 command can cancel the three- dimensional coordinate conversion mode (G68), the G69 mode is assumed when the program restart is performed.

14.11 Circle Cutting Function

During circle cutting, the tool moves from the center of a circle and cuts a workpiece along the circle as shown in Fig. 14.11 (a). The tool first moves in a 45°direction, and then moves along an arc of a circle having half the radius of the target circle. The tool then comes into contact with the workpiece and starts cutting. The tool can cut the workpiece without leaving any marks on it.

A single block of G code can specify the series of movements described above. During circle cutting, cutter compensation can be performed.

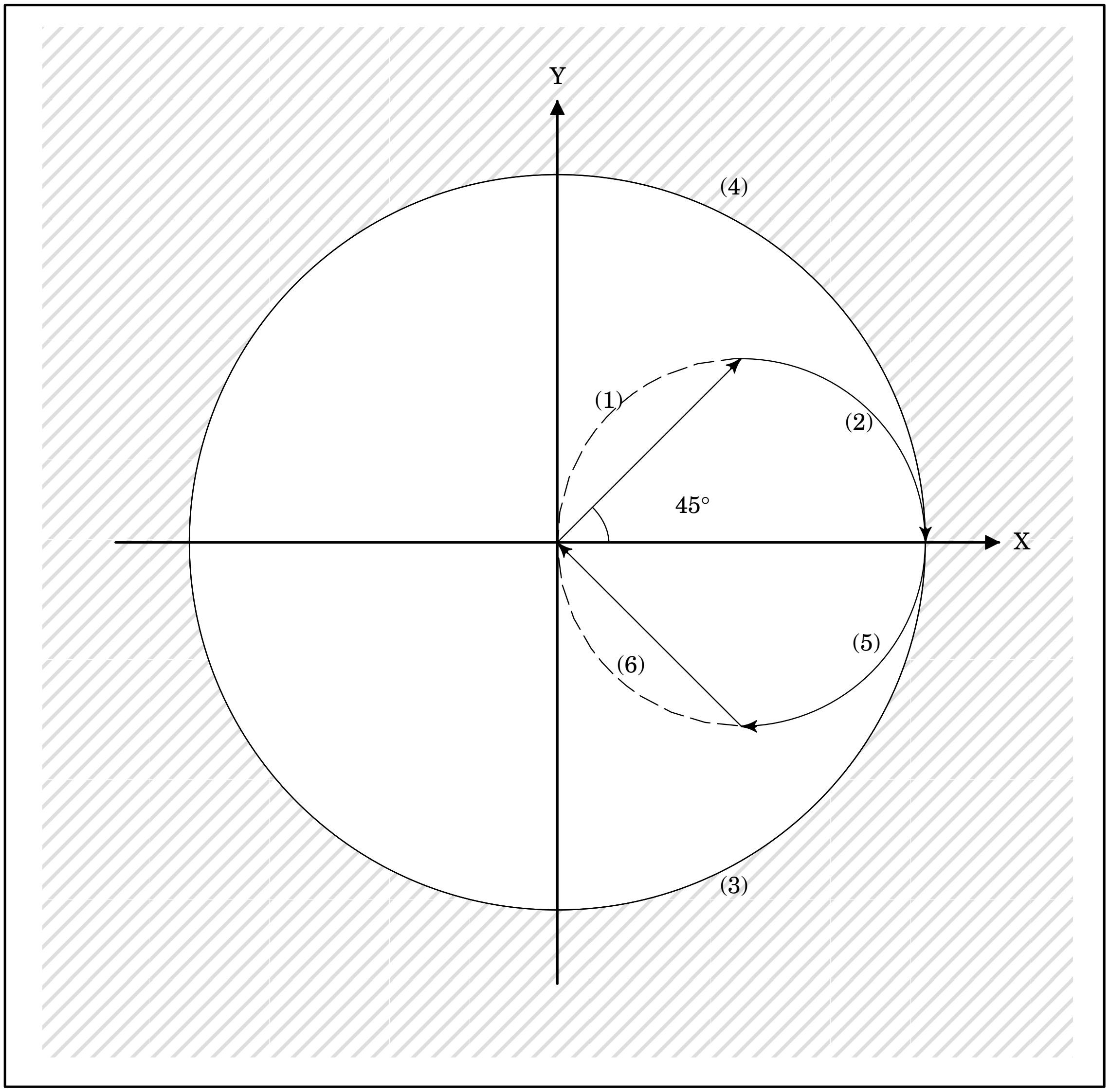


Fig. 14.11 (a)

(1) Command format

Circle cutting can be executed by specifying one of the following commands:

Circle cutting in the XY plane

G12.2 I_ D_ F_;

G13.2 I_ D_ F_;

Circle cutting in the YZ plane

G12.2 J_ D_ F_;

G13.2 J_ D_ F_;

Circle cutting in the ZX plane

G12.2 K_ D_ F_;

G13.2 K_ D_ F_;

where

G12.2 : Circle cutting in a clockwise direction

G13.2 : Circle cutting in a counterclockwise direction

I, J, K : Radius of a circle in each plane(*1)

D : Offset number (cutter compensation)

F : Feedrate

*1 A negative value can be specified for I, J, or K.

(2) Cutter compensation

If a D code is specified in the circle cutting command, cutter compensation is made during circle cutting. The tool is first moved in a 45°direction, allowing for cutter compensation as shown in Fig. 14.11 (b).

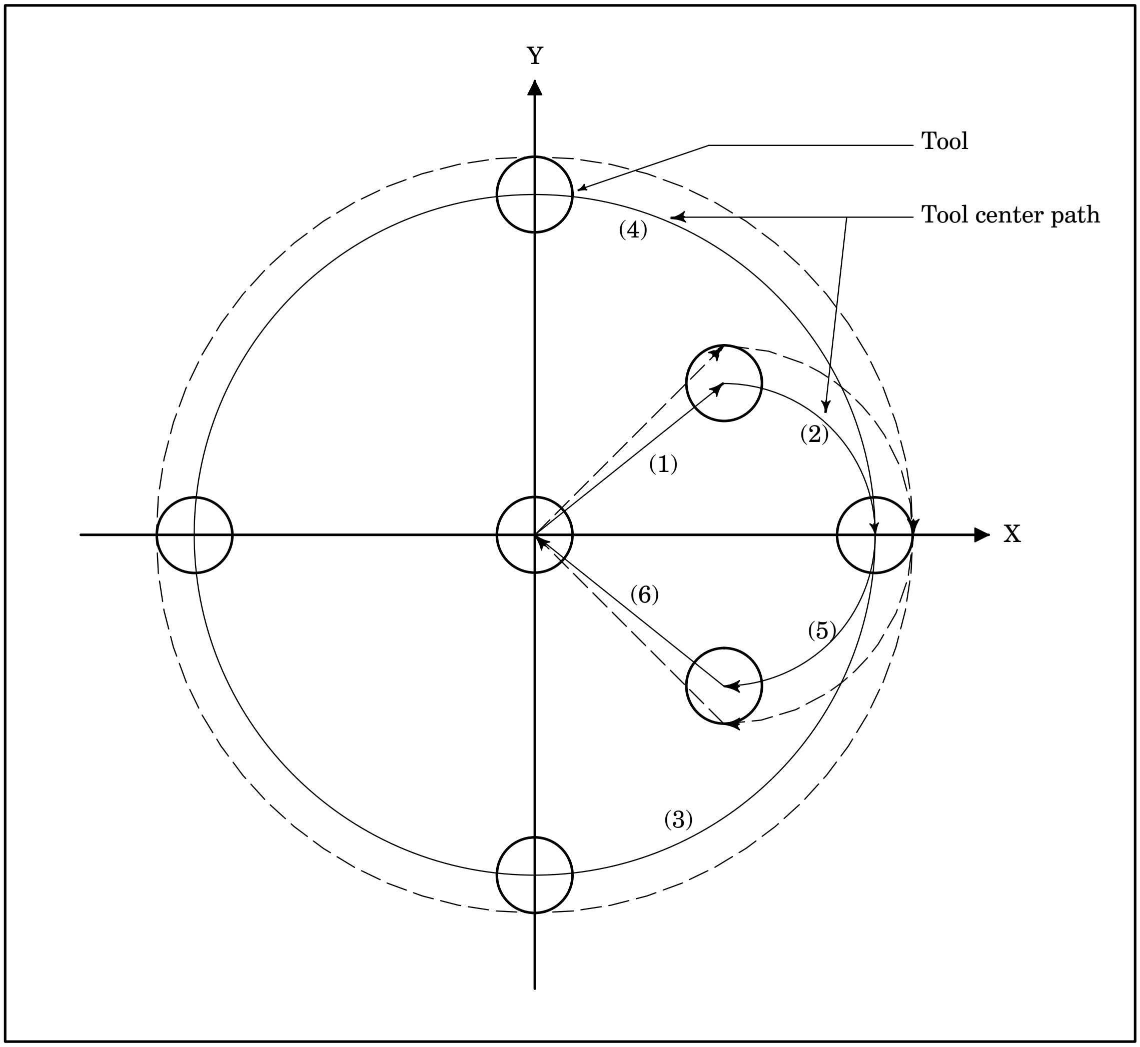


Fig. 14.11 (b)

(3) Radius of the circle

A negative value can be used for I, J, or K to specify the radius of the circle. When a negative value is specified, the tool starts moving in the opposite direction as shown in Fig. 14.11 (c).

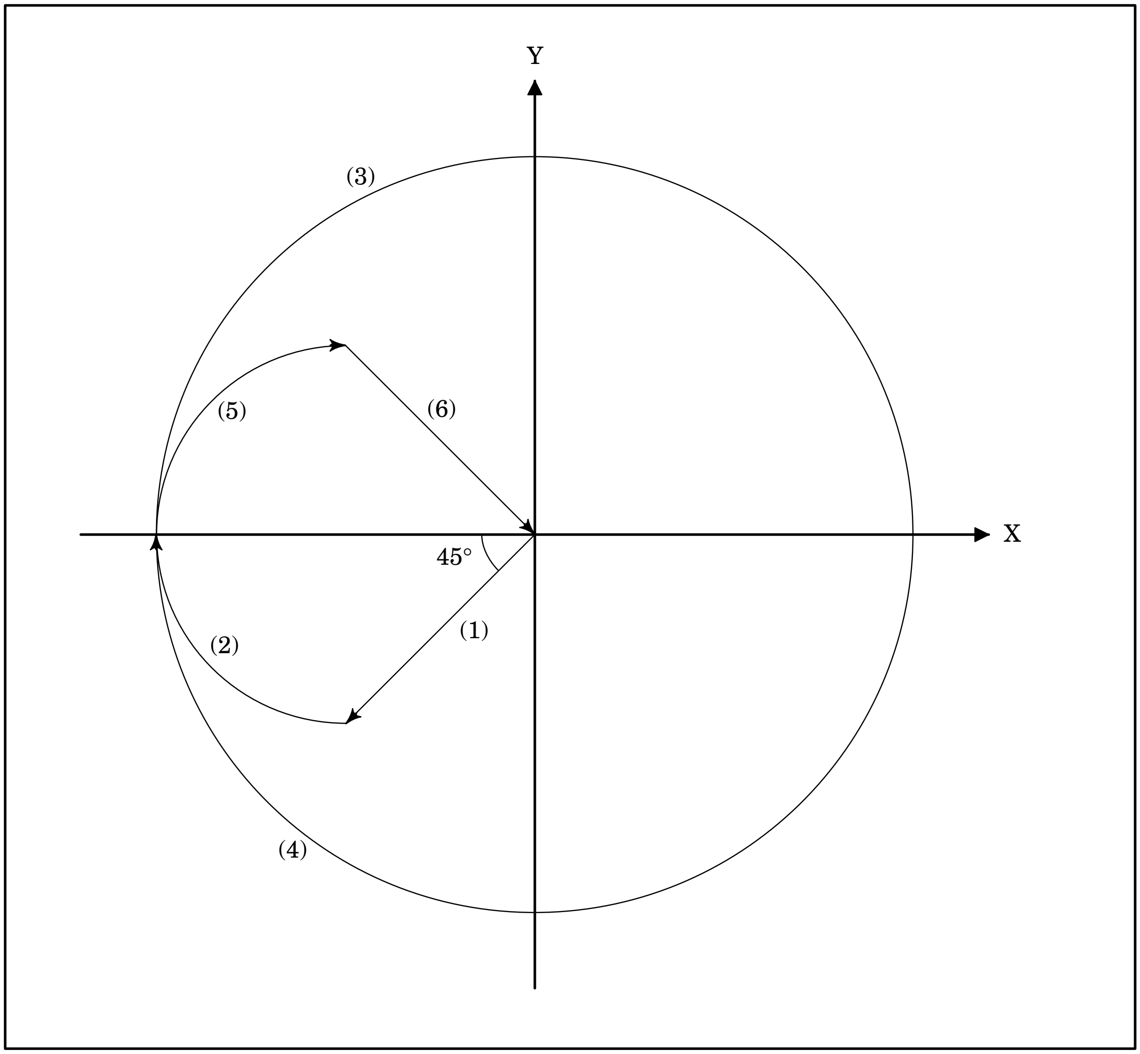


Fig. 14.11 (c)

(4) Single–block stop

During circle cutting, a single–block stop can be made at any of points (1) to (5), as shown in Fig. 14.11 (d).

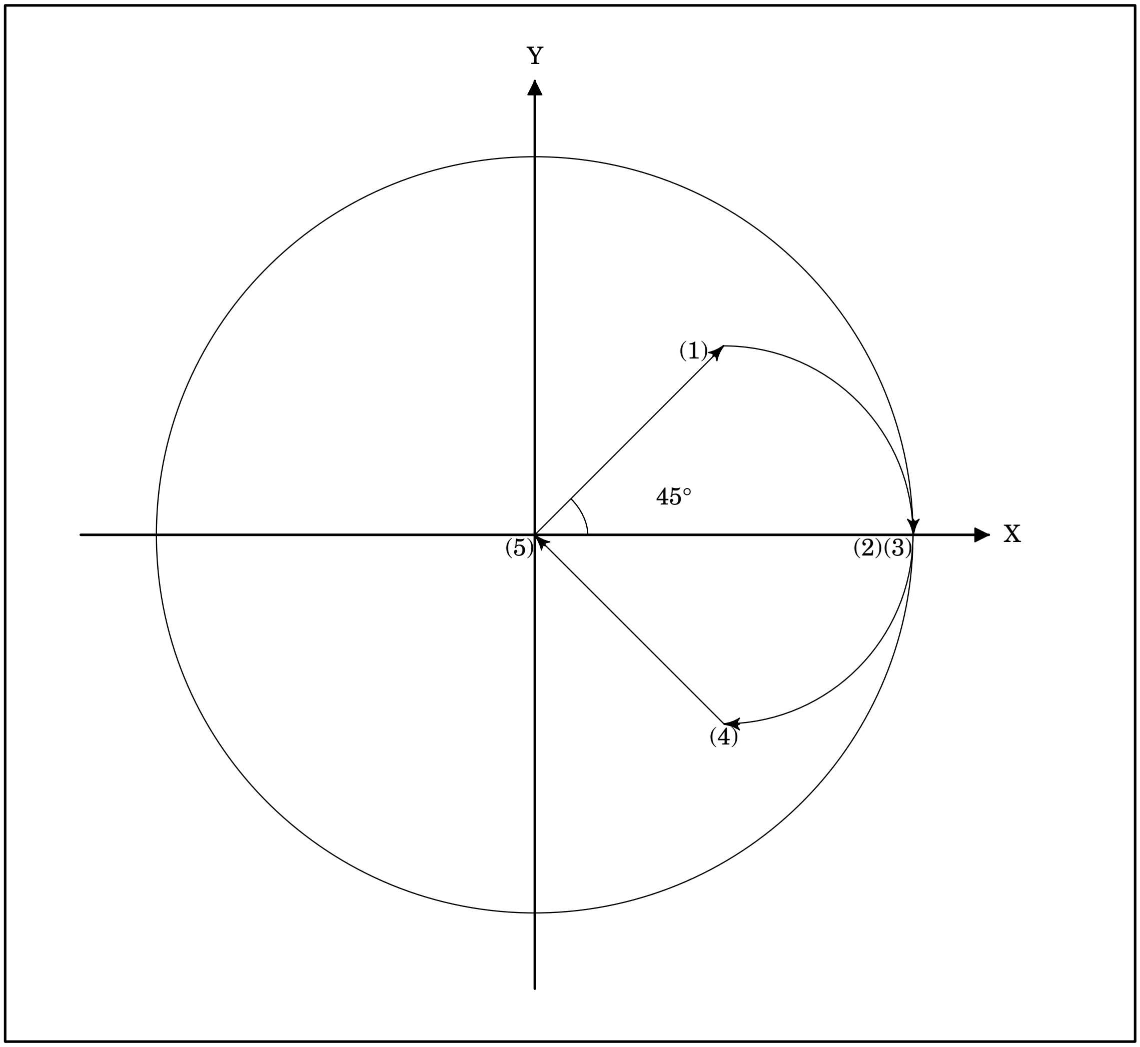


Fig. 14.11 (d)

NOTE 1 If no offset number is specified with the D code, the system assumes that the cutter compensation value is zero. If the continuous–state D code is specified, the offset number specified with the D code is assumed.

NOTE 2 When a feed hold occurs during circle cutting, the tool is stopped at that position.

NOTE 3 Before the circle cutting command is specified, the offset mode must be canceled.

NOTE 4 PS alarm 217 will occur if the cutter compensation value is greater than the radius of the target circle or if the radius of the target circle is set to 0 or is not specified.

NOTE 5 PS alarm 218 will occur if an axis command is specified in the same block as the circle cutting command as shown below:

G12.2 I100. D1 F100. X10.;

NOTE 6 This function cannot be used in coordinate rotation mode.

15.COMPENSATION FUNCTION

15.1 Tool Length Offset (G43,G44,G49)

The G43 and G44 commands are used to set the equipment to the tool length compensation mode. Specify along which axis tool length compensation is applied with axis address α in the G43 and G44 blocks.

$$\left.\begin{array}{l}\textbf{G43}\\ \textbf{G44}\end{array}\right\}\ \alpha\_\ \textbf{H}\_\ ;$$

The above command shifts the position of the end point specified with the move command for the α axis in a positive or negative direction by the value specified in offset memory. This function applies tool length compensation without the program being changed. This function stores the difference between the tool length assumed during programming and the length of the tool to be used for actual machining stored in offset memory.

Specify the direction of offset with the G43 or G44 command and the number of the offset memory storing the offset with an H code.

1) Direction of offset

   C43 + side offset

   C44 – side offset

   In any case of absolute or incremental commands, the offset amount that has been set into the offset memory assigned by H code is in G43, added to, and in G44, subtracted from the coordinate value of the terminal point of the α axis movement command. The coordinate value after the calculation becomes the terminal point. When the movement command for the α axis was omitted, it is taken in the same way as:

$$\left.\begin{array}{l}\textbf{G43}\\ \textbf{G44}\end{array}\right\}\ \textbf{G91}\alpha\textbf{OH}\ \text{———};$$

   and the tool moves by the offset amount in the + direction in G43 and in the – direction in G44. G43 and G44 being modal codes, are effective until another G code in the same group is programmed.

   The machine can be controlled with parameters so that it enters the G43 or G44 mode when the power is turned on.

2) Assignment of offset amount

   Assign offset number by H code. The offset amount that has been set in the offset memory is added to or subtracted from the programmed command value for the α axis. The offset amount may be set in the offset memory through the CRT/MDI panel.

NOTE	The offset value corresponding to offset No. 00 or H00 always means 0. It is impossible to set H00 to any other offset value.

3) Canceling tool length compensation

   To cancel the offset, command a G49 or assign offset H00. When H00 or G49 is commanded, the canceling action is taken immediately.

   When LXY, bit 4 of parameter No.6000, is set to 0 (tool length compensation is applied along the Z–axis), the offset can be canceled simply by specifying H00.

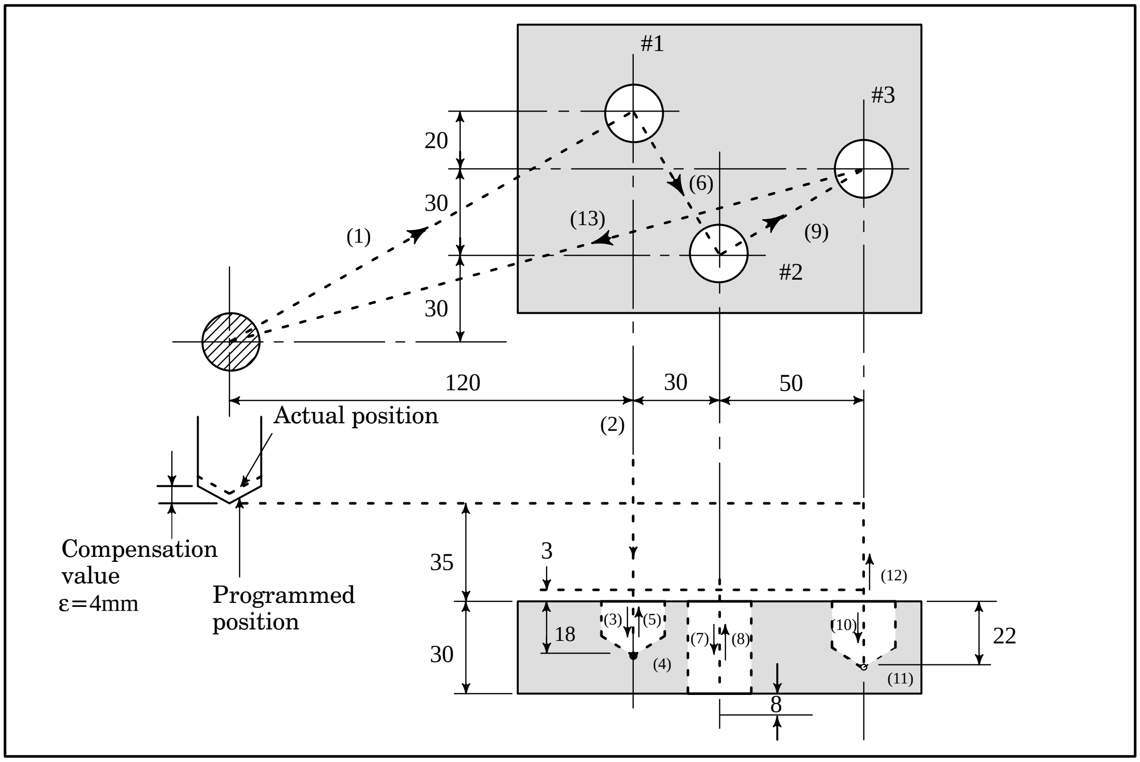
   When LXY, bit 4 of parameter No.6000, is set to 1, tool length compensation is applied along the axis specified in the program.

   Example)

   G43 X_ H_ ; (specified tool length compensation along the X–axis)
   G443 X_ H00; (Cancels tool length compensation along the X–axis)

4) Example of tool length compensation

(a) Tod length compensation (No.1, 2 and 3 borings)



H01 = –4.0 (Offset amount)

N1	G91	G00	X120.0	Y80.0;		(1)
N2		G43		Z–32.0	H01;	(2)
N3		G01		Z–21.0	F1000;	(3)
N4		G04			P2000;	(4)
N5		G00		Z21.0;		(5)
N6			X30.0	Y–50.0;		(6)
N7		G01		Z–41.0;		(7)
N8		G00		Z41.0;		(8)
N9			X50.0	Y30.0;		(9)
N10		G01		Z–25.0;		(10)
N11		G04			P2000;	(11)
N12		G00		Z57.0	H00;	(12)
N13			X–200.0	Y–60.0;		(13)
N14			M2;			

WARNING When the offset amount is changed due to a change of the offset number, the offset value changes to the new offset amount, the new offset amount is not added to the old offset amount.

H01 . . . . Offset amount 20.0
H02 . . . . Offset amount 30.0

G90 G43 Z100.0 H01; Z will move to 120.0
G90 G43 Z100.0 H02; Z will move to 130.0

NOTE 1 A D code cannot be used to specify the tool length compensation.

NOTE 2 Parameter setting enables application of offset along the Z–axis whichever value α is in the block for which G43 or G44 is specified.

NOTE 3 Tool length compensation can be applied along multiple axes. The following commands apply tool length compensation along the Z– and W–axes.

G43Z DDHDD; Tool length compensation applied along the Z–axis

G43WDDHDD; Tool length compensation applied along the W–axis

15.2 Tool Offset (G45 – G48)

By specifying G45 to G48, the movement distance of the axis specified on CNC tape, etc. can be expanded or reduced by the value set in the offset memory. Table 15.2 shows the g codes and their functions.

Table 15.2 Tool offset and G codes

G code	Function
G 45	Increase the movement amount by the value stored in the offset value memory.
G 46	Decreases the movement amount by the value stored in the offset value memory.
G 47	Increases the movement amount by twice the value stored in the offset value memory.
G 48	Decreases the movement amount by twice the value stored in the offset value memory.

Since these G codes are not modal, they are effective only for the specified block.

Once selected by D code, the offset value remains unchanged until another offset value is selected. Offset values can be set within the following range:

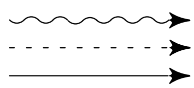
	mm input	inch input
Offset value	0 to ± 999.999 mm	0 to ± 999.999 inches
	0 to ± 999.999 deg	0 to ± 999.999 deg

This tool offset is also effective for additional axes. The D00 offset value is always zero.

Increase and decrease is performed in the move direction of an axis in which the move is performed. For absolute programming, increase and decrease are performed in the move direction from the end point of the previous block toward the position specified in the block containing G45 to G48.

1) G45 command (increase by the offset value)

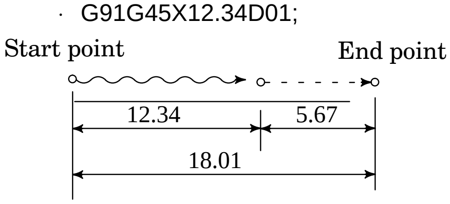
Move command value
Offset value
Actual movement



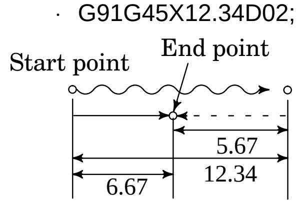
In command formats (a) to (d), D01, D02, and the direction in which the tool moves are as follows :

D01 offset : +5.67
D02 offset : –5.67
Direction : Along the X–axis

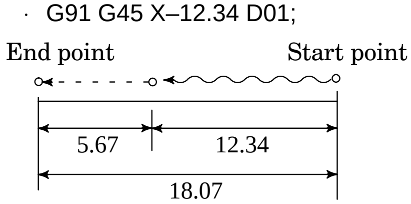
a) Move command: +12.34
Offset value: + 5.67
· G91G45X12.34D01;



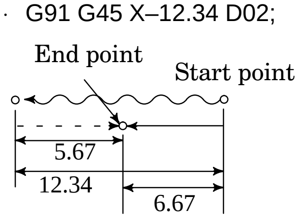
b) Move command: +12.34
Offset value: – 5.67
· G91G45X12.34D02;



c) Move command: –12.34
Offset value: + 5.67
· G91 G45 X–12.34 D01;



d) Move command: –12.34
Offset value: – 5.67
· G91 G45 X–12.34 D02;



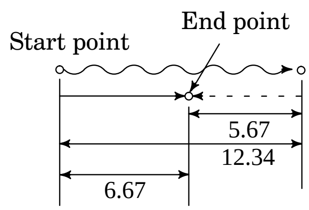
2) G46 command (decrease by offset value)

This is equivalent to reversing the positive or negative direction of the offset value specified in G45.

a) Move command: +12.34

Offset value: + 5.67

Format : G91G46X12.34D01; (offset number D01=+5.67, Move direction : X axis)



b) to d) Omitted

3) G47 command (double increase of offset value)

In command formats (a) to (d), D01, D02, and direction in which the tool moves are as follows:

D01 offset : +1.23

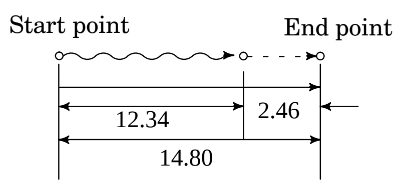
D02 offset : –1.23

Direction : Along the X–axis

a) Move command: +12.34

Offset value: + 1.23

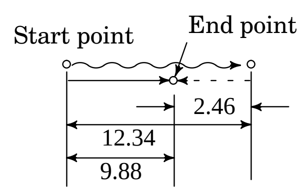
Format : G91G47X12.34D01;



b) Move command:+12.34

Offset value: – 1.23

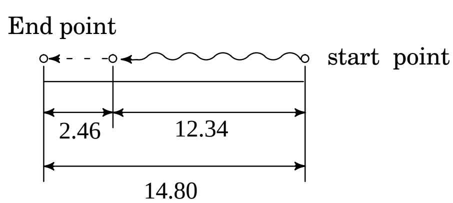
Format : G91G47X12.34D02;



c) Move command: –12.34

Offset value: + 1.23

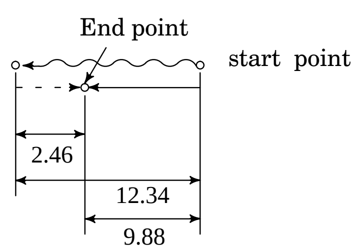
Format :G91 G47X–12.34 D01;



d) Move command: –12.34

Offset value: – 1.23

Format :G91 G47X–12.34 D02;



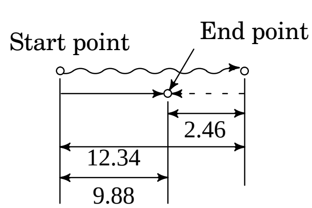
4) G48 command (double decrease of offset value)

This is equivalent to reversing the positive or negative direction of the offset value specified in G47.

a) Move command: +12.34

Offset value: + 1.23

Format : G91G48X12.34D01; (offset number D01=+1.23, Move direction : X axis)



b) to d) Omitted

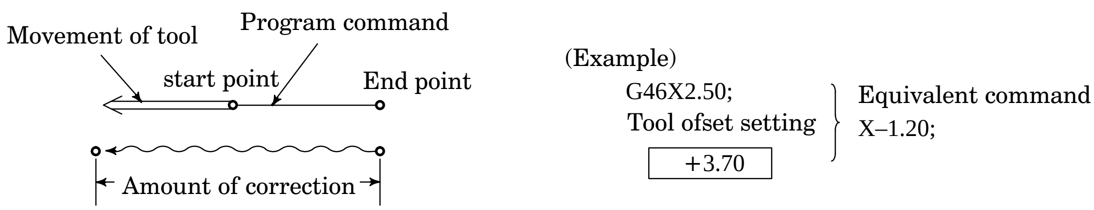
When the tool is moved by the offset value in incremental command (G91) mode, the movement value shall be specified as zero. In an absolute command (G90) mode, no operation is performed even if the movement value is specified as zero.

Example Offset value:
+12.34 (offset No. 01)

Equivalent command	G91G45X0D01;	G91G46X0D01;	G91G45X–0D01;	G91G46X–0D01;
CNC comand	X 12.34	X –12.34	X –12.34	X 12.34

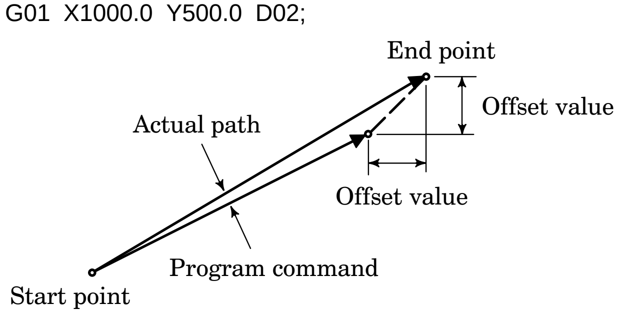
5) Notes on tool offset

WARNING When the specified direction is reversed by decrease as shown in the figure below, the tool moves in the opposite direction.

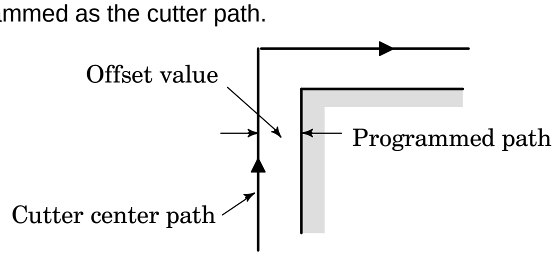


NOTE 1 When G45 to G48 is specified to n axes (n=1–6) simultaneously in a motion block, offset is applied to all n axes.

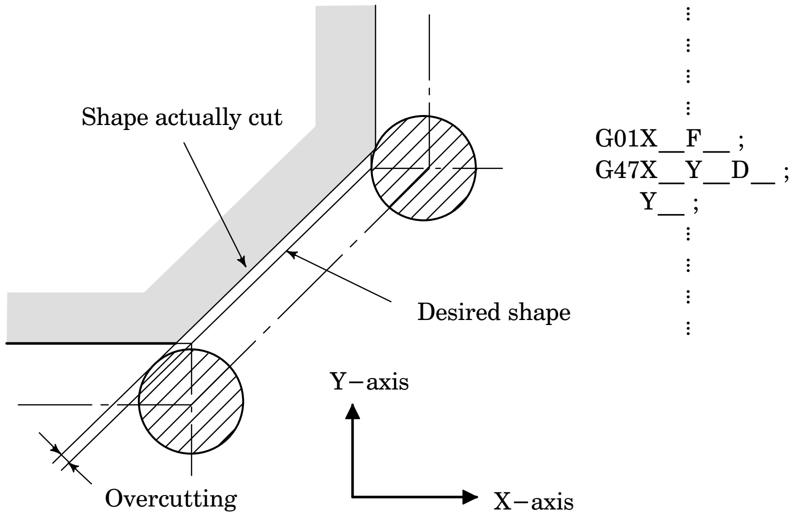
G45 specified for two axes
Move command: X1000.0Y500.0
Offset value : +200.0, offset No. 02
Program command:
G45 G01 X1000.0 Y500.0 D02;

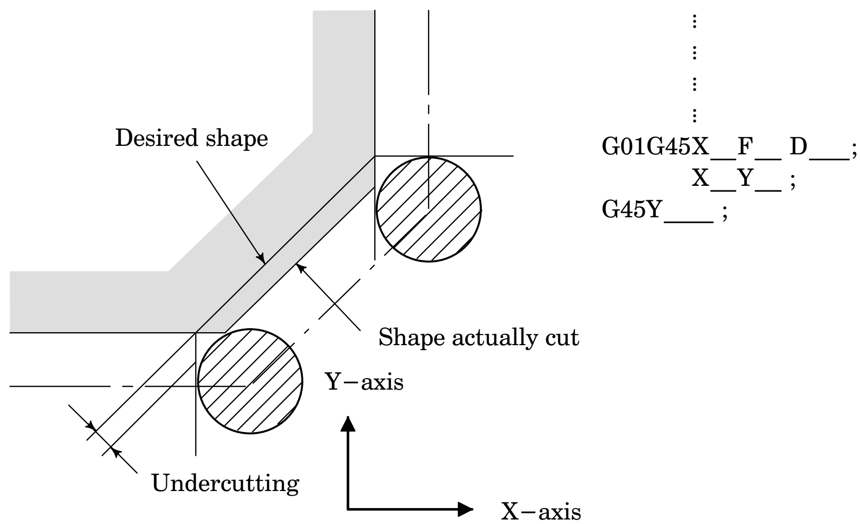


NOTE 2 When the cutter radius value is set in the offset memory, the work piece shape can be programmed as the cutter path.



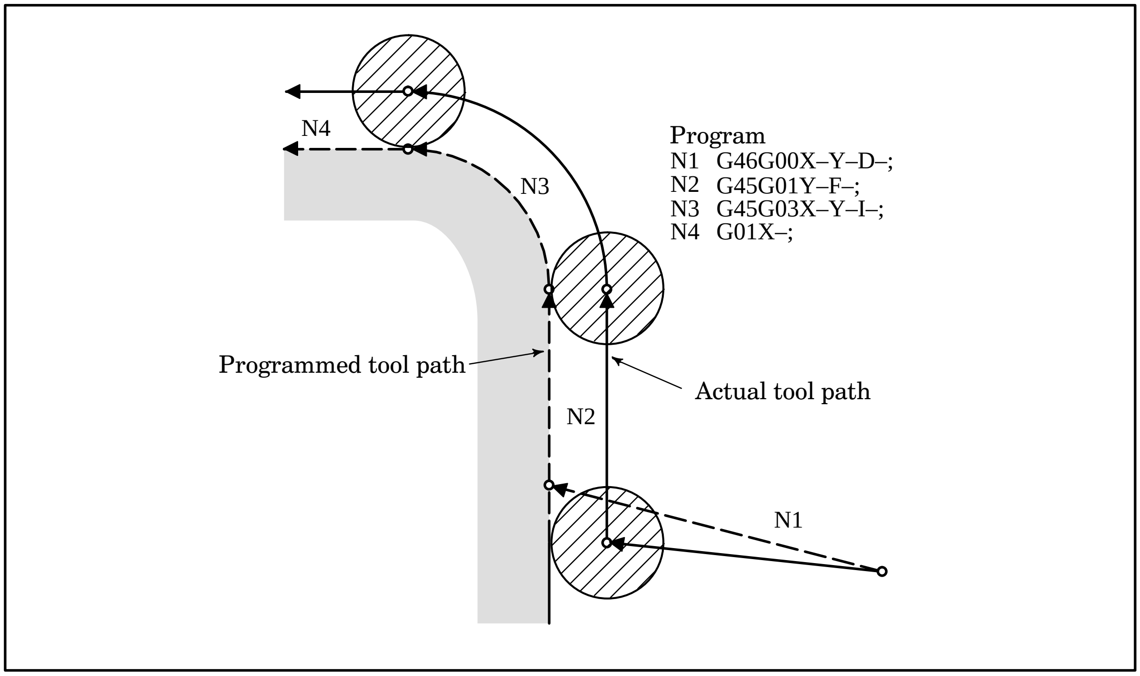
When the cutter is offset only for cutter radius or diameter in taper cutting, over cutting or undercutting occurs. Therefore, use cutter compensation (G40 or G42) shown in section 15.3.



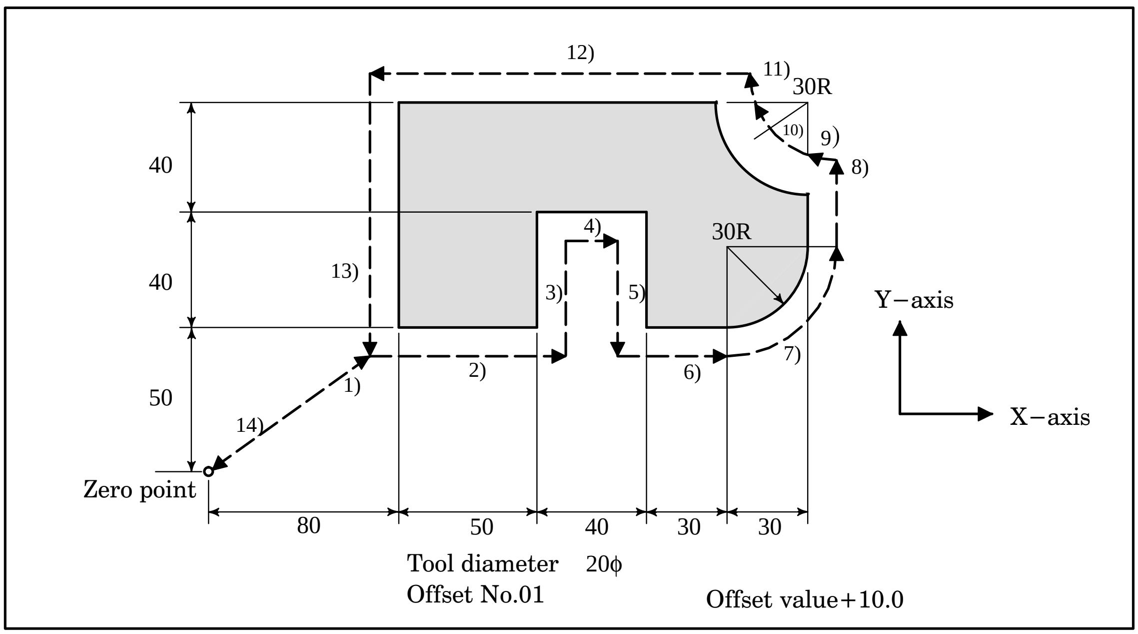


NOTE 3 Tool offset can be applied to circular interpolation (G02, G03) with the G45 to G48 commands only for 1/4 and 3/4 circles using addresses I, J and K by the parameter setting, providing that the coordinate rotation be specified at the same time. This function is provided for compatibility with the conventional CNC tape without any cutter compensation. The function should not be used when a new CNC program is prepared.

(a) Tool offset for circular interpolation



(b) Program using tool offset



Program

1) G91 G46 G00 X80.0 Y50.0 D01;
2) G47 G01 X50.0 F120;
3) Y40.0;
4) G48 X40.0;
5) Y–40.0;
6) G45 X30.0;
7) G45 G03 X30.0 Y30.0 J30.0;
8) G45 G01 Y20.0;
9) G46 X0; ······· Decreases toward the positive direction for movement amount “0”. The tool moves in the –X direction by the offset value.
10) G46 G02 X–30.0 Y30.0 J30.0;

11) G45 G01 Y0; ····· Increase toward the positive direction for movement amount “0”. The tool moves in the +Y direction by the offset value.

12) G47 X–120.0

13) G47 Y–80.0;

14) G46 G00 X–80.0 Y–50.0;

NOTE 4 D code should be used in tool offset mode (G45 to G48). However, H code can be used by setting the parameter because of compatibility with conventional CNC tape format. The H code must be used under tool length offset cancel (G49).

NOTE 5 G45 to G48 are ignored in canned cycle mode. Perform tool offset by specifying G45 to G48 before entering canned cycle mode and cancel the offset after releasing the canned cycle mode.

NOTE 6 G45 to G48 (tool offset) must not be used in the G41 or G42 (cutter compensation) mode.

NOTE 7 Do not use G45 to G48 when the coordinate system is rotated. When multiple axes are moved with coordinate rotation, coordinates are not extended or reduoed along the direction specified by the program for the same reason as in note1. The actual path may shift from that specified by the program.

15.3 Cutter Compensation C (G40 – G42)

15.3.1 Cutter compensation function

In the figure 15.3.1 , in order to cut a workpiece indicated as A with an R–radius tool, the path for the center of the tool is the B which is separated R distance from the A. The tool being separated some distance like this is called offset. By the cutter compensation function, the tool path being separated some distance (namely, offset) is computed.

Therefore, the workpiece shape is programmed with the cutter compensation mode by a programmer, and in machining, if the cutter radius (offset amount) is measured and set to the CNC, the tool path is offset (path B) regardless of the program.

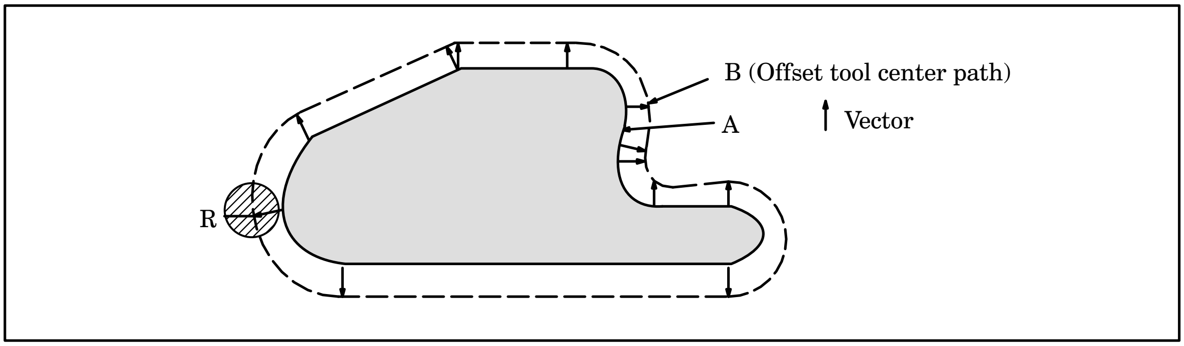


Fig. 15.3.1 Cutter compensation and vector

The methods for cutter compensation are classified into cutter compensation B and cutter compensation C. Only cutter compensation C is described here. Cutter compensation C is almost the same as cutter compensation B. They differ only in the following point. For cutter compensation B, an offset cannot be applied at an internal acute angle of up to 90,.

An appropriate arc must be specified for the internal acute angle when cutter compensation B is programmed.

15.3.2 Offset (D code)

The offset can be set in offset memory.

Preset the offset establishing a correspondence with the value of a D code (offset number) to be programmed on the CRT/MDI panel. Alternatively, punch the data in the tape to input it in the tape reader.

Offset 00, namely, the offset corresponding to D00 is always 0. The offset corresponding to D00 cannot therefore be set.

15.3.3 Offset vector

The offset vector (Fig.15.3.1) is the two dimensional vector that is equal to the amount assigned by D code. It is calculated inside the control unit, and its direction is up–dated in accordance with the progress of the tool in each block. This offset vector (hereinafter called vector) is produced inside the control unit in order to find out how much the tool motion should be offset, and is used to compute a path offset from the programmed path by the tool radius. The offset vector is deleted by reset.

This vector always follows the tool as its progress, and it is very important, when making a program, to understand the status of the vector. Read the following description carefully to understand how the vector is generated.

15.3.4 Plane selection and vector

Offset calculation is carried out in the plane determined by G17, G18 and G19, (G codes for plane selection.) This plane is called the offset plane. For instance, when the XY plane has been selected, the offset calculations are carried out using (X, Y) or (I, J) in the program tape and the vector is computed. The coordinate values of an axis not in the offset plane are not subject to the offset, and the command values for that axis on the program tape are used as they are.

In simultaneous 3 axes control, the tool path projected on the offset plane is compensated.

The offset plane is changed during the offset cancel mode. If it is performed during the offset mode, an alarm (No. 37) is displayed and the machine is stopped.

G code	Offset plane
G17	Xp – Yp plane
G18	Zp – Xp plane
G19	Yp – Zp plane

Xp : X–axis or an axis parallel to the X–axis
Yp : Y–axis or an axis parallel to the Y–axis
Zp : Z–axis or an axis parallel to the Z–axis

NOTE For details of plane selection, see Section 8.4.

Before specifying an offset plane containing an auxiliary axis, specify to which of the X–, Y–, and Z–axes the auxiliary axis is parallel using the parameter. Unless the auxiliary axis is parallel to one of the axes, an offset plane cannot be created.

To specify the offset plane containing an auxiliary axis, specify the auxiliary axis with a G code such as G17, G18, or G19.

i) G17 X__Y__ . . . . . . X–Y plane
ii) G17 U__Y__ . . . . . . U–X plane (The U–axis is parallel to the X–axis.)
iii) G17 Y__ . . . . X–Y plane
iv) G17 . . . . X–Y plane
v) G17 X__Y__U__ . . . Alarm
vi) G18 X__W__ . . . . . . X–W plane (The W–axis is parallel to the Z–axis.)

15.3.5 Deleting and generating vectors (G40, G41 and G42)

Using G40, G41 and G42, the deletion and generation of cutter compensation vectors is commanded. They are commanded simultaneously with G00, G01, G02 or G03 to define a mode which determines the amount and direction of offset vectors, and the direction of tool motion.

G code	Offset axis
G40	Cutter compensation cancel
G41	Cutter compensation left
G42	Cutter compensation right

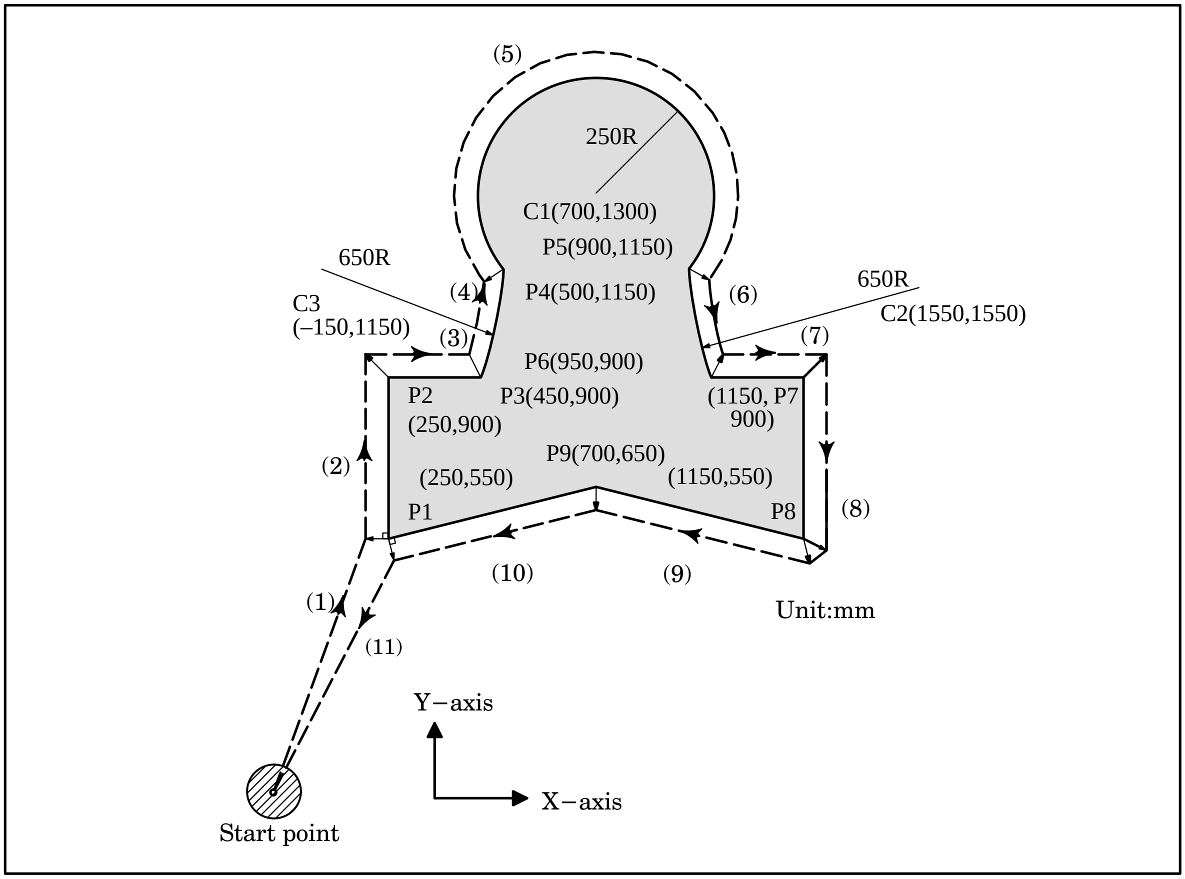
A G41 or G42 command causes the equipment to enter the offset mode. A G40 command causes the equipment to enter the cancel mode.

The offset procedure, for example, is explained in the next figure.

The block N1 is called the start–up. In this block, the offset cancel mode is changed to the offset mode (G41). At the end point of this block, the tool center is translated by the cutter radius in the direction being vertical to the next block path (from P1 to P2). The cutter compensation value is specified by D07, namely the offset number is 7, and G41 means the cutter compensation left. After being start–up, when the workpiece shape is programmed as P1 → P2 ..... → P8 → P9 → P1, the cutter compensation is performed automatically.

In the block N11 , the cutter returns to the start point by commanding G40 (offset cancel). At the end point of the block N10, the cutter center is translated vertically to the programmed path (from P9 to P1).

At the end of the program, the G40 (offset cancel) must be commanded.



In advance, the offset amount must be set with the offset number D07.

(1) G92 X0 Y0 Z0 ;
(2) N1 G90 G17 G00 G41 D07 X250.0 Y550.0 ;
(3) N2 G01 Y900.0 F150 ;
(4) N3 X450.0 ;
(5) N4 G03 X500.0 Y1150.0 R650.0 ;
(6) N5 G02 X900.0 R–250.0 ;
(7) N6 G03 X950.0 Y900.0 R650.0 ;
(8) N7 G01 X1150.0 ;
(9) N8 Y550.0 ;
(10) N9 X700.0 Y650.0 ;
(11) N10 X250.0 Y550.0 ;
(12) N11 G00 G40 X0 Y0 ;

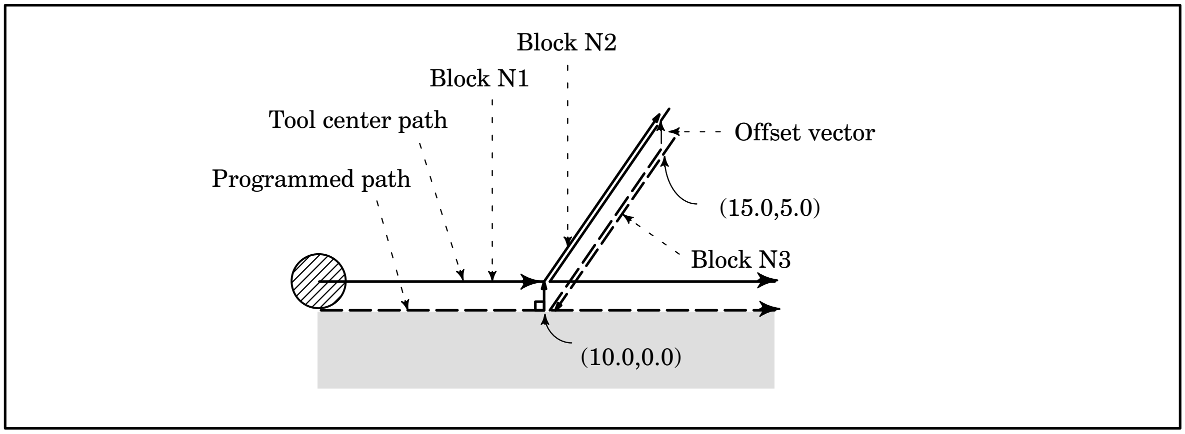
15.3.6 Vector holding (G38) and corner arc (G39)

Issuing G38 in the offset mode when the cutter compensation C function is effective enables the offset vector at the end point for the previous block to be held without calculating the intersection. Issuing G39 enables corner circular interpolation using the radius that is made equal to the offset at the corner.

(1) G38 (vector holding)

When **G38X__Y__Z__;** is issued in the offset mode, the offset vector at the end point for the previous block is held without creating an offset vector.

G38 is a single–shot code. When the next move command that does not contain the G38 command is executed, an offset vector is created again.



(Sample program)

.	(in the cutter compensation C mode, G90)
.	
N1 G38 X10.0 ;	The offset vector is held.
N2 G38 X15.0 Y 5.0 ;	The offset vector is held.
N3 G38 X10.0 Y0.0 ;	The offset vector is held.
N4 X20.0 ;	
.	
.	

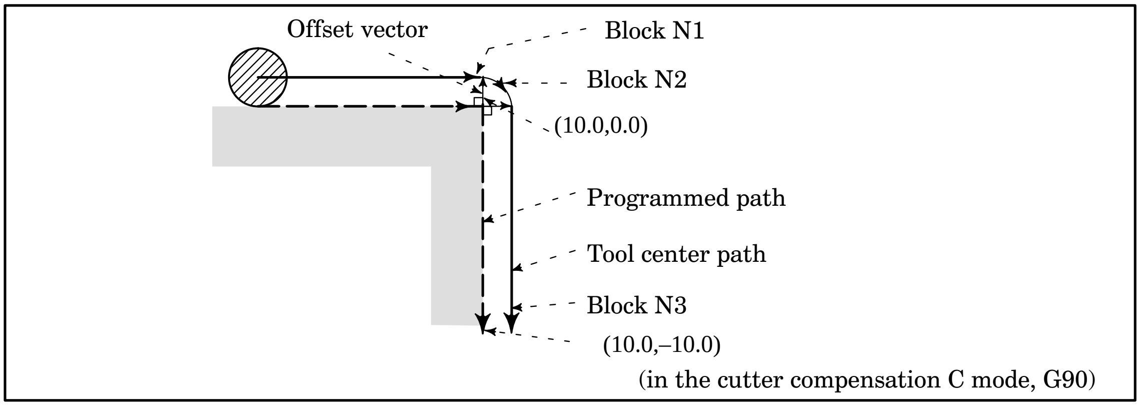
NOTE 1 Use G38 in the G00 or G01 mode.
NOTE 2 When the G38 block contains a move command, do not specify I, J, or K.
NOTE 3 When the G38 block contains not a move command but I, J, and K, an offset vector is not held. In this case, the block does not specify movement.

(2) G39(corner arc)

Issuing **G39;** or **G39 I__J__K__;** in the offset mode enables corner circular interpolation using the radius that is made equal to the compensation at the corner. Whether G41 or G42 is specified prior to G39 determines whether the arc is clockwise or counterclockwise. G39 is a single–shot code.

(i) G39 without I, J, and K

When G39; is issued, a corner arc is created such that the end–point vector of the arc is perpendicular to the start–point offset offset vector of the next block.

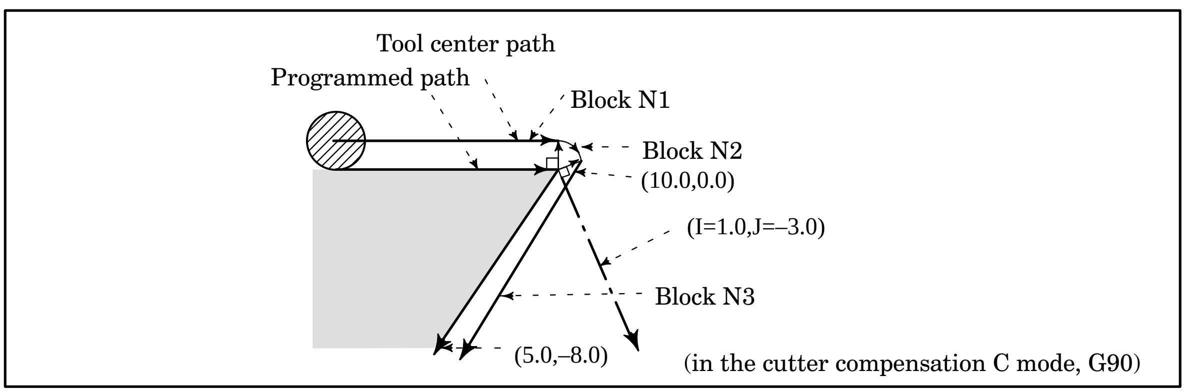


(Sample program)

```
      ·          (in the cutter compensation C mode, G90)
      ·
N1 X10.0 ;
N2 G39;        Corner circular interpolation
N3 Y–10.0 ;
      ·
      ·
```

(ii) G39 with I, J, and K

When G39 with I, J, and K is issued, a corner arc is created such that the end–point vector is perpendicular to the vector specified with I, J, and K.



(Sample program)

```
      ·  (in the cutter compensation C mode, G90)
      ·
N1 X10.0  ;
N2 G39 I1.0 J–3.0 ;      Corner circular interpolation
N3 X5.0 Y–10.0     ;
      ·
      ·
```

NOTE	No movement can be specified in the G39 block.

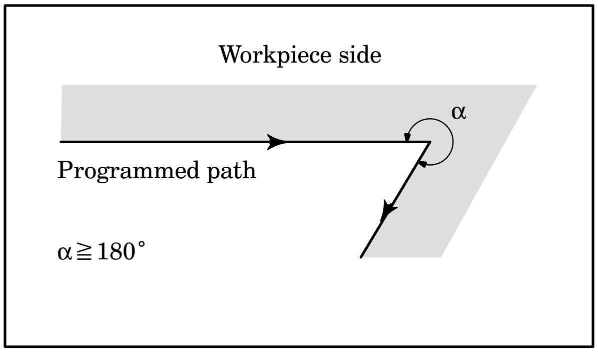
15.3.7 Details of cutter compensation C

In this item, the details of the cutter compensation C is explained.

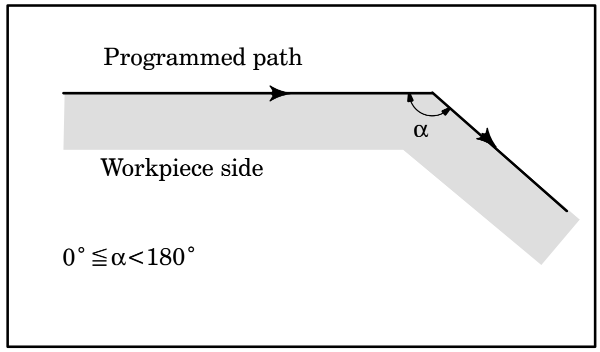
(a) Inner side and outer side

When an angle of intersection created by tool paths specified with move commands for two blocks is over 180°, it is referred to as "inner side." When the angle is between 0° and 180°, it is referred to as "outer side." See the following figures.

(1) Inner side



(2) Outer side



(b) Meaning of symbols

S : Stop point of a single block

L : Straight line

C : Arc

1) Cancel mode

At the beginning when power is applied the control is in the cancel mode. In the cancel mode, the vector is always 0, and the tool center path coincides with the programmed path. The cancel mode (G40) is programmed before the end of a program.

When the program ends in the offset mode, positioning cannot be made to the terminal point of the program, and the tool position will be separated from the terminal position by the vector value.

2) Start up

In the cancel mode, when a block which satisfies all of the following conditions is executed, the equipment enters the offset mode. This is called start up .

i) G41 or G42 has been commanded. Or, it has already been commanded, and the control enters the G41 or G42 mode.

ii) The offset number for cutter compensation is not 00.

iii) A move in any of the axes (I, J, K are excluded) in the offset plane has been commanded, and its commanded move is not 0.

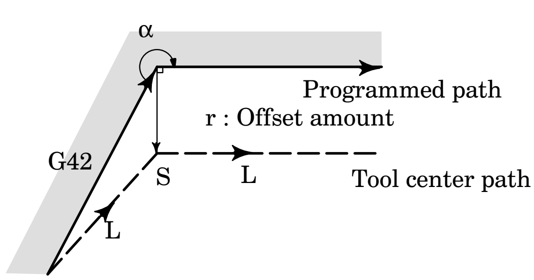
In the start up block, arc commands (G02, G03) are not allowed, as an alarm is generated and the tool stops. At start up, two blocks are read. After the first block is read and executed, the next block enters the cutter compensation buffer (which cannot be indicated)

In the meantime, in the case of single block mode, two blocks are read and the machine stops after the first block execution.

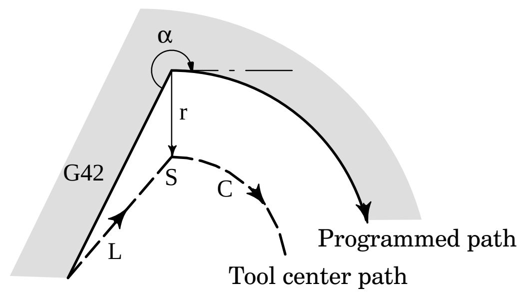
Thereafter, 2 blocks are read in advance normally, inside the CNC, there are three blocks, the block under execution, and the next two blocks which are entered into buffer.

a) When going around an inside corner (180° ≦ α)

i) Linear→Linear

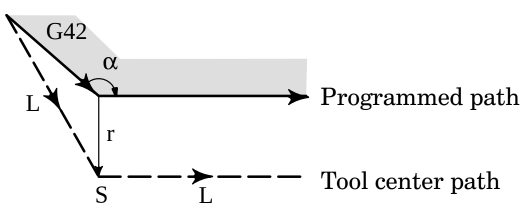


ii) Linear→Arc

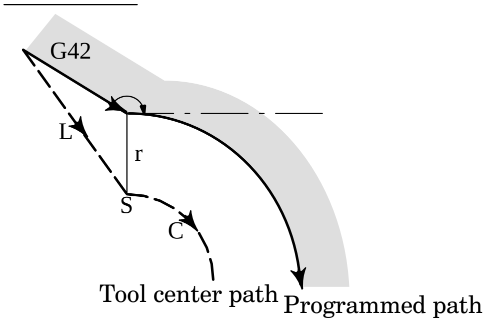


b) Whwn the tool goes around the outside of a corner at an obtuse angle (90° ≦ α ≦ 180°).
Tool path in start up or cancel has two types A and B, and they are selected by parameter CSU(No.6001#0).

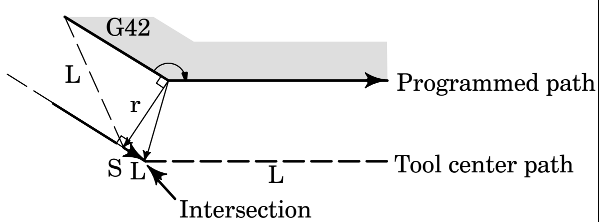
Type A i) Linear→Linear



ii) Linear→Arc

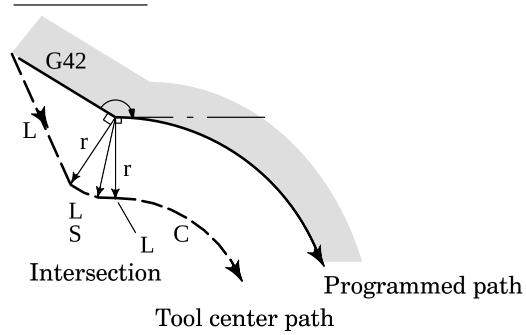


Type B i) Linear→Linear



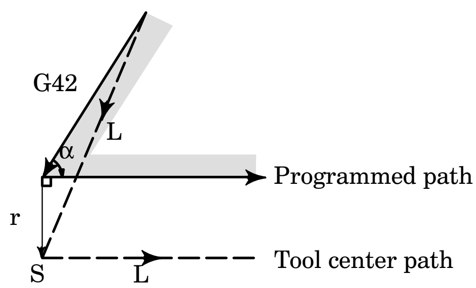
(Note) The intersection is the point where two offset paths calculated by continuous two blocks are intersected.

ii) Linear→Arc

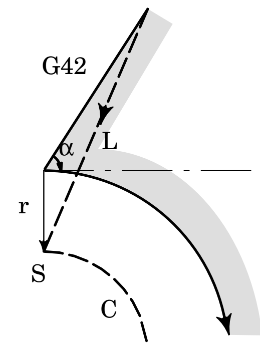


c) When going around the outside of an acute angle ($\alpha \leqq 90°$)
Tool path in start up or cancel has two types A and B,and they are selected by parameter CSU (No.6001#0).

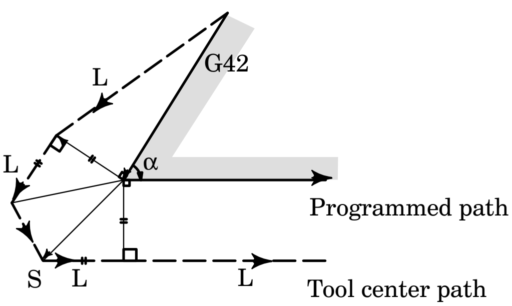
Type A i) Linear→Linear



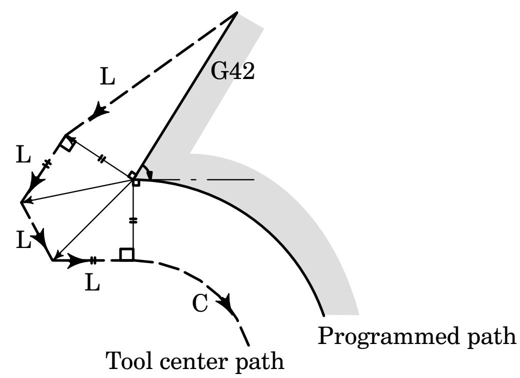
ii) Linear→Arc



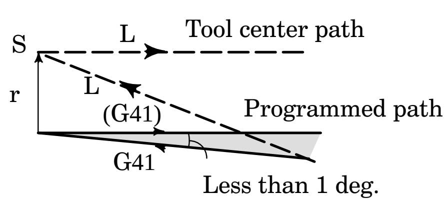
Type B i) Linear→Linear



ii) Linear→Arc



d) When the tool goes around the outside linear→linear at an acute angle less than 1 degree, compensation is performed as follows

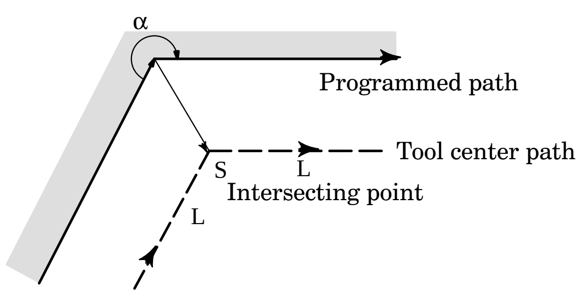


3) In offset mode

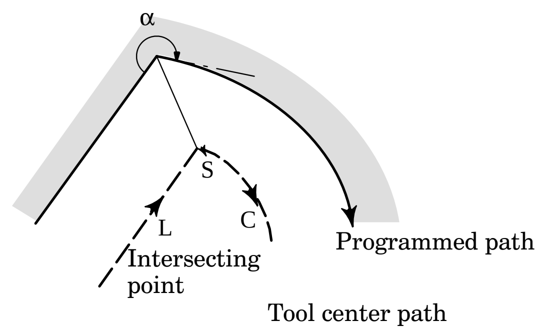
In the offset mode, offset is performed correctly, if non–positioning commands such as auxiliary functions or dwell would not be programmed in two or more successive blocks. Otherwise overcutting or under cutting will occur. Offset plane must not be changed during offset mode. Otherwise an alarm is displayed and the tool stops.

a) When going around the inside (180° ≦ α)

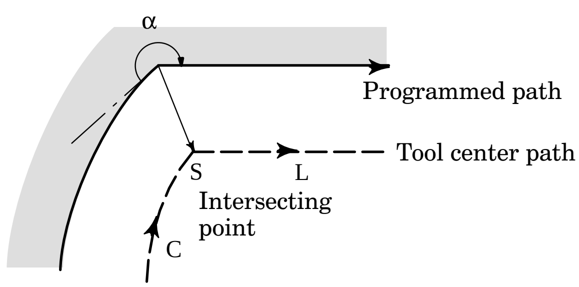
i) Linear→Linear



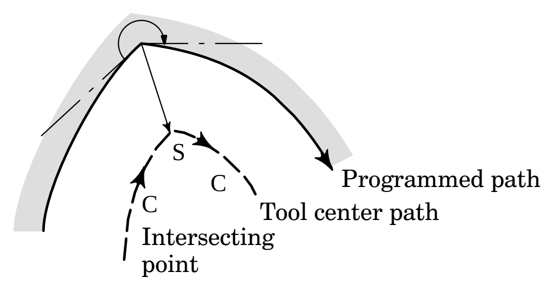
ii) Linear→Arc



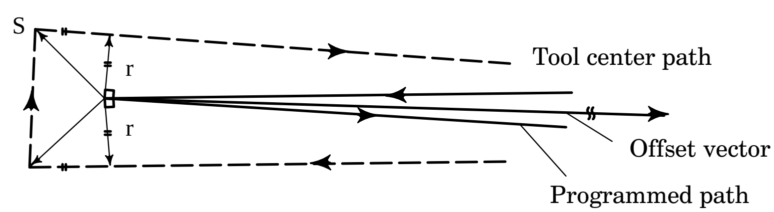
iii) Arc→Linear



iv) Arc→Arc



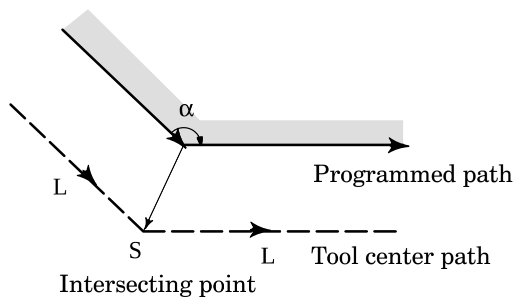
v) Straight line to straight line in the case in which it goes around at a narrow angle (less than 1°) and the offset vector became abnormally large.



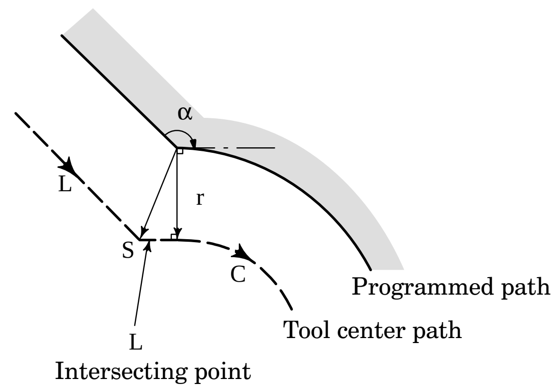
Also in case of arc to straight line, straight line to arc and arc to arc, the reader should infer in the same procedure.

b) When going around an outside corner at an obtuse angle($90^\circ \leqq \alpha < 180^\circ$)

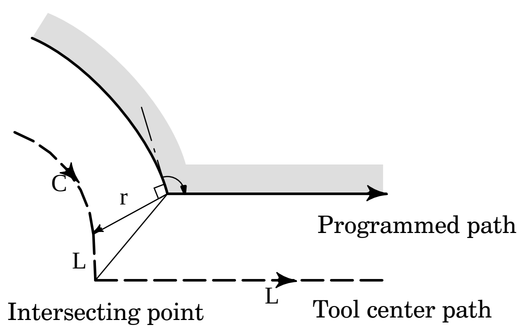
i) Linear→linear



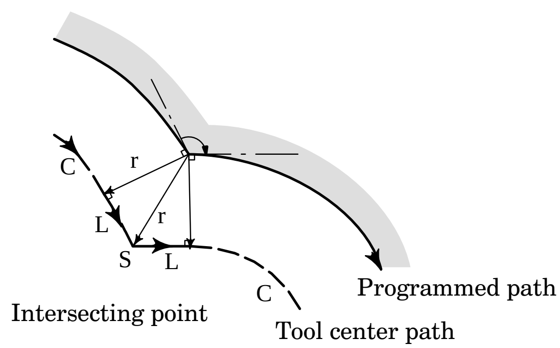
ii) Linear→Arc



iii) Arc→Linear

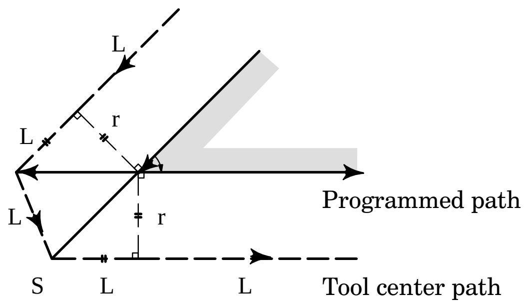


iv) Arc→Arc

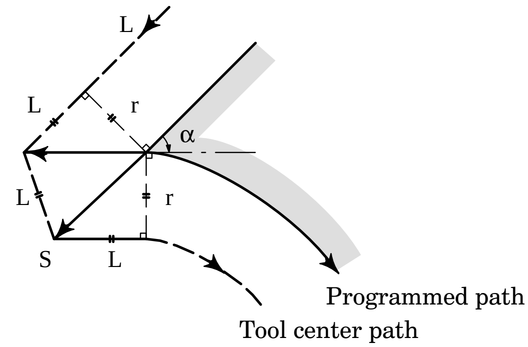


c) When going outside corner at an acute angle

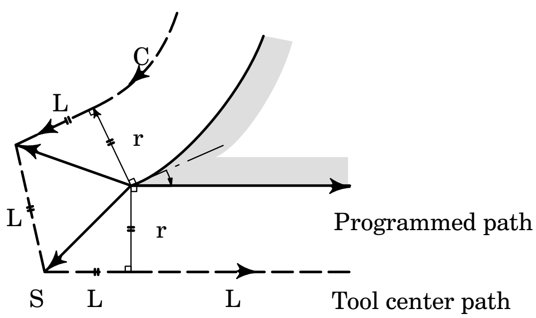
i) Linear→Linear



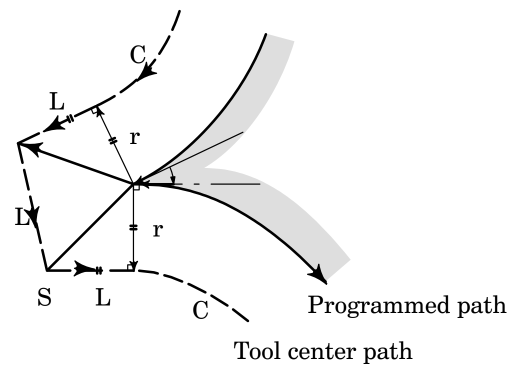
ii) Linear→Arc



iii) Arc→Linear

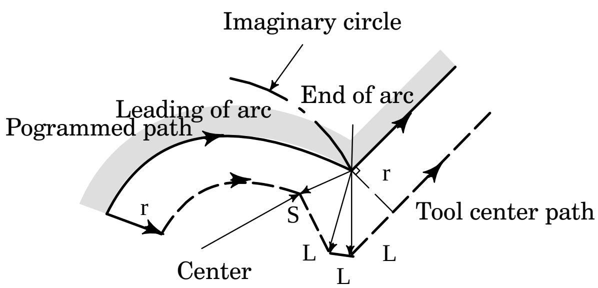


iv) Arc→Arc



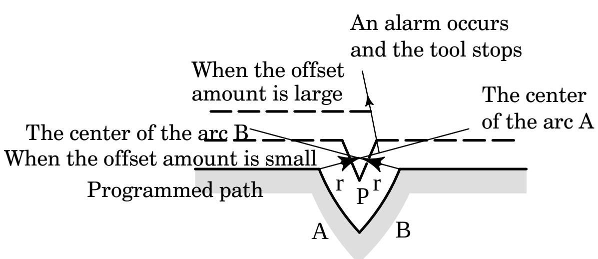
d) Exceptional cases

i) The end point of arc is not on the arc



If there is an arc, lead−in line as shown, assume an imaginary circle which passes through the arc end point and is concentrical to the arc, and draw vectors for compensation, on the assumption that the cutter compensation has been carried out for the imaginary circle. As a result, a path for the cutter center, is established according to the program that regards the arc lead−in line as a straight line. similarly consider a case of arc−arc.

ii) There is no inner intersecting point



In regard to the left figure, the intersecting point of the arcs is present on the offset path if the offset value is small, but it may disappear if the offset value is quite large. in this case, alarm (No.33) occurs at the end point of the preceding block, and the tool will stop.
Concerning the left figure, there is an intersecting point P of the offset paths for the arcs A and B, if the offset value is small, but the intersecting point cannot be found if the offset value is quite large.

4) Offset cancel

In the offset mode, when a block which satisfies any one of the following conditions is executed, the equipment enters the cutter compensation cancel mode, and the action of this block is called the offset cancel.

a) G40 has been commanded.

b) D00 has been commanded as the offset number for cutter compensation.

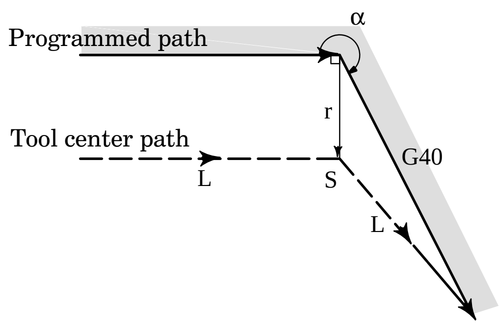
When performing offset cancel, circular arc commands (G02 and G03) are not available. If a circular arc is commanded, an alarm (No. 270) is generated and the tool stops.

In the offset cancel, the control executes the instructions in that block and the block in the cutter compensation buffer. In the meantime, in the case of a single block mode, after reading one block the control executes it and stops. By pushing the cycle start button once more, one block is executed without reading the next block.

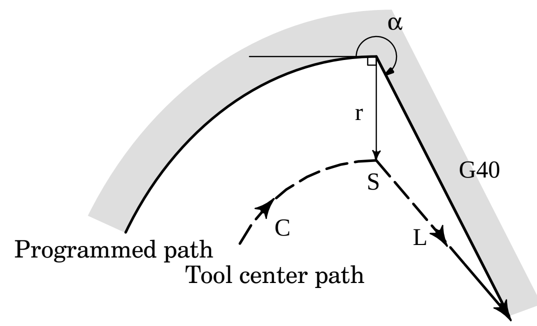
Then the control is in the cancel mode, and normally, the block to be executed next will be stored in the buffer register and the next block is not read into the buffer for cutter compensation.

a) When going around an insaide corner (α≧180°)

i) Linear→Linear

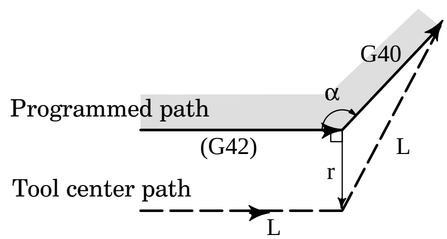


ii) Arc→Linear

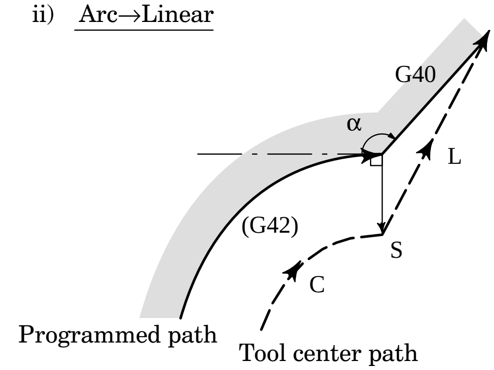


b) When going around an outside corner at an abtuse angle (90°≦α<180°)
Tool path has two types, A and B; and they are selected by parameter SUP (No.5003#0).

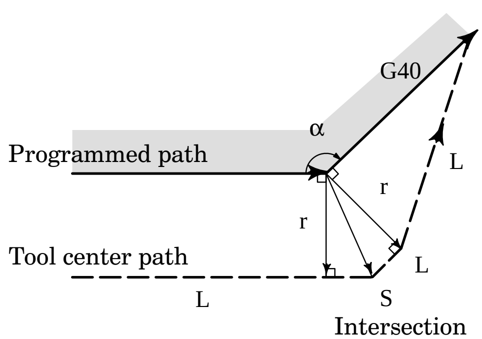
Type A i) Linear→Linear



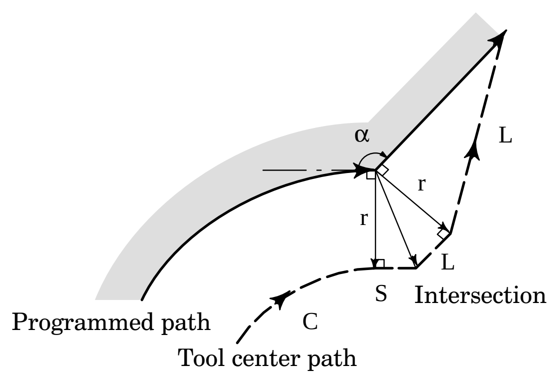
ii) Arc→Linear



Type B i) Linear→Linear

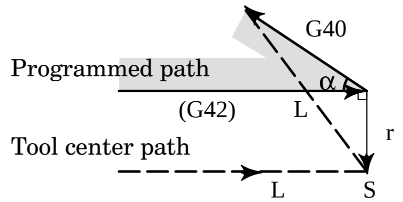


ii) Arc→Linear

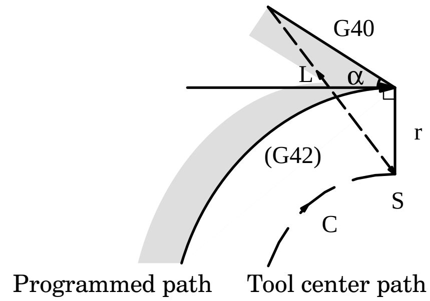


c) When going around an outside corner at an acute angle (α<90˚)
Tool path has two types, A and B:and they are selected by parameter SUP (No.5003#0)

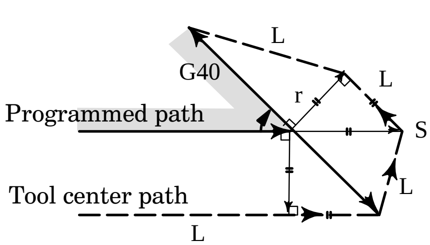
Type A i) Linear→Linear



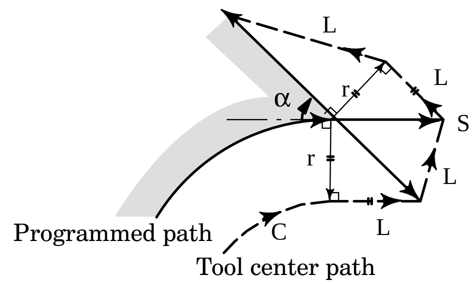
ii) Arc→Linear



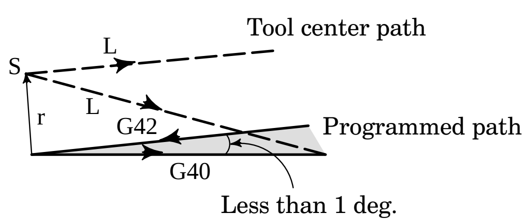
Type B i) Linear→Linear



ii) Arc→Linear



d) When the tool goes around the outside linear linear → at an acute angle less than 1 degree, compensation is performed as follows.

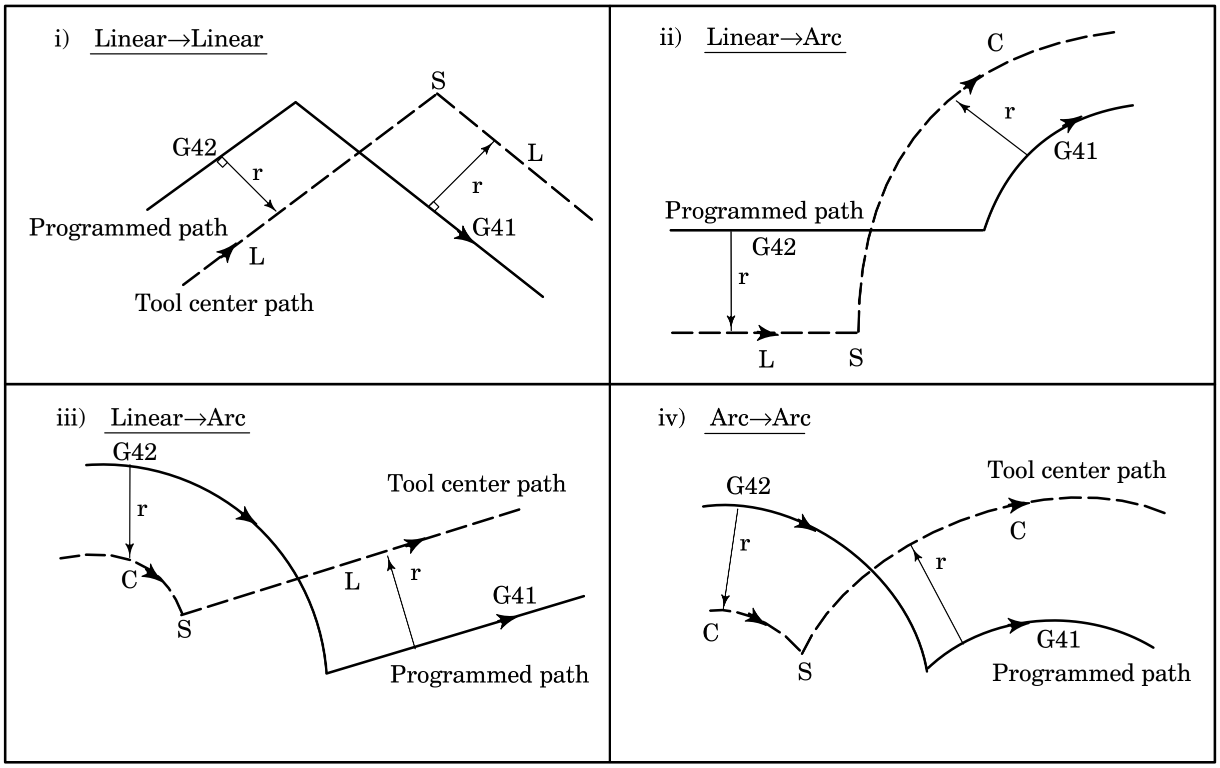


5) Change of offset direction in the offset mode

The offset direction is decided by G codes (G41 and G42) for cutter radius and the sign of offset amount is as follows.

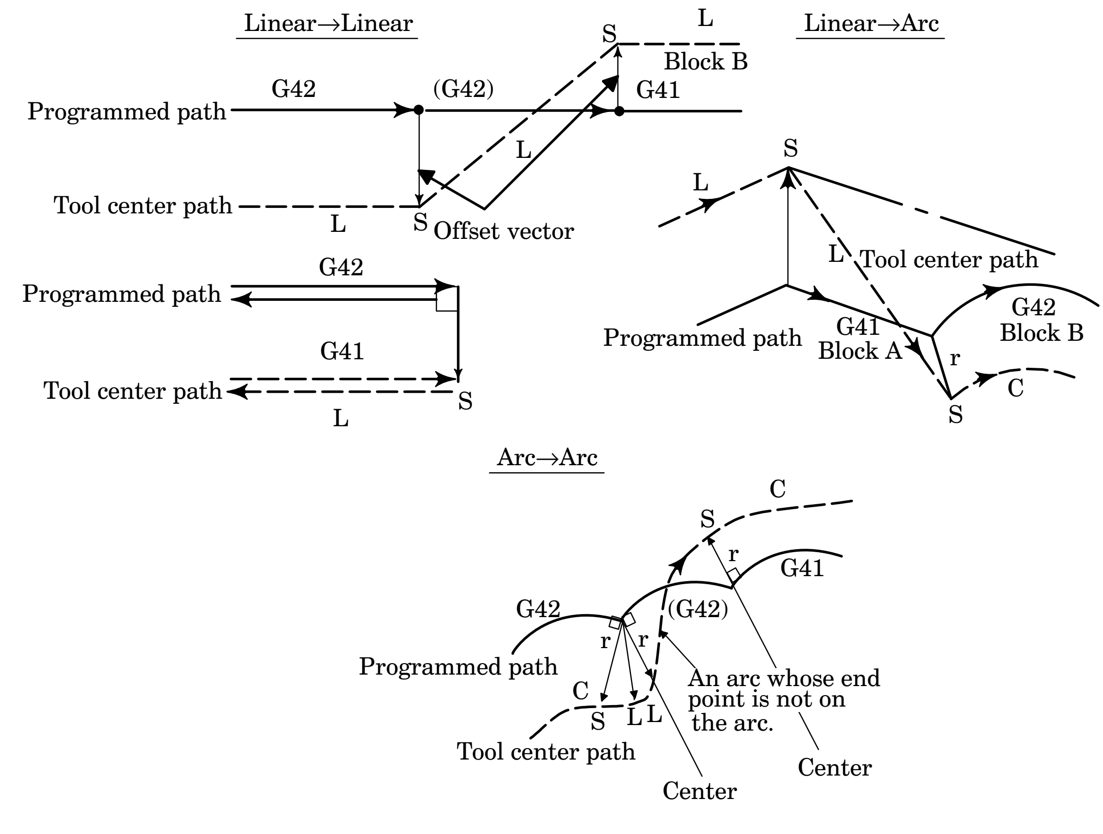
G code \ Sign of offset amount	+	–
G41	Left side offset	Right side offset
G42	Right side offset	Left side offset

The offset direction can be changed in the offset mode. If the offset direction is changed in a block, a vector is generated at the intersection of the tool center path of that block and the tool center path of a preceding block. However, the change is not available in the start up block and the block following it. (See items (i) to (iv).)



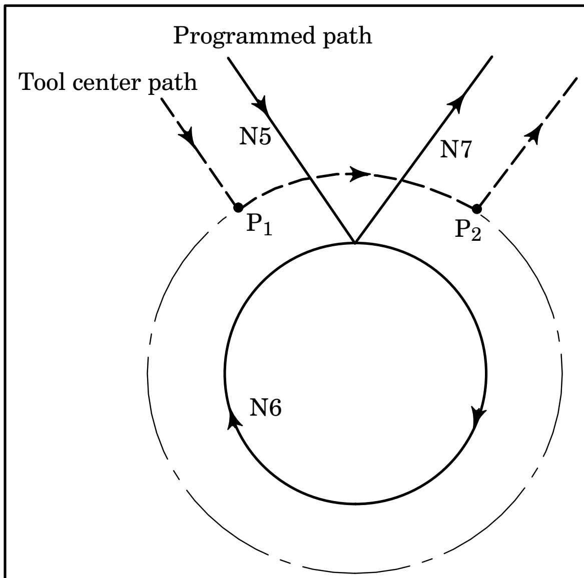
v) When an intersection is not obtained if offset is normally performed.

When changing the offset direction in block A to block B using G41 and G42, if intersection with the offset path is not required, the vector normal to block B is created at the start point of block B.



vi) When the length of tool center path becomes more than a circle because of cutter compensation

Normally there is almost no possibility of generating the above situation. However, when G41 and G42 are changed, or when a G40 was commanded with address I, J, and K the above situation can occur.



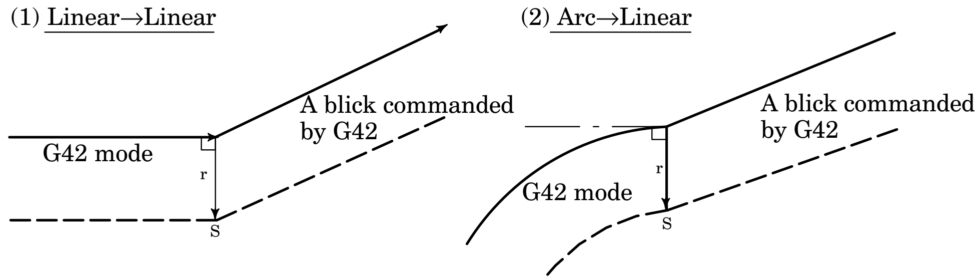
(G42)
N5 G01 G91 X500.0 Y–700.0;
N6 G41G02 J–500.0;
N7 G42 G01 X500.0 Y700.0;

In this case, the cutter compensation is not performed with more than one circle circumference; an arc is formed from P1 to P2 as shown. (Depending on the circumstances, an alarm may be displayed due to the "Interference Check" described later.) To execute a circle with more than one circumference, the circle must be specified in segments.

6) Cutter compensation G code in the offset mode

The offset vector can be set to form a right angle to the moving direction in the previous block, irrespective of machining on inner or outer wall, by commanding the cutter compensation G code (G41, G42) in the offset mode, independently. If this code is specified in a circular command, correct circular motion will not be obtained.

When the direction of offset is expected to reverse by the command of cutter compensation G code (G41, G42), refer to (5).

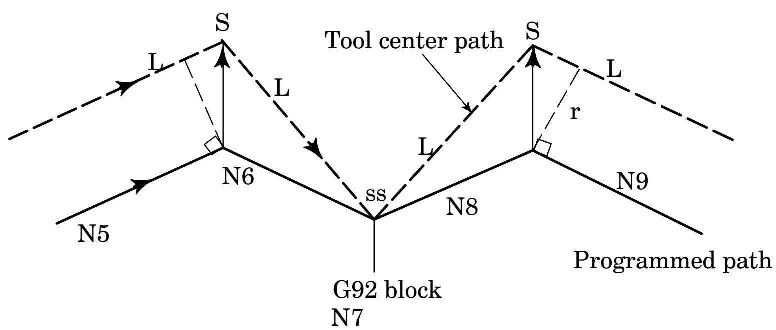


7) Command for temporary cancelling offset vector

During offset mode, if G92 (absolute zero point programming) is commanded,the offset vector is temporarily cancelled and thereafter offset mode is automatically restored (See figure above).

In this case, without movement of offset cancel, the tool moves directly from the intersecting point to the commanded point where offset vector is canceled. Also when restored to offset mode, the tool moves directly to the intersecting point.

(G41 mode)
N5 G91 G01 X 300.0 Z700.0;
N6 X–300.0 Z600.0;
N7 G92 X 100.0 Z200.0;
N8 G90 G01 X 400.0 Z800.0;



(G41 mode)
N5 G91 G01 X 300.0 Z700.0;
N6 X−300.0 Z600.0;
N7 G92 X 100.0 Z200.0;
N8 G90 G01 X 400.0 Z800.0;

8) A block without tool movement

The following blocks have no tool movement. In these blocks, the tool will not move even if cutter compensation is effected.

(1) M05; M code output

(2) S21; S code output

(3) G04 X100.0; Dwell

(4) G22 X100000; Cutting area setting

(5) G10 P01 R10.0; Offset value setting

(6) (G17) Z200.0; Move command not included in the offset plane.

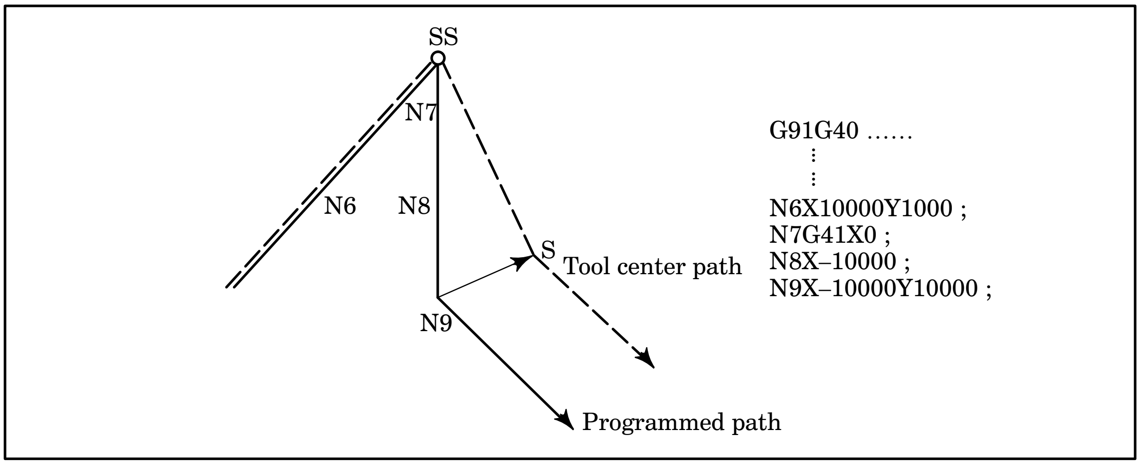
(7) G90; G code only

(8) G91 X0; Move distance is zero.

Commands (1) to (7) are of no movement.

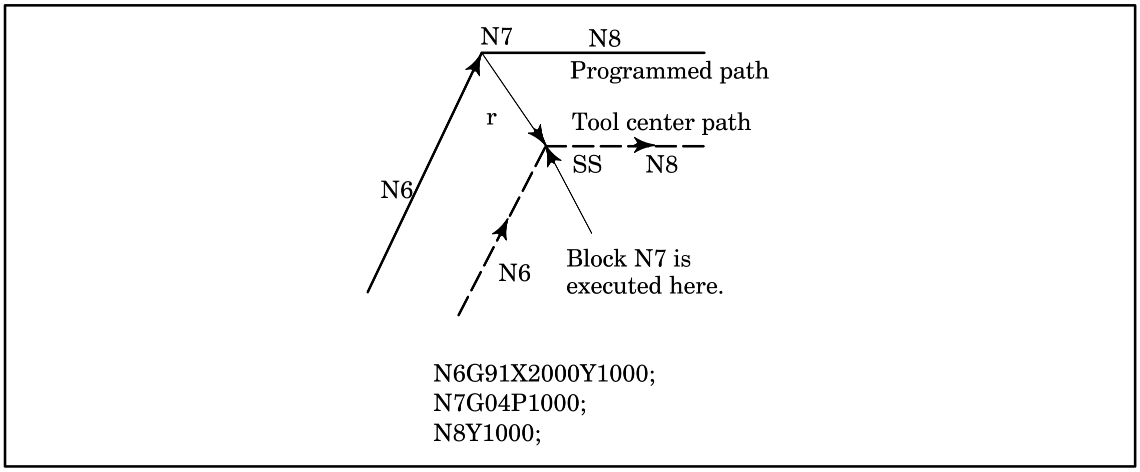
a) When commanded at start–up

If a block without tool movement is commanded at start–up, the offset vector is not produced.

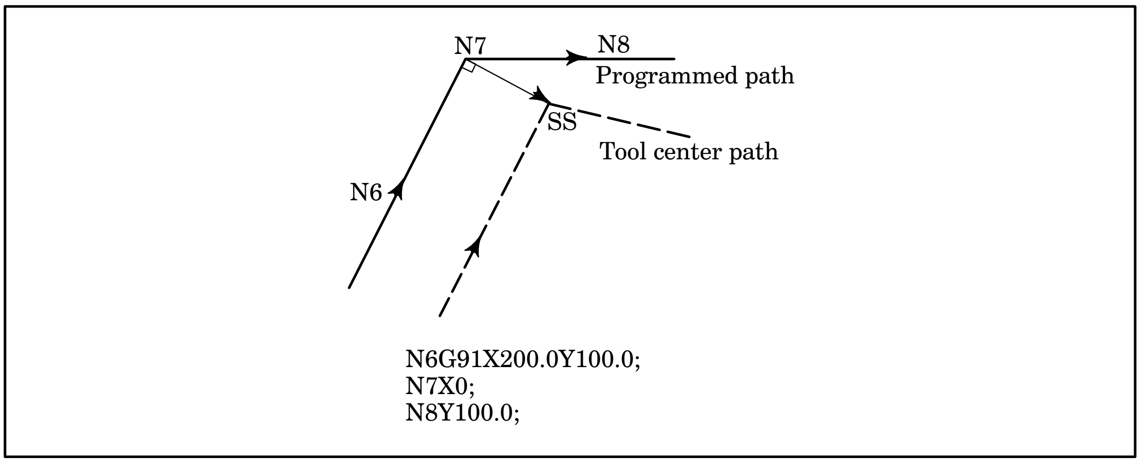


b) When commanded in offset mode

When a single block without tool movement is commanded in the offset mode, the vector and tool center path are the same as those when the block is not commanded. (Refer to item 3) Offset mode) This block is executed at the single block stop point.

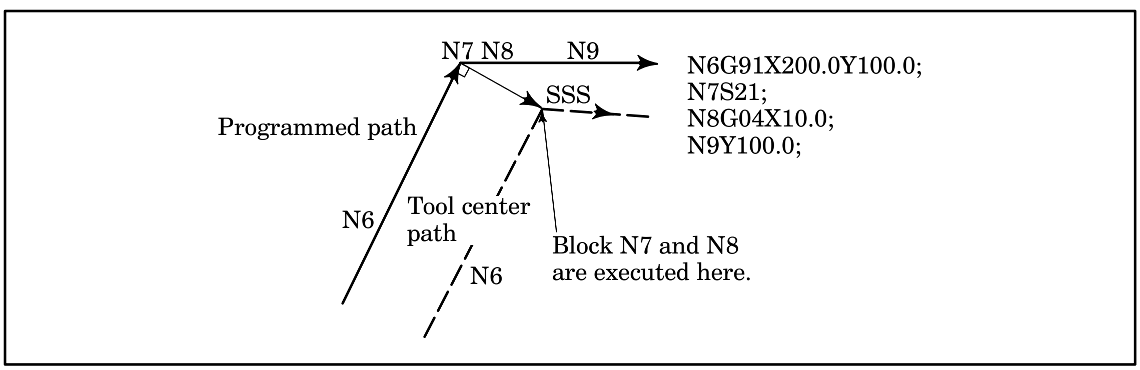


However, when the move distance is zero, even if the block is commanded singly, tool motion becomes the same as that when more than one block of without tool movement are commanded, which will be described subsequently.



Two blocks without tool movement should not be commanded consecutively.

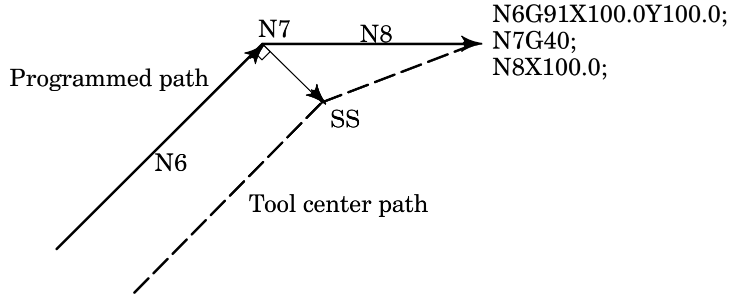
If commanded, a vector whose length is equal to the offset value is produced in a normal direction to tool motion in earlier block, so overcutting may result.



NOTE	SSS means that a single block is executed three times.

c) When commanded together with offset cancel

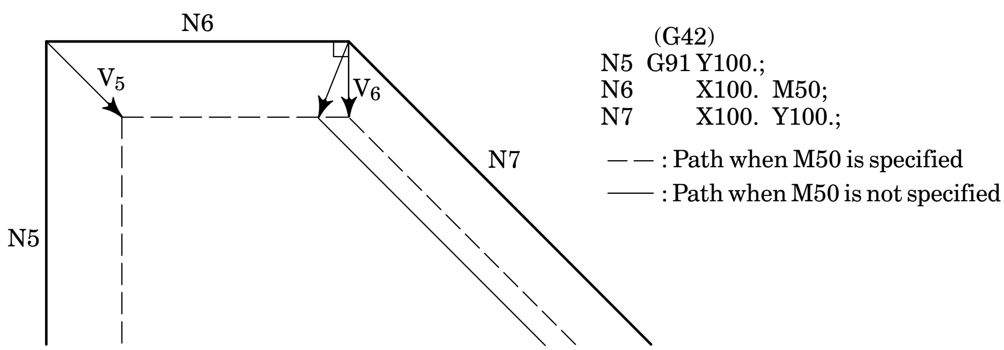
When offset cancel is specified in a block where movement is not specified, a vector is created whose length is equivalent to the specified offset in the direction perpendicular to the direction of movement specified in the preceding block.



9) Command for suppressing buffering in offset mode

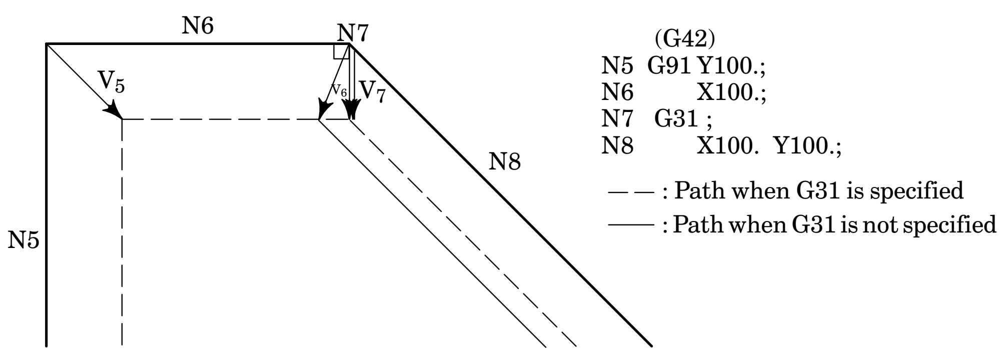
If a G or M code which suppresses buffering (advance reading) is specified in offset mode, intersection computation for determining an offset vector cannot be performed, so that a vector perpendicular to the movement specified in the current block is output at the end point of the block. The tool path after compensation, therefore, varies depending on whether such a G or M code is specified. Be careful to prevent overcutting of the workpiece as shown below.

(a) When a buffering suppression M code (M50) is specified together with a command which specifies movement on the offset plane



Because N7 cannot be read in advance, perpendicular vector V_6 is output at the end point of N6.

(b) When a buffering suppression G code (G31) is specified alone in a block which contains no move command on the offset plane

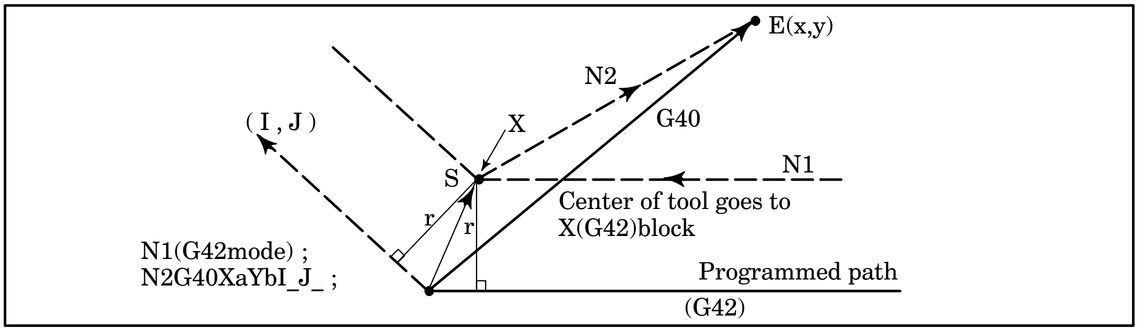


An intersection is computed between N6 and N8 because N7 involves no movement. Because N8 cannot be read in advance, however, perpendicular vector V_6 is output at the end point of N6. Because N7 involves no movement, the end point vector for the previous block (V_6) is held as V_7.

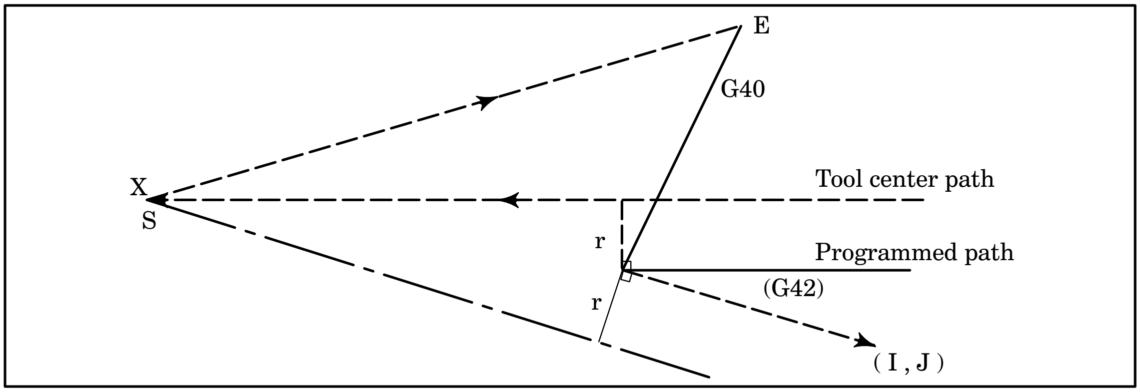
10) When a block contains a G40 and I ___ J ___ K ___ included in the offset plane

i) When the mode of the previous block is G41 or G42

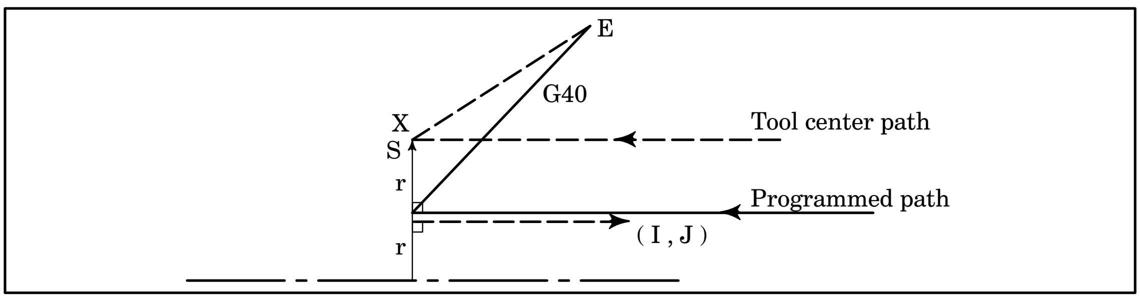
In this case the CNC assumes that the movement from the end point of the previous block in the direction of (I, J or K) has been commanded.



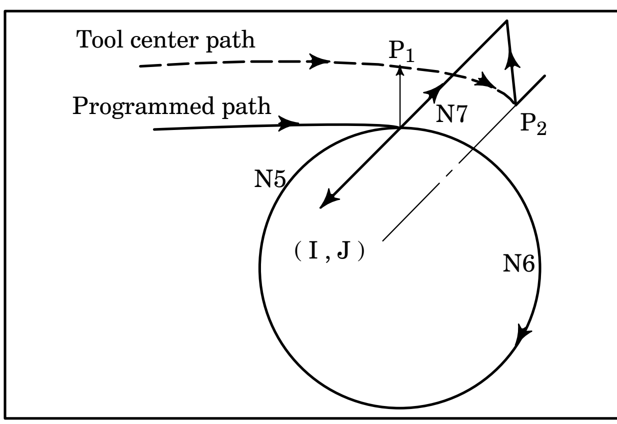
In this case, note that the CNC obtains an intersection of the tool path irrespective of whether inner or outer wall machining is specified



When an intersection is not obtainable, the tool comes to the position normal to the previous block at the end of the previous block.



ii) When the length of the tool center path become more than a circle.



(G41)
N5 G01 G91 X1000.0;
N6 G02 J–600.0;
N7 G40 G01 X500.0 Y500.0 I–10.0J–10.0;

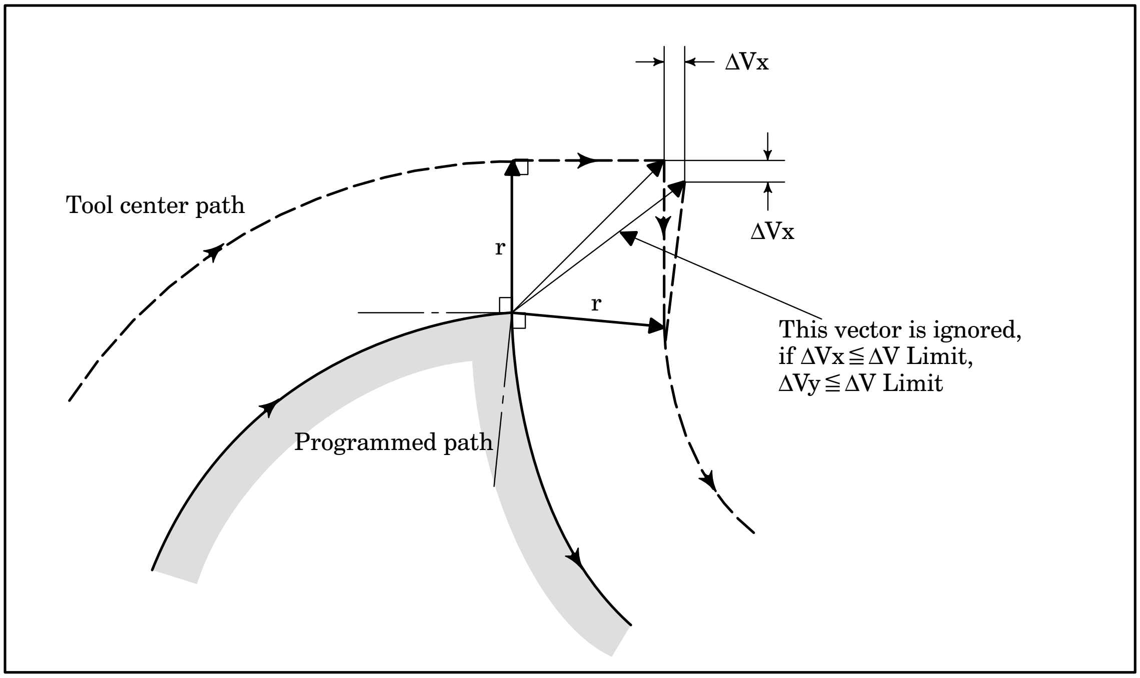
In the above case, the tool center path does not go around a circle but moves only from point P1 to P2 along an arc.

According to under some circumstance, an alarm may be generated by the interference check described later. (if it is desired to move a tool around a circle, a circle must be commanded with partitions.)

11) Corner movement

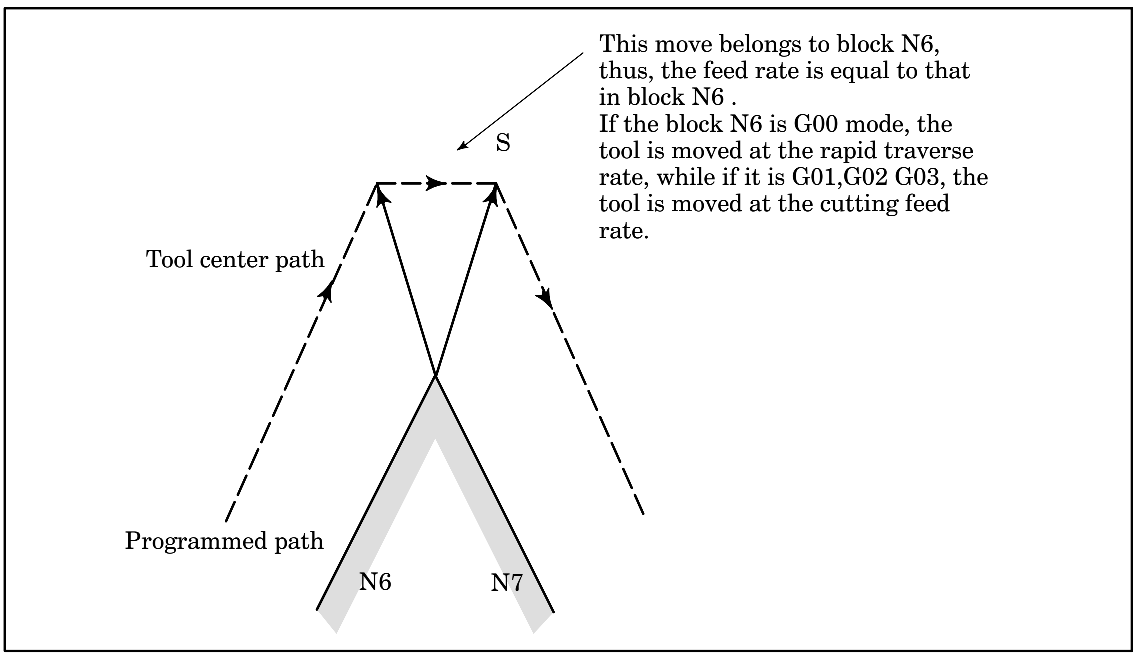
When two or more vectors are produced at the end of a block, the tool moves linearly from one vector to another. This movement is called the corner movement.

If these vectors almost coincide with each other, the corner movement isn't performed and the latter vector is ignored.

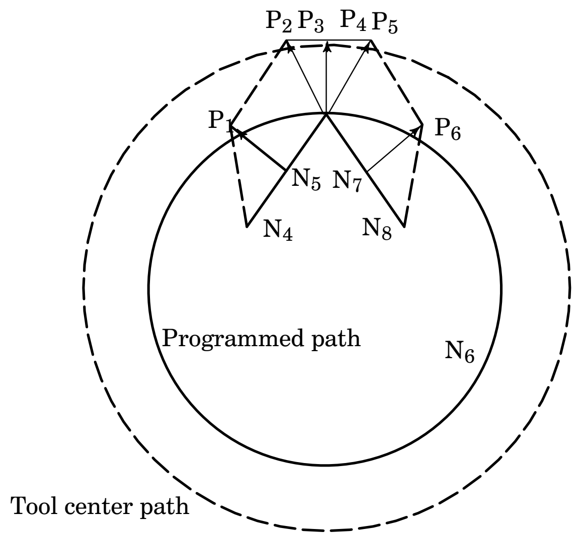


If ΔVx ≦ ΔVlimit and ΔVy ≦ ΔVlimit, the latter vector is ignored. The ΔVlimit is set in advance by parameter (No. 6010).

If these vectors do not coincide, a move is generated to turn around the corner. This move belongs to the latter block.



However, if the path of the next block is semicircular or more, the above function is not performed. The reason for this is as follows:




```
N4 G41 G91 G01 X150.0 Y200.0;
N5 X150.0 Y200.0;
N6 G02 J–600.0;
N7 G01 X150.0 Y–200.0;
N8 G40 X150.0 Y–200.0;
```

If the vector is not ignored, the tool path is as follows:

P0 → P1 → P2 → P3 → (Circle) → P4 → P5 → P6 → P7

But if the distance between P2 and P3 is negligible, the point P3 is ignored. Therefore, the tool path is as follows:

P0 → P1 → P2 → P4 → P5 → P6 → P7

Namely, circle cutting by the block N6 is ignored.

12) Interference check

Tool overcutting is called 'interference'. The interference check function checks for tool overcutting in advance. However, all interference can not be checked by this function. The interference check is performed even if overcutting does not occur.

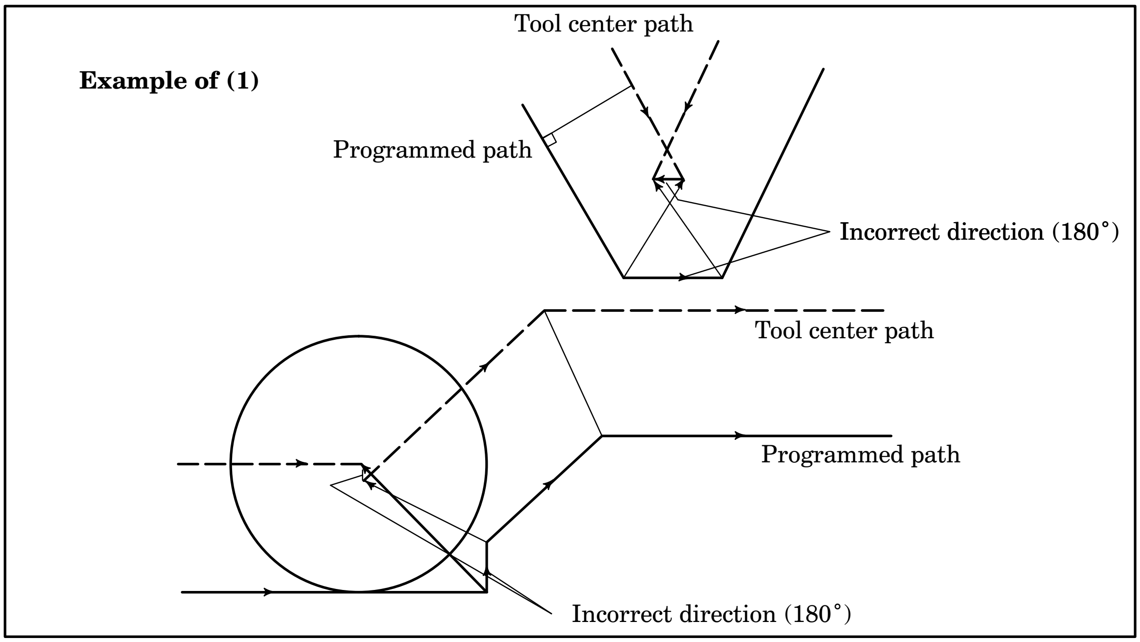
a) Reference conditions for interference

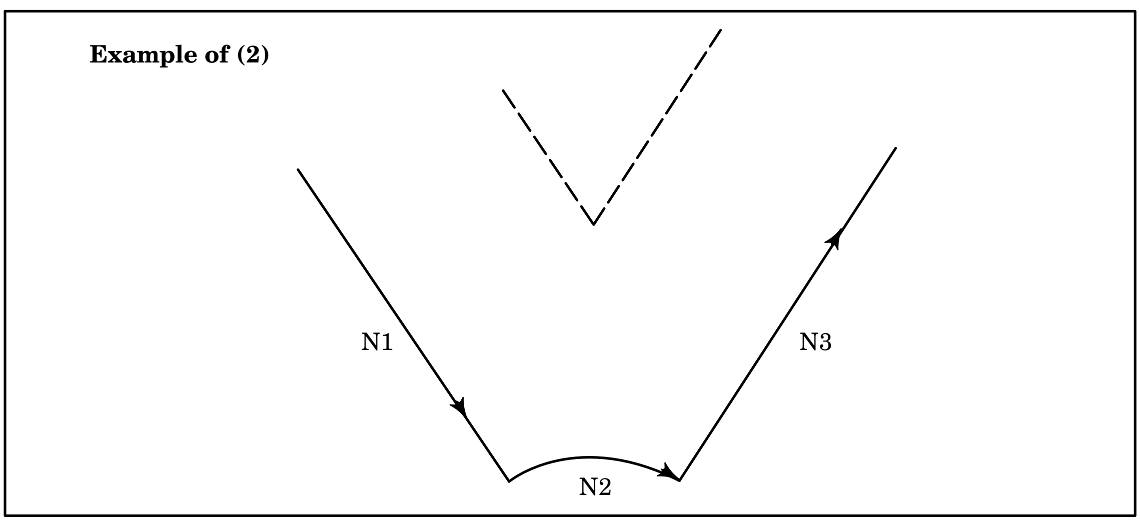
(1) The direction of the tool path is different from that of the programmed path. (From 90 degrees to 270 degrees between these paths).

(However, bit 1 (CNC) of parameter 6001 can be set so that the description of (1) is not checked.)

(2) In addition to the above condition, the angle between the start point and end point on the tool center path is quite different from that between the start point and end point on the programmed path in circular machining. (More than 180 degrees)

In the following example, the arc in block N6 is placed in the one quadrant. But after cutter compensation, the arc is placed in the four quadrants.





As shown in the figure above, when the traveled distance is set to 0 in block N2 because the offset is applied, an attempt is made to move the tool around the entire circumference if the arc is already specified in block N2. If an interference check is made in this state, an alarm occurs, and the machine stops.

b) Correction of interference in advance

(1) Removal of the vector causing the interference

When cutter compensation is performed for blocks A, B and C and vectors V1, V2, V3 and V4 between blocks A and B, and V5, V6, V7 and V8 between B and C are produced, the nearest vectors are checked first. If interference occurs, they are ignored.

But if the vectors to be ignored due to interference are the last vectors at the corner, they cannot be ignored.

Interference check between vectors V4 and V5

Interference ——— V4 and V5 are ignored

Check between V3 and V6

Interference ——— Ignored

Check between V2 and V7

Interference ——— Ignored

Check between V1 and V8

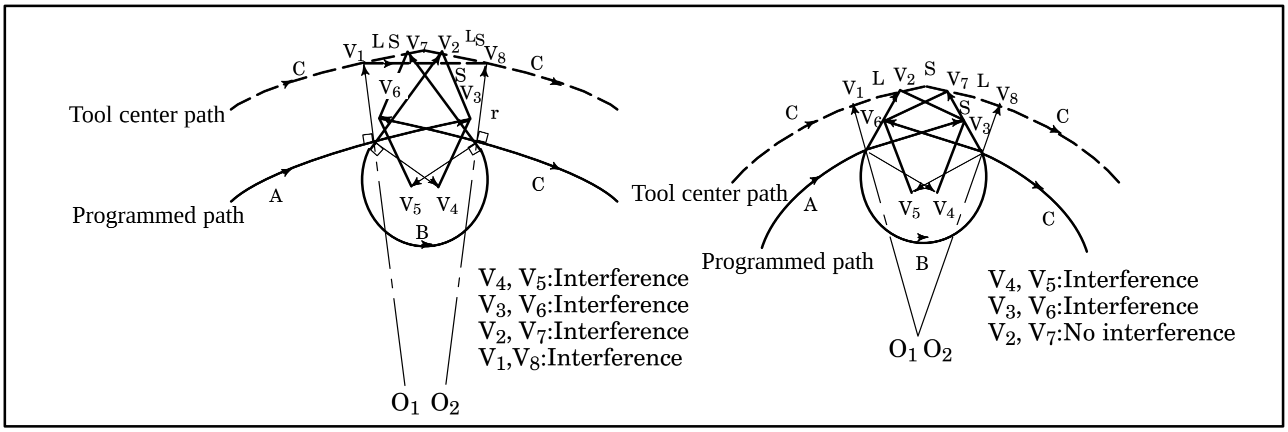
Interference ——— Cannot be ignored

If while checking, a vector with no interference is detected, subsequent vectors are not checked.

If block B is a circular movement, a linear movement is produced if the vectors are interfered.

Example 1 The tool moves lineraly from V1 to V8
If the tool is stopped by single block operation at block A, the tool center moves to V3.

Example 2 The tool moves linearly as follows:
Tool path: V1 $\rightarrow$ V2 $\rightarrow$ V7 $\rightarrow$ V8



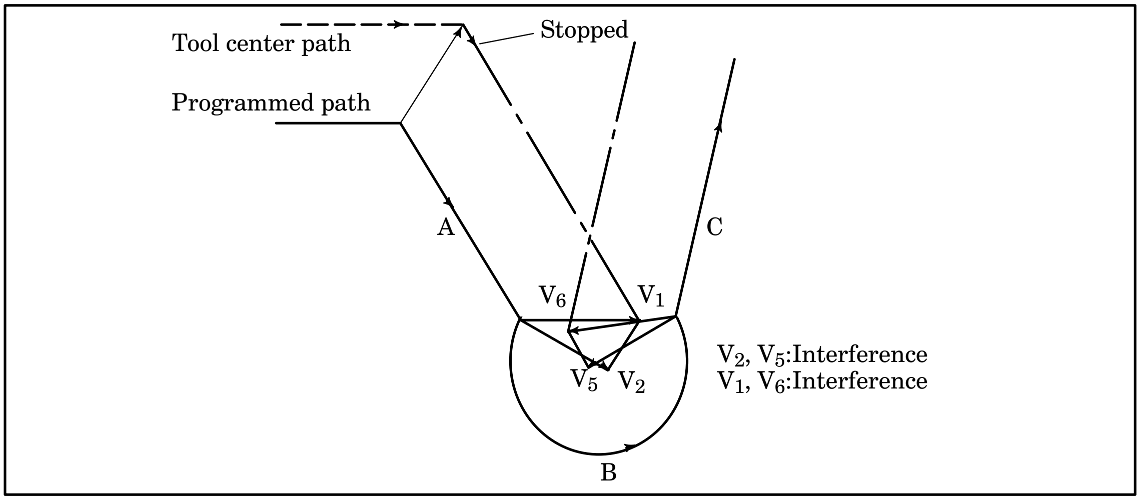
If the tool is stopped by single block operation at block A, the tool center moves to V3.

Then putting the operation into start moves the tool to V7 or V8.

(2) If the interference occurs after correction (1), the tool is stopped with an alarm.

If the interference occurs after correction (1) or if there are only one pair of vectors from the beginning of checking and the vectors interfere, the alarm (No.41) is displayed and the tool is stopped immediately after execution of the preceding block . (If the block is executed by the single block operation, the tool is stopped at the end of the block.)

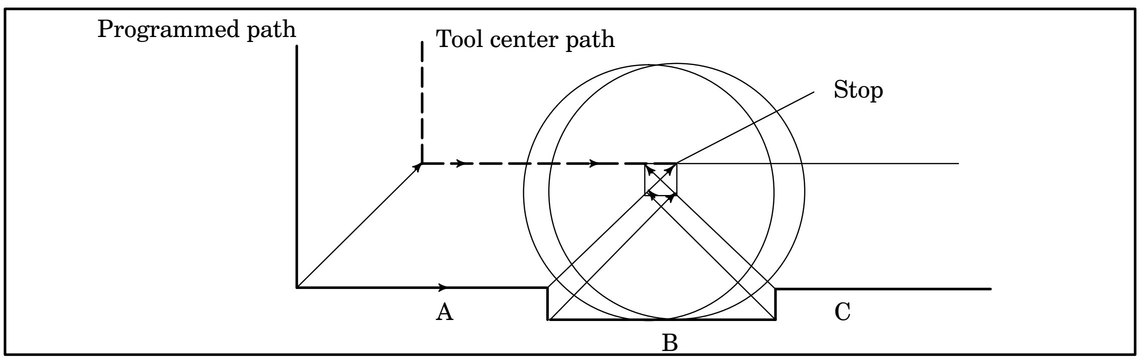
After ignoring vectors V2 and V5 because of interference, interference also occurs between vectors V1 and V6. The alarm is displayed and the tool is stopped.



c) When interference is supposed though there is no actual interference

Several examples will be given.

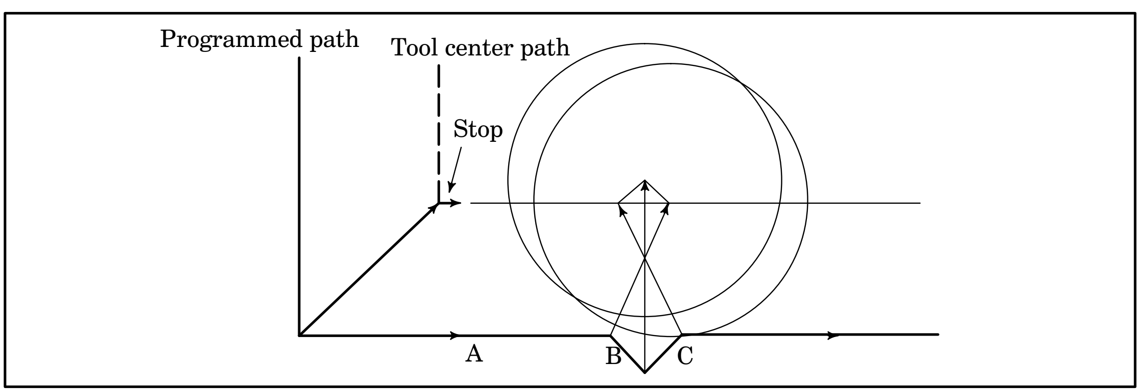
(1) Depression which is smaller than the offset value.



There is no actual interference, but since the direction programmed in block B is opposite to that of the path after cutter compensation the tool stops and an alarm is displayed.

(2) Groove which is smaller than the offset value.

Like (1) , the direction is reverse in block B.

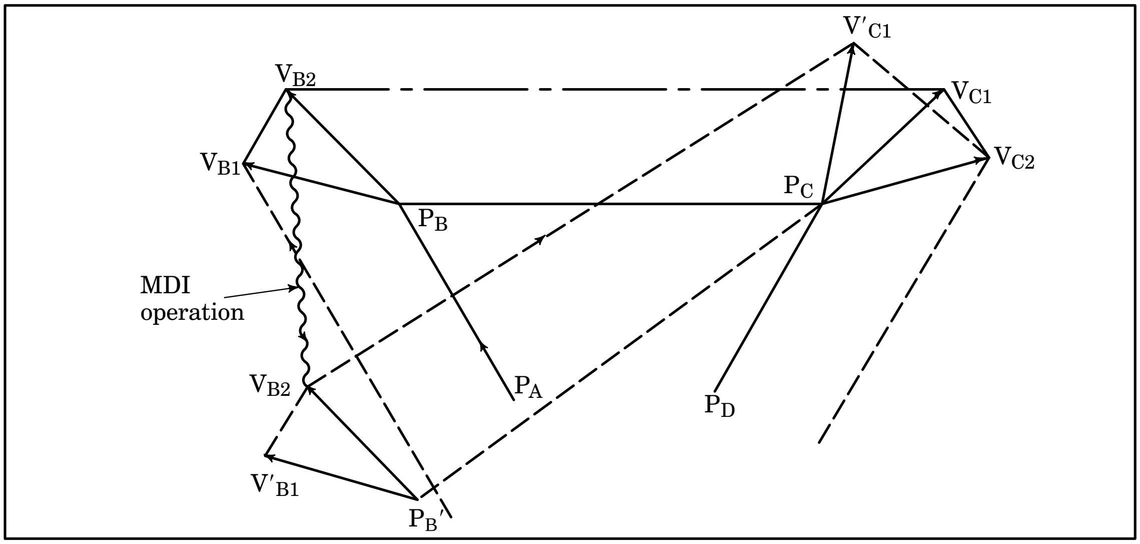


13) Input command from MDI

Cutter compensation is not performed for commands input from the MDI.

However, when automatic operation using the CNC tape composed of absolute commands is temporarily stopped by the single block function, MDI operation is performed, then automatic operation starts again, the tool path is as follows:

In this case, the vectors at the start point of the next block are translated and the other vectors are produced by the next two blocks. Therefore, from point Pc, cutter compensation is accurately performed.



When points PA, PB and PC are programmed in an absolute command, the tool is stopped by the single block function after executing the block from PA to PB and the tool is moved by MDI operation. Vectors VB1 and VB2 are translated to V'B1 and V'B2 and offset vectors are recalculated for the vectors VC1 and VC2 between block PB–PC and PC–PD.

However, since vector VB2 is not calculated again, compensation is accurately performed from point PC.

14) Manual operation

For manual operation during the cutter compensation, refer to Note 1 in Section III–4.7, " Manual Absolute ON and OFF."

15) General precautions on offset

a) Offset value command

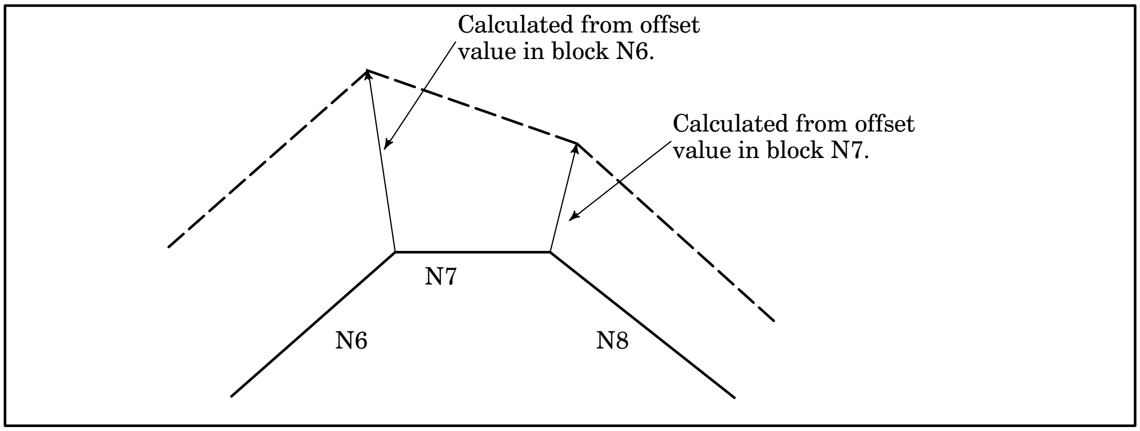
The offset value is commanded by a D code which specifies an offset value number.

Once a D code is commanded, it remains valid until another D code is commanded.

In addition to specifying the offset value for cutter compensation, the D code is also used to command offset value for tool offset.

b) Changing the offset value

In general, the offset value shall be changed in the cancel mode, when changing tools. If the offset value is changed in offset mode, the vector at the end point of the block is calculated for the new offset value.

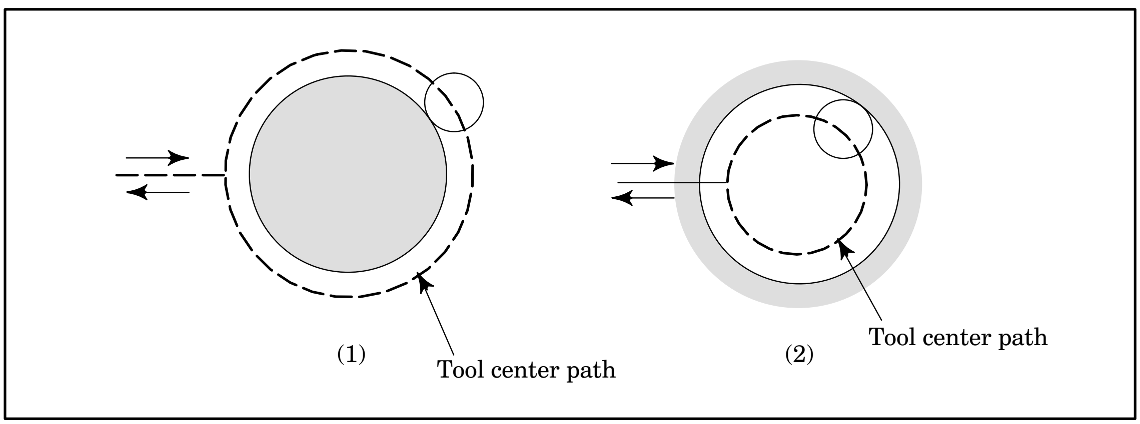


c) Positive/negative offset amount and tool center path

If the offset amount is negative (–), distribution is made for a figure in which G41's and G42's are all replaced with each other on the program. Consequently, if the tool center is passing around the outside of the workpiece, it will pass around the inside, and vice versa.

The figure below shows one example. Generally, the offset amount is programmed to be positive (+).

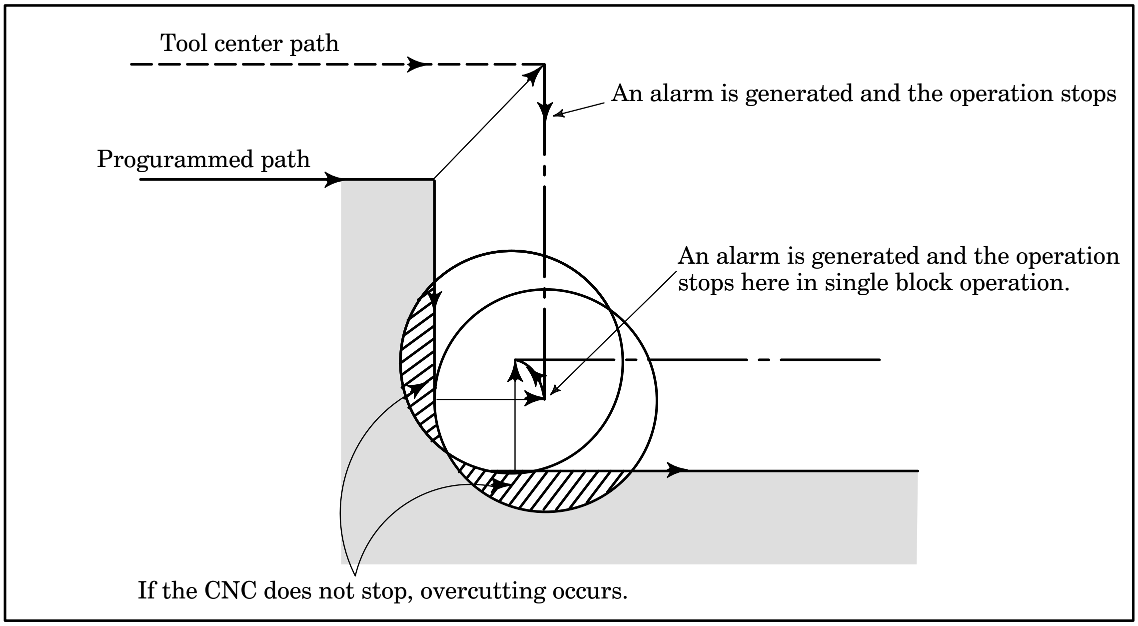
When a tool path is programmed as in (1), if the offset amount is made negative (–), the tool center moves as in (2), and vice versa. Consequently, the same tape permits cutting both male and female shapes, and any gap between them can be adjusted by the selection of the offset amount. (applicable if start–up and cancel is A type).



d) Overcutting by cutter compensation

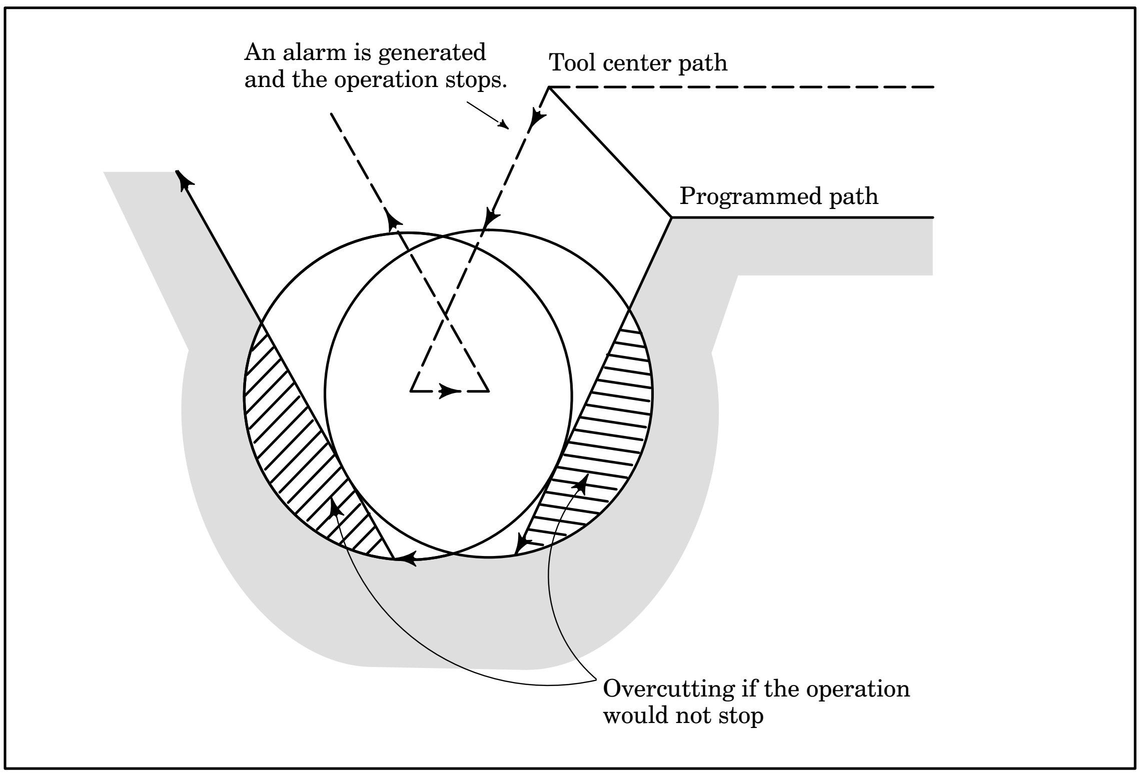
i) Machining an inside corner at a radius smaller than the cutter radius

When the radius of a corner is smaller than the cutter radius, because the inner offsetting of the cutter will result in overcuttings, an alarm is enerated and the CNC stops at the start of the block. In single block operation, the overcutting is gen erated because the tool is stopped after the block execution.



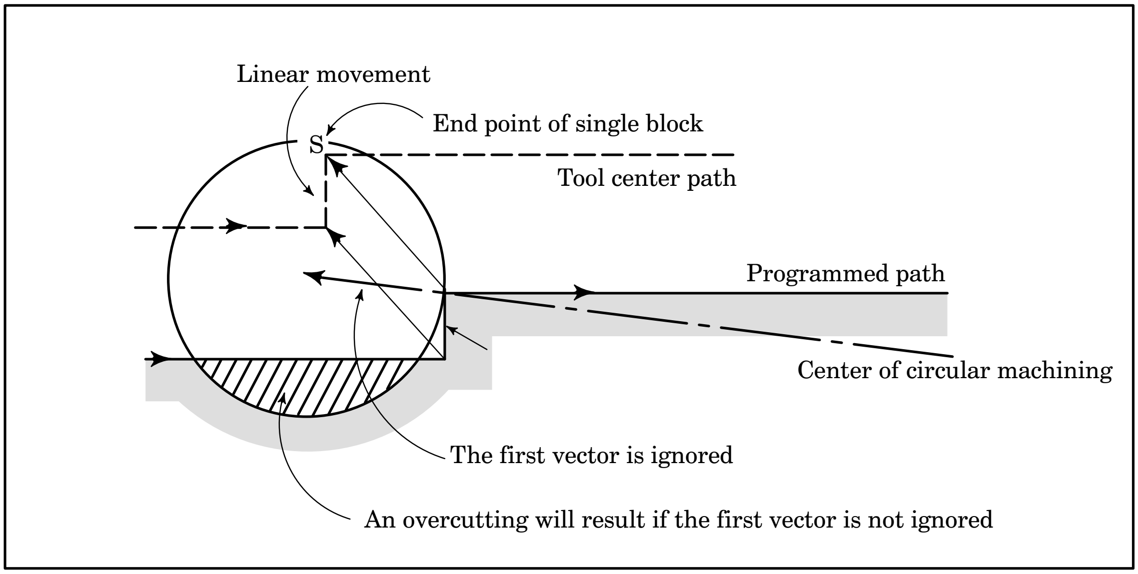
ii) Machining a groove smaller than the tool radius

Since the cutter compensation forces the path of the center of the tool to move in the reverse of the programmed direction, overcutting will result. In this case an alarm is generated and the CNC stops at the start of the block.



iii) When machining a step smaller than the tool radius

When machining of the step is commanded by circular machining in the case of a program containing a step smaller than the tool radius, the path of the center of tool with the ordinary offset becomes reverse to the programmed direction. In this case, the first vector is ignored, and the tool moves linearly to the second vector position. The single block operation is stopped at this point. If the machining is not in the single block mode, the cycle operation is continued. If the step is of linear, no alarm will be generated and cut correctly. However uncut part will remain.

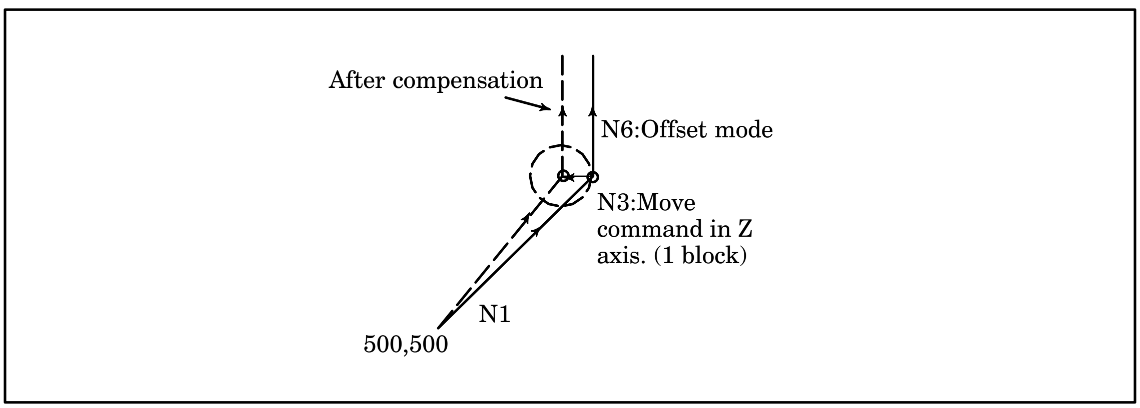


iv) The start up and the movement in Z axis on the cutter compensation C

It is usually used such a method that the tool is moved along the Z axis after the cutter compensation is effected at some distance from the workpiece at the start of the machining.

In the case above, if it is desired to divide the motion along the Z axis into rapid traverse and cutting feed, follow the procedure below.

```
N1  G91   G00  G41  X500.0  Y500.0  D1 ;
N3  G01   Z–300.0  F100 ;
N6        Y1000.0  F200 ;
```



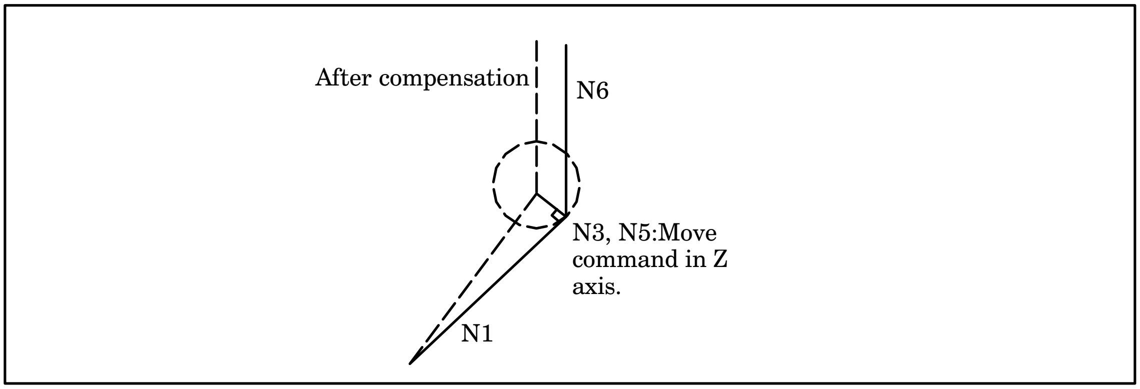
In the program example above, when executing block N1, blocks N3 and N6 are also entered into the buffer storage, and by the relationship among them the correct compensation is performed as in the figure above.

Then, if the block N3 (move command in Z axis) is divided as follows:

```
N1  G92 G00 G41 X500.0
    Y500.0 D1;
N3  Z–250.0;
N5  G01 Z–50.0 F100;
N6  Y1000.0 F200;
```

As there are two move command blocks not included in the selected plane and the block N6 cannot be entered into the buffer storage, the tool center path is calculated by the information of N1 in the figure above. That is, the offset vector is not calculated in start up and the overcutting may result.

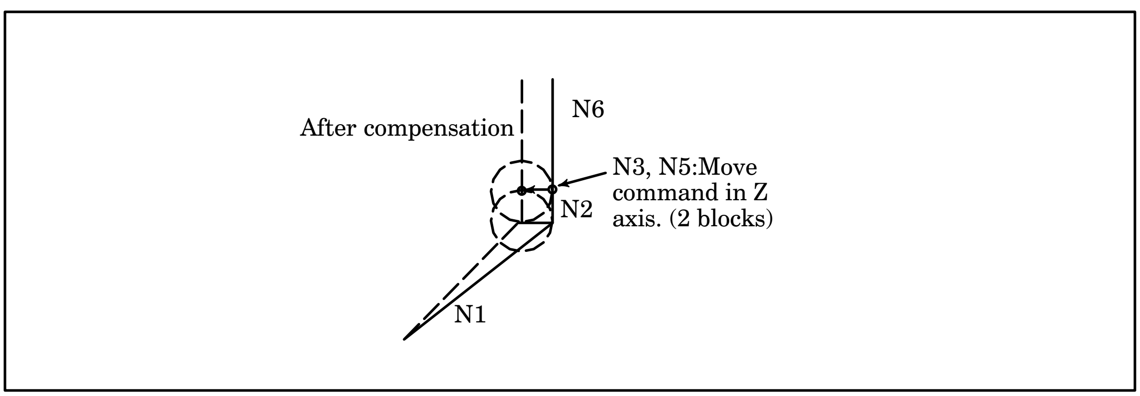
The above example should be modified as follows:



The move command in the same direction as that of the move command after the motion in Z axis should be programmed.

```
N1  G91 G00 G41 X500.0
    Y400.0 D1;
N2  Y100.0;
N3  Z–250.0;
N5  G01 Z–50.0 F100;
N6  Y1000.0 F200;
```

(The direction of the move command N2 is the same as that of N6.)



When executing the block N1, the blocks N2 and N3 are entered into the buffer storage and the correct compensation is performed by the relationship between N1 and N2.

15.4 Three–Dimensional Tool Offset (G40 and G41)

Cutter compensation C enables a two–dimensional offset to be applied on the selected plane. The three–dimensional tool offset function applies a three–dimensional offset by programming its direction.

15.4.1 Starting three–dimensional tool offset (G41)

When the following command is executed in the tool offset cancellation mode, the machine enters the three–dimensional tool offset mode.

G41XpDYpDZpDIDJDKDDD;

Xp: X–axis or an axis parallel to the X–axis

Yp: Y–axis or an axis parallel to the Y–axis

Zp: Z–axis or an axis parallel to the Z–axis

The three–dimensional space where a three–dimensional tool offset is applied depends on the axis address specified in the start block containing G41. When Xp, Yp, or Zp is not specified, X, Y, or Z is assumed.

Example When the X– and U–axes are parallel, the Y– and V–axes are parallel, or the Z– and W–axes are parallel

G41	X__I__J__K__;	X–Y–Z space
G41	U__V__Z__ I__J__K__;	U–V–Z space
G41	W__I__J__K__;	X–Y–W space

NOTE 1 To start the function, I, J, and K must be specified. If one of them is omitted, two–dimensional cutter compensation C is started.

NOTE 2 G41 is usually used. G42 also executes the function, but the offset direction is opposite to that for G41.

15.4.2 Canceling three–dimensional tool offset (G40)

When the following command is executed in the three–dimensional tool offset mode, the machine enters the tool offset cancellation mode.

To cancel the function together with movement, execute the following command:

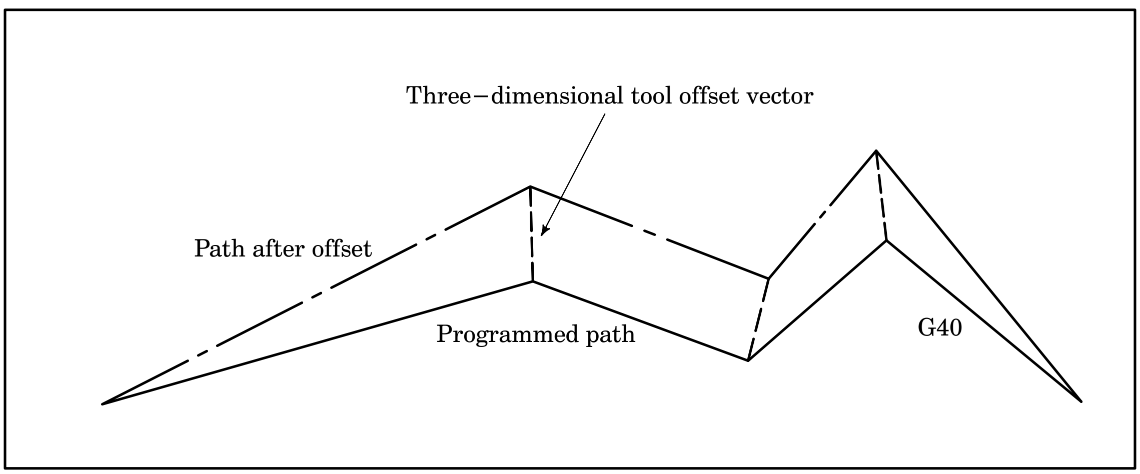
G40 Xp__Yp__Zp__; or Xp__Yp__Zp__ D00;

To cancel the specified vector, execute the following command:

G40; or D00;

15.4.3 Three–dimensional tool offset vector

The following three–dimensional tool offset vector is created at the end points of each block in the three–dimensional tool offset mode.



The vector is obtained by the following formulas:

$$Vx = \frac{i \cdot r}{P} \quad \text{(Vector component along the Xp–axis)}$$

$$Vy = \frac{j \cdot r}{P} \quad \text{(Vector component along the Yp–axis)}$$

$$Vz = \frac{k \cdot r}{P} \quad \text{(Vector component along the Zp–axis)}$$

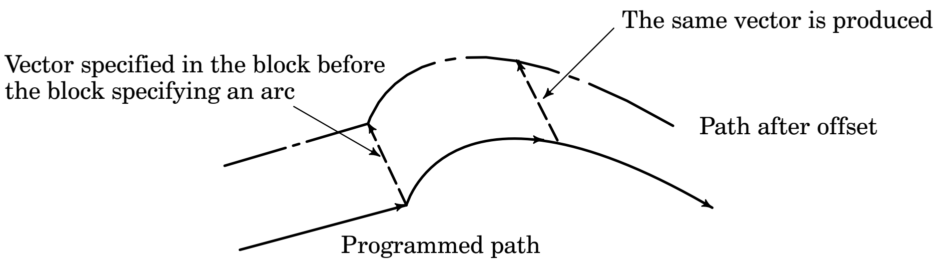
i, j, and k indicate the values of addresses I, J, and K specified in the block.

r indicates the offset corresponding to the specified offset number.

p is obtained by the following formula:

$$p = \sqrt{i^2 + j^2 + k^2}$$

NOTE 1 When a circular interpolation/helical cutting (G02 or G03) command is executed, the vector created in the preceding block is retained.



NOTE 2 Cancel the three–dimensional tool offset function before specifying reference position return (G28) or 2nd, 3rd, and 4th reference position return (G30).

NOTE 3 To program the direction of a three–dimensional offset vector as well as its magnitute, use the value of p in the following expressions as a constant (parameter 6011):

Expressions for finding the offset vector

$Vx = i{\cdot}r/p$
$Vy = j{\cdot}r/p$
$Vz = k{\cdot}r/p$

When the setting is 0, however, p is assumed as follows:

$$p = \sqrt{i^2 + j^2 + k^2}$$

The valid range of value p conforms to the following:

Input in mm	Input in inches
0 to ±99999.999mm	0 to ±9999.9999inches

NOTE 4 In the three–dimensional offset mode, the vector generated in the previous block is retained in the block where all of I, J, and K are omitted.

15.4.4 Relationship with other offset functions

(1) Tool length offset

A three–dimensional tool offset is applied to the programmed path first, then a tool length offset is applied.

(2) Tool offset

A tool offset command cannot be executed in the three–dimensional offset mode.

(3) Cutter compensation C

When cutter compensation is specified in the start block using all of addresses I, J, and K, a three–dimensional tool offset is applied. When it is specified using up to two addresses, cutter compensation C is applied. Three–dimensional tool offset cannot be specified in the cutter compensation C mode. Cutter compensation C cannot be specified in the three–dimensional offset mode.

15.5 Tool Compensation Amount

There are three tool compensation memories, A, B, and C. One of the memories is selected according to the nature of the compensation.

15.5.1 Tool compensation amount

The valid range of tool compensation amount can be selected using ORG and OFN of parameter 6002, OUF of parameter 6004, and ONM of parameter 6007.

ONM	OUF	OFN	ORG	Geometric compensation	Geometric compensation	Wear compensation	Wear compensation
				Input in mm	Input in mm	Input in incches	Input in inches
0	0	0	1	±999.99 mm	±99.999 inch	±99.99 mm	±9.999 inch
0	0	0	0	±999.999 mm	±99.9999 inch	±99.999 mm	±9.9999 inch
0	0	1	0	±999.9999 mm	±99.99999 inch	±99.9999 mm	±9.99999 inch
0	1	0	0	±99.99999 mm	±9.999999 inch	±9.99999 mm	±0.999999 inch
1	0	0	0	±9.999999 mm	±0.9999999inch	±0.999999 mm	±0.099999 inch

15.5.2 Extended tool compensation amount

Even if the optional extended tool compensation amount is equipped, the valid range of tool compensation amount can be selected using ORG and OFN of parameter 6002, OUF of parameter 6004 and ONM of parameter 6007. However, the number of compensation sets reduces by half. Its detail is as follows:

ONM	OUF	OFN	ORG	Geometric compensation		Wear compensation	
				Input in mm	Input in incches	Input in mm	Input in inches
0	0	0	1	±9999.99 mm	±999.999 inch	±999.99 mm	±99.999 inch
0	0	0	0	±9999.999 mm	±999.9999 inch	±999.999 mm	±99.9999 inch
0	0	1	0	±9999.9999 mm	±999.99999 inch	±999.9999 mm	±99.99999 inch
0	1	0	0	±9999.99999 mm	±999.999999 inch	±999.99999 mm	±99.999999 inch
1	0	0	0	±999.999999 mm	±99.9999999 inch	±99.999999 mm	±9.9999999 inch

15.5.3 Tool compensation memory A

The memory for geometric compensation and that for wear compensation are not separated in tool compensation memory A. Therefore, the sum of the geometric compensation amount and wear compensation amount is set in the memory.

In addition, the memory for cutter compensation (for D code) and that for tool length compensation (for H code) are not separated.

Example

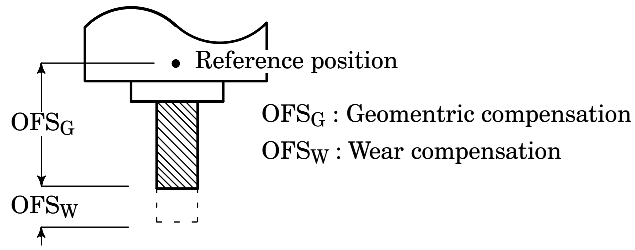
Offset No.	Compensation amount (geometric + wear)	D code/H code common
001	10.1	For D code
002	20.2	For D code
003	100.1	For H code

The setting range for the tool compensation amount is the same as that for the geometric compensation amount.

15.5.4 Tool compensation memory B

The memory for geometric compensation and that for wear compensation are separated in tool compensation memory B. The geometric compensation amount and wear compensation amount can thus be set separately.

However, the memory for cutter compensation (for D code) and that for tool length compensation (for H code) are not separated.



Example

Offset No.	Compensation amount (geometric + wear)	For wear compensation	D code/H code common
001	10.0	0.1	For D code
002	20.0	0.2	For D code
003	100.0	0.1	For H code

15.5.5 Tool compensation memory C

The memory for geometric compensation and that for wear compensation are separated in the tool compensation memory C. The geometric compensation amount and wear compensation amount can thus be set separately.

In addition, separate memories are provided for cutter compensation (for D code) and for tool length compensation (for H code).

Example

Offset No.	For D code		For H code	
	For geometric compensation	For wear compensation	For geometric compensation	For wear compenation
001	10.0	0.1	100.0	0.1
002	20.0	0.2	300.0	0.3

15.6 Number of Tool Compensation Settings

(1) 32 tool compensation settings
Applicable offset Nos. (D code/H code) are 0 to 32.
D00 to D32 or H00 to H32

(2) 99 tool compensation settings
Applicable offset Nos. (D code/H code) are 0 to 99.
D00 to D64 or H00 to H99

(3) 200 tool compensation settings
Applicable offset Nos. (D code/H code) are 0 to 200
D00 to D99 or H00 to H200

(4) 499 tool compensation settings
Applicable offset Nos. (D code/H code) are 0 to 499.
D00 to D200 or H00 to H499

(5) 999 tool compensation settings
Applicable offset Nos. (D code/H code) are 0 to 999.
D00 to D400 or H00 to H999

15.7 Changing the Tool Compensation Amount (Programmable Data Input) (G10)

The tool compensation amount can be set or changed with the G10 command.

When G10 is used in absolute input (G90), the compensation amount specified in the command becomes the new tool compensation amount. When G10 is used in incremental input (G91), the compensation amount specified in the command is added to the amount currently set.

Format

(1) For tool compensation memory A

G10 L11 P__R__;

where P__: Offset No.

R__: Tool compensation amount

(2) For tool compensation memory B

Setting/changing the geometric compensation amount

G10 L10 P__R__;

Setting/changing the wear compensation amount

G10 L11 P__R__;

(3) For tool compensation memory C

Setting/changing the geometric compensation amount for H code

G10 L10 P__R__;

Setting/changing the geometric compensation amount for D code

G10 L12 P__R__;

Setting/changing the wear compensation amount for H code

G10 L11 P__R__;

Setting/changing the wear compensation amount for D code

G10 L13 P__R__;

NOTE The L1 command may be used instead of L11 for format compatibility of the conventional CNC.

15.8 Scaling (G50,G51)

Scaling is commanded to figures specified by machining programs.

G51X____Y____Z____P____;

X,Y,Z : X,Y,Z coordinate values of scaling center

P : Scale factor (Least input increment: 0.001 or 0.00001 .... parameter selection

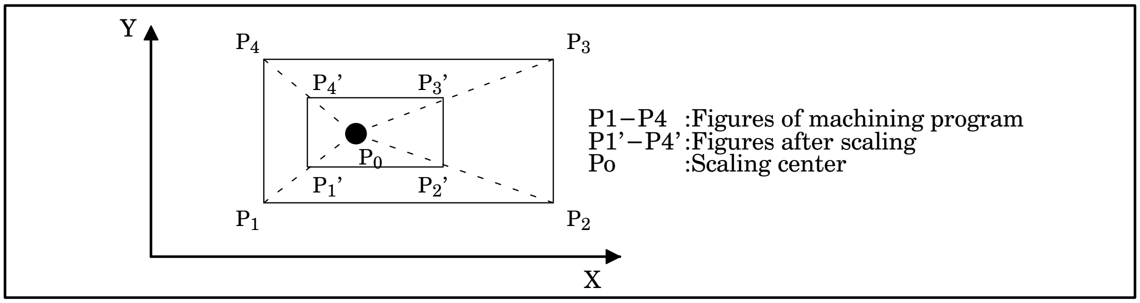
By this command, the subsequent move command is scaled by the scale factor specified by P, starting with the point specified by X,Y,Z as the scaling center.

This scaling mode is cancelled by G50.

G50 : Scaling mode cancel command G51 : Scaling mode command

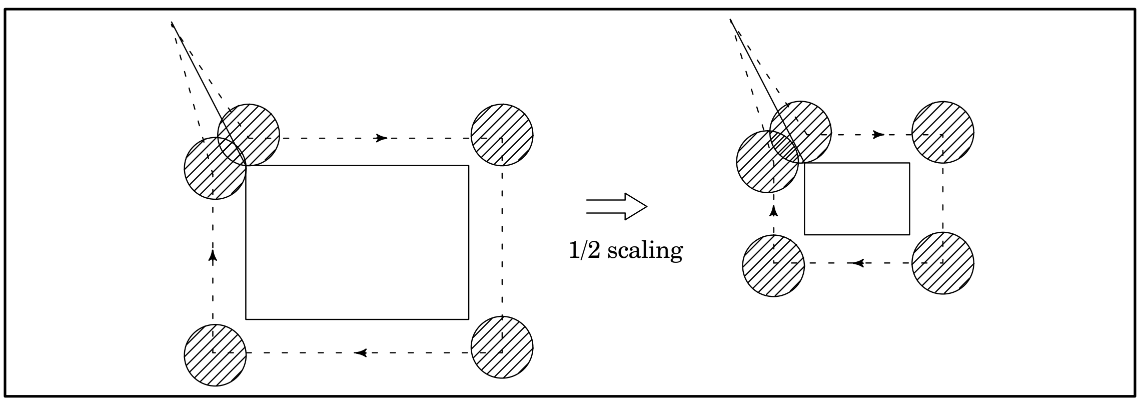
The scale factor can be specified within the following range:

0.00001 – 9.99999 or 0.001 – 999.999 times



If P is not specified, the scaling factor set by CRT/MDI is applicable. If X,Y,Z are omitted, the G51 command point serves as the scaling center.This scaling is not applicable to the offset quantities such as cutter compensation amount, tool

length compensation amount, tool offset amount, and others.



The scaling factor is common to all axes. Different scaling factors can, however, be specified for each axis using bit 4 (XSC) of parameter 7611. The scaling factors are specified for each axis with parameter 6421. Scaling is not applied to an axis whose scaling factor is specified as 0. The scaling factors for each axis cannot, however, be specified by the CNC program.

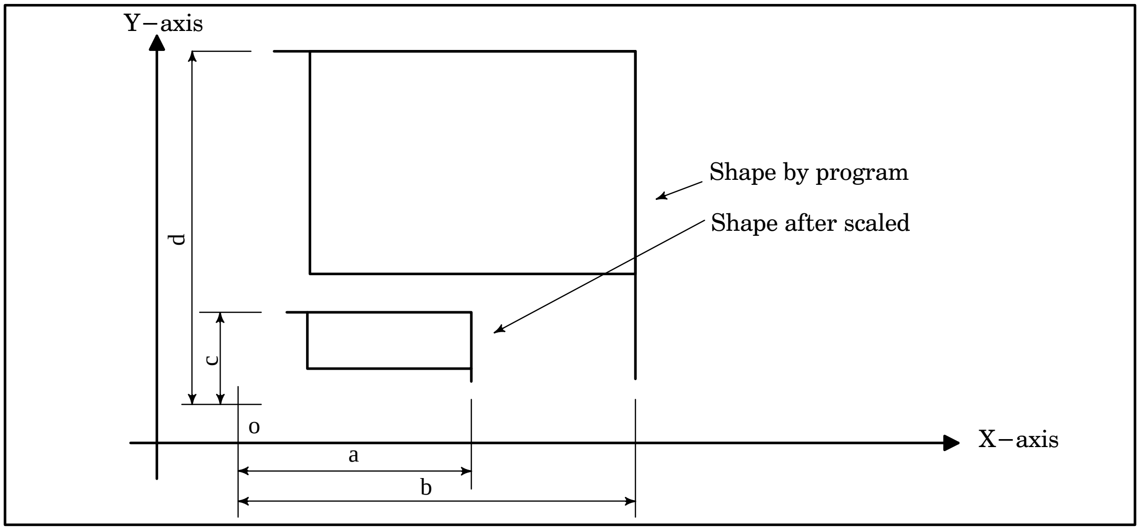


Fig. 15.8 Scaling of each axis

b/a : Magnification of X axis
d/c : Magnification of Y axis
o : Center of scaling

NOTE 1 Specify G51 in an independent block. Cancel G51 by G50 without fail after scaling.

NOTE 2 The position display represents the coordinate value after scaling.

NOTE 3 If a setting value is employed as a scale factor without specifying P, the setting value at G51 command time is employed as the scale factor,and a change of this value by another command, if any, is not effective.

NOTE 4 Parameter 12 specifies for each axis whether the scaling function is effective or ineffective.

NOTE 5 Scaling is not effective for manual operation, but is effective for the tape, memory, or MDI operations.

NOTE 6 Scaling is not applicable to the following movement in case of the canned cycle Z–axis movement.

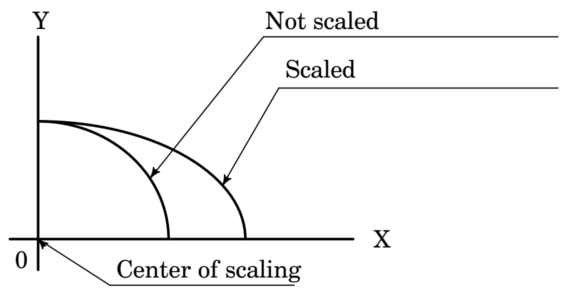
- Cutting depth Q and return relief of deep hole cycle (G83, G73).
- Fine boring (G76)
- Shift of X and Y axes in back boring (G87).

NOTE 7 Specify G27, G28, G29, G30, and G92 in G50 mode without fail.

NOTE 8 If scaling results are rounded by counting fractions of 5 and over as a unit and disregarding the rest, the move amount may become zero. In this case, the block is regarded as a no movement block, and therefore, it may affect the cutter movement by cutter compensation C. See 15.3 (8).

NOTE 9 A circle does not become an ellipse even when different scaling factors are applied to each axis during circular interpolation (G02 or G03).

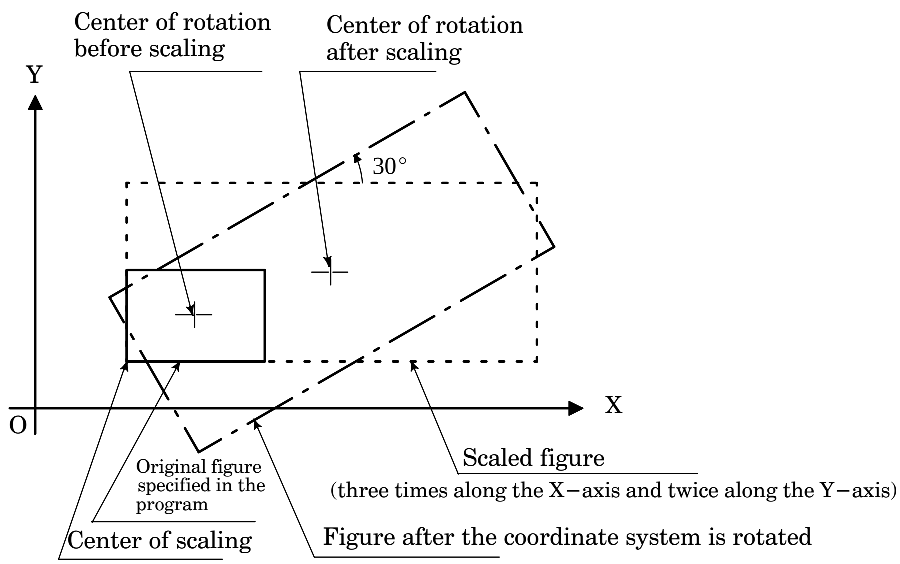
Example G90 G00 X0 Y100.0 ;
G51 X0 Y0 ;
(Specifies the factors of twice along the X–axis and once along the Y–axis.)
G02 X100. Y0 I0 J–100.0 F500 ;
The above command is equivalent to the following command:
G90 G00 X0 Y100.0 ;
G02 X200. Y0 I0 J–100.0 F500 ;
Circular interpolation changes to spiral interpolation because the end point does not lie on the radius.



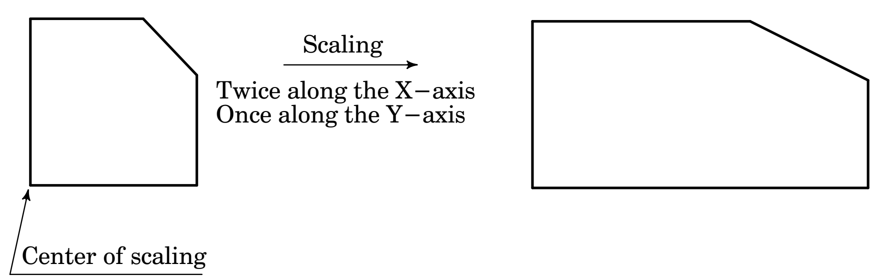
NOTE 10 When both scaling and coordinate system rotation are specified, the coordinate system is rotated after scaling is applied. In this case, scaling is effective for the center of rotation.

Example

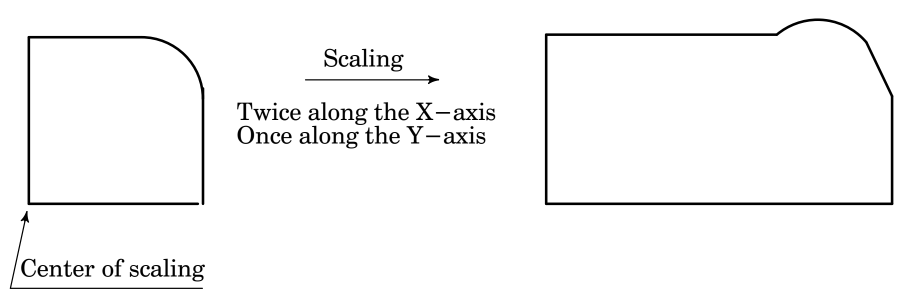
```
O0001 ;                             O100 ;
G90 G00 X20.0 Y10.0 ;               G01 X50.0 F500 ;
M98 P1000 ;                         G01 Y30.0 ;
G51 X20.0 Y10.0 ;                   G01 X20.0 ;
M98 P1000 ;                         G01 Y10.0 ;
G17 G68 X35.0 Y20.0 R3000 ;         M99 ;
M98 P1000 ;
G69 ;
G50 ;
M30 ;
```



NOTE 11 When chamfering at any angle is specified during scaling, the figure is scaled.



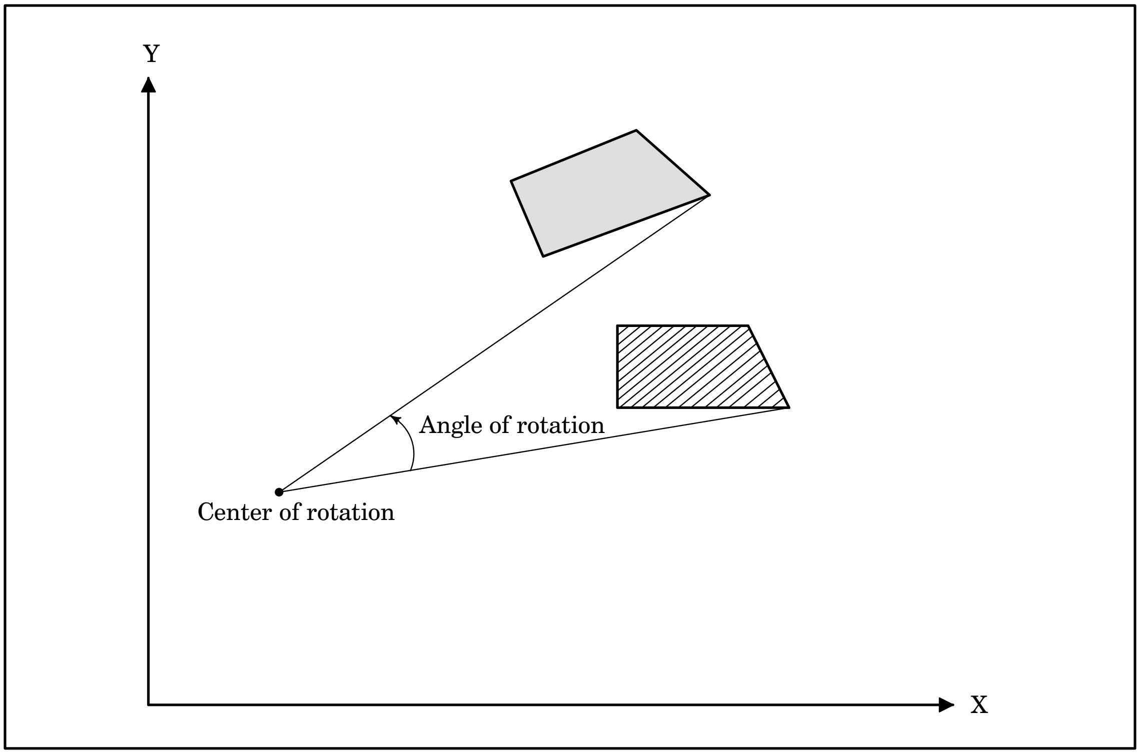
NOTE 12 When different scaling factors are specified for each axis, the end point of an arc and the radius are scaled. A rounded corner becomes a spiral circle, not an arc.



NOTE 13 The center of scaling is not assumed to be specified for the incremental command until the absolute position command is issued after the G51 block. The point specified with G51 is assumed to be the center of scaling.

15.9 Coordinate System Rotation (G68, G69)

A programmed shape can be rotated. By using this function it becomes possible, for example, to modify a program using a rotation command when a workpiece has been placed with some angle rotated from the programmed position on the machine.Further, when there is a pattern comprising some identical shapes in the positions rotated from a shape, the time required for programming and the length of the program can be reduced by preparing a subprogram of the shape and calling it after rotation.



15.9.1 Command format

(1) Coordinate system rotation command

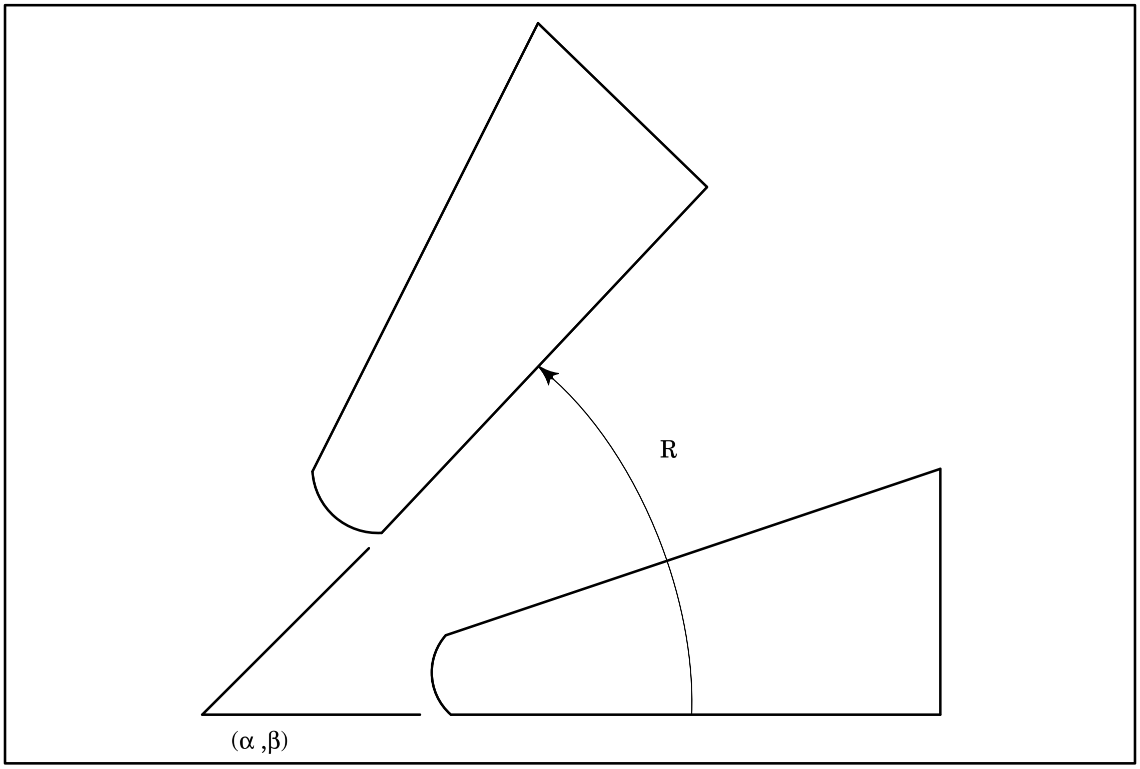
$$\left\{\begin{matrix}\textbf{G17}\\\textbf{G18}\\\textbf{G19}\end{matrix}\right\}\ \textbf{G68}\ \alpha\_\_\beta\_\_\textbf{R}\_\_;$$

(2) Coordinate system rotation cancel command G69

α,β :Designate two axes among X, Y and Z (coordinate value of rotational center) that correspond to G17, G18 and G19. (G90/G91 mode is effective.)

R : Angle of rotation ("+" is the CCW direction. Command with an absolute value. The relevant parameter enables the use of an incremental value.

After this command is specified, the subsequent commands are rotated by the angle specified with R around the point designated with α and β. Command the angle of rotation within the range of –360,000≦R≦ 360,000 (or –36,000,000 ≦ R ≦ 36,000,000) in 0.001 deg. or 0.00001 deg. (setting by parameter) increment.



The rotation plane depends on the plane (G17, G18, G19) selected when G68 is designated. G17, G18 or G19 is not required to be designated in the same block as G68.

When α and β are omitted, the position where G68 is commanded is set as the center of rotation.

As for the incremental position commands designated between the G68 block and a block with an absolute command; it is regarded that the center of rotation is not designated (that is, it is regarded that the point where G68 was designated is the center of rotation).

When R is omitted, the value set to parameter (setting input enable) is regarded as the angle. The coordinate system rotation is cancelled by G69;

G69 may be designated in the same block as the other commands. Tool offset, such as cutter compensation, tool length compensation, or tool offset, is performed after the coordinate system is rotated for the command program.

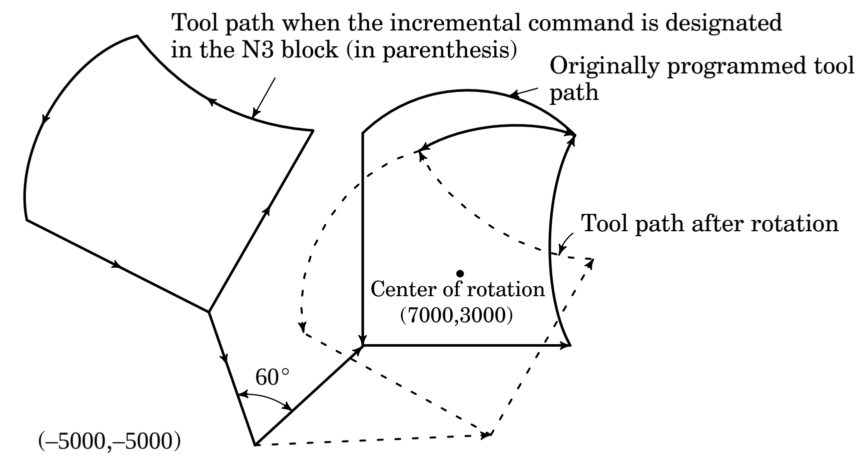
WARNING	If a movement command is specified with G69, the following incremental command is not executed as specified.

NOTE 1 When a value with a decimal point is designated for R, the position of the decimal point becomes the unit of angle

NOTE 2 Specify G68 in the G00 or G01 mode.

NOTE 3 Specify G27, G28, G29, or G30, and G92 in the G69 mode.

Example 1




```
N1  G92   X–5000.0     Y–5000.0  G69   G17;
N2  G68   X7000.0      Y3000.0R60000.0;
N3  G90   G01   X0     Y0  F200;
    (G91  X5000.0      Y5000.0)
N4  G91   X10000.0;
N5  G02   Y10000.0     R10000.0;
N6  G03   X–10000.0   I–5000.0   J–5000.0;
N7  G01   Y–10000.0;
N8  G69   G90   X–5000.0  Y–5000.0  M02;
```

NOTE Do not change the selected plane in the G68 mode. Be sure to specify G17, G18, or G19 in the G69 mode.

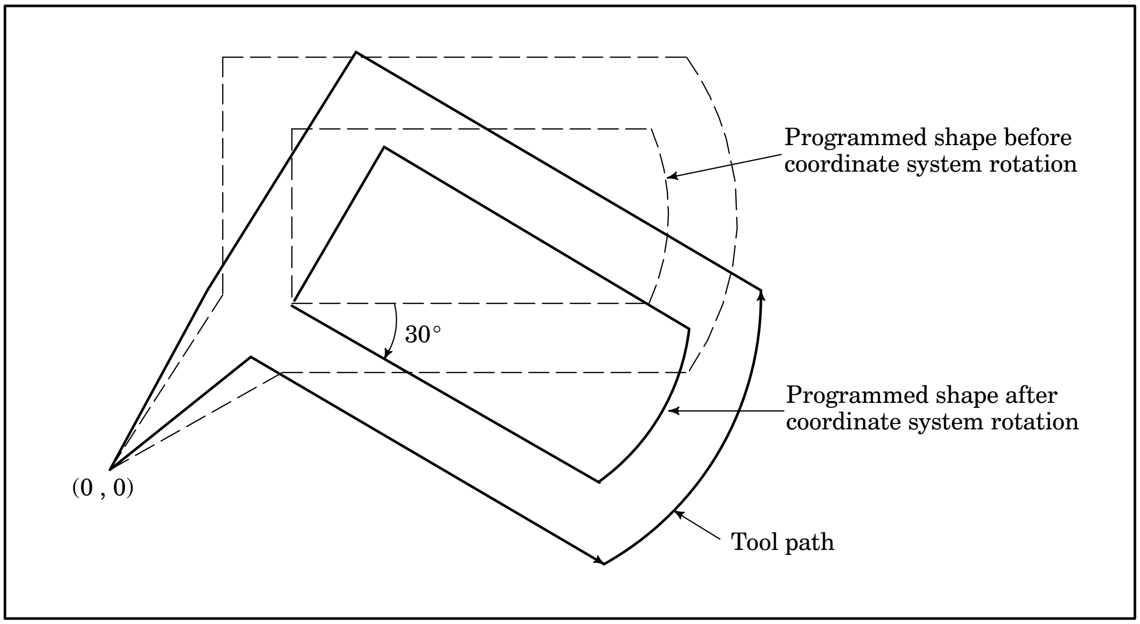
15.9.2 Relationship to other functions

1) Cutter Compensation C

It is possible to specify G68 and G69 within cutter compensation C. The rotation plane must coincide with the plane of cutter compensation C.

Example 1

N1 G92 X0 Y0 G69 G01;
N2 G42 G90 X1000.0 Y1000.0 F1000 D01;
N3 G68 R–30000.0;
N4 G91 X2000.0;
N5 G03 Y1000.0 R1000.0 J500.0;
N6 G01 X–2000.0;
N7 Y–1000.0;
N8 G69 G40 G90 X0 Y0 M30;



2) Coordinate system rotation in the scaling mode.

If a coordinate system rotation command is executed in the scaling mode (G51 mode), the coordinate value (α, β) of the rotation center will also be scaled, but not the rotation angle (R). When a move command is issued, the scaling is applied first and then the coordinates are rotated. Therefore, the commands must be specified in the order shown below for programming.

Example 1

G51 . . . . ; (Scaling mode)

G68 . . . . ; (Coordinate system rotation ON)

⋮

G69 . . . . ; (Coordinate system rotation OFF)

G50 . . . . ; (Scaling mode cancel)

A coordinate system rotation command (G68) should not be issued in cutter compensation C mode (G41, G42) on scaling mode (G51). The coordinate system rotation command should always be specified prior to setting the cutter compensation C mode.

Example 2

(Cutter compensation cancel)

G51 . . . . ; (Scaling mode)

G68 . . . . ; (Coordinate system rotation ON)

⋮

G41 . . . . ; (Cutter compensation C)

⋮

Example 3

G92X0Y0;

G51 X3000.0 Y1500.0 P500 ;

G68 X2000.0 Y1000.0 R45000.0 ;

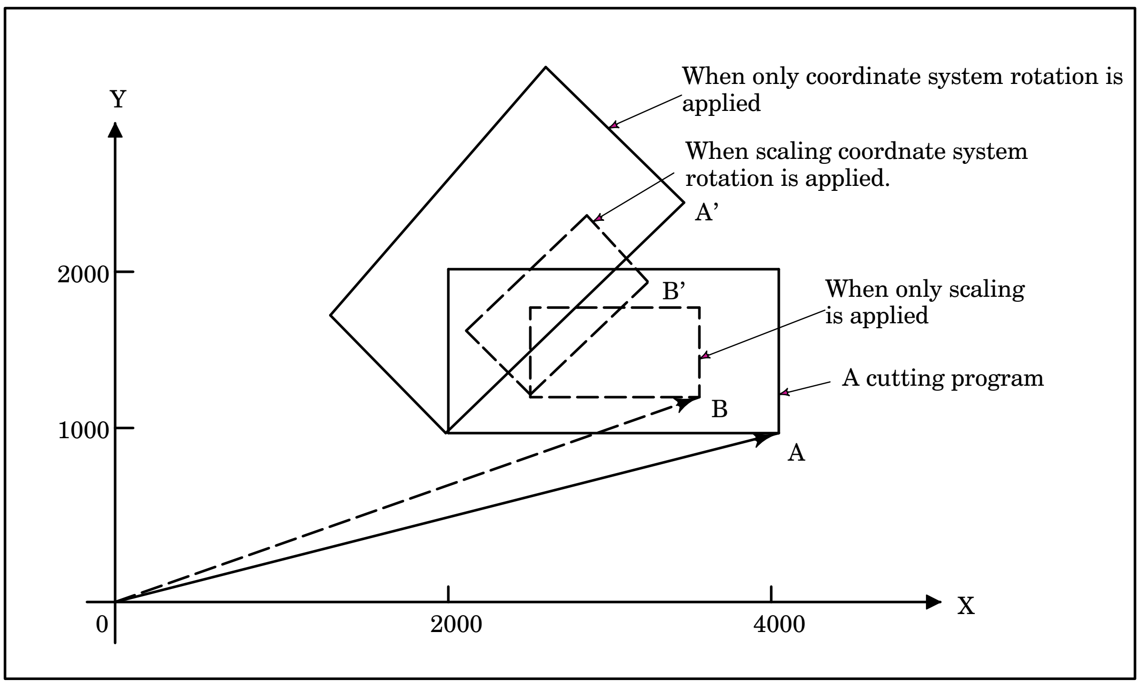
G01 X4000.0 Y1000.0 ;

Y1000.0 ;

X–2000.0 ;

Y–1000.0 ;

X2000.0 ;



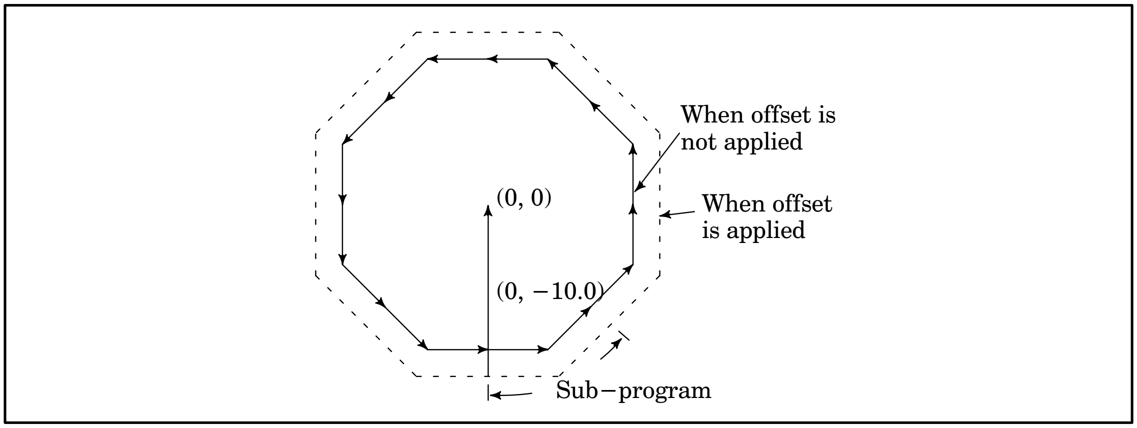
3) Repetitive Commands

It is possible to store one program as a subprogram and recall subprogram by changing the angle.

Example 1

If the rotation angle is as specified by G90 or G91 when parameter 6400#0 is set to 1

```
G92 X0 Y0 G69 G17 ;
G01 F200 H01 ;
M98 P2100 ;
M98 P072200 ;
G00 G90 X0 Y0 M30 ;
O 2200 G68 X0 Y0 G91 R45.0 ;
G90 M98 P2100 ;

M99 ;
O 2100 G90 G01 G42 X0 Y–10.0 ;
X4.142
X7.071 Y–7.071 ;
G40 ;
M99 ;
```



15.10 Tool Offsets Based on Tool Numbers

Cutter compensation data, tool length compensation data, and the tool pot number can be set for a specific tool number (T code). Up to 300 sets of data can be set. If a certain tool number is specified, the pot number corresponding to that tool number is output as a T code to the PMC. If cutter compensation or tool length compensation is specified, compensation is performed using the cutter compensation or tool length compensation data set for the tool number.

(1) Setting tool data

Tool data can be set by manually entering the data on the MDI panel or by specifying the data in a program.

Format for specifying tool data in a program

Registering new tool data after clearing the existing tool data

```
G10 L70 ;
T – – P – – R – – K – – ;
T – – P – – R – – K – –
– – – – – – – – – – –
T – – P – – R – – K – –
G11 ;
```

G10L70	:	Clear existing tool data, then start registration.
T	:	Tool number (0 to 99999999)
P	:	Pot number (0 to 9999)
R	:	Cutter compensation data
K	:	Tool length compensation
G11	:	End registration.

Adding tool data to existing tool data

```
G10 L71 ;
T – – P – – R – – K – –
T – – P – – R – – K – –
– – – – – – – – – – –
T – – P – – R – – K – –
G11 ;
```

G10L71 : Start registering the additional data.

> **NOTE 1** For the allowable range of cutter compensation and tool length compensation data, see Section 15.5.
>
> **NOTE 2** If the same tool number is registered twice, data for the tool number registered last becomes valid.

(2) Deleting registered tool data

Registered tool data can be deleted either manually on the MDI panel or by specifying the appropriate instructions in the program. For how to delete data manually, see III.10.1.

When deletion of tool data is programmed, one of the following is specified:

– Deletion with a tool number specified
– Deletion with a pot number specified
– Deletion with a tool number and pot number specified

If a pot number is specified, all the tool data for the specified pot number is deleted.

The format of the instructions for deleting tool data is as follows:

```
G10L72 ;  Start deleting tool data.
T––    ;  Delete tool data for the specified tool number.
   :
P––    ;  Delete all tool data for the specified pot number.
   :
T–– P–– ; Delete tool data for the specified tool number and pot number.
   :
T–– P–– ; Delete tool data for the specified tool number and pot number.
G11    ;  End deleting tool data.
```

where,

G10L72 : Start of tool data deletion
T–– : Tool number (0 to 99999999)
P–– : Pot number (0 to 9999)
G11 : End of tool data deletion

If an attempt is made to delete the data of a tool currently selected, an alarm occurs. When a tool number and pot number are specified, the tool data is not deleted unless the tool data associated with both numbers exists.

Sample program

```
NO1 G01L72  ;
N0 T01  ;      Deletes tool data for T01.
N0 P20  ;      Deletes all tool data for P20.
N04 T04P20;  An alarm occurs.
N05 T04P30;  Deletes tool data for T04 and P30.
N06 G11   ;
```

(3) Outputting a tool pot number

When a tool number (T code) is specified, the corresponding tool pot number is extracted from the tool data file and is output to the machine as a T code.
If a tool having the same pot number as another tool is set, an asterisk (*) is indicated for the pot number on the tool data screen. If a tool marked * is specified, its pot number is output as a T code, and a signal is output indicating that the same pot is specified for another tool.

Example

[Tool data setting]

```
N01   G10 L70 ;
N02   T01 P10 R12.0 K11.0 ;
N03   T02 P20 R22.0 K21.0 ;
N04   T03 P20 R32.0 K31.0 ;
N05   T04 P30 R42.0 K32.0 ;
N06   G11 ;
N07   M02 ;
```

[Machining program]

N01 T01 ; Tool pot number 10 which corresponds to T01 is output as a T code.

N02 T02 ; Tool pot number 20 which corresponds to T02 is output as a T code. At the same time, a signal is output to indicate that the pot number is duplicated.

(4) Cutter compensation and tool length compensation

If an M code to change tools (NOTE3) is specified, the cutter compensation data and tool length compensation data for the tool number (T code) specified previously become valid.

Sample program

[Tool data setting]

```
N01   G10 L70 ;
N02   T01 P10 R12.0 K11.0 ;
N03   T02 P20 R22.0 K21.0 ;
N04   G11 ;
N05   M02 ;
```

[Compensation using tool data when M06 is specified]

```
N01   T01 ;       (Tool pot number 10 corresponding to T01 is output as a T code.)
N20   M06 ;       (The cutter compensation data and tool length compensation data for T01 become
                  valid.)
N30   G43—— ;     (Tool length compensation is performed using the compensation data for T01, 11.0.)
N40   T02 ;       (Tool pot number 20 corresponding to T02 is output as a T code.)
N50   G41—— ;     (Cutter compensation is performed using the compensation data for T01, 12.0.)
N60   G40G49 ;
N70   M06 ;       (The cutter compensation data and tool length compensation data for T02 become
                  valid.)
N80   G43—— ;     (Tool length compensation is performed using the compensation data for T02, 21.0.)
```

NOTE 1 Use parameter No. 2429 to specify an M code for changing tools.

NOTE 2 If an M code for changing tools and a tool number (T code) are specified in the same block, how the program is executed depends on the settings of CT1 and CT2 of tool life management parameter No. 7401 as listed in the table below:

CT2	CT1	Changing tools	Description
0	0	Method A	The compensation data for the tool number (T code) specified previously becomes valid, and the tool pot number corresponding to the tool number is output. The compensation data for this tool number does not become valid only when the M code for changing tools is specified.
1	0	Method B	The compensation data for the tool number (T code) specified previously becomes valid, and the tool pot number corresponding to the tool number is output. It becomes valid for the next M code for changing tools.
0	1	Method C	Same as method B
1	1	Method D	The tool pot number corresponding to the tool number is output, and the corresponding compensation data becomes valid immediately.

Example When the M code for changing tools is M06

(1) Tool changing method A (Specifying the tool return number)

N01 T10 ; (The tool pot number corresponding to T10 is output.)

N02 M06T11 ; (The compensation data for T10 becomes valid, and the tool pot number corresponding to T11 is output.)

N03 M06 ; (The compensation data for T10 does not become valid.)

When the tool number (T11) is specified in the same block as M06, the M code for changing tools, M06, does not make the compensation data for the tool number, T11, valid.

(2) Tool changing methods B and C

N01 T10 ; (The tool pot number corresponding to T10 is output.)

N02 M06T11 ; (The compensation data for T10 becomes valid, and the tool pot number corresponding to T11 is output.)

N03 M06T10 ; (The compensation data for T11 becomes valid, and the tool pot number corresponding to T10 is output.)

(3) Tool changing method D

N01 T10M06 ; (The tool pot number corresponding to T10 is output, and the compensation data for T10 becomes valid.)

N02 T11M06 ; (The tool pot number corresponding to T11 is output, and the compensation data for T11 becomes valid.)

NOTE 3 Bit TLN in parameter No. 2203 determines whether the T codes displayed on the program check screen and T codes that can be read through system variables are handled as tool pot numbers or tool numbers. A T code appears on the program check screen when execution of the block immediately following the tool number command has been completed. When a system variable is used, a T code can be read from the block immediately following the tool number command.

NOTE 4 The tool position can be compensated by specifying its offset number using H and D codes. However, to compensate the tool length or cutter by specifying their offset numbers using H and D codes, it is necessary to set the TOFS bit in parameter No. 11. In this case, neither tool–number–based tool pot number output nor compensation data for a tool number can be used. Before parameter No. 11 can be set, the system must be reset.

(4) Relation to other functions

(i) Tool life management

(a) Data specified using D codes nor H codes cannot be registered as tool life data.

(b) Neither the H99 nor D99 command can be used.

(c) The length of the T code for tool life management data has been changed to 8 digits. Also, a tool number (tool group) can now be specified using 8 digits. The memory length remains unchanged.

(d) A tool number–based tool offset is associated with tool life management as follows: A tool number (T code) selected by tool life management is used as the tool number to specify a tool number–based tool offset.

An example is shown below.

Tool life management data

```
GROUP01  LIFE300   COUNT20
@0001        0002
```

Tool number–based tool offset data

NO	TOOL	POT #	LENGTH	RADIUS
001	1	0100	123.456	789.012

Program (When the number of a tool to which tool–life management is not applied is 100 and the M code for changing tools is M06)

(1) N1 T101 ;

(2) N2 M06 ;

(a) N1 T101 selects group 01 of tool life management.

Tool life management then selects tool number @0001.

For a tool number–based tool offset, the pot number for T1 (100) is output to the machine as a T code.

(b) N2 M060 starts life management (i.e., monitoring the usable life of tools) for tool @0001. The compensation data for T1 becomes valid as a tool number–based tool offset.

(e) The number of a tool to which tool–life management is not applied can now be specified using up to 8 digits.

Data number	Data
7441	Number of a tool to which tool–life management is not applied (four high–order digits)

Parameter input

Data format: Word type

Data unit : Integer

Data range: 0 to 9999 (0 to 99999999, combined with No. 7440)

Specify the four high–order digits of the number of a tool to which tool–life management is not applied.

For example, to set the number to 5000000, set the following:

Parameter No.7440	0	(four low–order digits)
Parameter No.7441	5000	(four high–order digits)

(ii) External data input/output

Use of the external data input/output function allows input/output of tool data. Note that this function is available only when the BMI interface is used.

(iii) Automatic tool length measurement (G37)

When automatic tool length measurement is specified, the tool length compensation data for the valid tool number is updated.

(iv) System variable (custom macro)

System variables for tool offset cannot be used to read or change the compensation data of tool number–based tool offsets.

(v) NC window

The NC window cannot be used to read or change the tool data of tool number–based tool offsets.

15.11 Programmable Parameter Input

The setting of parameters and pitch error data can be specified on tape. This enables:

1. Setting parameters such as pitch error compensation data used for changing attachments
2. Changing parameters, such as the maximum cutting speed and cutting–time constant, according to the machining conditions

(1) Format

G10 L50 ;
N – – R – – – ;
N – – P – – – R – – – ;
:
N – – R – – – ;
G11 ;

where,

G10L50 : Parameter input mode
G11 : Parameter input mode cancel
N––– : Parameter number (or pitch error data number + 10000)
P––– : Axis number (for axis parameters)
R––– : Parameter setting (or pitch error data)

WARNING 1 Before changing the pitch error compensation data or backlash compensation data, disable pitch error compensation or backlash compensation (return to the machine zero point). If the data is changed while compensation is enabled, the machine position will shift.

WARNING 2 If changing of the grid shift is specified, the machine moves by the difference between the new grid shift and previous grid shift.

WARNING 3 Cancel the canned cycle mode before using this function, otherwise the machine starts drilling.

NOTE 1 In the parameter input mode, other NC statements cannot be specified.

NOTE 2 Do not use a decimal point in address R. Otherwise, the value actually set varies according to the increment system for the reference axis.

When a custom macro variable is used, the same problem occurs, so take care.

NOTE 3 In the parameter input mode, a fixed–point number cannot be specified in address R.

NOTE 4 The following parameters can be set using this function:

0000, 0010 to 0012, 0020 to 0023, 0100 to 0107, 0111 to 0114
1000, 1004, 1005, 1020 to 1022, 1030, 1031, 1220 to 1226, 1241 to 1243, 1400, 1401, 1410, 1412 to 1414, 1420 to 1428, 1450 to 1461, 1493, 1600, 1620 to 1631, 1800, 1802, 1825 to 1832, 1834, 1850, 1851
2000, 2002, 2003, 2010, 2011, 2014 to 2016, 2020, 2021, 2030 to 2033, 2049, 2200 to 2204, 2291, 2311 to 2318, 2321 to 2328, 2331 to 2338, 2341 to 2348, 2351 to 2358, 2361 to 2368, 2371 to 2378, 2381 to 2388, 2400 to 2402, 2410 to 2418, 2900
5001 to 5003, 5011, 5013, 5110 to 5112, 5120 to 5122, 5130 to 5132, 5140 to 5142, 5150 to 5152, 5160 to 5162, 5200, 5210, 5220 to 5225, 5600 to 5605, 5610 to 5628, 5631 to 5637, 5640 to 5648, 5660 to 5664, 5670, 5680 to 5683, 5691 to 5698, 5701, 5702, 5721
6000 to 6002, 6010, 6011, 6200 to 6202, 6210 to 6220, 6240, 6400, 6410, 6411, 6610 to 6614, 6820
7000 to 7002, 7010 to 7017, 7031 to 7034, 7050 to 7059, 7071 to 7089, 7110, 7220 to 7202, 7211 to 7214, 7300, 7311 to 7313, 7321 to 7323, 7331 to 7333, 7600, 7601, 7603, 7605, 7607, 7632, 7633, 7682, 7683, 7701, 7702
8000, 8010

15.12 Tool Length Compensation along the Tool Axis

The tool axis does not always lie along the Z–axis. When the tool axis is a rotation axis, this function allows the tool length to be compensated along the tool axis. In a program in which the rotation of the tool axis is specified, machining cannot be performed without this function if the actual tool length differs from the programmed length. It becomes necessary to re–create the program so as to conform to the actual tool length.

Using this function eliminates the need to re–create the program even when the tool length varies.

15.12.1 Format

G43.1X__Y__Z__H__A(B)__C__;

G43.1 : Tool length compensation along the tool axis enabled

G49 : Tool length compensation along the tool axis disabled

G43.1 and G49 are continuous–state G codes.

If a rotation axis or an H code is specified following G43.1, the compensation vector calculations are performed again. When using an H code, specify an offset number for which tool length compensation is set.

15.12.2 Vectors for tool length compensation along the tool axis

In the mode for tool length compensation along the tool axis, the following vectors are generated at the end of each block.

The rotation axes are set with the BC bit of parameter No. 6001 and the TLAX bit of parameter No. 7550.

a) For the A and C axes (with the tool axis along the Z–axis)

$Vx = L_C \times SIN(a) \times SIN(c)$

$Vy = –L_C \times SIN(a) \times COS(c)$

$Vz = L_C \times COS(a)$

b) For the B and C axes (with the tool axis along the Z–axis)

$Vx = L_C \times SIN(b) \times COS(c)$

$Vy = L_C \times SIN(b) \times SIN(c)$

$Vz = L_C \times COS(b)$

c) For the A and B axes (with the tool axis along the X–axis)

$Vx = L_C \times COS(b)$

$Vy = L_C \times SIN(b) \times SIN(a)$

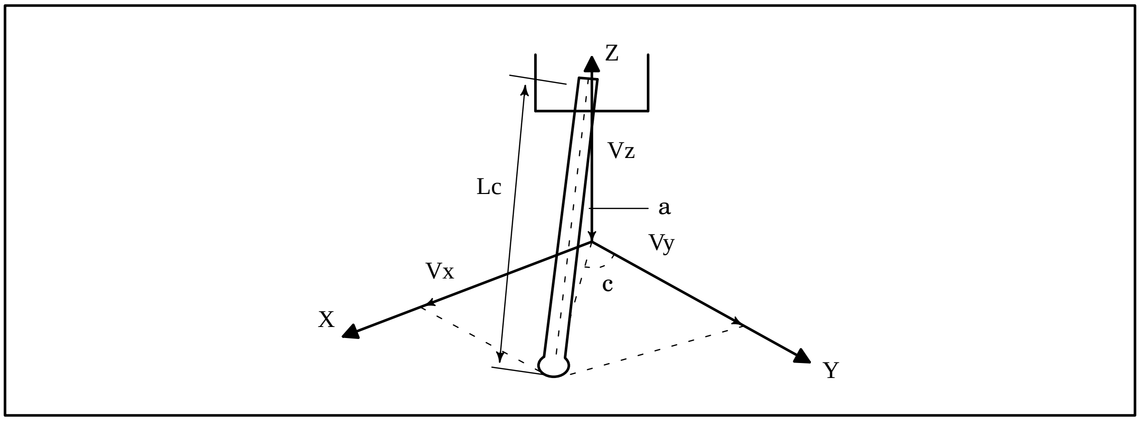
$Vz = –L_C \times SIN(b) \times COS(a)$

Vx, Vy, Vz : Tool length compensation vectors along the X–, Y, and Z–axes

Lc : Tool length compensation data (offset)

a, b, c : Absolute coordinates of the A, B, and C axes in the program coordinate system

A and C axes

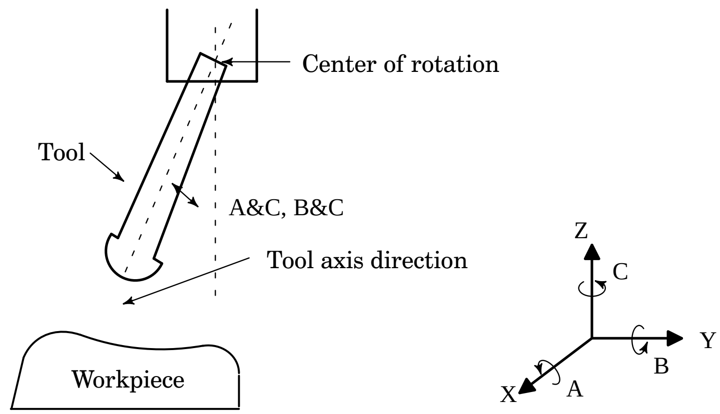


NOTE

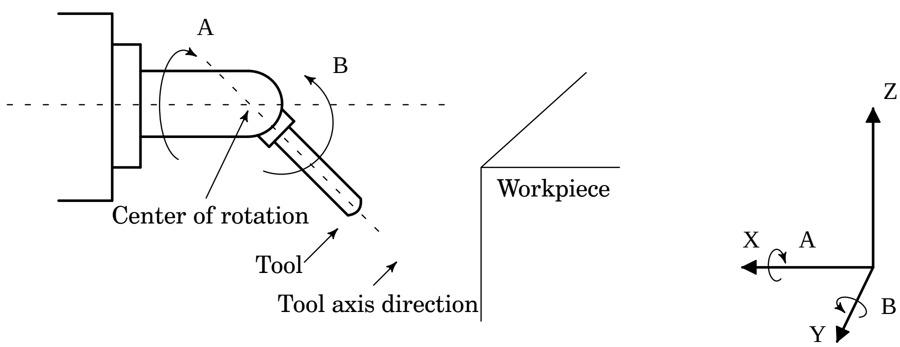
(1) Automatic return to the reference position (G28, G29, G30)
In the mode for tool length compensation along the tool axis (G43.1), none of the three commands, G28, G29, and G30, can be specified.

(2) Tool length compensation (G43, G44, G49)
The G43.1 command for specifying tool length compensation along the tool axis belongs to the same group of G codes as those for normal tool length compensation, G43 and G44. This means that these two modes for tool length compensation cannot be applied at the same time. To cancel tool length compensation, use G49 for both modes.

(3) Tool length compensation along the tool axis is applicable to machines having one of the following axis configurations, where the tool axis rotates around the center of one tool:
A and C axes
B and C axes



A and B axes (with the tool axis along the X–axis)



15.13 Rotary Table Dynamic Fixture Offset

The workpiece coordinate system is set after the position of a workpiece placed on a rotary table is measured. In a conventional system, however, if the rotary table rotates before cutting is started, the position of the workpiece must be measured again and the workpiece coordinate system must be reset accordingly.

The rotary table dynamic fixture offset function saves the operator the trouble of re–setting the workpiece coordinate system when the rotary table rotates before cutting is started. With this function the operator simply sets the position of a workpiece placed at a certain position on the rotary table as a reference fixture offset. If the rotary table rotates, the system automatically obtains a current fixture offset from the angular displacement of the rotary table and creates a suitable workpiece coordinate system. After the reference fixture offset is set, the workpiece coordinate system is prepared dynamically, wherever the rotary table is located.

The zero point of the workpiece coordinate system is obtained by adding the fixture offset to the offset from the workpiece reference point.

(1) Setting the data

(1) Setting a group of three parameters which specify one rotation axis and two linear axes constituting the plane of rotation

In each group, specify the number of the rotation axis as the first parameter and the numbers of the linear axes as the second and third parameters. The rotation in the normal direction about the rotation axis must agree with the rotation from the positive side of the linear axis set as the second parameter to the positive side of the linear axis set as the third parameter.

Example Suppose that a machine has four axes, X, Y, Z, and C. The X–, Y–, and Z–axes form a right–handed coordinate system. The C–axis is a rotation axis. When viewed from the positive side of the Z–axis, a rotation in the normal direction about the C–axis is treated as the counterclockwise rotation around the Z–axis.

For this machine, specify the parameters as follows:

First parameter : 4 (C–axis)

Second parameter : 1 (X–axis)

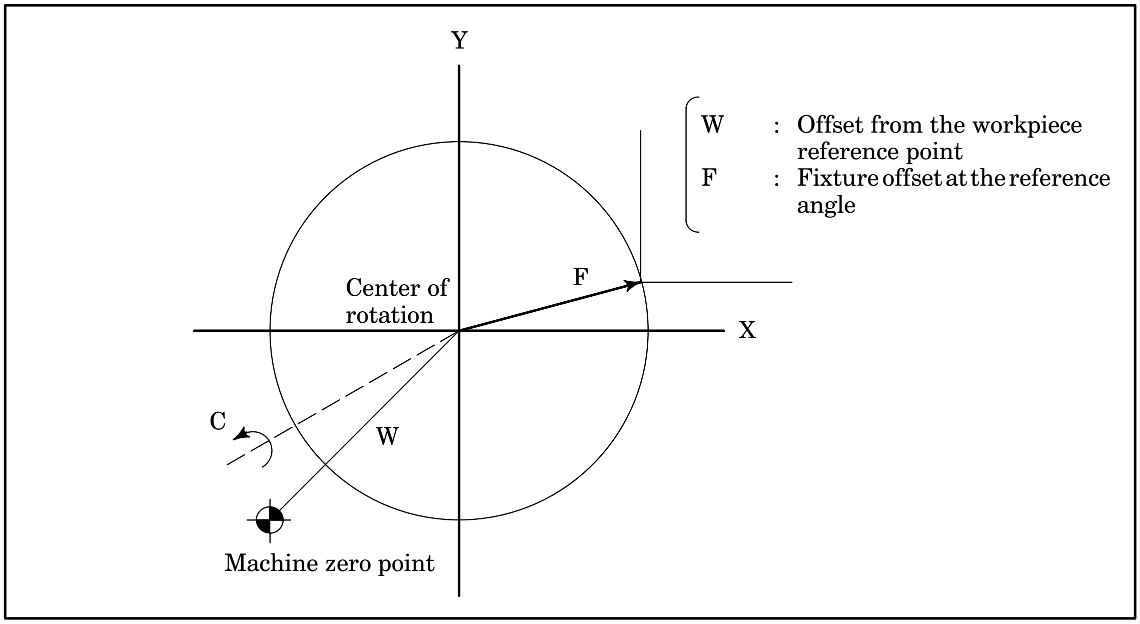
Third parameter : 2 (Y–axis)

Up to three groups of parameters can be set. In calculation of the fixture offset, the data of the rotation axis specified in the first group is calculated first. Then, the data of the second and third groups are calculated.

If a machine has two or more rotation axes and the plane of rotation depends on the rotation about another rotation axis, the plane of rotation is set when the angular displacement about the rotation axis is 0 .

(2) Setting the reference angle of the rotation axis and the corresponding reference fixture offset

Set the reference angle of the rotation axis and the fixture offset that corresponds to the reference angle.



Set the data on the fixture offset screen. Eight groups of data items can be specified.

(3) Setting a parameter for enabling or disabling the fixture offset of each axis (bit 0 of parameter 1007)

For the axis for which the fixture offset is enabled, set the parameter to 1. This need not be specified for a rotation axis.

(4) Setting the type of fixture offset (bit 1 of parameter 6004)

Specify whether to cause a movement according to the increment or decrement of the fixture offset vector when the vector changes (when G54.2 is specified or when a rotation axis movement occurs in the G54.2 mode).

When 0 is set, the movement is made. (The current position on the workpiece coordinate system does not change. The position on the machine coordinate system changes.)

When 1 is set, the movement is not made. (The current position on the workpiece coordinate system changes. The position on the machine coordinate system does not change.)

(2) Command specification

G54.2 Pn ; [n:Number of the fixture offset (1 to 8) or value 0]

When this command is specified, a fixture offset is calculated from the current angular displacement and the data of n. The fixture offset becomes valid.

If n is set to 0, the fixture offset becomes invalid.

If a movement is made in the G54.2 mode about the rotation axis processed with the fixture offset, the vector is calculated again.

(3) Operation at reset

The NCM bit (bit 7 of parameter 2401) determines whether the fixture offset is canceled at reset.

When the bit is set to 1, the vector before the reset is retained.

When the bit is set to 0, the vector is cleared. The increment or decrement of the vector will never cause a movement, regardless of the value set in bit 1 of parameter 6004.

(4) Sample program and operation

Parameter 6068 = 4 (C–axis)

Parameter 6069 = 1 (X–axis), bit 0 of parameter 1007 (X) = 1
(The offset is valid for the X–axis.)

Parameter 6070 = 2 (Y–axis), bit 0 of parameter 1007 (Y) = 1
(The offset is valid for the Y–axis.)

Parameters 6071 to 6076 = 0

Bit 1 of parameter 6004 = 0
(When bit 1 of parameter 6004 is set to 1, the values in square brackets ([ ]) are calculated.)

Data of fixture offset 1 (n = 1)

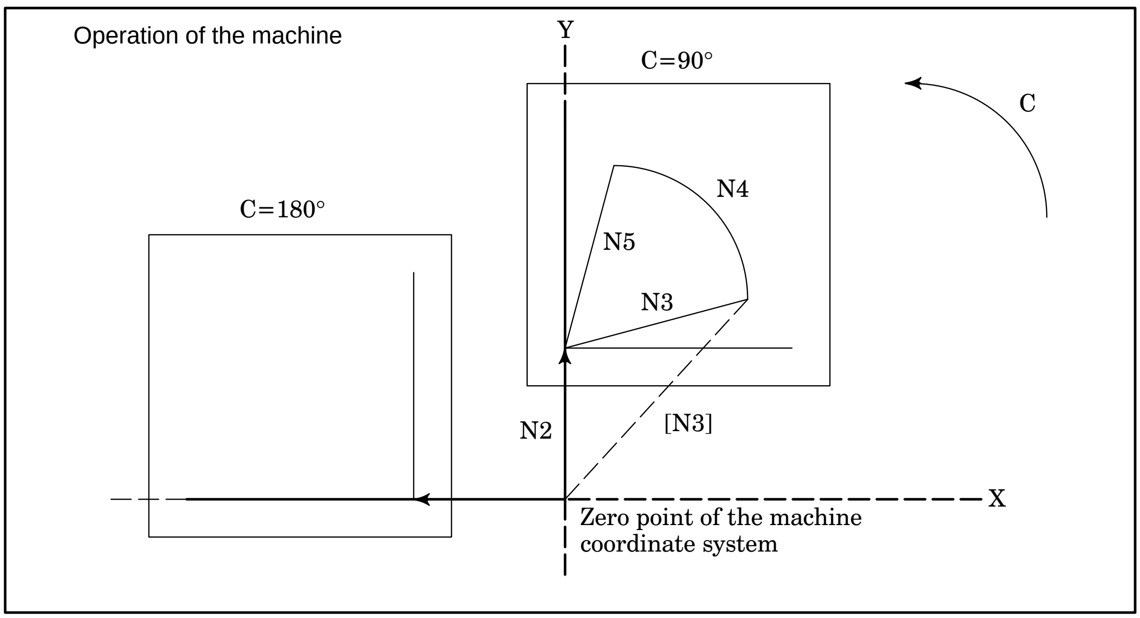
C = 180.0 (reference angle)

X = –10.0

Y = 0.0

When these parameters and data are set, the machine operates as shown below:

Coordinates / Program	Position on the workpiece coordinate system (ABSOLUTE)			Position on the machine coordinate system (MACHINE)			Fixture offset		
	X	Y	C	X	Y	C	X	Y	C
N1 G90 G00 X0 Y0 C90.;	0	0	90.	0	0	90.	0	0	0
N2 G54.2 P1;	0 [0	0 –10.	90. 90.]	0 [0	10. 0	90. 90.]	0 [0	10. 10.	0 0]
N3 G01 X10. Y2. F100.;	10.	2.	90.	10.	12.	90.	0	10.	0
N4 G02 X2. Y10. R10.;	2.	10.	90.	2.	20.	90.	0	10.	0
N5 G01 X0 Y0; ...	0	0	90.	0	10.	90.	0	10.	0



When G54.2 P1 is specified in the N2 block, the fixture offset vector (0, 10.0) is calculated. The vector is handled in the same way as the offset from the workpiece reference point. The current position on the workpiece coordinate system is (0, –10.0).

If bit 1 of parameter 6004 is set to 0, the tool is moved according to the vector. The resultant position on the workpiece coordinate system is (0, 0), the position before the command is specified.

(5) Fixture offset screen

Display the fixture offset screen by pressing the <OFFSET> soft key, then the <FOFS> soft key.
Fixture offset screen for the 9–inch CRT or MDI panel

```
FIXTURE OFFSET            O0100 N00010
     ACT ( P = 01 )     NO. 02
      X      0.0000      X      0.0000
      Y     10.0000      Y      0.0000
      Z      0.0000      Z      0.0000
      C      0.0000      C      0.0000
     NO. 01             NO. 03
      X    –10.0000      X      0.0000
      Y      0.0000      Y      0.0000
      Z      0.0000      Z      0.0000
      C   –180.0000      C      0.0000


>
 MDI   *** STOP **** *** ***08:40:00  LSK
INPUT       INPUT+   PUNCH   INP_NO  +
```

Place the cursor on a desired field by pressing the cursor keys and the <INP_NO> soft key. Enter a desired value and press the <INPUT> and <INPUT+> keys.
The number of the current fixture offset (P) and the fixture offset vector are displayed at the top left of the screen (ACT).

(6) Inputting and outputting the fixture offset

The setting of a program and external data can be input and output as described below:

(1) Setting the reference fixture offset by G10

G10 L21 Pn IP;

n :Number of fixture offset
IP :Reference fixture offset or reference angle of each axis

This command sets the reference fixture offset or reference angle in a program.
When the command is executed in the G90 mode, the specified value is set directly.
When the command is executed in the G91 mode, the specified value plus a value set before the execution is set.

NOTE	The programmable data input function (G10) is required.

(2) Reading and writing the data by a system variable of a custom macro

The reference fixture offset or reference angle can be read and written by the following system variables:
15001+20*(n–1)+(m–1)

n :Number of fixture offset (1 to 8)
m :Axis number (1 to the number of controlled axes)

NOTE	The custom macro function is required.

(3) Reading and writing the data by the PMC window

The data can be read and written as a system variable of a custom macro by the PMC window.

NOTE	The NC window function and custom macro function are required.

(4) Outputting the data to an external device

By selecting <PUNCH> on the fixture offset screen, the data can be output to a Floppy Cassette or other external device via RS–232C.
The data is output in the G10 form without a program number. After the output data is cataloged as a program, it can be input by executing the program.

NOTE	The reader/punch interface function and programmable data input function (G10) are required.

(7) Calculating the Fixture Offset

(1) Relationship between the rotation axis and linear axes (when A = B = 0)

First group : 5 (B–axis), 1 (X–axis), 3 (Z–axis)

Second group : 4 (A–axis), 3 (Z–axis), 2 (Y–axis)

Third group : 0, 0, 0

(2) Reference angle and reference fixture offset

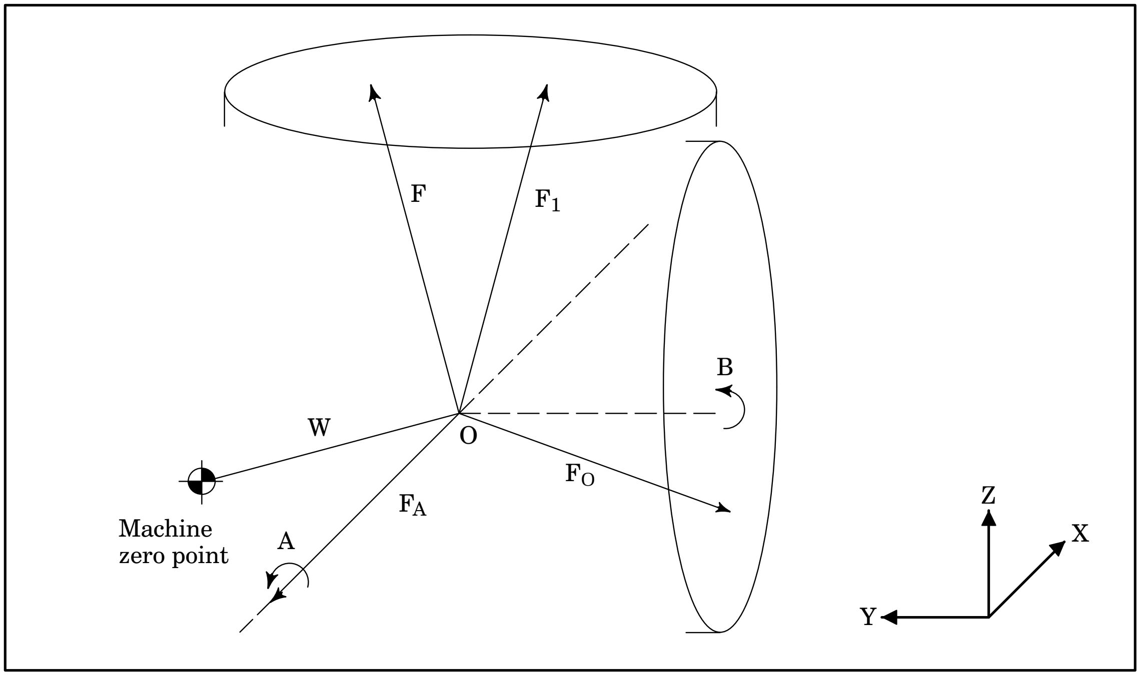
X: F_{OX}

Y: F_{OY}

Z: F_{OZ}

A: ϕ_O

B: ϕ_O When these are specified, the data is calculated as follows:



$F_A = (F_{AX}, F_{AY}, F_{AZ})$

$F_1 = (F_{1X}, F_{1Y}, F_{1Z})$

$F = (F_X, F_Y, F_Z)$ Then,

$$\begin{pmatrix} F_{AX} \\ F_{AY} \\ F_{AZ} \end{pmatrix} = \begin{pmatrix} \cos(-\theta_O) & 0 & -\sin(-\theta_O) \\ 0 & 1 & 0 \\ \sin(-\theta_O) & 0 & \cos(-\theta_O) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \sin(-\phi_O) & \cos(-\phi_O) \\ 0 & \cos(-\phi_O) & -\sin(-\phi_O) \end{pmatrix} \begin{pmatrix} F_{OX} \\ F_{OY} \\ F_{OZ} \end{pmatrix}$$

$$\begin{pmatrix} F_{1X} \\ F_{1Y} \\ F_{1Z} \end{pmatrix} = \begin{pmatrix} \cos(\theta) & 0 & -\sin(-\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \begin{pmatrix} F_{AX} \\ F_{AY} \\ F_{AZ} \end{pmatrix}$$

$$\begin{pmatrix} F_X \\ F_Y \\ F_Z \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \sin(\phi) & \cos(\phi) \\ 0 & \cos(\phi) & -\sin(\phi) \end{pmatrix} \begin{pmatrix} F_{1X} \\ F_{1Y} \\ F_{1Z} \end{pmatrix}$$

(8) Buffering in fixture offset mode

Parameter No. 6006 (NMFOFS) can be used to enable buffering for subsequent blocks even if a G54.2 command is specified or a rotation axis command related to a fixture offset is specified in G54.2 mode.

– When parameter No. 6006 (NMFOFS) = 0

1) Buffering is suppressed for a G54.2 command, or any rotation axis command related to a fixture offset that is specified in G54.2 mode. In other words, buffering for subsequent blocks is not performed until the execution of that block has been completed.

2) If any of parameters No. 6068 to 6076 or the reference fixture offset is changed in G54.2 mode, the new data becomes valid when G54.2 is next specified.

– When parameter No. 6006 (NMFOFS) = 1

1) If any of parameters No. 6068 to 6076 or the reference fixture offset is changed in G54.2 mode, the new data becomes valid in the block that is buffered next.

NOTE 1 If one of parameters 6068 to 6076 or the reference fixture offset is changed in the G54.2 mode, the new data becomes valid the next time G54.2 is specified.

NOTE 2 It depends on the current continuous–state code of the 01 group whether a change in the fixture offset vector causes a movement. If the system is in a mode other than the G00 or G01 mode (G02, G03, etc.), the movement is made temporarily in the G01 mode.

NOTE 3 When the automatic operation is stopped using the SBK stop function during the G54.2 mode and a manual movement is made about the rotation axis, the fixture offset vector does not change. The vector is calculated when a rotation axis command or G54.2 is specified for automatic operation or MDI operation.

If a manual intervention is made when bit 1 of parameter 2402 is set to 1 and the manual absolute switch is on, a rotation axis command in the incremental mode (G91 mode) calculates the vector using the coordinates in which the manual intervention is not reflected.

Example N1 G90 G00 C10.0;
N2 G54.2 P1 ;
After these commands are executed, a manual intervention is made while the manual absolute switch is on. A movement of +20.0 is made about the C–axis.
N3 G91 C30.0;
If this command is specified after the operation is resumed, the coordinate of the C–axis on the workpiece coordinate system becomes 60.0. When the fixture offset is calculated, the coordinate of the C–axis is assumed to be 40.0.
If the INC bit (bit 1 of parameter 2402) is set to 0 or the G90 mode is selected when N3 is speci fied, the coordinate of the C–axis is assumed to be 30.0, which is the specified value, in calculation.

NOTE 4 When a movement about the rotation axis processed with the fixture offset is specified in the G54.2 mode, the vector is calculated based on the coordinate of the rotation axis at the end of the block. A movement is then made on the workpiece coordinate system to the position pointed to by the vector.

NOTE 5 In calculation of the fixture offset, the coordinate of the rotation axis on the workpiece coordinate system is used. If a tool offset or another offset is applied, the coordinate before the offset is used. If the mirror image function or scaling function is executed, the coordinate before the function was executed is used.

NOTE 6 If the following commands are specified for the rotation axis in the G54.2 mode, the fixture offset vector is not calculated:

Command related to the machine coordinate system : G53

Command specifying a change of the workpiece coordinate system:
G54 to G59, G54.1, G92, G52

Command specifying a return to the reference position : G27, G28, G29, G30, G30.1

NOTE 7 High–speed machining (G10.3, G65.3) can be carried out in the G54.2 mode. If this changes the coordinate of the rotation axis, the fixture offset vector is not calculated again.

NOTE 8 The rotation axis used for polar coordinate interpolation (G12.1) cannot be set as the rotation axis for the fixture offset.

NOTE 9 When the fixture offset function is used, the length of tape usable for storage is reduced approximately 1.5 m.

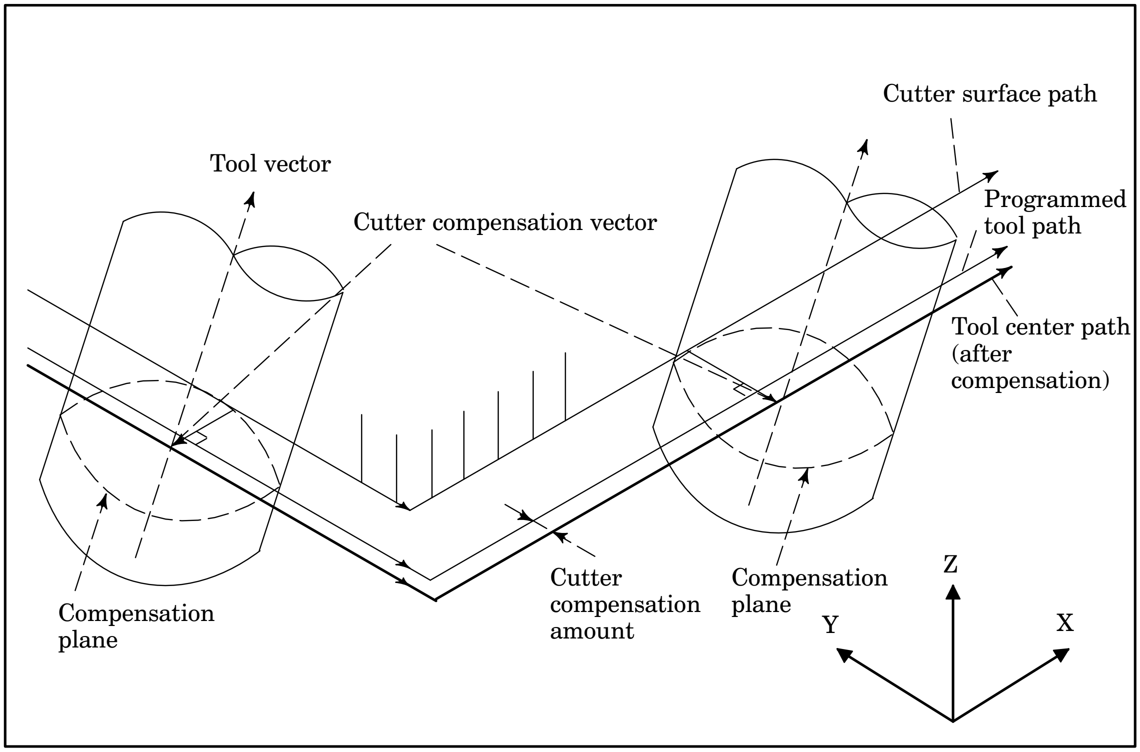
NOTE 10 In the G54.2 mode, background drawing is inhibited.

15.14 Three–Dimensional Cutter Compensation

The three–dimensional cutter compensation function is used with machines that can control the direction of tool axis movement by using rotation axes (such as the B– and C–axes). This function performs cutter compensation by calculating a tool vector from the positions of the rotation axes, then calculating a compensation vector in a plane (compensation plane) that is perpendicular to the tool vector.

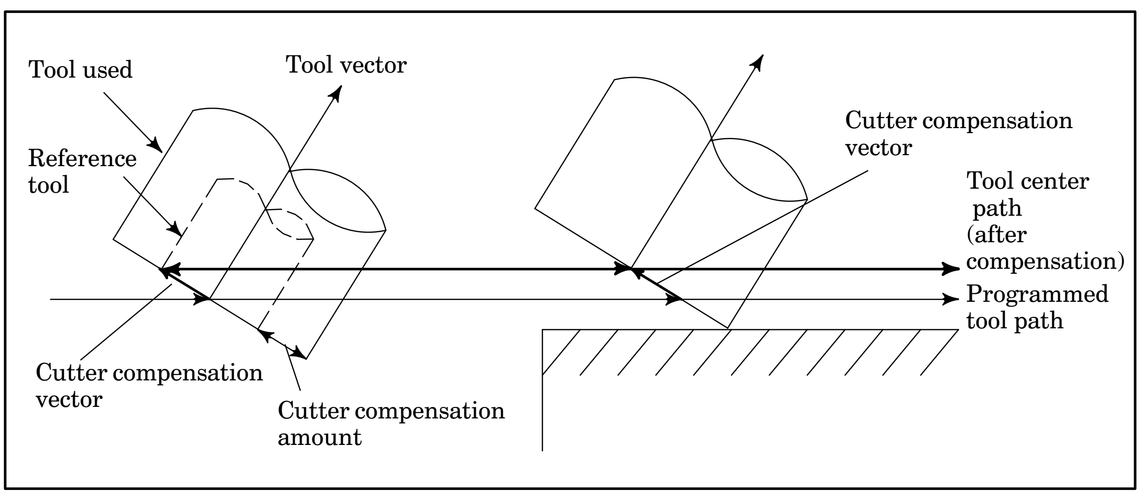
There are two types of cutter compensation: Tool side compensation and leading edge compensation. Which is used depends on the type of machining.

Example Tool side compensation



Leading edge compensation is performed when a workpiece is machined by the edge of the tool. In leading edge compensation, the tool is shifted automatically by the distance of the tool radius along the line where the plane formed by the tool vector and the movement direction and a plane perpendicular to the tool axis direction intersect.

Example Leading edge compensation



15.14.1 Tool side compensation

(1) Command specification

The command below enables tool side compensation:

G41.2 X_ Y_ Z_ D_ ; (tool compensation to the left as viewed from behind the tool)

G42.2 X_ Y_ Z_ D_ ; (tool compensation to the right as viewed from behind the tool)

The following command cancels tool side compensation:

G40 X_ Y_ Z_ ;

WARNING	When selecting type C for the operation at tool side compensation start–up or cancellation, do not specify movement such as X_ Y_ Z_ in G41.2 or G42.2 blocks.

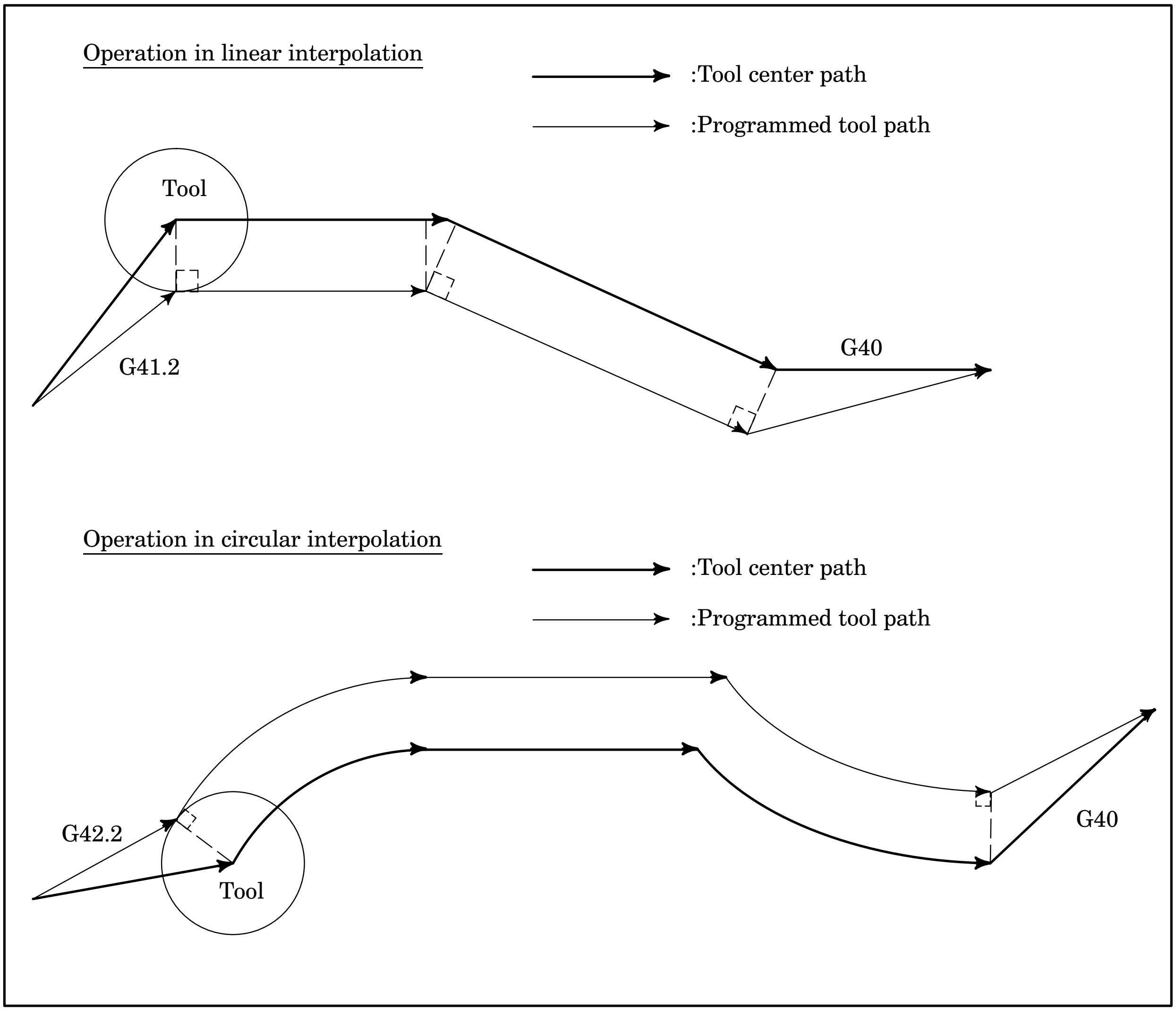
(2) Operation at compensation start–up and cancellation

At tool side compensation start–up or cancellation, one of three types of operation takes place. One of the three can be selected by setting the CSU bit of parameter 6001 and the SUC bit of parameter 6003. The figures below show the programmed tool path and the actual tool path in the compensation plane (a plane perpendicular to the tool axis).

(1) Type A

Type A operation is similar to cutter compensation C as shown below.

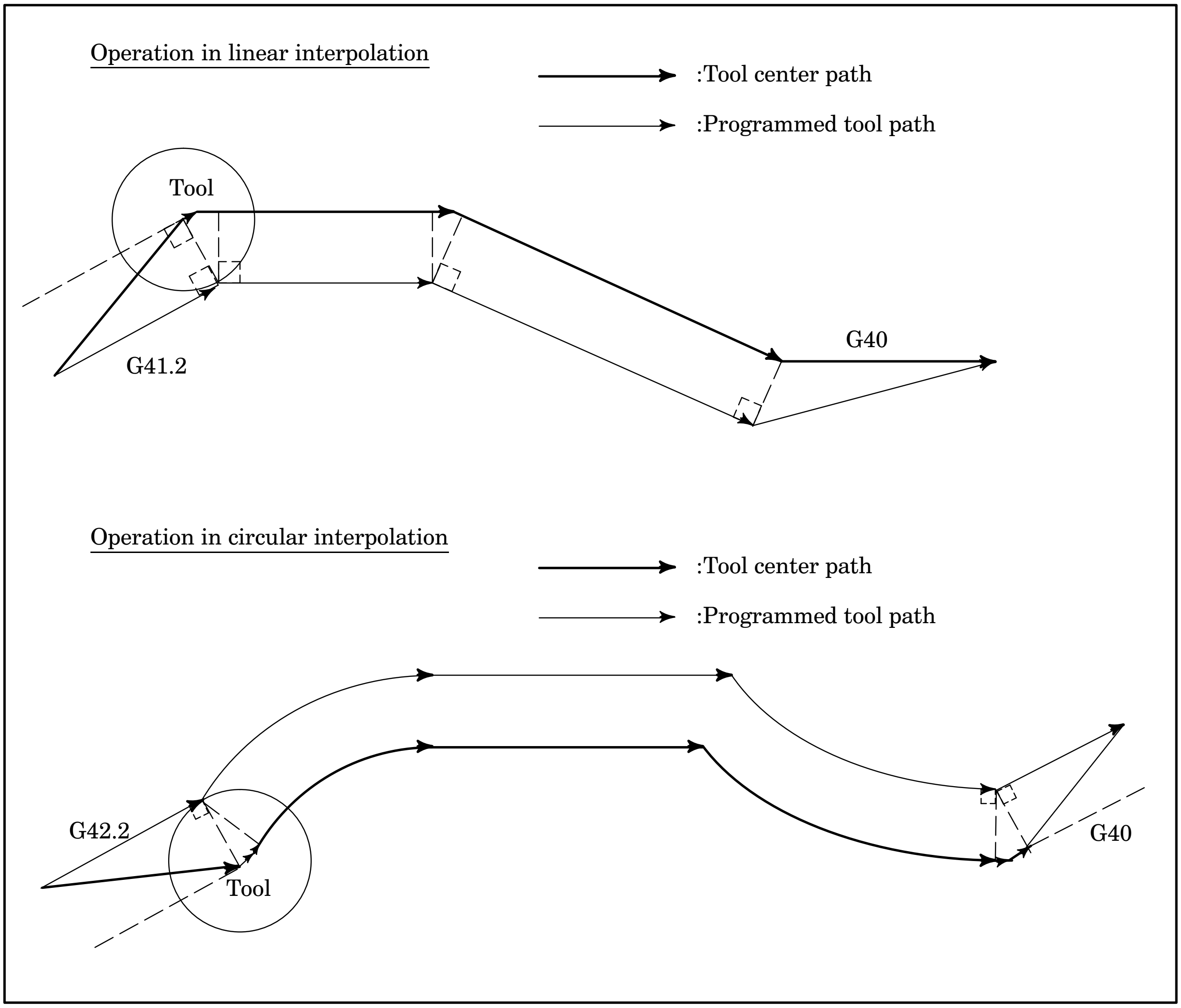
(For details, see Subsection 15.3.7.)



(2) Type B

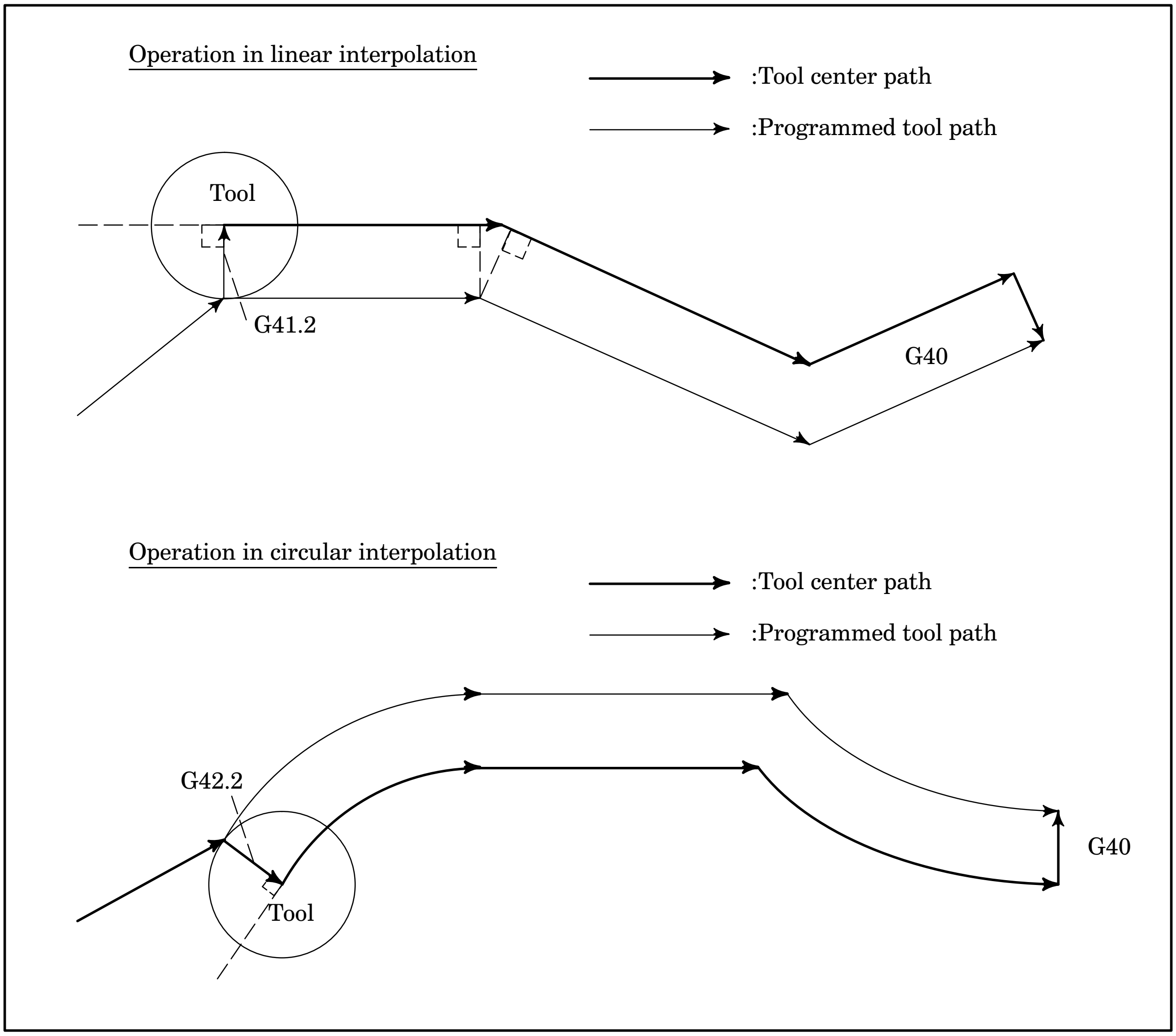
Type B operation is similar to cutter compensation C as shown below.

(For details, see Subsection 15.3.7.)



(3) Type C

As shown in the following figures, when G41.2, G42.2, or G40 is specified, a block is inserted which moves the tool perpendicularly to the movement direction specified in the next block by the distance of the tool radius.



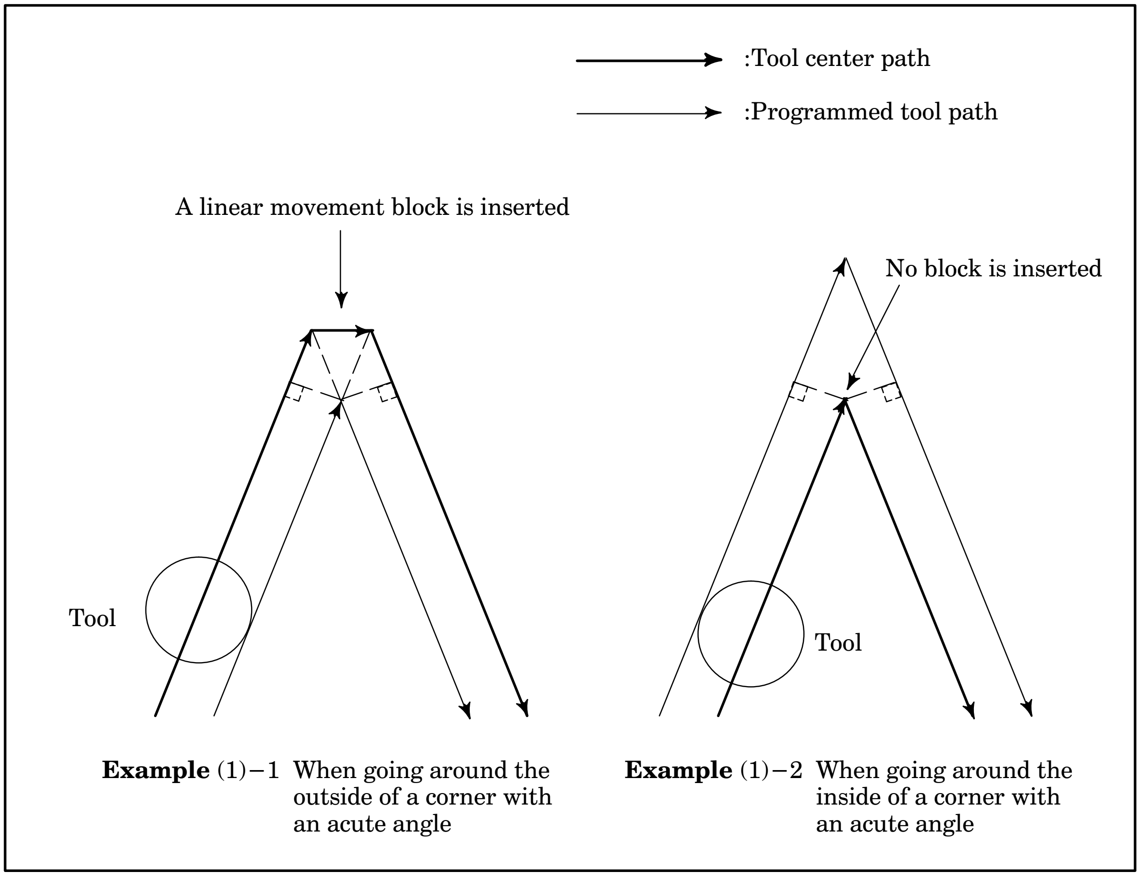
For type C operation, the following conditions must be satisfied when tool side compensation is started up or canceled:

- The block containing G40, G41.2, or G42.2 must be executed in the G00 or G01 mode.
- The block containing G40, G41.2, or G42.2 must have no move command.
- The block after the block containing G41.2 or G42.2 must contain a G00, G01, G02, or G03 move command.

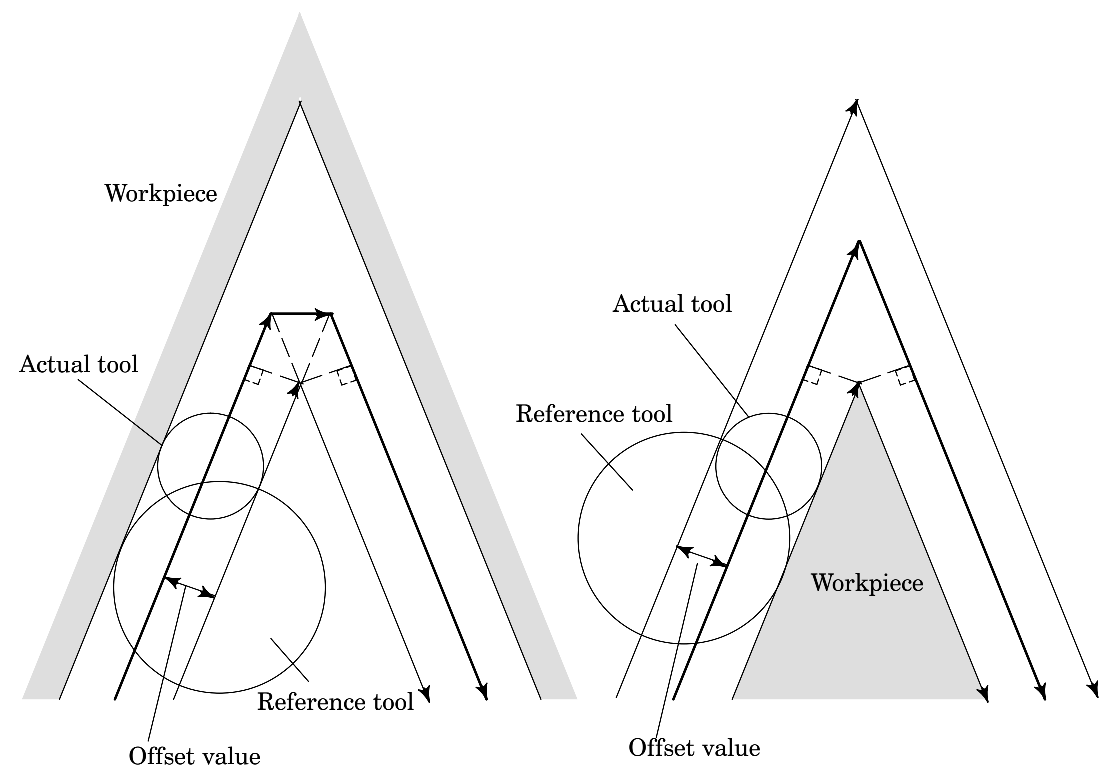
(3) Operation in the compensation mode

Changing of offset directions and offset values, holding of vectors, interference checks, and so on are performed in the same way as for cutter compensation C. (For details, see Subsection 15.3.7.) G39 (corner rounding) cannot be specified. So, note the following:

(1) When the tool center path goes outside the programmed tool path at a corner, a linear movement block, instead of an arc movement block, is inserted to move the tool around the corner. When the tool center path goes inside the programmed tool path, no block is inserted.



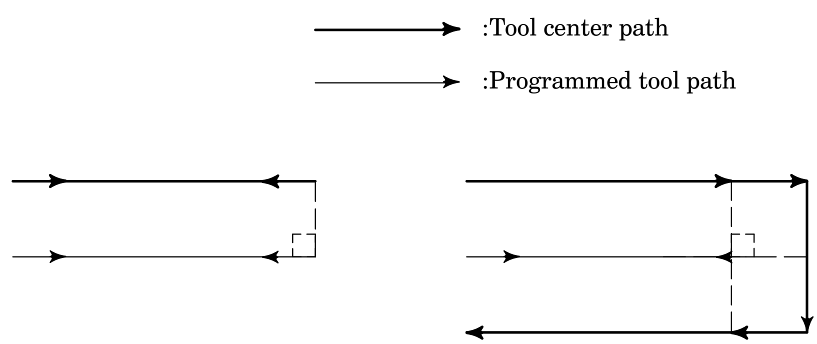
In the above examples, the term "inside" means that the tool center path is positioned inside the programmed tool path at a corner, and "outside" means that the tool center path is positioned outside the programmed tool path. In Example (1)–3, the relationship between the tool center path and the programmed tool path is the same as in Example (1)–1; the tool center path is positioned outside the programmed tool path. Example (1)–4 has the same relationship as Example (1)–2, where the tool center path is positioned inside the programmed tool path.



Example (1)−3 **Example** (1)−4

(2) When the tool moves at a corner, the feedrate of the previous block is used if the corner is positioned before a single–block stop point; if the corner is after a single–block stop point, the feedrate of the next block is used.

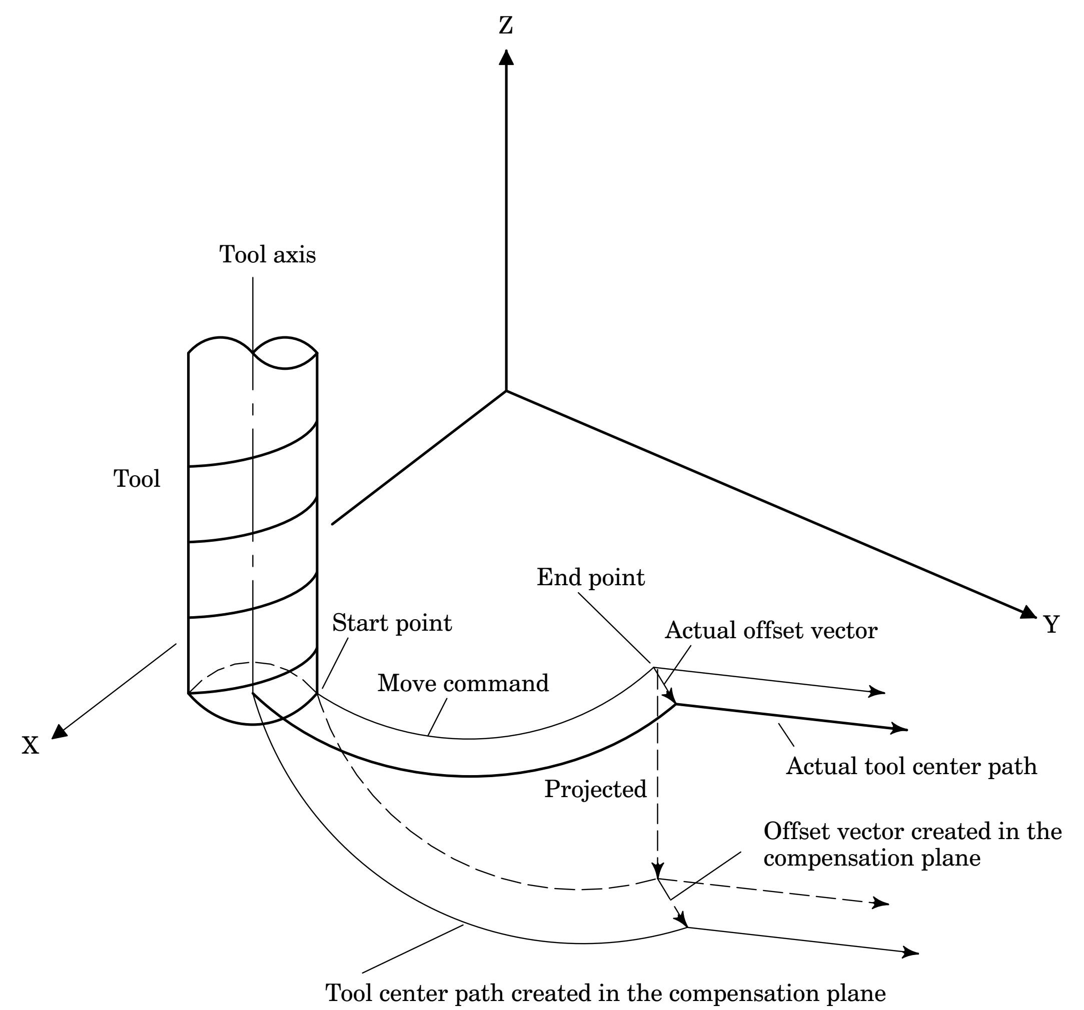
(3) When a command is specified to make the tool retrace the path specified in the previous block, the tool path can match the locus of the previous block by changing the G code to change the offset direction. If the G code is left unchanged, the operation shown in Example (3)–2 results:



Example (3)−1 Tool movement when changing G41.2 to G42.2
(G41.2mode)
G91 G01 X100.0 ;
G42.2 X−100.0 ;

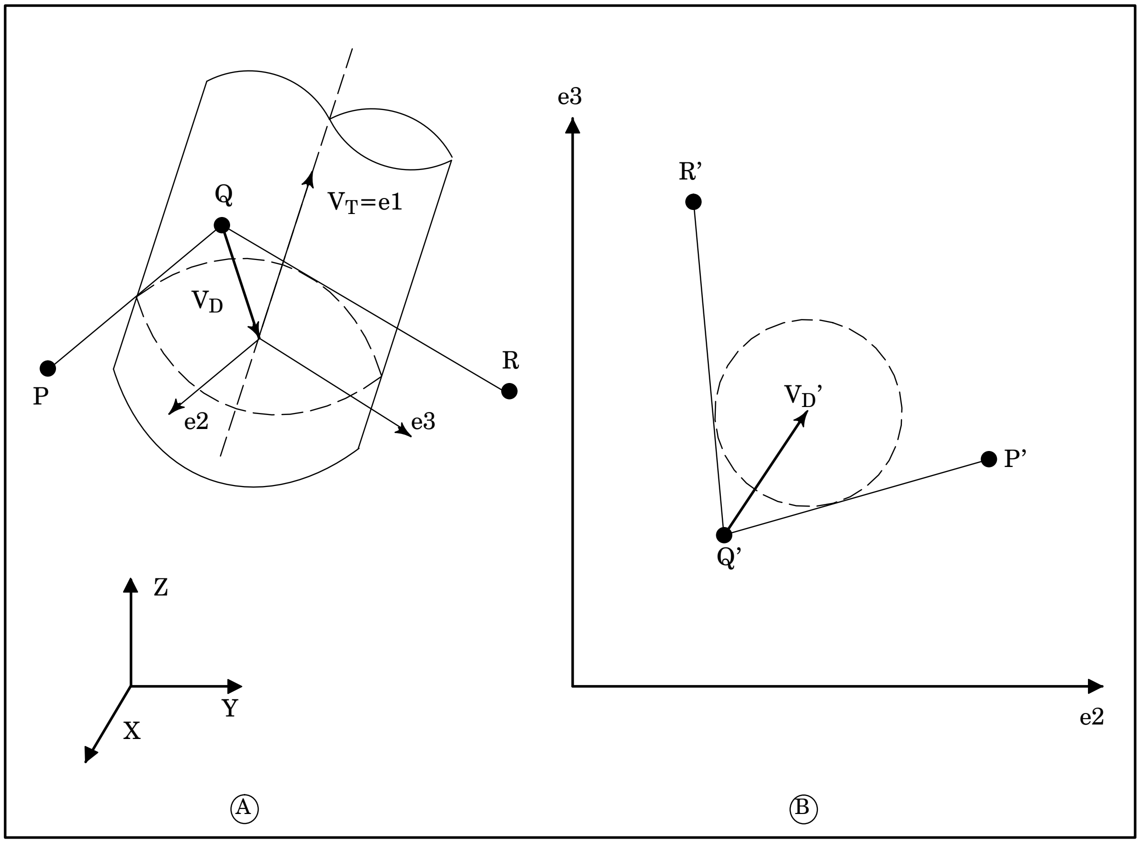
Example (3)−2 Tool movement when the G code is left unchanged
(G41.2mode)
G91 G01 X100.0 ;
X−100.0 ;

(4) Even when the tool movement changes linear to circular (helical), circular (helical) to linear, or circular (helical) to circular (helical), the start, end, and center points of a circular (helical) movement are projected on the compensation plane that is perpendicular to the tool axis, a compensation vector is calculated for the plane, then the vector is added to the originally specified position to obtain the command position. Then the tool is moved linearly or circularly (helically) to the obtained command position. In this case, the tool moves circularly (helically) in the currently selected plane. The tool does not move circularly (helically) in the compensation plane. Therefore, when the tool compensation is for circular movement, the compensation plane must be the XY, YZ, or ZX plane.



Compensation plane = XY plane

(4) Calculations



In Ⓐ, cutter compensation vector V_D at point Q is calculated as follows:

(1) Calculating the tool vector (V_T)

(2) Calculating the coordinate conversion matrix (M)

Coordinate systems are defined as follows:

· Coordinate system C1: {O; X, Y, Z}

Cartesian coordinate system whose fundamental vectors are the following unit vectors along the X–, Y–, and Z–axes:

(1, 0, 0)

(0, 1, 0)

(0, 0, 1)

· Coordinate system C2: {O; e2, e3, e1}

Cartesian coordinate system whose fundamental vectors are the following unit vectors:

e2

e3

e1

where, e2, e3, and e1 are defined as follows:

$$\left(\begin{array}{l} e1=V_T \\ e2=b2/\ |b2|,\ b2=a2-(a2,e1)\cdot\ e1 \\ e3=b3/\ |b3|,\ b3=a3-(a3,e1)\cdot\ e1-(a3,e2)\cdot\ e2 \end{array} \right)$$

a2 is an arbitrary vector linearly independent of e1, and a3 is an arbitrary vector linearly indepen dent of e2 and e1.

The coordinate conversion matrix M from coordinate system C1 to C2, and the coordinate conversion matrix M^{-1} from coordinate system C2 to C1 are expressed as:

$$M=\begin{pmatrix} e2 \\ e3 \\ e1 \end{pmatrix} \quad , \quad M\text{–}1= {}^{t}e2\ \ {}^{t}e3\ \ {}^{t}e1$$

(3) Converting coordinates from coordinate system C1 to coordinate system C2

The coordinates of the start and end points P and Q of a block and coordinates of the end point R of the next block in coordinate system C1 are converted to coordinates P', Q', and R' in coordinate system C2, respectively, by using the following expressions:

P'=M · P
Q'=M · Q
R'=M · R

(4) Calculating the intersection vector (V_D') in the compensation plane {O; e2, e3}

In the coordinates in coordinate system C2 obtained in (3), two components (the e1 component, the component of the tool direction, is excluded) are used to calculate intersection vector V_D' in the compensation plane. (See B) The e1 component of V_D' is assumed to be always 0. The calculation is similar to the calculation of cutter compensation C. Although one vector is obtained in this example, up to four vectors may be calculated.

(5) Converting the intersection vector from coordinate system C2 to coordinate system C1

From the following expression, vector V_D' in coordinate system C2 is converted to vector VD in coordinate system C1:

VD = M–1 · V_D'

Vector VD is the compensation vector in the original XYZ coordinate system.

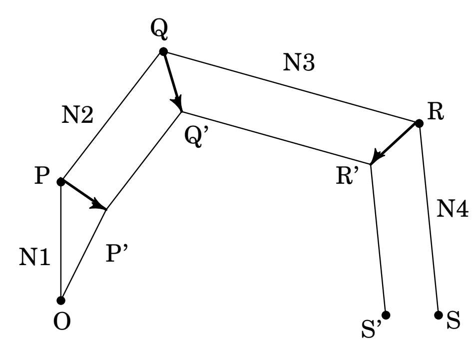
NOTE	If the difference between the e2 and e3 components between two points (components in the compensation plane) is smaller than the value set in parameter 6114 when the intersection vector is calculated in (4), that block is assumed as a block involving no movement. In this case, the intersection is calculated using the coordinates of the next block.

(5) Calculation used when the compensation plane is changed

(1) When a rotation axis and linear axis are specified at the same time

Program (1)

```
...
G90 G00 X0 Y0 Z0 B0 C0 ;
G01 F1000 ;
N1 G42.2 Xp Yp Zp Bp Cp D1 ;
N2 Xq Yq Zq Bq Cq ;
N3 Xr Yr Zr Br Cr ;
N4 Xs Ys Zs Bs Cs ;
...
```



As shown in program (1), when a rotation axis and linear axis are specified in the same block in the G41.2 or G42.2 mode (the compensation plane changes frequently), the cutter compensation vector is calculated using the coordinates of the rotation axis at each point at which the vector is obtained.

Example Vector calculation at the end point (Q) of block N2

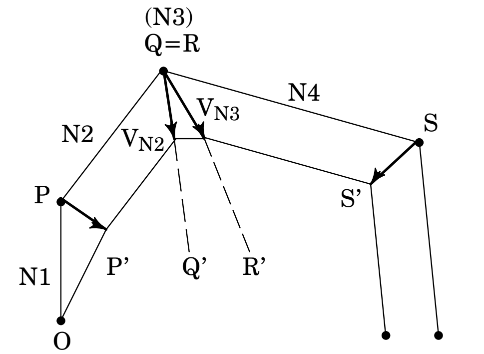
· The tool vector (V_T) and coordinate conversion matrix (M) are calculated using the coordi nates (Bq, Cq) of the rotation axis at point Q.

· The cutter compensation vector is calculated using the resultant coordinates into which three points, P, Q, and R, are converted by matrix M.

(2) When a rotation axis is specified alone

Program (2)

```
...
G90 G00 X0 Y0 Z0 B0 C0 ;
G01 F1000 ;
N1 G42.2 Xp Yp Zp D1 ;
N2 Xq Yq Zq ;
N3 Br Cr ;
N4 Xs Ys Zs ;
...
```



When a rotation axis is specified alone in the G41.2 or G42.2 mode as shown in program (2) (the compensation plane changes), the cutter compensation vector is calculated as follows:

Example Vector calculation at the end point (Q) of block N2

- The tool vector (V_T) and coordinate conversion matrix (M_{N2}) are calculated using the coor dinates (B = 0, C = 0) of the rotation axis at point Q.
- The cutter compensation vector (V_{N2}) is calculated using the resultant coordinates into which three points, P, Q, and S, are converted by matrix M_{N2}.

Vector calculation at the end point of block N3

- The coordinate conversion matrix (M_{N3}) is calculated using the coordinates (Br, Cr) of the rotation axis at point R.
- The cutter compensation vector (V_{N3}) is calculated from the following expression:

$V_{N3} = M_{N3}–1(M_{N2} \cdot V_{N2})$

15.14.2 Leading edge compensation

(1) Command specification

The following command enables leading edge compensation:

G41.3 D_ ;

The command below cancels leading edge compensation:

G40 ;

NOTE 1 G41.3 can be specified only in the G00 or G01 mode. In a block containing G41.3 or G40, only the addresses D, O, and N can be specified.

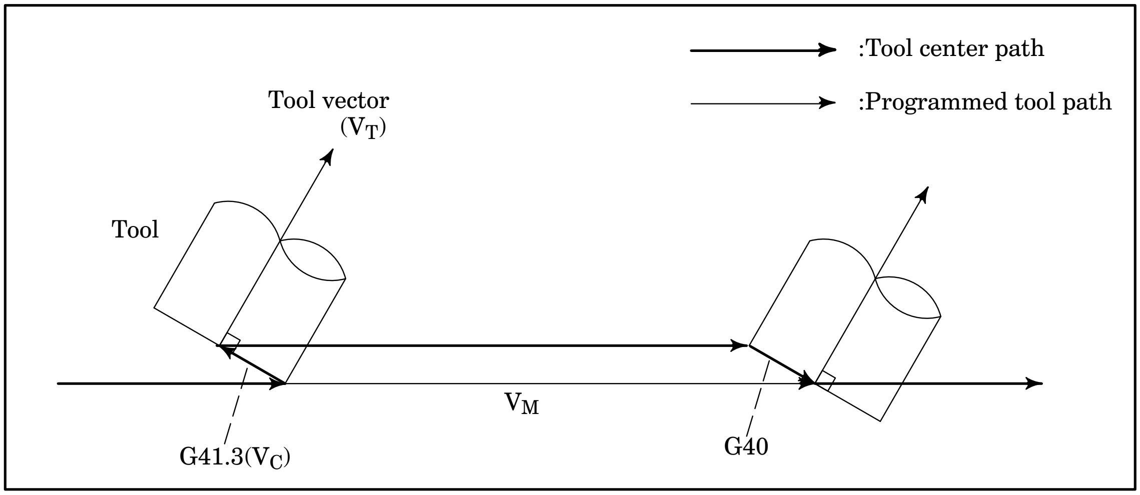
NOTE 2 The block after a block containing a G41.3 command must contain a move command. In that block, however, tool movement in the same direction as the tool axis direction or the opposite direction cannot be specified.

NOTE 3 No continuous–state G code that belongs to the same group as G00 and G01 can be specified in the G41.3 mode.

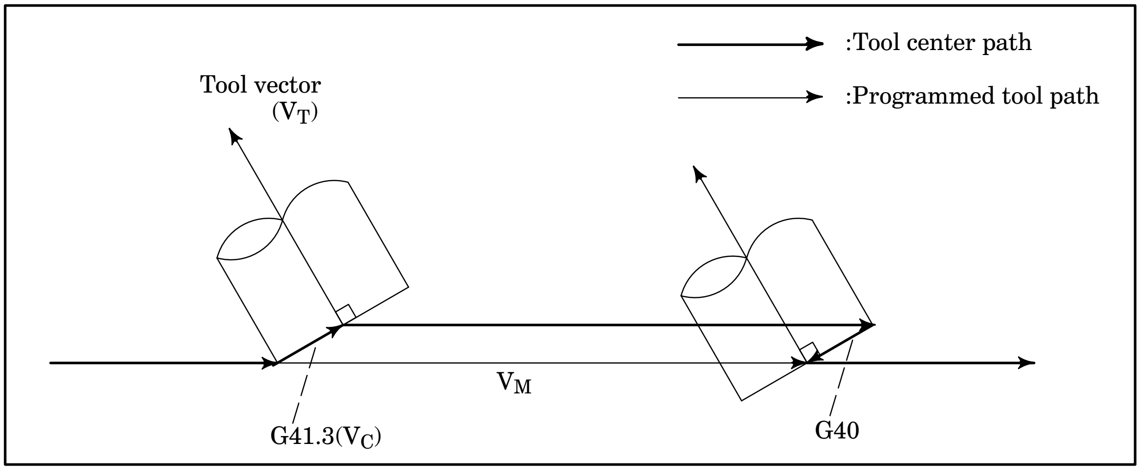
(2) Operation at compensation start–up and cancellation

Unlike tool side compensation the operation performed at leading edge compensation start–up and cancellation does not vary. When G41.3 is specified, the tool is moved by the amount of compensation (V_C) in the plane formed by the movement vector (V_M) of the block after a G41.3 block and the tool vector (V_T) obtained at the time of G41.3 specification. The tool movement is perpendicular to the tool vector. When G40 is specified, the tool is moved to cancel V_C. The following illustrates how the compensation is performed.

(1) When the tool vector is inclined in the direction the tool moves

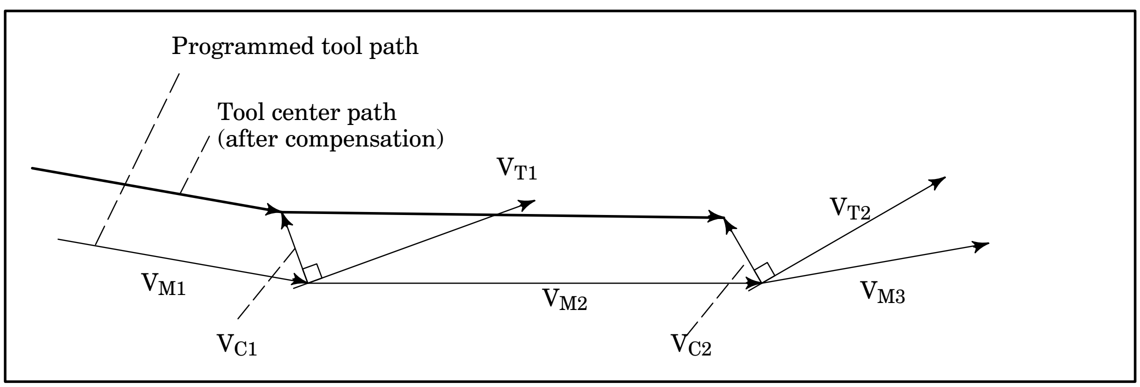


(2) When the tool vector is inclined in the direction opposite to the direction the tool moves



(3) Operation in the compensation mode

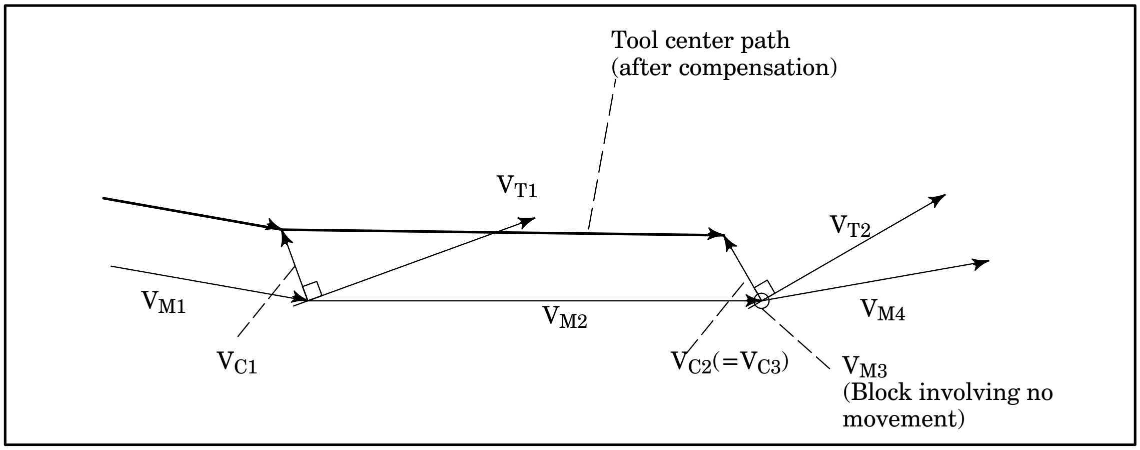
The tool center moves so that a compensation vector (V_C) perpendicular to the tool vector (V_T) is created in the plane formed by the tool vector (V_T) at the end point of each block and the movement vector (V_M) of the next block.



V_{Mn} : Movement vector of block n
V_{Tn} : Tool vector at the end of block n
V_{Cn} : Compensation vector of block n (that lies in the $V_{Tn} - V_{Mn+1}$ plane, and is perpendicular to V_{Tn})

If a G code or M code that suppresses buffering is specified in the compensation mode, however, the compensation vector created immediately before the specification is maintained.

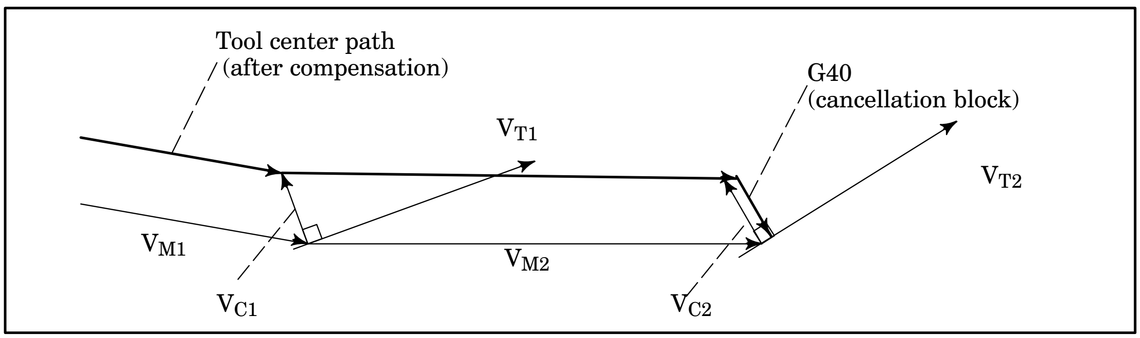
When a block involving no movement (including a block containing a move command for a rotation axis only) is specified, the movement vector of the block after the block involving no movement is used to create a compensation vector as shown below.



(If block 3 involves no movement, the compensation vector of block 2 (V_{C2}) is created so that it is perpendicular to V_{T2} and lies in the plane formed by the movement vector (V_{M4}) of block 4 and the tool vector (V_{T2}) at the end point of block 2.)

If two or more successive blocks involve no movement, the previously created compensation vector is maintained. However, such specification should be avoided.

In the block immediately before the compensation cancel command (G40), a compensation vector is created from the movement vector of that block and the tool vector at the end point of the block as shown below:



(The compensation vector (V_{C2}) of block 2 is created so that it is perpendicular to V_{T2} and lies in the plane formed by the tool vector (V_{T2}) at the end point of block 2 and the movement vector (V_{M2}) of block 2.)

(4) Calculations

In leading edge compensation, the compensation vector is calculated as follows:

(1) Tool vector

The tool vector (V_{Tn}) of block n is calculated using the expression for obtaining the tool length compensation amount for nutating rotary head tool length compensation , on the assumption that the tool compensation amount is 1.0. (See Subsection 2.1.4.)

(2) Movement vector

The movement vector (V_{Mn+1}) of block n+1 is obtained from the following expression:

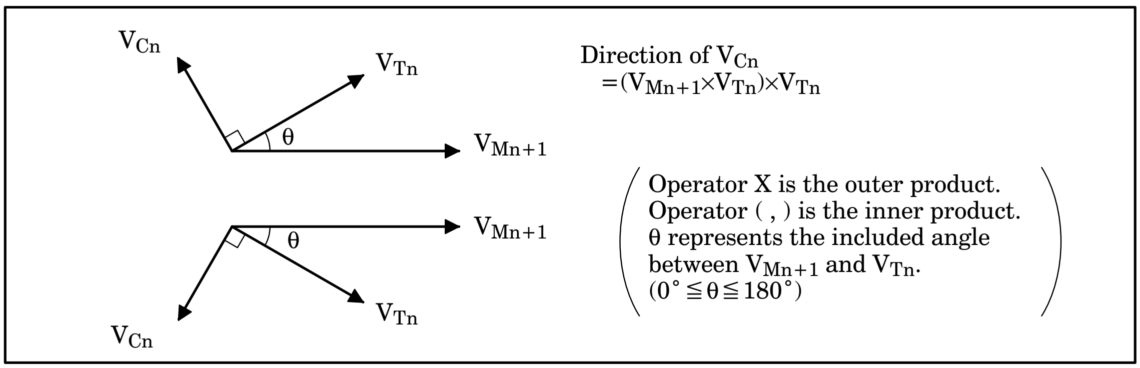
$$V_{Mn+1}=\begin{bmatrix} X_{n+1}-X_n \\ Y_{n+1}-Y_n \\ Z_{n+1}-Z_n \end{bmatrix}$$

(Xn : Absolute X coordinate of the end point of block n
Yn : Absolute Y coordinate of the end point of block n
Zn : Absolute Z coordinate of the end point of block n)

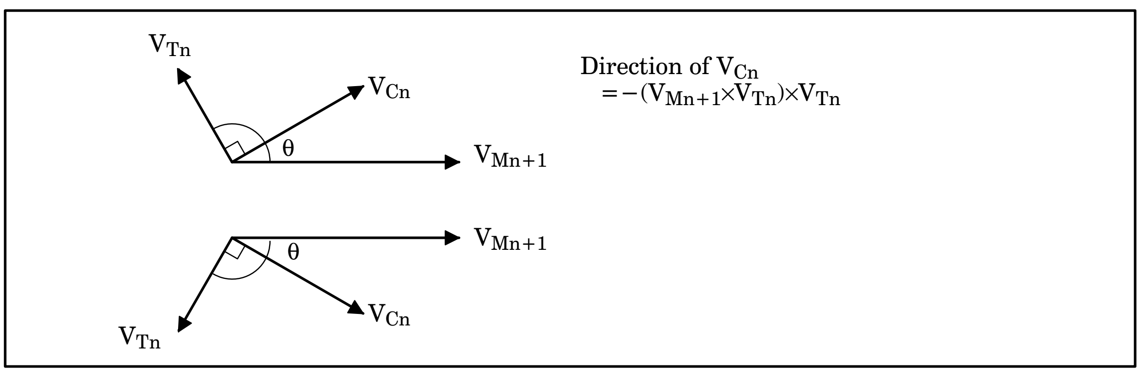
(3) Compensation vector

The direction of the compensation vector (V_{Cn}) of block n is defined as follows:

(I) (V_{Mn}+1, V_{Tn}) > 0 (0°<θ< 90°)



(II) (V_{Mn+1}, V_{Tn}) < 0 (90°<θ< 180°)



The compensation vector (V_{Cn}) of block n is calculated from V_{Tn} and V_{Mn+1} as described below.

Let $V_{Tn} = \begin{bmatrix} V_{Tx} \\ V_{Ty} \\ V_{Tz} \end{bmatrix}$, $V_{Mn+1} = \begin{bmatrix} V_{Mx} \\ V_{My} \\ V_{Mz} \end{bmatrix}$, and the offset value = R.

Let the vector obtained by $(V_{Mn+1} \times V_{Tn}) \times V_{Tn}$ be V.

$$V = \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} V_{Tz}(V_{Mz}V_{Tx}-V_{Mx}V_{Tz}) - V_{Ty}(V_{Mx}V_{Ty}-V_{My}V_{Tx}) \\ V_{Tx}(V_{Mx}V_{Ty}-V_{My}V_{Tx}) - V_{Tz}(V_{My}V_{Tz}-V_{Mz}V_{Ty}) \\ V_{Ty}(V_{My}V_{Tz}-V_{Mz}V_{Ty}) - V_{Tx}(V_{Mz}V_{Tx}-V_{Mx}V_{Tz}) \end{bmatrix}$$

That is,

$0^\circ < \theta < 90^\circ$

$$V_{Cn} = \frac{R}{\sqrt{V_x^2+V_y^2+V_z^2}} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}$$

$90^\circ < \theta < 180^\circ$

$$V_{Cn} = \frac{-R}{\sqrt{V_x^2+V_y^2+V_z^2}} \begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix}$$

(5) Compensation performed when θ is approximately 0°, 90°, or 180°

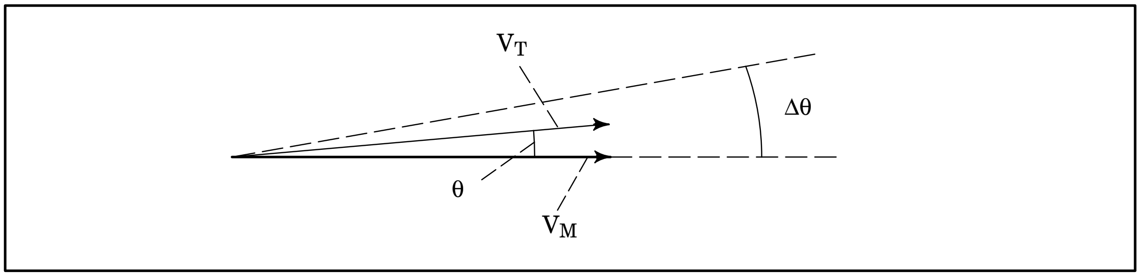
When the included angle θ between V_{Mn+1} and V_{Tn} is regarded as 0°, 180°, or 90°, the compensation vector is created in a different way. So, when creating an NC program, note the following points:

(1) Setting a variation range for regarding θ as 0 , 180 , or 90

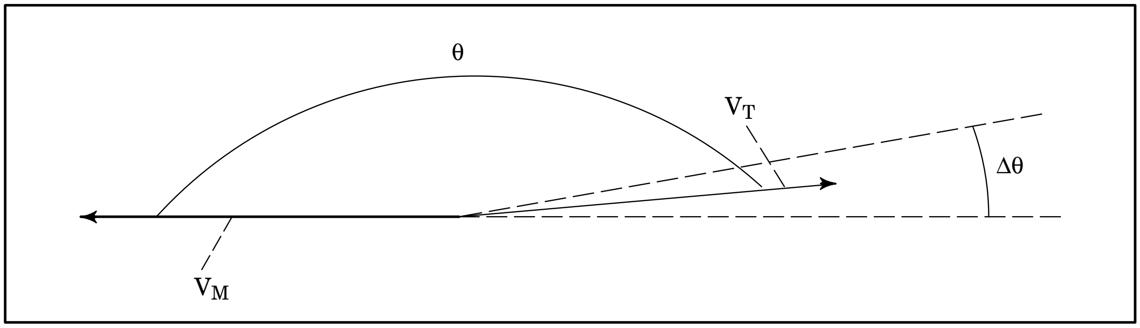
When the included angle (θ) between the tool vector (V_T) and the movement vector (V_M) becomes approximately 0°, 180°, or 90°, the system regards θ as 0°, 180°, or 90°, then creates a compensation vector which is different from the normal compensation vector. The variation range used for regarding θ as 0°, 180°, and 90°is or in parameter 6115 in 0.001–degree units.

For example, suppose that the angle set in parameter 6115 is Δθ.

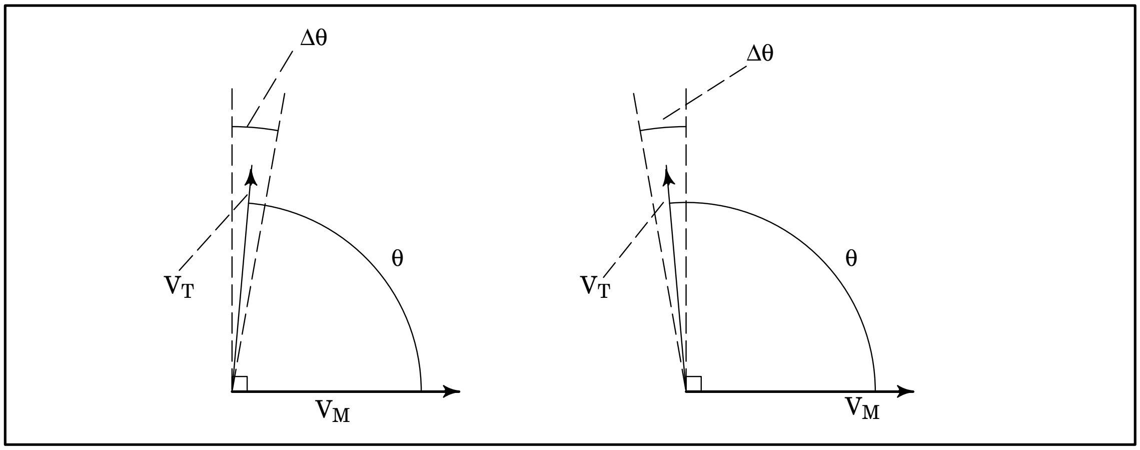
· If 0 <θ< Δθ, θ is regarded as 0°.



· If (180 – Δθ) <θ< 180, θ is regarded as 180°.



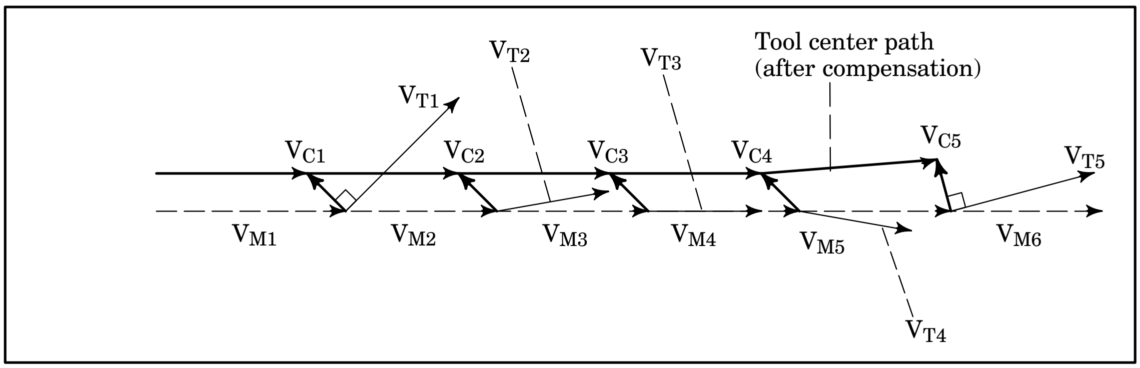
· If (90 – Δθ) <θ< (90 +Δθ),θ is regarded as 90°.



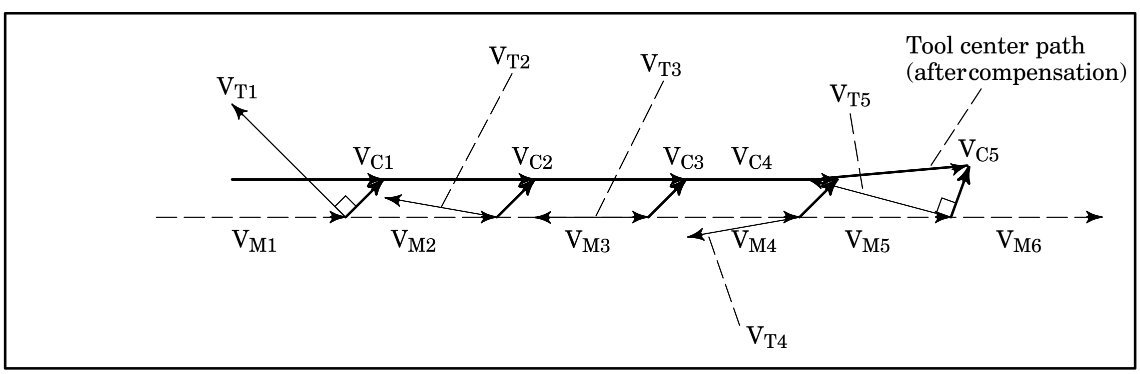
(2) Compensation vector when θ is regarded as 0°or 180°

If θ is regarded as 0°or 180°when G41.3 is specified to start leading edge compensation, alarm PS998 ”G41.3 ILLEGAL START UP” is issued. This means that the tool vector of the current block and the movement vector of the next block must not point in the same direction or in opposite directions at start–up.

The previously created compensation vector is maintained other than at start–up at all times.



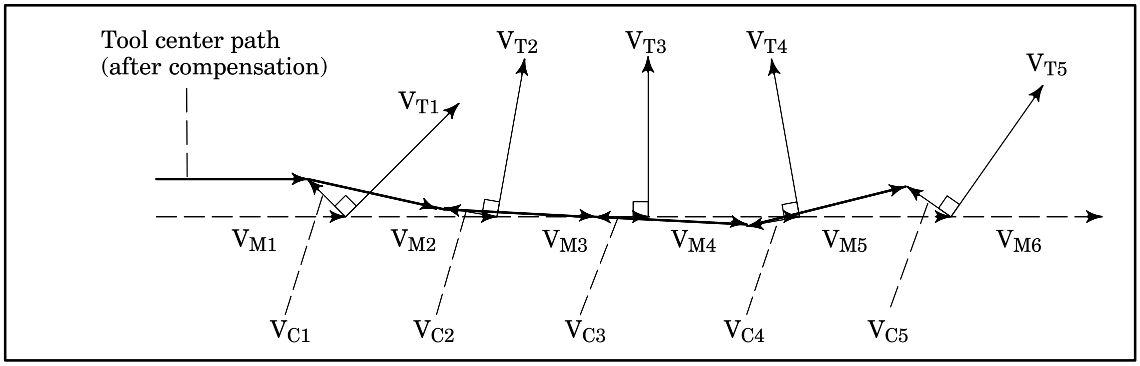
(If the included angles between V_{T2} and V_{M3}, V_{T3} and V_{M4}, and V_{T4} and V_{M5} are regarded as 0°, the compensation vector V_{C1} of block 1 is maintained as the compensation vectors V_{C2}, V_{C3}, and V_{C4}, of blocks 2, 3, and 4, respectively.)



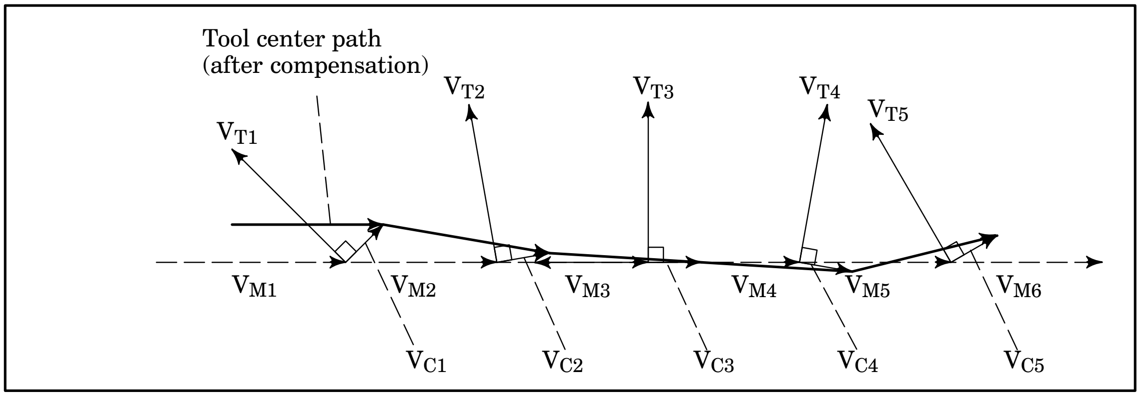
(If the included angles between V_{T2} and V_{M3}, V_{T3} and V_{M4}, and V_{T4} and V_{M5} are regarded as 180 , the compensation vector V_{C1} of block 1 is maintained as the compensation vectors V_{C2}, V_{C3}, and V_{C4} of blocks 2, 3, and 4, respectively.)

(3) Compensation vector when θ is regarded as 90°

If the previous compensation vector (V_{Cn-1}) points in the opposite direction (($V_{Mn} \times V_{Tn-1}$) $\times$ V_{Tn-1} direction) to V_{Mn} with respect to V_{Tn-1}, the current compensation vector (V_{Cn}) is created so it points in the ($V_{Mn+1} \times V_{Tn}$) $\times$ V_{Tn} direction. If the previous compensation vector (V_{Cn-1}) points in the same direction (($-V_{Mn} \times V_{Tn-1}$) x V_{Tn-1} direction) as V_{Mn} with respect to V_{Tn-1} , the current compensation vector (V_{Cn}) is created so it points in the $-(V_{Mn+1} \times V_{Tn}) \times V_{Tn}$ direction.



(If the included angles between V_{T2} and V_{M3}, V_{T3} and VM4, and V_{T4} and V_{M5} are regarded as 90°, like the compensation vector V_{C1} of block 1, the compensation vectors V_{C2}, V_{C3}, and V_{C4} of blocks 2, 3, and 4 are created so they point in the opposite direction ($V_{Mn+1} \times V_{Tn}$) $\times$ V_{Tn} direction) to V_{Mn+1} with respect to V_{Tn}.)



(If the included angles between V_{T2} and V_{M3}, V_{T3} and V_{M4}, and V_{T4} and V_{M5} are regarded as 90°, like the compensation vector V_{C1} of block 1, the compensation vectors V_{C2}, V_{C3}, and V_{C4} of blocks 2, 3, and 4 are created so they point in the $-(V_{Mn+1} \times V_{Tn}) \times V_{Tn}$ direction (as mentioned above).)

15.14.3 Notes

(1) G41.2, G42.2, G41.3, and G40 are continuous–state G codes that belong to the same group. Therefore, the G41.2, G42.2, and G41.3 modes cannot exist at the same time.

(2) Specify a canned cycle command and reference position return command in the compensation cancel mode (G40).

(3) In the compensation mode, never successively specify two or more blocks that involve no movement. Blocks involving no movement include:

- M05 ; . . . . M code output
- S21 ; . . . . S code output
- G04 X1000 ; . . . . Dwell
- G22 X100000 ; . . . . . . Machining area setting
- G10 P01 R100 ; . . . . . Offset value setting
- G90 ;, O10 ;, N20 ; . . Block containing no move command
- Blocks regarded as involving no movement according to parameter 6114 (for tool side compensation only)

(4) Resetting the system in a compensation mode (G41.2, G42.2, or G41.3) always results in the cancellation mode (G40).

15.15 Designation Direction Tool Lenght Compensation

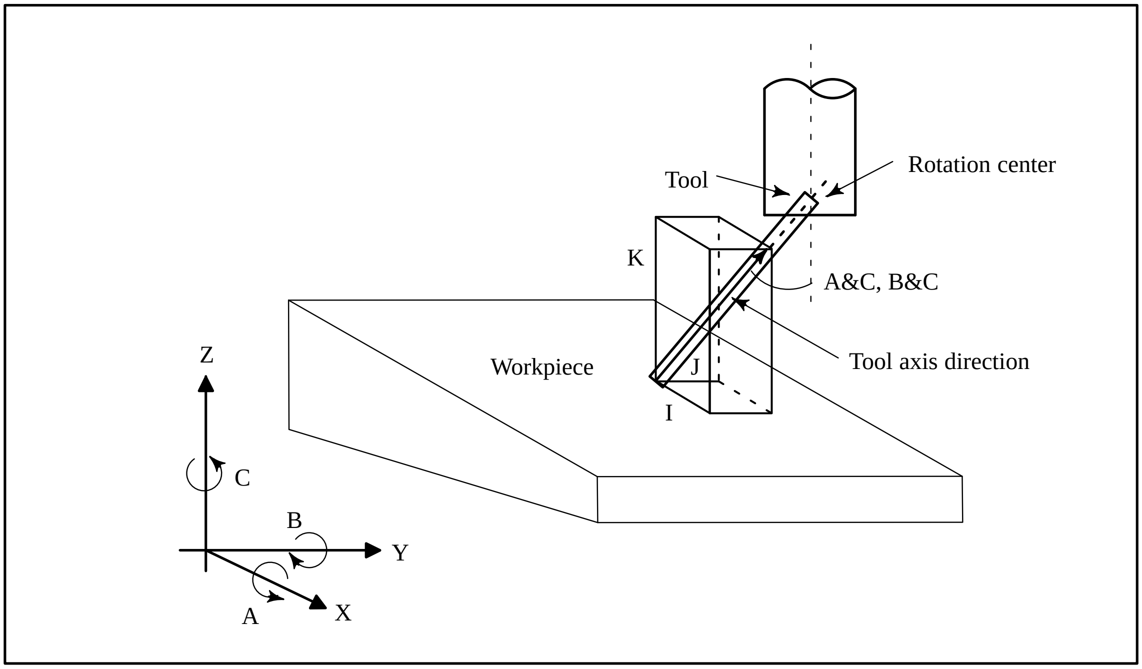
In a five–axis machine tool having three basic axes and two rotation axes for turning the tool, tool length compensation can be applied in the direction of the tool axis.

The tool axis direction is specified with I, J, and K; a move command for the rotation axes is not specified directly. When I, J, and K are specified in designation direction tool length compensation mode, the following operation is performed automatically:

1. The basic three axes operate so that tool length compensation is applied using the offset specified by the D code in the direction specified by I, J, and K. (Compensation is applied in the same way as for the three–dimensional tool compensation function).
2. The two rotation axes operate so that the tool axis is oriented in the direction specified by I, J, and K. (This specifications manual explains this operation.)

Machine configuration example

A and C axes or B and C axes (the tool axis corresponds to the Z–axis.)



15.15.1 Command Format

1) Starting designation direction tool length compensation

When the following command is executed in cutter compensation cancel mode (G40 mode), the system enters designation direction tool length compensation mode:

G41 Xp_ Yp_ Zp_ I_ J_ K_ D_ ;
Xp : X–axis or an axis parallel to the X–axis
Yp : Y–axis or an axis parallel to the Y–axis
Zp : Z–axis or an axis parallel to the Z–axis

The three–dimensional space in which designation direction tool length compensation is executed is determined by the axis addresses specified in the startup block containing G41. If Xp, Yp, or Zp is omitted, the X–, Y–, or Z–axis is assumed as the axis to be used for designation direction tool length compensation, respectively.

The linear axes and rotation axes move so that the tool rotation center is shifted by the offset specified by the D code, in the direction specified by I, J, and K.

In designation direction tool length compensation mode, movement along axes other than the rotation axes is performed at a specified feedrate. Movement about the rotation axes can also be included by setting bit 2 of parameter 7711.

Example When X and U, Y and V, and Z and W are parallel

G41 X_ I_J_ K_ ; X–Y–Z space
G41 U_ V_ Z_ I_J_ K_ ; U–V–Z space
G41 W_ I_J_ K_ ; X–Y–W space

WARNING 1 Bit 0 of parameter 7711 specifies whether the startup command is to be used to perform the designation direction tool length compensation or to set three–dimensional tool compensation mode.

WARNING 2 In the startup block, I, J, and K must be specified. If I, J, or K is omitted, ordinary cutter compensation is applied.

WARNING 3 An offset number is selected using address D. Then, the offset corresponding to the selected offset number is regarded as the distance ι from the tool tip to the tool rotation center.

WARNING 4 Generally, the G41 command is used for startup. Note that when the G42 command is used for startup, an offset is applied in the opposite direction to that applied by G41.

NOTE In designation direction tool length compensation mode, never specify a command for the rotation axes. If such a command is issued, alarm PS805 is output.

2) Canceling designation direction tool length compensation

When the following command is executed in designation direction tool length compensation mode, the system enters designation direction tool length compensation cancel mode:

a) When designation direction tool length compensation mode is canceled while movement is being performed (example when the target rotation axes for the designation direction tool length compensation function are the B– and C–axes)

G40 X_ Y_ Z_ B_ C_ ;
or
D00X_ Y_ Z_ B_ C_ ;

b) When only the compensation vector amount is canceled

G40 ;
or
D00 ;

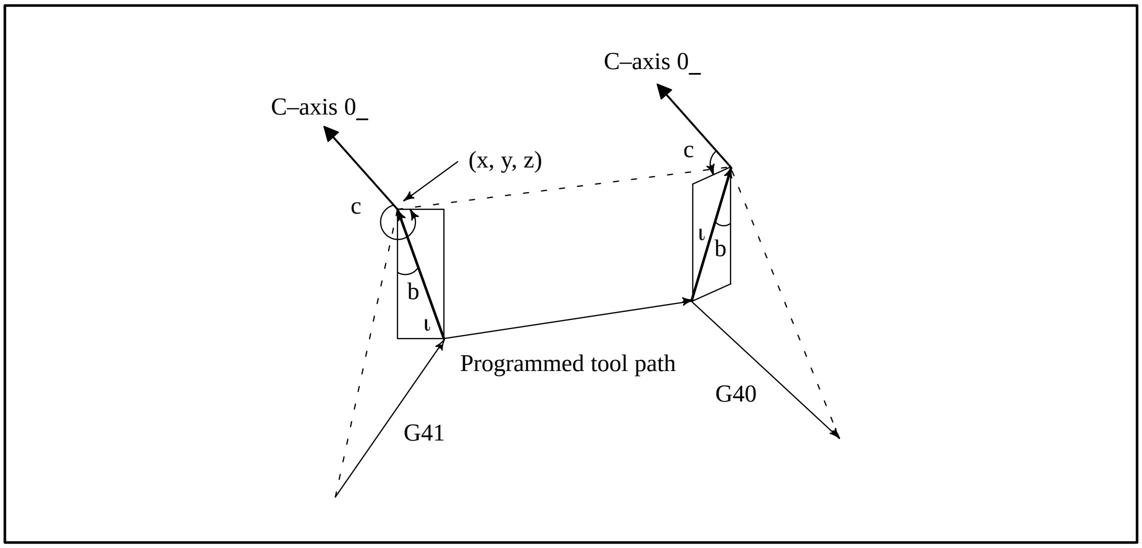
In this case, target rotation axes B– and C–axes for designation direction tool length compensation are not moved.

Sample program in which the target rotation axes for designation direction tool length compensation are the B– and C–axes

```
G92  X_  Y_  Z_  B_  C_  ;   ......  Specify a coordinate system
G41  X_  Y_  Z_  I_J_  K_  D_  ;
              .......  Startup (enter designation direction tool length compensation mode)
X_  Y_  Z_  I_J_  K_  ;
  .
  .
  .
G40  X_  Y_  Z_  B_  C_  ;
              .......  Cancel (cancel designation direction tool length compensation mode)
```

15.15.2 Movement in Each Block

This section explains the end point of each block in designation direction tool length compensation mode.



(1) When positioning (G00) or linear interpolation (G01) is performed, the tool rotation center is obtained from the following expressions. (The expressions for b and c are used for a machine whose tool axis corresponds to the Z–axis, and whose rotation axes correspond to the B– and C–axes.)

$$x = X + \frac{I}{\sqrt{I^2 + J^2 + K^2}} \times \iota$$

$$y = Y + \frac{J}{\sqrt{I^2 + J^2 + K^2}} \times \iota$$

$$z = Z + \frac{K}{\sqrt{I^2 + J^2 + K^2}} \times \iota$$

$$b = \tan^{-1}\frac{\sqrt{I^2 + J^2}}{K}$$

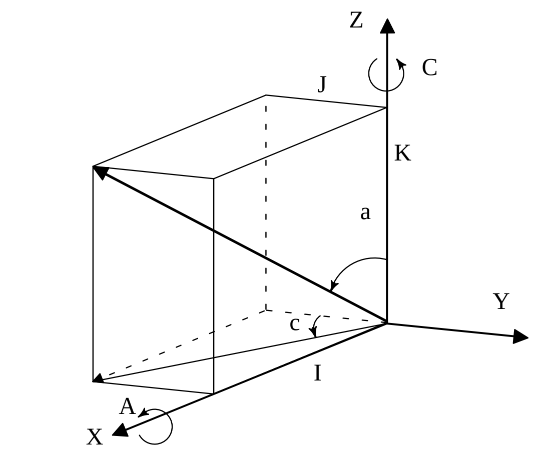
$$c = \tan^{-1}\frac{J}{I}$$

x, y, z : Tool rotation center position (absolute coordinates)
b, c : Rotation axes (absolute coordinates)
X, Y, Z : Tool tip position (absolute coordinates)
I, J, K : Specified I, J, and K
ι : Distance from the tool tip to tool rotation center

If none of I, J, and K are specified in a block specified in designation direction tool length compensation mode, the vector generated in the previous block is maintained. If any of I, J, or K is omitted, the omitted I, J, or K is assumed to be 0.

In a machine configuration that can use designation direction tool length compensation, the angles of the rotation axes are calculated as follows:

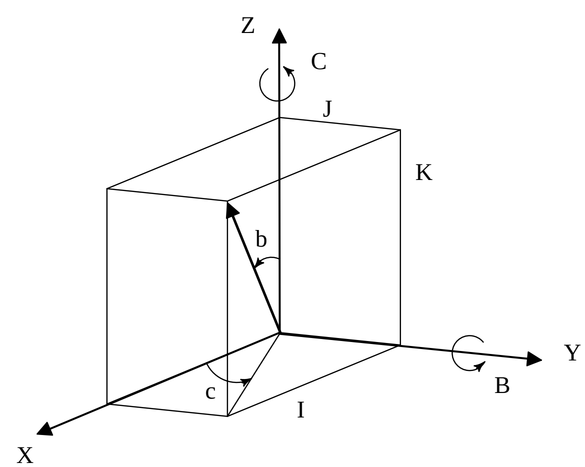
(a) When the rotation axes are the A– and C–axes, and the tool axis is the Z–axis



$$a = \tan^{-1}\frac{\sqrt{I^2 + J^2}}{K}$$

$$c = \tan^{-1}\frac{I}{-J}$$

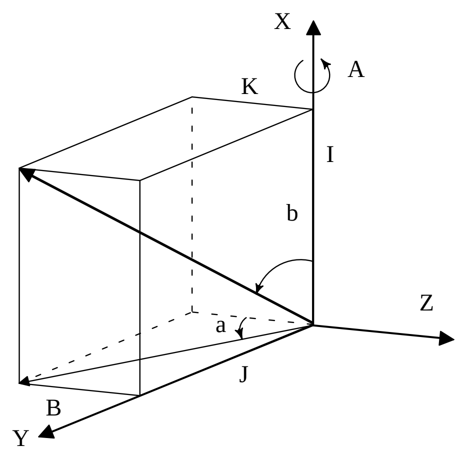
(b) When the rotation axes are the B– and C–axes, and the tool axis is the Z–axis



$$b = \tan^{-1}\frac{\sqrt{I^2 + J^2}}{K}$$

$$c = \tan^{-1}\frac{J}{I}$$

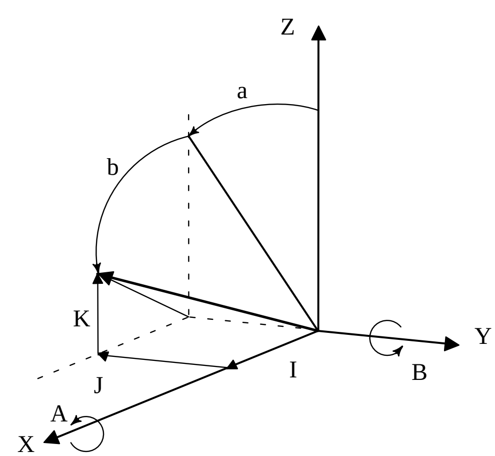
(c) When the rotation axes are the A– and B–axes, and the tool axis is the X–axis



$$b = \tan^{-1}\frac{J}{-K}$$

$$b = \tan^{-1}\frac{\sqrt{J^2 + K^2}}{I}$$

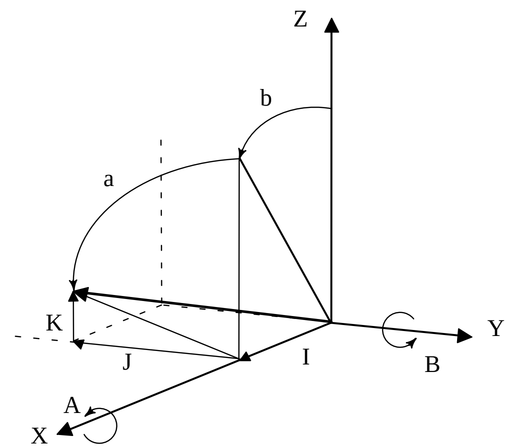
(d) When the rotation axes are the A– and B–axes, and the tool axis is the Z–axis (master axis: B–axis)



$$a = \tan^{-1}\frac{-J}{\sqrt{I^2 + K^2}}$$

$$b = \tan^{-1}\frac{I}{K}$$

(e) When the rotation axes are the A– and B–axes, and the tool axis is the Z–axis (master axis: A–axis)



$$b = \tan^{-1}\frac{-J}{K}$$

$$b = \tan^{-1}\frac{I}{\sqrt{J^2 + K^2}}$$

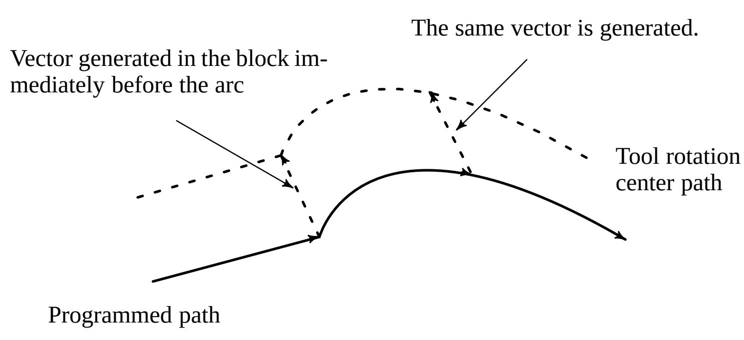
WARNING 1 From the current position to the point obtained above, the tool is moved along the five axes simultaneously.

WARNING 2 In designation direction tool length compensation mode, coordinate system rotation and scaling are performed for the tool tip position.

WARNING 3 If the absolute value of the C–axis travel (γ) exceeds 180_, a short–cut is taken as shown below, such that the actual amount of travel becomes γ'.

γ 180 : γ' = –360_ γ
γ –180 : γ' = 360_ γ

(2) When circular interpolation or helical interpolation (G02 or G03) is performed, I, J, and K have no meaning for the designation direction tool length compensation function. In this case, the designation direction tool length compensation vector generated in the previous block is maintained as is.



(3) Before specifying the following commands, cancel designation direction tool length compensation mode:

– Reference position return check (G27)
– Automatic return to reference position (G28)
– 2nd reference position return (G30)
– Automatic return from reference position (G29)

15.15.3 Relationship with Other Offset Functions

(1) Tool length offset

For a programmed tool path, the tool rotation center is obtained. Tool length offset is applied to the tool rotation center.

(2) Tool offset

In designation direction tool length compensation mode, tool offset cannot be specified.

(3) Cutter compensation C

When addresses I, J, and K are all specified in a startup block, designation direction tool length compensation mode is entered. When one or more addresses are omitted, cutter compensation C is performed.

This means that the designation direction tool length compensation function cannot be specified in cutter compensation C mode. Also, in designation direction tool length compensation mode, cutter compensation C cannot be specified.

(4) Tool length compensation in tool axis direction

In designation direction tool length compensation mode, tool length compensation in the tool axis direction cannot be specified.

15.15.4 Specifying Offset Vector Components

When the following parameter is set, x, y, and z can be obtained from the following expressions. Therefore, in addition to the offset vector direction, the components of the offset vector can be programmed using I, J, and K.

$$x = X + \frac{I}{S} \times 1$$

$$y = Y + \frac{J}{S} \times 1$$

$$z = Z + \frac{K}{S} \times 1$$

S : Parameter setting (No. 6011)

If the parameter (No. 6011) is set to 0, S is expressed as follows:

$$S = \sqrt{I^2 + J^2 + K^2}$$

Data No.	Data
6011	Denominator constant in 3–dimensional tool compensation

Setting input

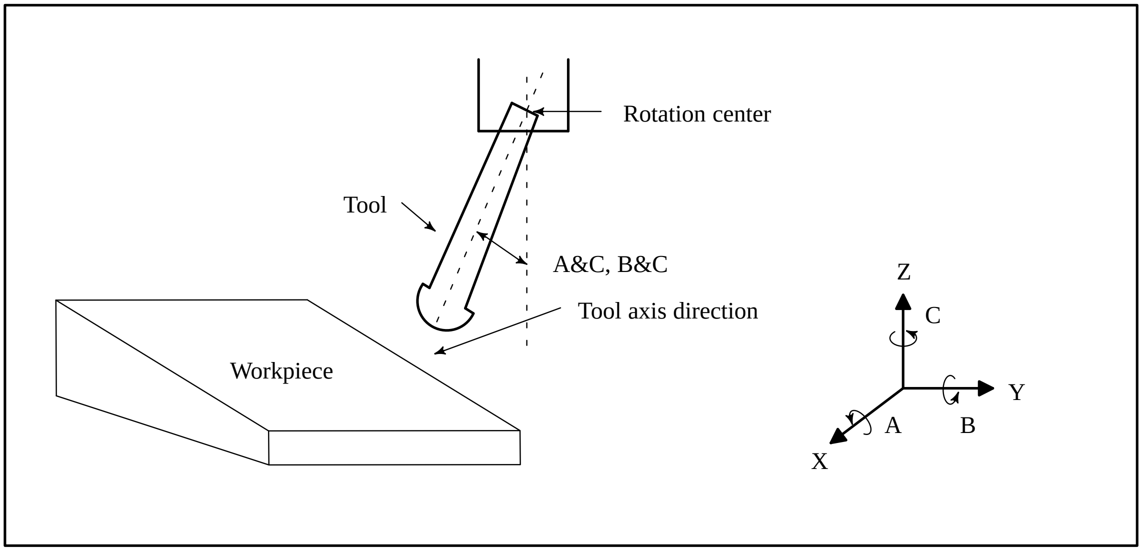
Valid range : 0 to ±999999999

Units : Minimum input increment

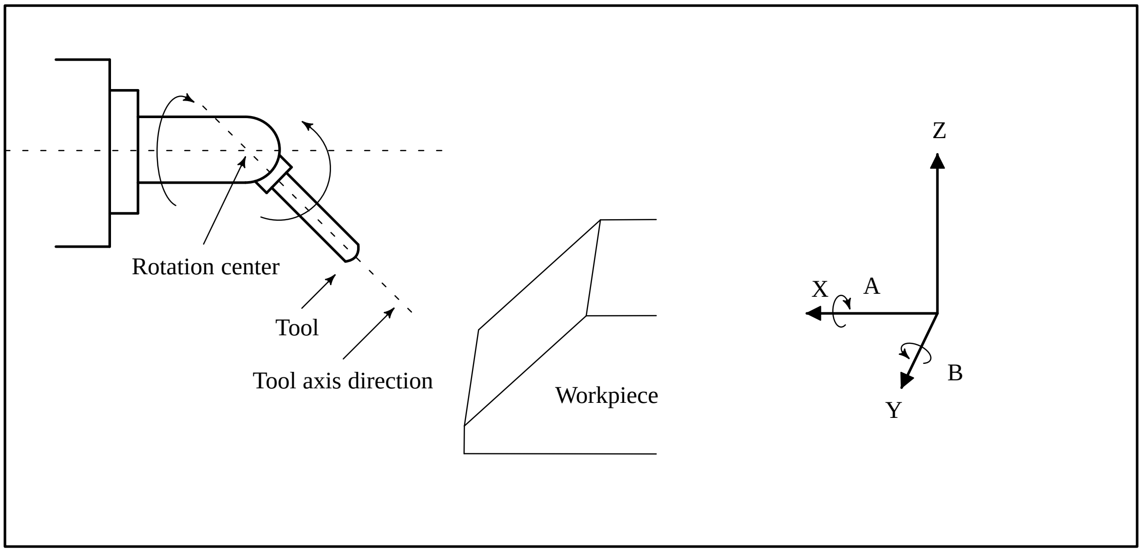
15.15.5 Notes

1) The designation direction tool length compensation function is supported only when one of the following axis configurations is used:

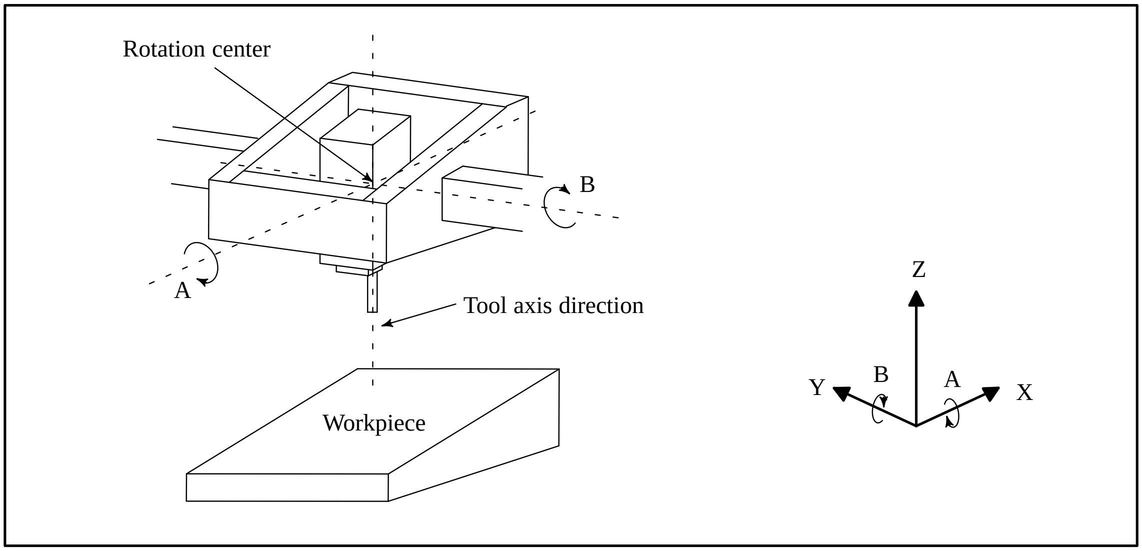
(a) A– and C–axes or B– and C–axes (when the tool axis is the Z–axis)



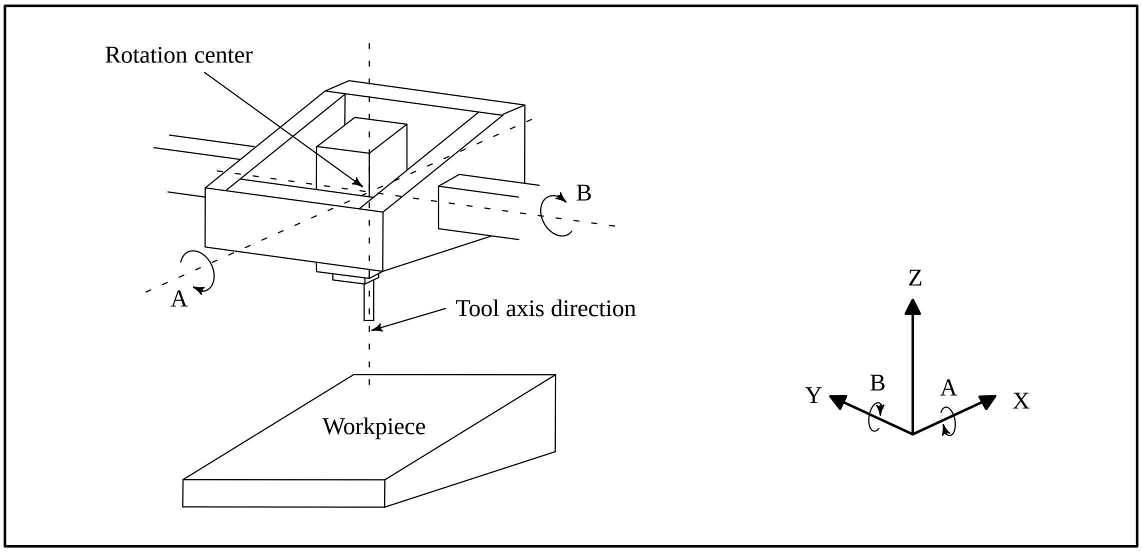
(b) A– and B–axes (when the tool axis is the X–axis)



(c) A– and B–axes (when the tool axis is the Z–axis, and the B–axis is the master axis)



(d) A– and B–axes (when the tool axis is the Z–axis, and the A–axis is the master axis)



16.MEASUREMENT FUNCTIONS

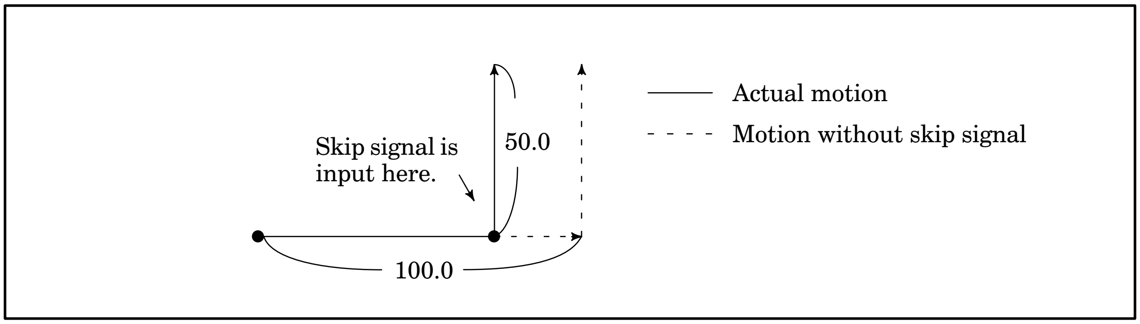
16.1 Skip Function (G31)

Linear interpolation can be commanded by specifying axial move following the G31 command, like G01. If an external skip signal is input during the execution of this command, execution of the command is interrupted and the next block is executed. G31 is a one–shot command and is effective only in the block in which it is specified. The motion after the input of the skip signal depends on whether the next block is an incremental or absolute command.

1) When the next block is an incremental command:

   The motion in the next block is performed incrementally from the position at which it is interrupted by the skip signal.

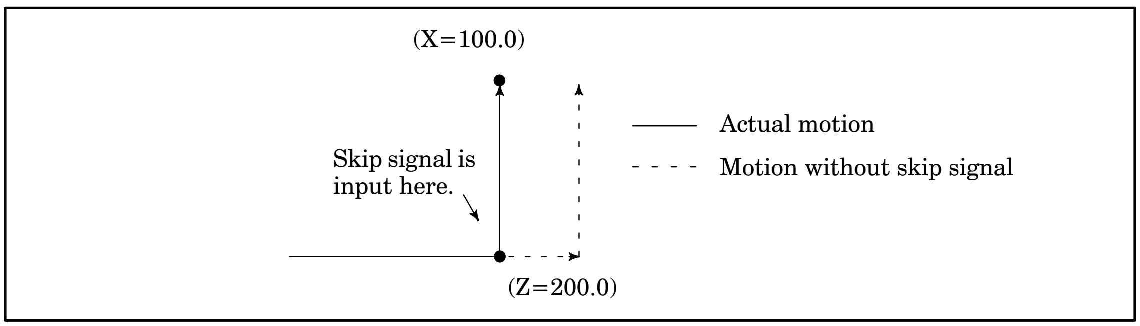
   Example G31 G91 X100.0 F100;
   Y 50.0;



2) When the next block is an absolute command with only one axis specified:

   The specified axis moves to the commanded position. The axis which was not specified remains at the position when the skip signal is input.

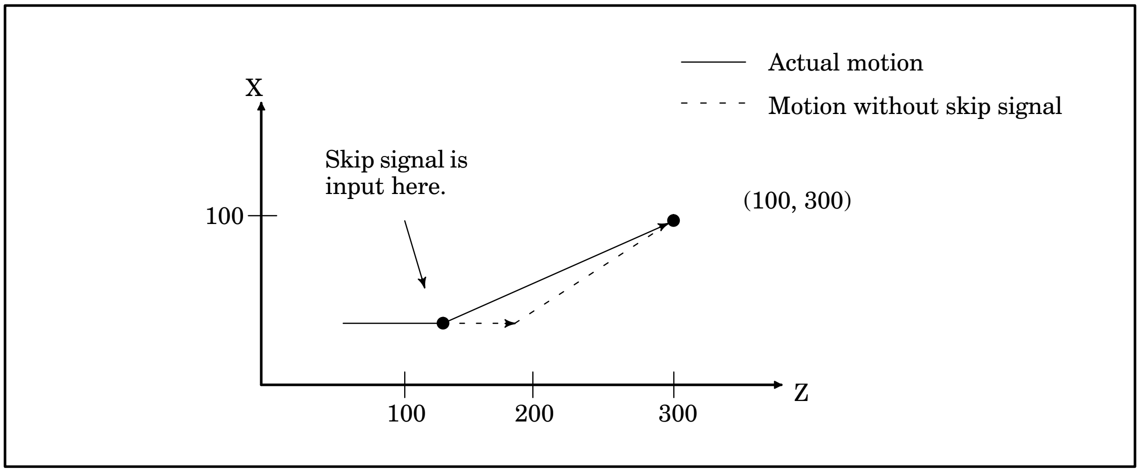
   Example G31 G90 X200.00 F100;
   Y100.0;



3) When the next block is an absolute command with two axes specified:

The tool moves to the position specified by next block even if the skip signal is input.

Example G31 G90 X200.0 F100;
X100.0 Z300.0;



The feedrate of the block in which G31 is specified may be one of the following, depending on the setting of the SKF bit in parameter No. 1400:

(a) Feedrate specified by F (The feedrate may be specified either in or before the block containing the G31 code.)

(b) Feedrate set in parameter No. 1428

The coordinate values when the skip signal is turned on can be used in a custom macro because they are stored in the custom macro system variable #5061 to #5066, as follows:

#5061 . . . X axis coordinate value

#5062 . . . Y axis coordinate value

#5063 . . . Z axis coordinate value

#5064 . . . 4th axis coordinate value

#5065 . . . 5th axis coordinate value

#5066 . . . 6th axis coordinate value

The skip function can be used when the move end distance is not known as in the standard–size feed in grinders.

WARNING If G31 command is issued while cutter compensation C is applied, an P/S alarm of No.035 is displayed. Cancel the cutter compensation with the G40 command before the G31 command is specified.

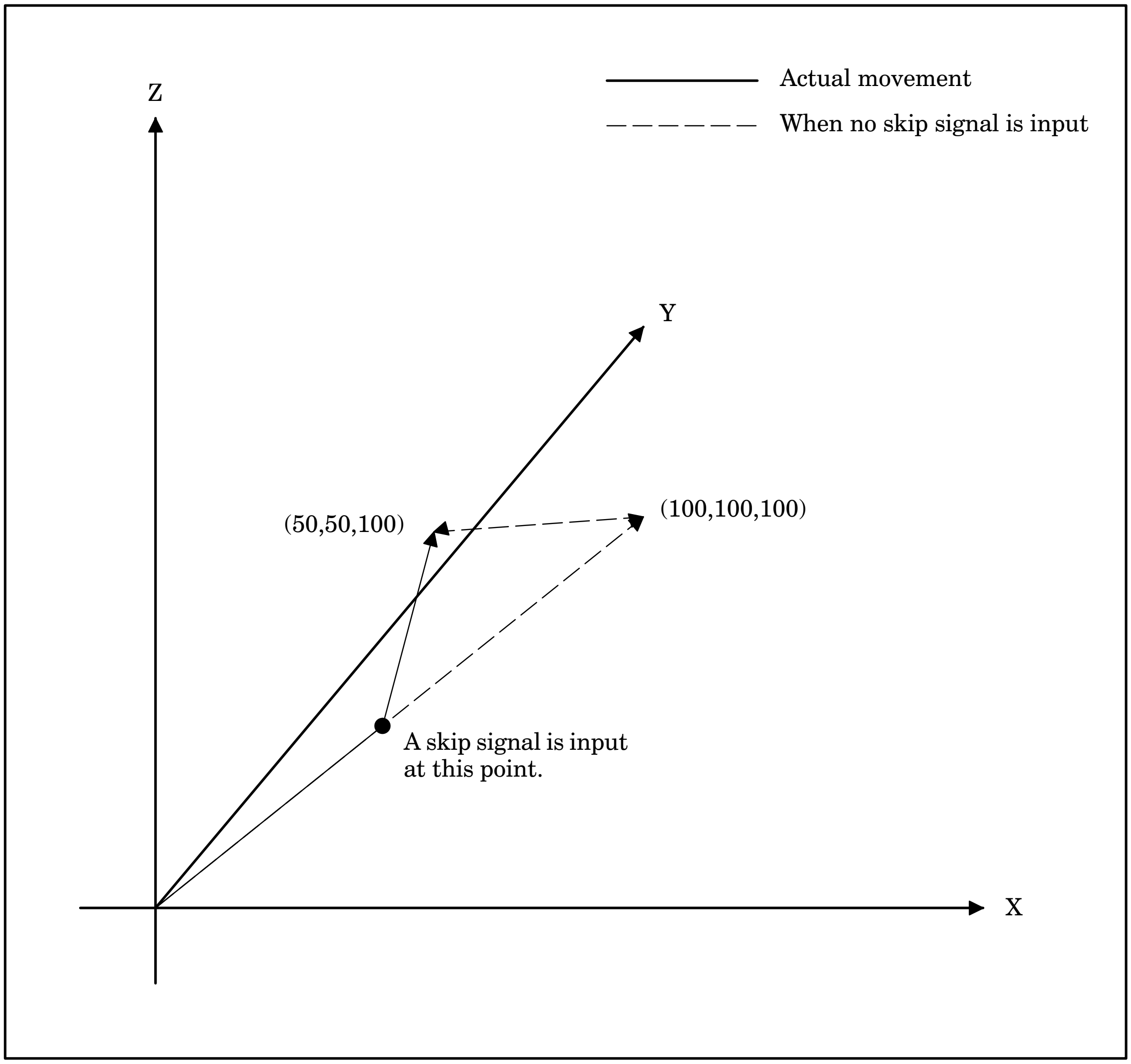
NOTE Override Dry run, and automatic acceleration/deceleration are ineffective for skip function. However, it is possible to make effective them by parameter setting (SKPF, parameter No. 0015).

16.2 Skipping the Commands for Several Axes

Move commands can be specified for several axes at one time in a G31 block. If an external skip signal is input during such commands, the command is canceled for all specified axes and the next block is executed.

The position for each specified axis where a skip signal is input is set in the macro variable for the axis (#5061 to #5066).

Example G31 G90 X100.0 Y100.0 Z100.0 ;
X50.0 Y50.0 Z100.0 ;



16.3 Multistage Skip (G31.1 to G31.4)

If G31.1, G31.2, G31.3, or G31.4 is specified in a block, the coordinates when the corresponding skip signal goes on are stored in custom macro variables. At the same time, the remaining traveled distance in that block is skipped. If the G04 command (dwell) is specified in a block, it is possible to set a parameter so that when a skip signal goes on, the remaining dwell is skipped.

The correspondence between the three skip signals and G codes is selected by parameters. The signals can correspond to G codes on a one–to–one basis, or one skip signal can correspond to more than one G code.

Example In machining such as grinding, the end point of machining is not programmed. Instead, a signal such as a machining condition signal sent from the machine specifies skipping, then execution proceeds to the next block.

Suppose that the following procedure is used for machining:

(1) Feed at a feedrate of 10 mm/min until machining condition 1 is met.

(2) Feed at a feedrate of 3 mm/min until machining condition 2 is met.

(3) Dwell until machining condition 3 is met.

Suppose the machining conditions, skip signals, and G codes correspond to each another as follows:

Machining condition 1 – skip signal 1 – G31.1

Machining condition 2 – skip signal 2 – G31.1, G31.2

Machining condition 3 – skip signal 3 – G31.1, G31.2, G04

Then the following program fragment is coded:

```
N1G31.1X100.0F10.0 ; (feed)
N2G31.2X100.0F3.0 ; (feed)
N3 G04 X100.0 : (dwell)
```

If machining condition 2 is met from the beginning, execution starts with the N3 block (dwell).

Delay and variation in the detection of skip signal input is 0 ms to 2 ms in the NC only (independent of the PC). In high–speed skip signal input, this value drops to 0.1 ms or less, resulting in measurement with high accuracy. For details, refer to the manual provided by the machine tool builder.

16.4 Automatic Tool Length Measurement (G37)

G37α ;

By issuing G37 the tool starts moving to the measurement position and keeps on moving till the measuring position reach signal from the measurement device is output. Movement of the tool is stopped when the tool head reaches the measurement position.

Difference between coordinate values when tool has reached the measurement position and coordinate value commanded by G37 is added to the tool length compensation amount currently used.

1) Coordinate system

When moving the tool to a position for measurement, the coordinate system must be set in advance. (the workpiece coordinate system for programming is used in common).

2) Moving to measurement position

A movement to a measurement position is performed by specifying as follows in the MDI, or memory mode.
G37 α ....;

α; an axis among X, Y, or Z

In this case, the measurement position should be α – (absolute command).

Execution of this command moves the tool at the rapid traverse rate toward the measurement position, reduces the feedrate halfway, then continuous to move it until the approach end signal from the measuring instrument is issued. When the tool tip reaches the measurement position, the measuring instrument sends an approach end signal to the CNC which stops the tool.

NOTE	Approach end signal 1 is used for the measurement position signal.

3) Offset

The current tool offset amount is further offset by the difference between the coordinate value (α) when the tool has reached the measurement position and the value of xa, ya, or za specified under G37 command.

Offset amount x= current offset amount x + (α – xa)

Offset amount z= current offset amount z + (α – za)

Offset amount y= current offset amount y + (α – ya)

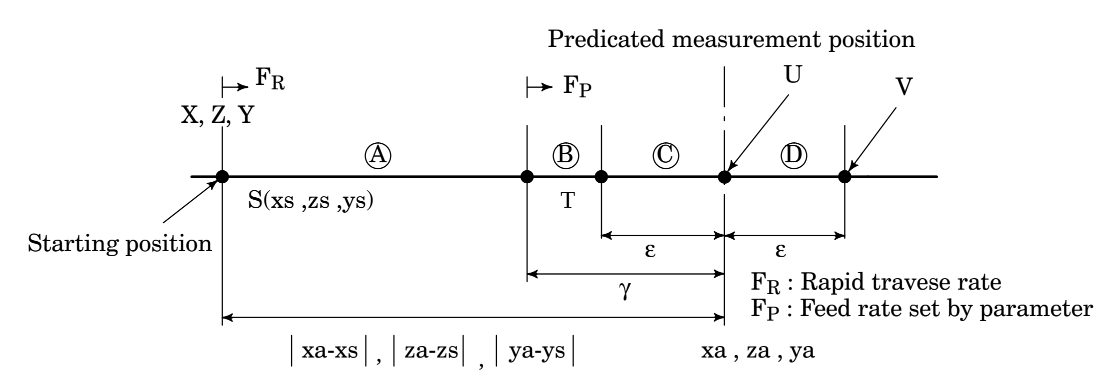
Xa : Programmed measurement point along the X–axis

Ya : Programmed measurement point along the Y–axis

Za : Programmed measurement point along the Z–axis

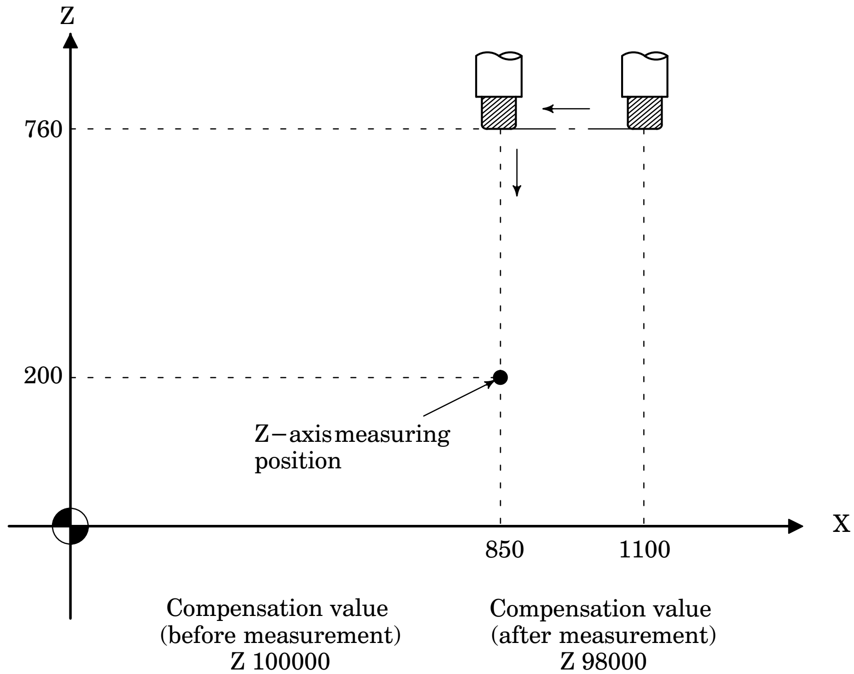
These offset values can be manually changed from MDI.

4) Feedrate and alarm



The tool, when moving from the starting position toward the measurement position predicated by xa or za or ya with G37, is fed at the rapid traverse rate across area A. Then the tool stops at point T (xa – γx or za – γz or ya – γy) and moves at the measurement feedrate set by parameter across areas B, C, and D. If the approach end signal turns on during movement across area B, an alarm is generated. If the approach end signal does not turn on before point V, the tool stops at point V and alarm is generated.

Program Example



NOTE 1	In case of compensation memory A	Change of compensation value
NOTE 2	In case of compensation memory B	Change of wear compensation value
NOTE 3	In case of compensation memory C	Change of wear compensation value for H code

G92 Z760.0 X1100.0 ;

Programming of absolute zero point (Coordinate system setting)

G90 G43 G00 X850.0 H1 ;

Select offset number 1

Moves some distance away from the Z measurement position

Moves to the measurement position If the tool has reached the measurement position at Z198.0; since the correct measurement position is 200 mm, the offset amount is altered by 198.0 – 200.0 = 2.0 mm.

G00 Z204.0 ;

Retracts a little along the Z axis

WARNING When a manual movement is inserted into a movement at a measurement feedrate, return the tool to the position before the inserted manual movement for restart.

NOTE 1 When an H code is specified in the same block as G37, an alarm is generated. Specify H code before the block of G37.

NOTE 2 Measurement speed, γ, and ε are set as parameters by machine tool builder. ε must be a positive number such that satisfies $\gamma > \varepsilon$.

NOTE 3 ε must always be an integer and satisfy the following condition : $\gamma > \varepsilon$.

NOTE 4 The approach end signal is monitored usually every 2 ms. The following measuring error is generated:

ERRmax. : Fm $\times$ 1/60 $\times$ Ts/1000

where

Ts: Sampling period,
For usual 2 (msec)

ERRmax. : maximum measuring error (mm)

Fm : measurement feedrate (mm/min.)

When Fm = 100 mm/min., for example, ERRmax. = 0.003mm

NOTE 5 The tool stops a maximum of 16 ms after the approach end signal is detected. But the value of the position when the approach end signal was detected (note the value when the tool stopped) is used for α and β (see (3)) to determine the offset amount. The overrun for 16 ms is:

$$\left\{\begin{array}{l}\text{Qmax.} = \text{Fm} \times 1/60 \times 16/1000 \\ \text{Qmax. : maximum overrun (mm)} \\ \text{Fm : measurement feedrate (mm/min.)}\end{array}\right.$$

NOTE 6 To use this function, always set EV0 of parameter 6000 to 0.
Parameters 7200, 7300, 7311, 7321, 7331, and 7600 are used as the parameters for automatic tool compensation. For details of parameters, see the table.

16.5 Torque Limit Skip Function

This function drives an axis with a torque limit applied to the feed motor, and carries out a skip operation when the torque limit is reached, for example, because the axis is pushed against a stop position.

16.5.1 Program

G31 P99 α [amount of movement] F [speed]; α : Axis address

G31 P98 α [amount of movement] F [speed]; α : Axis address

A cutting feed command similar to G01 can be implemented by issuing a move command after G31P99 (or G31P98) with a limit placed on the motor torque (using a command such as a torque limit in the PMC window). This move command can be used only for one axis at a time.

If a motor torque limit is reached or a skip signal (or high–speed skip signal) is issued during execution of the G31P99 command, the rest of the operation of the G31P99 command is canceled, and the next block is executed.

If a motor torque limit is reached during execution of the G31P98 command, the rest of the operation of the G31P98 command is canceled, and the next block is executed. (The SKIP signal does not affect the G31P98 operation.)

Both G31P99 and G31P98 commands are one–shot type. They are effective only in the block in which they are specified.

Example

```
O0001 ;
......
M□□ ;                  ← A torque limit is specified by the PMC via a window.
......
G31 P99 V200. F100 ;   ← Torque limit skip command
......
G01 X100. F500 ;       ← Move command with a torque limit placed.
......
M△△ ;                  ← The torque limit is canceled by the PMC.
......
M30 ;
%
```

16.5.2 Operation

(1) If a torque limit is not specified before G31P99/P98 (if the torque limit override value is 0% or 100%), the move command is executed without carrying out a skip operation based on the torque limit.

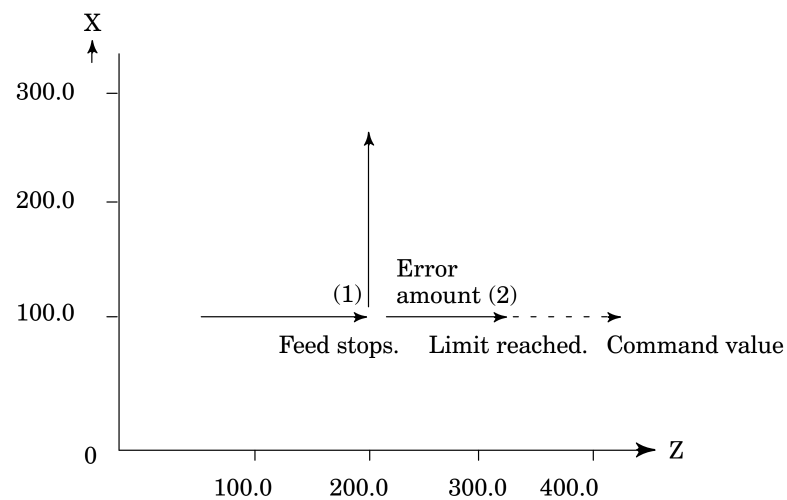
(2) A skip operation occurs even if a skip signal is input during execution of the G31P99 command.

(3) A skip operation does not occur even if a skip signal is input during execution of the G31P98 command.

(4) When G31P99/P98 is specified, the coordinates of the axis are assigned to the custom macro system variable (data No. 5061, etc.) when a skip operation ends. (Bit 1 of parameter No. 7203 can be used to switch the feed stop position and limit reached position.)

Example M□□ ;
N01 G31 P99 Z400.0 F100 ;
N02 G01 X300.0 F400 ;
M△△ ;

While the above program is being executed, the axis stops at Z200.0 in block N01, and a torque limit reached signal or skip signal is received after a while.



16.5.3 Notes

WARNING 1 Specify a torque limit before G31P99/P98. If G31P99/P98 is executed without specifying a torque limit, the move command will be executed without carrying out a skip operation.

WARNING 2 Do not specify G31P99/P98 for an axis under synchronous control such as simple synchronous control.

WARNING 3 Before specifying G31P99/P98, cancel tool–tip radius compensation (for model T or TT) or cutter compensation C (for model M) using G40 if such compensation is effective.

NOTE 1 Do not use G31P99/P98 in consecutive blocks.

NOTE 2 If no axis command or axis commands for more than one axis are issued in a block that contains G31P99/P98, alarm PS150, "G31 FORMAT ERROR" is issued.

NOTE 3 The torque limit reached signal (TRQLn) is always output regardless of this optional function.

NOTE 4 Generally, for any skip function, the higher the move speed is, the larger is the difference between the position assigned to a system variable and the actual skip position. Moreover, if the speed is changed during a move operation, the difference becomes larger. So, do not use an override.

17.CUSTOM MACRO

Request:
Machine tool builders: You are requested to attach your custom macro program tape or program list to the CNC unit without fail.
If it is necessary to replace part program storage memory due to a failure, FANUC servicemen or end user operators in charge of maintenance should know the contents of your custom macro for the purpose of repairing the trouble immediately.

A function covering a group of instructions is stored in memory the same as a subprogram. The stored function is represented by one instruction, so that only the representative instruction need be specified to execute the function. This group of registered instructions is called a "custom macro body" and the representative instruction is called a "custom macro instruction". The custom macro body may simply be called a macro. And the custom macro instruction may be called a macro call command.

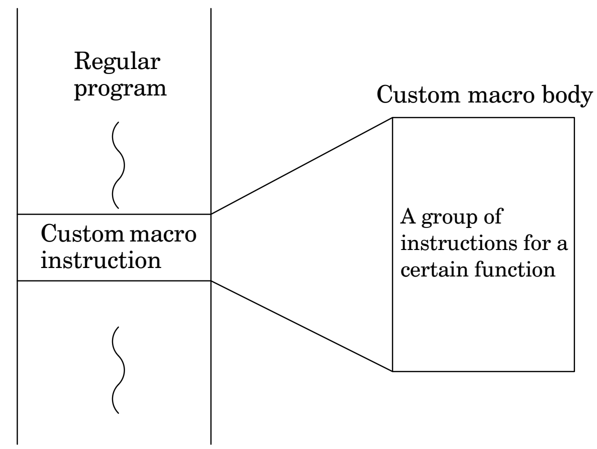


Fig. 17.(a) Custom macro instruction and custom macro body

Programmers need only remember representative macro instructions without having to remember all the instructions in a custom macro body.

The three most significant points on custom macros are that variables can be used in the custom macro body, operations can be performed on variables and actual values can be assigned to the variables in custom macro instructions.

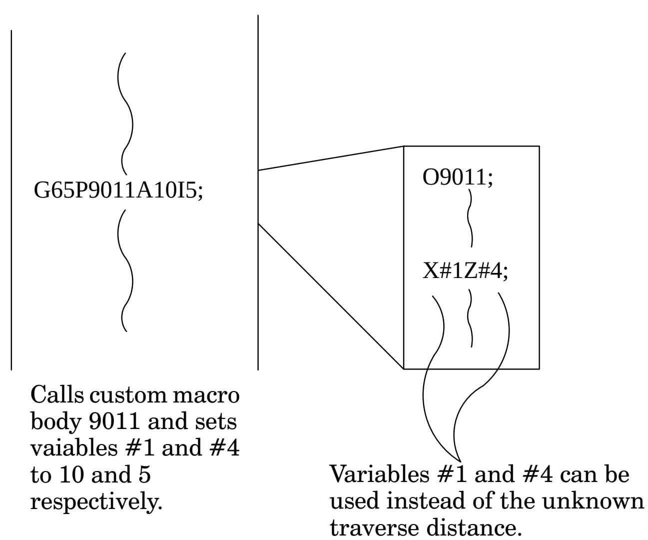
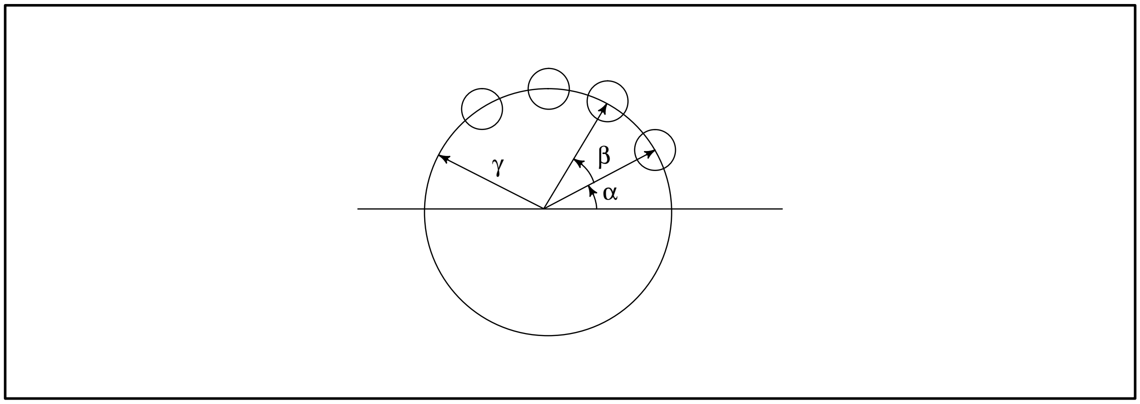


Fig. 17.(b) Custom macro instruction and variables

This means that a function of general use can be formed when programming a certain function as a custom macro. That is, programs can be written using variables for data that might change or be unknown. This can be further applied to group technology.

Similar workpieces can be collected as a group and a universal custom macro body can be programmed using variables applicable to each group. In this way, programming is not required for the workpieces in the group. The programmer only need to assign actual values to the variables.



Bolt hole circles as shown in the above figure can be made easily.

Once a custom macro body for the bolt hole circle is programmed and registered, the CNC can operate as if it has the bolt hole circle cutting function.

Programmers can use the bolt hole circle function by using the following command only:

(Example of calling bolt hole circle)

G65P p R r A α B β K k ;

P : Macro number of bolt hole circle

r : Radius

α : Start angle

β : Angle between circles

k : Number of circles

17.1 Macro Call Command (Custom Macro Command)

A macro can be called from a single block, or modally from each block in the call mode.

> **NOTE** In MDI mode, a macro call instruction can be executed only during an MEM operation with no nesting.

17.1.1 Simple calls

When the following command is executed, the custom macro body (for the specified block only) identified by P (program number) is called.
G65 P(program number) L(iteration times) <argument assignment>;
When it is necessary to transfer arguments to a custom macro body, the argument is specified by <argument assignment>. The following two types of <argument assignment> can be specified. The argument mentioned here is the actual numerical value assigned to a variable.

> **NOTE** G65 must be specified before arguments in the G65 block. The negative sign and the decimal point can be used regardless of addresses in <argument assignment >.

a) Argument assignment I

A____ B____ C____ D____ .....Z____

An argument can be assigned for all addresses except G, L, N, O, and P. Assignment need not be made in alphabetical order. Specification is made according to word address format. Addresses not required may be omitted.
However, when I, J, and K are used, assignment must be made in the alphabetical order.

B____ A____ D____ ..... I____ K____ ..... valid
B____ A____ D____ ..... J____ I____ ..... invalid

Addresses assigned in argument assignment I and the number of the variable in custom macro body correspond as follows:

Table 17.1.1(a) Correspondence of addresses of the argument assignment I and variables in the macro

address of the argument assignment I	Variable in custom macro body
A	#1
B	#2
C	#3
D	#7
E	#8
F	#9
H	#11
I	#4
J	#5
K	#6
M	#13
Q	#17
R	#18
S	#19
T	#20
U	#21
V	#22
W	#23
X	#24
Y	#25
Z	#26

b) Argument assignment II

A __ B __ C __ I __ J __ K __ I __ J __ K __ ....

In addition to the fact that arguments can be assigned in addresses A, B, and C, a maximum of ten sets of arguments can be set for addresses I, J and K.

When several numbers are assigned in the same address, they must be assigned in the determined sequence.

Addresses not required can be omitted.

Addresses assigned in argument assignment II and the number of the variable used in the macro correspond as follows:

Table 17.1.1(b) Correspondence of addresses of the argument assignment II and variables in the macro

Address of the argument assignment II	Variable in user macfro body
A	#1
B	#2
C	#3
I_1	#4
J_1	#5
K_1	#6
I_2	#7
J_2	#8
K_2	#9
I_3	#10
J_3	#11
K_3	#12
I_4	#13
J_4	#14
K_4	#15
I_5	#16
J_5	#17
K_5	#18
I_6	#19
J_6	#20
K_6	#21
I_7	#22
J_7	#23
K_7	#24
I_8	#25
J_8	#26
K_8	#27
I_9	#28
J_9	#29
K_9	#30
I_{10}	#31
J_{10}	#32
K_{10}	#33

Suffixes 1 to 10 of I, J and K indicate the sequence of the assigned set.

c) Coexistence of argument assignment I and II

No alarm is generated even if arguments of both assignment I and II are specified in the same block with a G65 command.

If an argument of type I and an argument of type II are specified to the same variable, the argument specified later is effective.

Example G65 A1.0 B2.0 I–3.0 I4.0 D5.0 P1000 ;

(Variable)
#1 : 1.0
#2 : 2.0
#3 :
#4 : –3.0
#5 :
#6 :
#7 : ~~4.0~~ 5.0

In this example, even if arguments I4.0 and D5.0 are specified to variable #7, the latter is effective.

d) Number of decimal places assumed when an argument is specified without the decimal point

A decimal point can be used for arguments regardless of the address. The following table shows the number of decimal places assumed when no decimal point is specified.

Address		Input in mm	Input in inches
D,E,H,M,S,T		0	0
A,B,C		α (*1)	α+1 (α) (*1,2)
F	G99 mode	2	4
	G98 mode	0(1) (*4)	2
I,J,K,Q,R,U,V,W,X,Y,Z		α (*1)	α+1 (*1)

NOTE 1 Value α is specified as follows according to the setting of the increment system for parameter 1004:

Increment system (α)	IS–A (2)	IS–B (3)	IS–C (4)

When IPR–X of parameter 1004 is 1 (increment system: X10), 1 is subtracted from each value α for IS–A and IS–B, namely each of the above values.

NOTE 2 When the addresses are used as rotation axis addresses, the position of the decimal point is indicated by the value enclosed in parentheses.

NOTE 3 When SLE of parameter 2402 is 1 (the E code specifies the screw pitch), the position of the decimal point is indicated by the value enclosed in parentheses.

NOTE 4 When F41 of parameter 2400 is 1 (when the unit of the F code specifying feed per minute in millimeter is 0.1 mm/min), the position of the decimal point is indicated by the value enclosed in parentheses.

NOTE 5 The number of decimal places is 0 when entering floating–point decimal numbers (when bit 0 (DPI) of parameter 2400 is 1).

17.1.2 Continuous–state calling

The methods for continuous–state calling are classified into move command calling (G66) and each–block calling (G66.1).

The following command specifies one of these macro calling mode.

G66 (or G66.1)P(program number) L (number of times the program is called)<argument>;

<argument> is specified in the same way as for simple calling.

The following command cancels the macro calling mode:

G67;

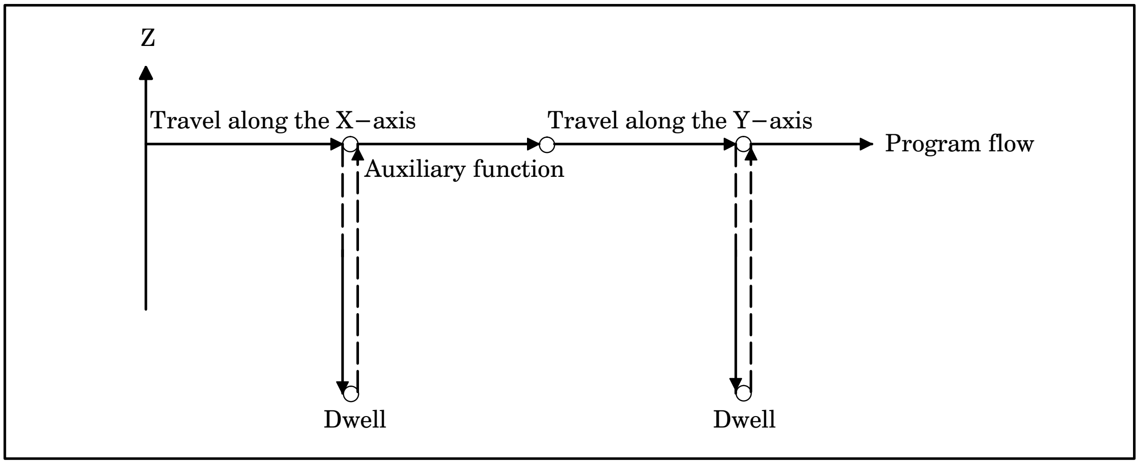
NOTE 1	G66 or G66.1 must be specified before all arguments in the G66 or G66.1 block. A sign and decimal point can be used in <argument> regardless of addresses.
NOTE 2	The G66 or G66.1 block must be specified together with the G67 block in the same program.

(a) Move command calling (G66)

In this mode, the specified macro is called after a move command is executed in the block containing the move command.

Example 1 Drilling cycles

Drilling cycles are executed at each point where the tool is positioned.



G66 P9082 R (point R) Z (point Z) X (dwell) ;

```
X______;
M______;
Y______;
 .
 .
 .
 .
G67;
```

Drilling cycles are executed in the blocks containing move commands.

The macro is coded as follows for incremental programming.

```
O9082 ;
G00 Z#18 ;
G01 Z#26 ;
G04 X#24 ;
G00 Z–[ ROUND [ #18] + ROUND [#26] ] ;
M99 ;
```

(b) Each–block calling (G66.1)

In this mode, the specified macros are unconditionally called at every block containing a CNC command. Except for the O and N codes and G codes other that the last one, none of the commands in any of the blocks are executed. Instead, they become arguments.

CNC command blocks that contain an O or N command have the same effect as the next block. CNC command blocks that do not contain an O or N code have the same effect as a block with G65P__ specified at the start.

For example,

N001 G01 G91 X100 Y200 D1 R1000; has the same effect as N001 G65P 100 G01 G91 X100 Y200 D1 R100; in the G66.1P100; mode.

However, note the following:

(1) G66.1 block

(a) A macro is called even in the G66.1 block.

(b) The correspondence between the addresses of the arguments and the variables is the same as for simple calling.

(2) The block after the G66.1 block and subsequent blocks where macros are called (not including the G66.1 block)

(a) G, P, and L are specified as new arguments. The correspondence is as follows:

G : Variable 10

L : Variable 12

P : Variable 16

However, the data is generally restricted by the input format of the CNC command.

For example, G1000P0.12L–4 cannot be issued.

(b) When multiple G codes are specified, the last G code is used as an argument.

(c) O and N codes and the G codes other than those in group 00 are used as continuous–state information in the next and subsequent blocks.

NOTE 1 Each block is assumed to be a CNC command unless it contains an O or N code. In this case, each–block calling is performed. When N is specified after an address other than O or N, N is used as an argument. N corresponds to variable 14.

NOTE 2 Neither S, T, nor the address for the second auxiliary function can be specified in the G66.1 mode.

Example If the following commands are issued when bit 0 of parameter 7000 = 1 (subprogram calling with a T code), alarm PS 093 occurs.

G66.1P1000;

T12;

17.1.3 Macro call using G codes

A G code can be set by a parameter to call a macro. That is, instead of specifying

N____G65P ΔΔΔΔ<argument assignment> ; the following simple command can be used.

N____ Gxx <argument assignment> ;

The correspondence between the calling G code xx and the program number ΔΔΔΔ of the called macro must be set as a parameter.

A calling G code and the program number ΔΔΔΔ of a called macro are set in a parameter.

The program number for macros ranges from 9010 to 9019.

Up to ten G01 to G999 commands can be used to call macros. G00, G65 to G67 cannot be used. These G codes cannot be specified in a macro called with a G code. These G codes cannot be specified in a subprogram called with an M code or a T code. Set parameters 7050 to 7059.

When a negative value is specified, continuous–state calling is selected.

When a parameter is –11, for example, continuous–state calling is performed in the G11 block. Parameter setting depends on whether continuous–state calling is performed in the G66 or G66.1 block.

7050	G code for calling macro 9010
7051	G code for calling macro 9011
⋮	⋮
7059	G code for calling macro 9019

17.1.4 Custom macro call with M code

Macro can be called with an M code set by a parameter, namely, the following command:

N ____ G65 P ΔΔΔΔ <argument assignment> ;

is equivalent to the following command:

N ____ M x x <argument assignment> ;

The correspondence between the M code x x which executes macro calling and the macro program No. ΔΔΔΔ accessed must be set as a parameter (No. 7080 to 7089). In the same way as subprogram calling with an M code, no MF or M code is sent. When the M codes are specified in the programs called by macro calling with a G code or subprogram calling with M, S, T, and B codes, no macros are called, and the M codes are treated as usual M codes.

Up to ten M codes from M01 to M97 can be used for macro calling.

Specify the following parameters:

7080	M code for calling macro 9020
7081	M code for calling macro 9021
⋮	⋮
7089	M code for calling macro 9029

NOTE These M codes are different from usual M codes and they must be commanded at the start of a block (just after the sequence No., if there is).

17.1.5 Subprogram call with M code

An M code can be set by a parameter to call a subprogram. That is, instead of

N ____ G ____ X ____ Y ____ M98PΔΔΔΔ ;

the following simple command can be specified.

N ____ G ____ X ____ Y ____ Mxx;

As for M98, the instruction is displayed on the program check screen, but MF and M codes are not transmitted. The correspondence between the calling M code xx and the program number ΔΔΔΔ of the called subprogram must be set as a parameter. Up to nine M codes can be used for a subprogram call. When these M codes are specified in a macro called with a G code or in a subprogram called with an M code or a T code, the subprogram is not called, but these M codes are treated as ordinary M codes.Use M codes other than M00, M01, M02, and M30 for these M codes.

Specify the following parameters:

7071	M code for calling subprogram 9001
7072	M code for calling subprogram 9002
⋮	⋮
7079	M code for calling subprogram 9009

17.1.6 Subprogram call with T code

If parameter is set in advance, a subprogram can be called with a T code.

N ____ G ____ X ____ Z ____ T t;

is equivalent to the following two blocks.

#149=t;

N ____ G ____ X ____ Z ____ M98 P9000;

T code t is stored as an argument in common variable #149. The T code is displayed on the program check screen, but TF and T codes are not transmitted. When this T code is specified in a macro called with a G code, or in a subprogram called with an M or T code, the subprogram is not called; but this T code is treated as ordinary T code.

Specify the following parameter:

7000								TCS

NOTE When TCS of parameter 7000 is 1, the block where a T code is specified calls a subprogram.

17.1.7 Subprogram calling with an S code

A parameter can be set to enable subprogram calling with an S code.

N___ G___ X___ Z___ .....Ss;

This block performs the same operation as the following two blocks:

#147=s;

N___ G___ X___ Z___ M98P9029;

Ss is stored in common variable 147 as an argument.

Although S codes are displayed on the program check screen, no SF or S code is sent.

When the S codes are specified in the macros called by macro calling with G codes or the subprograms called by subprogram calling with M, S, T, and B codes, no subprograms are called, and the S codes are treated as usual S codes.

Specify the following parameter:

7000							SCS	

NOTE When SCS of parameter 7000 is 1, the block where an S code is specified calls a subprogram.

17.1.8 Subprogram calling with a second auxiliary function code

A parameter can be set to enable a subprogram to be called with the second auxiliary function code specified in parameter 1030.

N____G____X____Z____......Bb; (B: second auxiliary function code)

This block executes the same operation as the following two blocks:

#146=b;

N____G____X____Z____......M98P9028;

Bb is stored in common variable 146 as an argument.

Although the second auxiliary function codes are displayed on the program check screen, no BF or second auxiliary function code is sent.

When second auxiliary function codes are specified in the macros called by G codes or the subprograms called by M, S, T, or B codes, no subprograms are called, and the second auxiliary function codes are treated as usual second auxiliary function codes.

Specify the following parameter:

7000						BCS		

NOTE When BCS of parameter 7000 is 1, the block where a B code is specified calls a subprogram.

17.1.9 Difference between M98 (subprogram call) and G65 (custom macro body call)

a) G65 can include arguments; M98 cannot.

b) M98 is used to branch to a subprogram after executing a command other than M, P or L in the block; G65 is used to branch only.

c) When a M98 block includes an address other than O, N, P and L, execution of the block stops as single block stop, a G65 block does not.

d) G65 changes the level of local variable; M98 does not. That is, #1 specified before G65 is one thing and #1 in the calling custom macro body is another.

   #1 specified before M98 is the same as #1 in the calling subprogram.

e) Up to four G65 calls, including G66, can be made in addition, M98 calls can be made up to eight calls together with calls by G65 and G66.

17.1.10 Multiplex calls

1) Multiplex calls

Similarly to a subprogram called from another subprogram, a macro can be called from another macro.

The multiplicity should be less than or equal to four including simple and modal calls.

2) Multiplex modal calls

In modal calls, the specified macro is called each time a motion command is executed. When several modal macros are specified, the next macro is called each time a motion command in the first macro is executed. Macros are successively called from those assigned later.

Example

G66 P9100;

Z 1000.0; (1–1)

G66 P9200;

Z 15000.0; (1–2)

G67; : P9200 cancelled

G67; : P9100 cancelled

Z–25000.0; (1–3)

M30;

O 9100;

X 5000.0; (2–1)

M99;

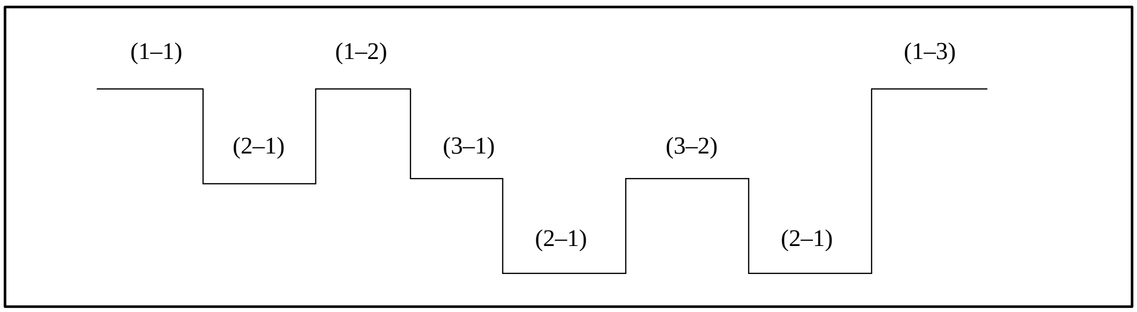
O 9200;

Z 6000.0; (3–1)

Z 7000.0; (3–2)

M99;

Sequence of execution (A block without a motion command is omitted in this chart.)

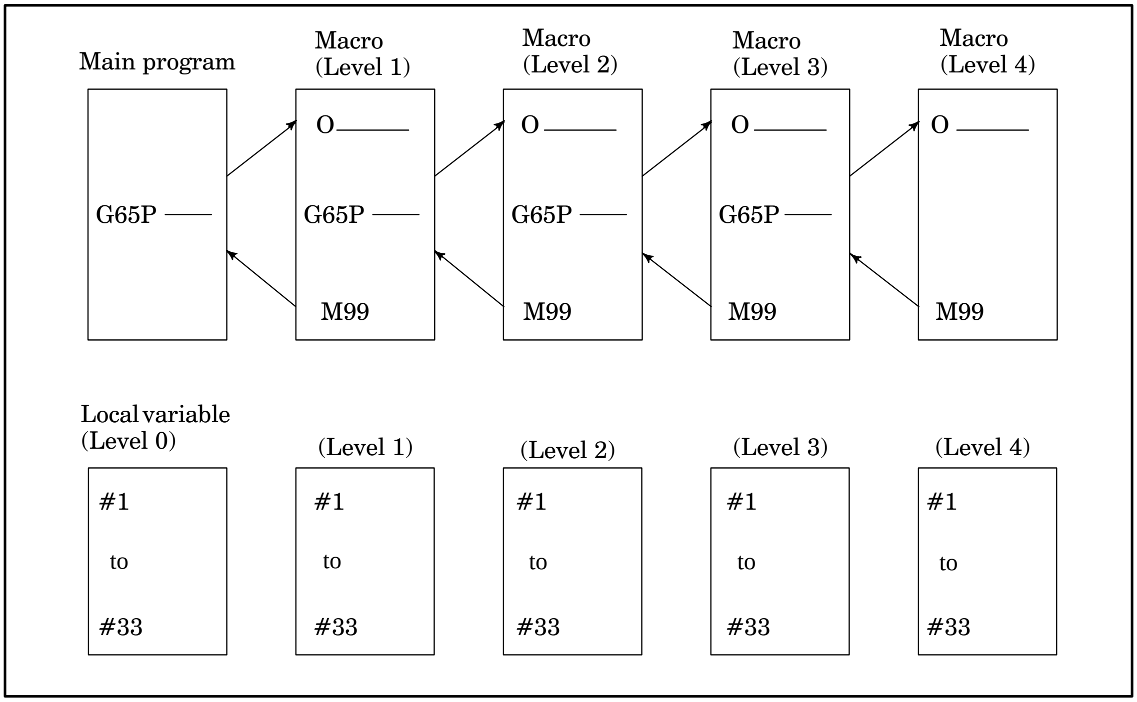


NOTE	A modal macro is not called after (1–3), which is not in macro call mode.

3) Custom macro level and local variable

When a macro is called with G65, G66, G66.1, or a G code which calls a macro, the level of the macro increases by one. As a result, the level of the local variable also increases by one. Namely, the relationship between the macro call and local variable is as follows.

(See 17.2.3–(1) for local variables.)



1) The main program is provided with #1 to #33 local variables (level 0).
2) When the macro (level 1) is called with G65, etc., the local variable (level 0) of the main program is stored, and #1 to #33 local variables (level 1) for the macro (level 1) are prepared. The arguments from the main program are taken in these variables.
3) The local variables (level 1, 2, 3) are stored each time the macros (level 2, 3, 4) are called, and new local variables (level 2, 3, 4) are prepared.
4) When the operation returns from each macro with M99, the local variables (level 0, 1, 2, 3) stored in (2) and (3) are set in the same conditions as when they were stored.

17.2 Creation of Custom Macro Body

17.2.1 Custom macro body format

The format of a custom macro body is as shown below.

```
O□□□□;(Program Number);
┌──────────────────────────────┐
│ Commands                     │
│ (Variables, arithmetic       │
│ operation and control        │
│ instruction can be specified.)│
└──────────────────────────────┘
M99;
```

Program numbers are determined as follows:

(1) O0001 – O7999 : for programs that can be registered, cancelled and edited as desired.

(2) O8000 to O8999 : Programs that cannot be registered, deleted, or edited unless settings are specified in the programs

(3) O9000 to O9019 : Macros having special calling formats

(4) O9020 to O9899 : Programs that cannot be registered, deleted, or edited unless parameters are specified in the programs

(5) O9900 to O9999 : Programs for operating the robot

Variables can be specified in a custom macro when the custom macro is programmed. Operation and control commands can also be used.

The actual values corresponding to the variables are specified in a macro calling command. See Subsection 17.1.1.

17.2.2 Variables

Variables can be used in the macro instead of numerical data. The user can assign any value (within the allowed range) to them. Using variables allow custom macros to become much more flexible than the conventional sub–routines.

Several variables can be used, and each variable is identified by a variable number.

1) Variable expressions

A variable is composed of the code # and a number as shown below.

#i(i=1,2,3,4.....)

Example 1 #5
#109
#1005

The following format can also be used where numbers are replaced by #[<Formula>]

#[<Formula>]

Example 2 #[#100]
#[#1001–1]
#[#6/2]

Variable #i explained hereafter can always be replaced with variable # [<Formula>] .

2) Quotation variables

The numerical value following an address can be replaced with a variable.

<address>#i or <address>–#i indicates that the value of the variable or the value multiplned by –1 is substituted for the command value of the address.

Example 3 F#33 If #33 = 1.5, it is the same as F1.5.

Z–#18 If #18 = 20.0, it is the same as Z–20.0.

G#130 If #130 = 3.0, it is the same as G3.

a) Using a variable with addresses/,:, O and N is prohibited, (i.e.,:#27 or N#1 cannot be used).

The value of n(n =1 to 9) in an optional block skip/n cannot be used as a variable.

b) A variable number cannot be replaced by a variable. When 5 in #5 is replaced with #30, it does not become ##30 but #[#30].

c) The value of a variable cannot exceed the maximum set for each address.

For example, when #140 = 120, G#140 exceeds the maximum.

d) When a variable is used for address data, its value is rounded to the range of the numerical values usable for each address.

e) By using <Formula>, as explained later, a numerical value following the address can also be replaced with <Formula>. <address>[<formula>] or <address>–[<Formula>] indicates that the value of the <formula> or the value of the <formula> multiplied by –1 is substituted for the command value of the address. Note that a constant with no decimal point used between brackets is assumed to have a decimal point at its end.

Example 4 X[#24+#18*COS[#1]]

Z–[#18+#26]

3) Undefined variables

The value of a variable which has not yet been defined is called <vacant>.

Variable #0 is used for a variable that is always <vacant>.

An undefined variable has the following nature:

a) Quotation

When an undefined variable is quoted, the address itself is also ignored.

When #1 =<vacant>	When #1 = 0
G90X100Y#1 ↓ G90X100	G90X100Y#1 ↓ G90X100YO

b) Operation

<vacant> is the same as 0 except when replaced by <vacant>.

When #1 =<vacant>	When #1 = 0
#2 = #1 ↓ #2 =<vacant>	#2 = #1 ↓ #2 = 0
#2 = #1 * 5 ↓ #2 = 0	#2 = #1 * 5 ↓ #2 = 0
#2 = #1 + #1 ↓ #2 = 0	#2 = #1 + #1 ↓ #2 = 0

c) Conditional expressions

<vacant> differs from 0 only for EQ and NE.

When #1 =<vacant>	When #1 = 0
#1 EQ #0 ↓ Established	#1 EQ #0 ↓ Not established
#1 NE 0 ↓ Established	#1 NE 0 ↓ Not established
#1 GE #0 ↓ Established	#1 GE #0 ↓ Not established
#1 GT O ↓ Not established	#1 GT O ↓ Not established

d) Display and setting variable values

Variable values can be displayed on the CRT screen. Variable values can also be set in the MDI mode. As far as operation, refer to III 10.4. Custom macro variable values.

17.2.3 Types of variables

Variables are classified into local variables, common variables and system variables, depending on the variable number. Usage and properties are different for each type of variable.

1) Local variable #1 to #33

The local variable is a variable locally used in the macro. That is, a local variable #i used in the macro and called at one point in time, is different from #i used in the macro (whether it is the same macro or not) called at another point in time. Accordingly, when macro K is called from macro J, as in a multiplex call, a local variable used in macro J is not destroyed by being used in macro K.

A local variable is used for an argument transfer. For information on the correspondence to the argument address, refer to section 17.1 for Macro Call Command. A local variable without a transferred argument is vacant in its initial status and can be used freely.

2) Common variable #100 – #199, #500 – #999

Just as a local variable is used locally in the macro, a common variable is in common use throughout the main program, throughout each subprogram called from the main program, and throughout each macro. That is, #i used in a certain macro is the same as #i used in another macro. Accordingly, the calculated value of a common variable #i in a certain macro can be used in another macro.

(a) Common variable group A

Common variables 100 to 149 and 500 to 549 can be used.

When the power is turned off, common variables 100 to 149 are cleared, but common variables 500 to 549 are retained.

(b) Common variable group B

Common variables 100 to 199 and 500 to 599 can be used.

When the power is turned off, common variables 100 to 199 are cleared, but common variables 500 to 599 are retained.

(c) Common variable group C

Common variables 100 to 199 and 500 to 699 can be used.

When the power is turned off, common variables 100 to 199 are cleared, but common variables 500 to 699 are retained.

NOTE	The usable length of the storage tape is reduced by 2.2 m.

(d) Common variable group D

Common variables 100 to 199 and 500 to 999 can be used.

When the power is turned off, common variables 100 to 199 are cleared, but common variables 500 to 999 are retained.

NOTE	The usable length of the storage tape is reduced by 7.4 m.

3) System variables

Use of a system variable is fixed in the system.

System variables

Variable number	Variable number
#1000 to #1035: DI for macros	#5041 to #5050: Workpiece coordinates
#1100 to #1135: DO for macros	#5061 to #5070: Skip signal position
#2000 to #2999: Tool compensation amount	#5081 to #5090: Tool compensation amount
#3000 and #3006: Macro alarm and program stop	#5101 to #5110: Servo deviation
#3001 and #3002: Clock	#5201 to #5210: Workpiece offset (external)
#3003 and #3004: Automatic operation control	#5221 to #5230: Workpiece offset (G54)
#3007: Mirror image	#5241 to #5250: Workpiece offset (G55)
#3011, #3012: Clock	#5261 to #5270: Workpiece offset (G56)
#3901, #3902: No. of parts machined, No. of parts required	#5281 to #5290: Workpiece offset (G57)
#4000: Main program No.	#5301 to #5310: Workpiece offset (G58)
#4001 to #4120: Continuous–state information (blocks read in advance)	#5321 to #5330: Workpiece offset (G59)
#4201 to #4320: Continuous–state information (blocks currently executed)	#7001 to #7950: Additional workpiece coordinate system workpiece offset
#5001 to #5010: Position of block end–point	#10001 to #13999: Tool offset
#5021 to #5030: Machine coordinates	

a) Interface signals #1000 to #1031, #1032, and #1033 to #1035, #1100 to #1155 and #1132, #1133 to #1135

Output signal

Interface output signals can be issued by assigning values to system variables #1100 to #1132.

System variable	Number of signals	Interface input signal	System variable	Number of signals	Interface input signal
#1100	1	2^0 UO000	#1116	1	2^{16} UO016
#1101	1	2^1 UO001	#1117	1	2^{17} UO017
#1102	1	2^2 UO002	#1118	1	2^{18} UO018
#1103	1	2^3 UO003	#1119	1	2^{19} UO019
#1103	1	2^4 UO004	#1120	1	2^{20} UO020
#1105	1	2^5 UO005	#1121	1	2^{21} UO021
#1106	1	2^6 UO006	#1122	1	2^{22} UO022
#1107	1	2^7 UO007	#1123	1	2^{23} UO023
#1108	1	2^8 UO008	#1124	1	2^{24} UO024
#1109	1	2^9 UO009	#1125	1	2^{25} UO025
#1110	1	2^{10} UO010	#1126	1	2^{26} UO026
#1111	1	2^{11} UO011	#1127	1	2^{27} UO027
#1112	1	2^{12} UO012	#1128	1	2^{28} UO028
#1113	1	2^{13} UO013	#1129	1	2^{29} UO029
#1114	1	2^{14} UO014	#1130	1	2^{30} UO030
#1115	1	2^{15} UO015	#1131	1	2^{31} UO031

NOTE Only UO000 to UO015 are provided for the 6M/3M interface.

Systen variable	Number of signals	Interface input signal
#1132 #1133 #1134 #1135	32 32 32 32	UO000AUO031 UO100AUO131 UO200AUO231 UO300AUO331

value of variable	Output signal
1	Contact closed (HIGH)
0	Contact open (LOW)

32 points output signals can be issued at one time by assigning a value to the system variable #1132 to #1135.

$$\#1132 = \sum_{i=0}^{30} \#[1100 + i]^{*} 2i - \#1031 \times 2^{31}$$

$$\#[1132 + n] = \sum_{I=0}^{30} \{2^{i} * Vi\} - 2^{31} \times V_{31}$$

When UOni is LOW, Vi=0

When UOni is HIGH, Vi=1

n is 0 to 3.

The last values of the system variable #1100 to #1132 issued are stored as 1.0 or 0.0.

NOTE When a value other than 1.0 or 0.0 is assigned to #1100 to #1115, <vacant> is assumed as 0 and values other than <vacant> and 0 are assumed as 1. A value under 0.00000001 is indefinite.

Input signal

The status of the interface input signal is determined by reading the values of the system variables #1000 to #1032.

System variable	Number of signals	Interface input signal	System variable	Number of signals	Interface input signal
#1000	1	2^{0} UI 000	#1016	1	2^{16} UI 016
#1001	1	2^{1} UI 001	#1017	1	2^{17} UI 017
#1002	1	2^{2} UI 002	#1018	1	2^{18} UI 018
#1003	1	2^{3} UI 003	#1019	1	2^{19} UI 019
#1004	1	2^{4} UI 004	#1020	1	2^{20} UI 020
#1005	1	2^{5} UI 005	#1021	1	2^{21} UI 021
#1006	1	2^{6} UI 006	#1022	1	2^{22} UI 022
#1007	1	2^{7} UI 007	#1023	1	2^{23} UI 023
#1008	1	2^{8} UI 008	#1024	1	2^{24} UI 024
#1009	1	2^{9} UI 009	#1025	1	2^{25} UI 025
#1010	1	2^{10} UI 010	#1026	1	2^{26} UI 026
#1011	1	2^{11} UI 011	#1027	1	2^{27} UI 027
#1012	1	2^{12} UI 012	#1028	1	2^{28} UI 028
#1013	1	2^{13} UI 013	#1029	1	2^{29} UI 029
#1014	1	2^{14} UI 014	#1030	1	2^{30} UI 030
#1015	1	2^{15} UI 015	#1031	1	2^{31} UI 031

NOTE Only UI000 to UI015 are provided for the 6M/3M interface.

Systen variable	Number of signals	Interface input signal
#1032	32	UI000AUI031
#1033	32	UI100AUI131
#1034	32	UI200AUI231
#1035	32	UI300AUI331

value of variable	Input signal
1	Contact closed (HIGH)
0	Contact open (LOW)

Sine the variable value read is 1.0 or 0.0 regardless of the unit system, the unit system must be considered in preparing a macro.

The reading system variable #1032 to #1035 is used to read 32 points input signals at one time.

$$\#1032 = \sum_{I=0}^{30} \#[1000 + i] \times 2^{i} - \#1031 \times 2^{31}$$

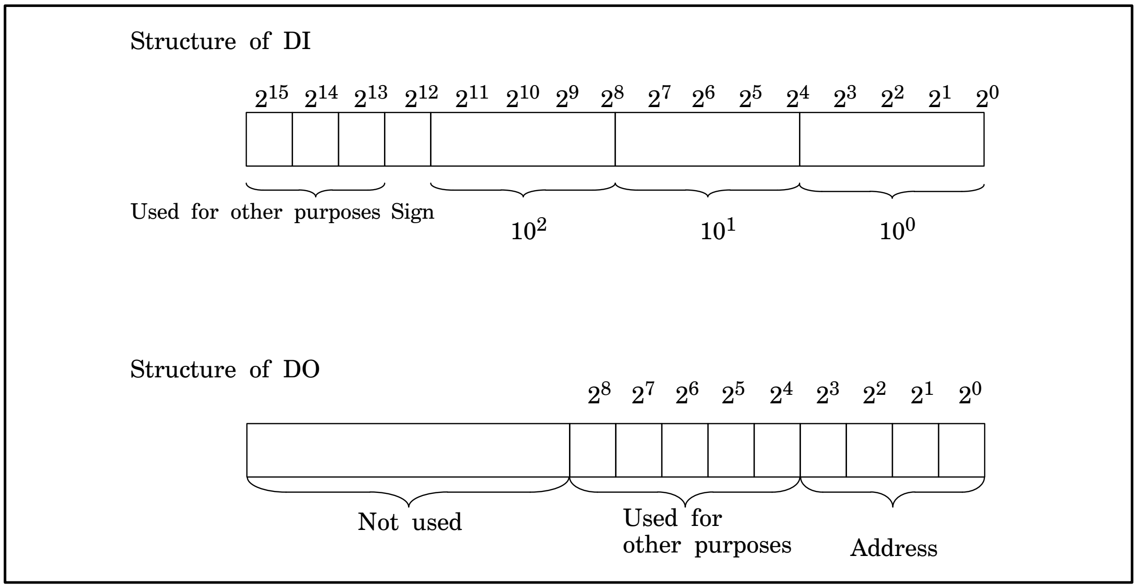
$$\#[1032 + n] = \sum_{I=0}^{30} \{ 2^{i} \times Vi\} - 2^{31} \times V_{31}$$

When the UIni signal is low, Vi = 0.
When the UIni signal is high , Vi = 1.
n is 0 to 3.

System variables #1000 to #1035 cannot be used as a left side term in a calculation command.

Example 1

1) Macro used to read three digits of BCD data with a sign by address switching Structure of DI Configuration of DI



Macro call command

G65P9100D (address);

The custom macro body is specified as follows:

O9100;

#1132=#1132 AND 496 OR#7; : address feed out.

G65P9101T60; : Timer macro

#100=BIN[#1032 AND 4095]; reading in of 3–digit BCD data.

IF[#1012EQ 0] GOTO 9100; (Sign)

#100=– #100;

N9100 M99;

2) Read eight types of signed 6–digit BCD data (three digits to the left of the decimal point and three digits to the right of the decimal point) by address switching in #101.

Machine tool composition

When DO 2^0= 0, three digits to the right of the decimal point

= 1, three digits to the left of the decimal point

When DO2^3 to 2^1 = 000, data No. 1

= 001, data No. 2

:

= 111, data No.8

Macro call command

G65P9101D (data number);

The custom macro body is specified as follows:

O9100;

G65P9100D[#7*2+1]:

#101=#100;

G65P9100D[#7*2];

#101=#101+#100/1000;

M99;

b) Tool offsets #2000 to #2800, #10001 to #13999

The tool offset can be determined by reading a tool offset system variable (#2001 to #2800, #10001 to #13999). It can also be modified by assigning a value to a system variable #i.

Numbers of system variables for tool offsets in memory C if the maximum number of offsets is 200

Offset Number	H code		D code	
	Geometric	Wear	Geometric	Wear
1	#2001	#2201	#2401	#2601
2	#2002	#2202	#2402	#2602
3	#2003	#2203	#2403	#2603
⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮
199	#2199	#2399	#2599	#2799
200	#2200	#2400	#2600	#2800

NOTE 1 System variables 2001 to 2400 are used in memory B irrespective of whether the code for offset is an H or D code.

NOTE 2 System variables 2001 to 2200 are used in memory A irrespective of whether the code for offset is an H or D code and whether the tool compensation is of geometric compensation or wear compensation.

NOTE 3 Variable 2000 is read only and is always 0.

When the number of tool offsets exceeds 200, the following system variables are used:

Tool offset memory A

Offset number	Geometric and wear
1	#10001
⋮	⋮
999	#10999

Tool offset memory B

Offset number	Geometric	Wear
1	#10001	#11001
⋮	⋮	⋮
999	#10999	#11999

Tool offset memory C

Offset Number	H code		D code	
	Geometric	Wear	Geometric	Wear
1	#10001	#11001	#12001	#13001
⋮	⋮	⋮	⋮	⋮
999	#10999	#11999	#12999	#13999

Offsets 1 to 200 can be referenced with system variables 2000 to 2800.

c) System variables for workpiece zero point offset values #5201–#5330, #7001–#7950

By reading the values of system variables #5201–#5326, and #7001–#7955, workpiece zero point offset values are determined.

By assigning values to these variables, workpiece zero point offset values can be changed.

Variable numbers 2500 to 2906 in system 6 can be used by setting bit #F6W in parameter 7001 to as appropriate. In this case, neither tool offset 2500 nor subsequent offsets can be used.

System variable number	Control axis	Additional workpiece coordinate system number
#5201 #5202 ⋮ #5210	1st–axis enternal offset from the workpice reference point 2nd–axis enternal offset from the workpice reference point 10th–axis enternal offset from the workpice reference point	External offsets from the workpice reference point to be applied to all coordinate systems
#5221 #5222 ⋮ #5230	1st–axis enternal offset from the workpice reference point 2nd–axis enternal offset from the workpice reference point 10th–axis enternal offset from the workpice reference point	G54
#5241 #5242 ⋮ #5250	1st–axis enternal offset from the workpice reference point 2nd–axis enternal offset from the workpice reference point 10th–axis enternal offset from the workpice reference point	G55
#5261 #5262 ⋮ #5270	1st–axis enternal offset from the workpice reference point 2nd–axis enternal offset from the workpice reference point 10th–axis enternal offset from the workpice reference point	G56
#5281 #5282 ⋮ #5290	1st–axis enternal offset from the workpice reference point 2nd–axis enternal offset from the workpice reference point 10th–axis enternal offset from the workpice reference point	G57
#5301 #5302 ⋮ #5310	1st–axis enternal offset from the workpice reference point 2nd–axis enternal offset from the workpice reference point 10th–axis enternal offset from the workpice reference point	G58
#5321 #5322 ⋮ #5330	1st–axis enternal offset from the workpice reference point 2nd–axis enternal offset from the workpice reference point 10th–axis enternal offset from the workpice reference point	G59

Like those in the standard workpiece coordinate system, the workpiece zero point offset in an additional workpiece coordinate system can be treated as a system variable. The system variable numbers are listed below.

System variable number	Control axis	Additional workpiece coordinate system number
#7001 #7002 ⋮ #7010	Workpiece zero point offset along the first axis Workpiece zero point offset along the second axis Workpiece zero point offset along the tenth axis	1 (G54.1P1)
#7021 #7022 ⋮ #7030	Workpiece zero point offset along the first axis Workpiece zero point offset along the second axis Workpiece zero point offset along the tenth axis	2 (G54.1P2)
. . .	. . .	
#7941 #7942 ⋮ #7950	Workpiece zero point offset along the first axis Workpiece zero point offset along the second axis Workpiece zero point offset along the tenth axis	48 (G54.1P48)

System variable number = 7000 + (coordinate system number –1) *20 + axis number

Coordinate system number: 1 to 48

Axis number: 1 to 10

d) Alarm #3000

When detecting an error in the macro, an alarm can be generated. When an alarm number is specified in the system variable #3000, the alarm lamp is turned on and the alarm status is entered after the proceeding block is processed.

#3000=n (ALARM MESSAGE);

($0 \leqq n \leqq 999$)

An alarm message of less than 26 characters can be specified in the section between control–out and control–in.

The alarm screen displays ”MCn ALARM MESSAGE”.

e) Clock #3001, #3002

The clock time can be determined by reading the values of the system variables #3001, #3002. The time can be preset by assigning a value to the system variable.

Kind	System variable	Unit time	At the time of power–on	counting
Clock 1	#3001	1 msec.	Reset to 0	Always
Clock 2	#3002	1 hour time	Same as at the time of power–off	While STL signal is on.

The accuracy each clock is within 16 msec.

Example Timer

Macro call command

G65P9101 T (Awaiting time) msec;

This macro may be specified as follows:

```
O9101;
#3001 = 0; (Initial setting)
WHILE[#3001 LE #20] DO1;
(Wait for the prescribed length of time)
END1;
M99;
```

f) The suppression of the single block stop and the wait for the auxiliary function end signal in #3003.

When the following values are assigned to the system variable #3003, the single block stop function is suppressed and execution advances from one block to the next without awaiting the end signal (FIN) of auxiliary functions (S, T, and M). When the end signal is not awaited, the distribution end signal (DEN) is not transmitted. Be careful not to specify a subsequent auxiliary function without awaiting the end signal.

#3003	Single block stop	Auxiliary function end signal
0	Enabled	Awaited
1	Disabled	Awaited
2	Enabled	Not awaited
3	Disabled	Not awaited

Example Drill cycle (For incremental programming) (Equivalent to G81)

Macro call command

G65 P9081 L (Repetition time) R (R point) W (Z point);

The custom macro body is specified as follows:

```
O9081 ;
#3003= 1;
G00   W#18 ;
G01   W#23 ;
G00Z– [ROUND [ #18] + ROUND [ #23] ] ;
#3003= 0 ;
M99 ;
```

The single block stop is not executed. #18 corresponds to R and #23 corresponds to N

NOTE The state of #3003 is deared by resetting.

g) Feed hold, feed rate override, and exact stop check suppression specified in #3004

When the following values are assigned to the system variable #3004, feed hold and feed rate override are suppressed for subsequent blocks and the exact stop check is not performed. When you press the feed hold button during execution of a block for which feed hold has been suppressed the block stops as single block stop. Howere, if the single block stop has also been suppressed, the tool does not stop at the block.

#3004	Feed hold	Feed rate override	Exact stop check
0	○	○	○
1	×	○	○
2	○	×	○
3	×	×	○
4	○	○	×
5	×	○	×
6	○	×	×
7	×	×	×

○: Effective, ×: Suppressed

NOTE 1 The state of #3004 is cleared by resetting.

NOTE 2 When the exact stop function is disabled, no exact stop check is made in the blocks where exact stops must originally be carried out, such as those for cutting feed and positioning. Note the contents of item (iii) in Subsection 17.2.7 and set system variable 3004 so that the exact stop function is enabled in the blocks where exact stops are generally carried out.

h) Program stop at the specified block with the message displayed in #3006.

Specifying #3006=1(MESSAGE); in the macro enables the program to be stopped after the current and preceding blocks are executed. Programming a message of up to 26 characters enclosed in control–in and control–out symbols enables the message to be displayed on the page for external operator messages.

i) Mirror image (system variable 3007)

Reading system variable 3007 enables the mirror images of each axis to be checked. As shown below, each axis corresponds to each bit.

	15	14	13	12	11	10	9	8 (bit)
#3007 upper							10	9

	7	6	5	4	3	2	1	0 (bit)
#3007 lower	8	7	6	5	4	3	2	1 axis

Each bit that is set to 0 indicates that mirror image is disabled for the corresponding axis.
Each bit that is set to 1 indicates that mirror image is enabled for the corresponding axis.

Example When system variable 3007 is 3, the mirror image is effective in the 1st and 2nd axes.

j) Main program number #4000

System variable	Object	Explanation
#4000	Main program number	Reads the main program number.

The main program number can always be read through system variable #4000 from any subprogram level.

NOTE 1 The main program number is the number of the program that started first.

NOTE 2 If an O–number is specified from the MDI during execution of the main program, or more than one O number is found during tape mode operation, the value of #4000 is changed to the newly specified O number.

NOTE 3 If no program is registered, or no O number is specified in tape mode, #4000 is set to 0.

NOTE 4 Even if #4000 is used when drawing is being performed in the background for a system such as the FANUC Series 15–MFB, the current main program number can be read.

NOTE 5 If high–speed distribution is specified (by setting bit 5 of parameter No. 0 to 0) in DNC operation with a remote buffer, #4000 cannot be used.

k) Continuous–state information (system variables 4001 to 4130 and 4201 to 4330)

Reading the values of system variables 4001 to 4130 enable continuous–state commands specified in the preceding blocks of the currently buffered block to be checked.

Reading the values of system variables 4201 to 4330 enable the continuous–state commands in the block currently being executed to be checked.

The unit when the continuous–state commands are specified is used.

System variable	Continuous–state information in the preceding block	System variable	Continuous–state information in the block currently being executed
#4001	G code in group 01	#4201	G code in group 01
:	:	:	:
#4020	G code in group 20	#4220	G code in group 20
#4025	G code (group 25)	#4225	G code (group 25)
#4026	G code (group 26)	#4226	G code (group 26)
#4102	B code	#4302	B code
#4107	D code	#4307	D code
#4108	E code	#4308	E code
#4109	F code	#4309	F code
#4111	H code	#4311	H code
#4113	M code	#4313	M code
#4114	Sequence number	#4314	Sequence number
#4115	Program number	#4315	Program number
#4119	S code	#4319	S code
#4120	T code	#4320	T code Additional workpiece coordinate system number
#4130	Additional workpiece coordinate system number	#4330	

NOTE Preceding block and block currently being executed

The CNC reads the blocks ahead of the block currently being executed in the machining program. The block currently being executed is generally different from the block the CNC is currently reading (see Subsection 17.2.7). The preceding block refers to the block immediately before the block currently being read, i.e., the block programmed immediately before the last of the blocks for which system variables 4001 to 4130 are specified.

An example is shown below:

Example

```
O1234 ;
N10 G00 X200.0 Y200.0 ;
N20 G01 X1000.0 Y1000.0 F10 ;
        :
        :
N50 G00 X500.0 Y500.0 ;
N60 #1= #4001 ;
```

Assume that the CNC is currently executing N20, and that, as shown above, the CNC has already read N20 to N60 (probably in the multi–buffer mode).

The block currently being executed is N20. The preceding block is N50.

Therefore, the continuous–state information for group 01 specified in the block currently being executed is G01. The continuous–state information for group 01 specified in the preceding block is G00.

When N60 #1=#4201; is specified, #1 = 1.

When N60 #1=#4001; is specified, #1 = 0.

l) Positional information #5001 to #5110

Ststem variable	Position information	Read–out during movement
#5001 #5002 #5003 #5004 #5005 #5006 #5007 #5008 #5009 #5010	1st axis block end position (ABSIO) 2nd axis block end position (ABSIO) 3rd axis block end position (ABSIO) 4th axis block end position (ABSIO) 5th axis block end position (ABSIO) 6th axis block end position (ABSIO) 7th axis block end position (ABSIO) 8th axis block end position (ABSIO) 9th axis block end position (ABSIO) 10th axis block end position (ABSIO)	Allowed
#5021 #5022 #5023 #5024 #5025 #5026 #5027 #5028 #5029 #5030	1st axis current position (MBSMT) 2nd axis current position (MBSMT) 3rd axis current position (MBSMT) 4th axis current position (MBSMT) 5th axis current position (MBSMT) 6th axis current position (MBSMT) 7th axis current position (MBSMT) 8th axis current position (MBSMT) 9th axis current d position (MBSMT) 10th axis current position (MBSMT)	Not allowed
#5041 #5042 #5043 #5044 #5045 #5046 #5047 #5048 #5049 #5050	1st axis current position (ABSOT) 2nd axis current position (ABSOT) 3rd axis current position (ABSOT) 4th axis current position (ABSOT) 5th axis current position (ABSOT) 6th axis current position (ABSOT) 7th axis current position (ABSOT) 8th axis current position (ABSOT) 9th axis current position (ABSOT) 10th axis current position (ABSOT)	Not allowed
#5061 #5062 #5063 #5064 #5065 #5066 #5067 #5068 #5069 #5070	1st axis skip signal position (ABSKP) 2nd axis skip signal position (ABSKP) 3rd axis skip signal position (ABSKP) 4th axis skip signal position (ABSKP) 5th axis skip signal position (ABSKP) 6th axis skip signal position (ABSKP) 7th axis skip signal position (ABSKP) 8th axis skip signal position (ABSKP) 9th axis skip signal position (ABSKP) 10th axis skip signal position (ABSKP)	Allowed

Ststem variable	Position information	Read–out during movement
#5081 #5082 #5083 #5084 #5085 #5086 #5087 #5088 #5089 #5090	1st axis tool position offset 2nd axis tool position offset 3rd axis tool position offset 4th axis tool position offset 5th axis tool position offset 6th axis tool position offset 7th axis tool position offset 8th axis tool position offset 9th axis tool position offset 10th axis tool position offset	Not allowed
#5101 #5102 #5103 #5104 #5105 #5106 #5107 #5108 #5109 #5110	Deviated 1st axis servo position Deviated 2nd axis servo position Deviated 3rd axis servo position Deviated 4th axis servo position Deviated 5th axis servo position Deviated 6th axis servo position Deviated 7th axis servo position Deviated 8th axis servo position Deviated 9th axis servo position Deviated 10th axis servo position	Not allowed

NOTE Reading up to the 10th axis is possible, but variables for more than the number of controlled axes (servo) are undifined.

Abbreviation	ABSIO	ABSMT	ABSOT	ABSKP
Meaning	End point coordinate of the preceding block	Command present coordinate	Command present coordinate	Psition where skip signal has turned on in the G31 block
Coordinate system	Workpiece coordinate system	Machine coordinate system	Workpiece coordinate system	Workpiece coordinate system
Tool offset Tool length offset, Cutter compensation	Not considered Tool top position	Considered Tool standard position	Considered Tool standard position	Considered Tool standard position

NOTE 1 The tool offset is not the in effect just before the block is executed; it is the offset for the current block.

NOTE 2 Skip signal position is the end point of the G31 block if the skip signal is not turned on in the block.

NOTE 3 The end point (ABSIO) of the block including the skip command (G31) is the position when the skip signal is turned on or the end point of the block when the skip signal is not turned on.

Example The tool moves to a point appropriate to the machine tool (point distant by Xp, Zp from a reference point) through a programmed intermediate point; and after processing a sequence of operations, returns to the original point.

Macro call command

```
G65 P9300 X(Intermediate point) Y(Intermediate point) Z(Intermediate point);
```

The custom macro body is specified as follows:

```
O9300;
#1=#5001;
#2=#5002;
#3=#5003;
G00 X#24, Z#26;
G04; (Move is interrupted to read #5021 to #5023)
G91 X[Xp – #5021] Y[Yp – #5022] Z[Zp – #5023];
            .
            . (processing)
X#24 Y#25 Z#26;
X#1 Y#2
Z#3;
```

m) Clock information #3011, #3012

It is possible to know the year, month, day, hour, minute, and second by reading system variables

Kind	System variable
Year, Month,Day	#3011
Hour, Minute, Second	#3012

Example When it is March 20, 1991 4:17 5” PM

#3011 = 19910320, #3012 = 161705

Refer to III–11.11 clock.

n) Accumulated number of machined parts and number of necessary parts (system variables 3901 and 3902)

The function for displaying the operating time and the number of parts enables the number of necessary parts and the accumulated number of machined parts to be displayed on the CRT screen (see Section 11.13 in Part III). When the accumulated number of machined parts agrees with the number of necessary parts, a signal is output to the machine (PMC unit).

The system variables containing the accumulated number of parts and the number of necessary parts can be read and written.

Type	System variable
Accumulated number of parts	#3901
Number of necessary parts	#3902

17.2.4 Specifying and displaying system variable names

Issuing the following command enables system variables 500 to 519 to be assigned names consisting of up to eight characters:

SETVNn[$\alpha1$ $\alpha2$...... $\alpha8$, $\beta1$ $\beta2$...... $\beta8$, ...];

n is the number of the first system variable to be named.

$\alpha1$ $\alpha2$... $\alpha8$ is the name of variable n. $\beta1$ $\beta2$... $\beta8$ is the name of variable n+1. Subsequent variable names correspond to ascending variable numbers in this way.

Separate the character strings by a comma (,). All the codes that can be used as significant information in the program can be used, except the control–in code, control–out code, comma (,), EOB, EOR, and colon for a program number (:). The system variable names are retained when the power is turned off.

The system variable numbers, names, and data are displayed in that order.

```
MACRO      VAL :
 NO.       NAME           DATA
 0500     ABCDEFGH      – 1234.5678
 0501     COUNTER         00020.000
 0502     POINTER
 0503     1ST           – 0000.4025
 0504     2ST             00004.500
 0505                     124000.00
 0
 0506
 0507
 0508     START
 0509
 0510     TOOL–PT         000045.00
```

17.2.5 Arithmetic commands

A variety of arithmetic operations can be performed on variables. An arithmetic command must be specified the same as in general arithmetic expressions.

#i=<Formula>

<Formula>, the right–hand–side of an arithmetic command is a combination of constants, variables, functions and operators. A constant can be used instead of #i, #j, and #k. A constant without a decimal point used in <Formula> is considered to have a decimal point at the end.

1) Definition and substitution of variables

   #i=#j Definition, substitution

2) Addition arithmetic

   #i=#j + #k Sum
   #i=#j – #k Subtraction
   #i=#j OR #k Logical sum (at every bit of 32 bits)
   #i=#j XOR #k Exclusive OR (at every bit of 32 bits)

3) Multiplication arithmetic

   #i=#j×#k Product
   #i=#j/#k Quotient
   #i=#j AND #k Logical product (at every bit of 32 bits)
   #i=#j MOD #k Remainder (After #j and #k are rounded off to integers, the remainder is obtained. If #j is negative, #i is negative.)

4) Functions

#i=SIN [#j]	Sine (degree unit)
#i=COS [#j]	Cosine (degree unit)
#i=TAN [#j]	Tangent (degree unit)
#i=ATAN [#j]/[#k]	Arctangent (degree unit) (#i=–180.0 to 180.0)
#i=SQRT [#j]	Square root
#i=ABS [#j	Absolute value
#i=BIN [#j]	Conversion from BCD to BIN
#i=BCO [#j]	Conversion from BIN to BCD
#i=ROUND [#j]	Rounding off
#i=FIX [#j]	Discard fractions less than 1
#i=FUP [#j]	Add 1 for fractions less than 1
#i=ACOS [#j]	Inverse cosine
#i=ASIN [#j]	Inverse sine
#i=LN [#j]	Natural logarithm
#i=EXP [#j]	Exponential function whose base is e (= 2.718 ...)
#i=ADP [#j]	Addition of a decimal point

Add Decimal Point (ADD) function

By specifying ADP [#n] (n=1 to 33), the deoimal point can be placed in arguments transferred to a subprogram without a decimal point.

Example When ADP [#24] is specified in a subprogram which is called with G65xxxxX10;, the decimal point is placed at the end of the argument, that is, after 10. This function is useful in creating a subprogram without considering the increment system used. When CVA of parameter No.7000 is set to 1, this argument is converted to 0.01 when it is transferred to the subprogram. In this case, the ADP function cannot be used.

NOTE To ensure the compatibility of programs, it is recommended to place the deoimal point in argument designation at macro calling without using the ADP function.

How to use function ROUND

(1) If function ROUND is employed in an arithmetic operation command or in an IF or WHILE conditional expression, the figure in function ROUND is rounded off as is ordinary data with a decimal point.

Example

#1=ROUND [1.2345];

#1 becomes 1.0.

IF[#1 LE ROUND[#2]] GOTO 10;

ROUND[#2] is 4.0 if #2 is 3.567.

(2) If function ROUND is employed in a command to an address, it is rounded off to the least input increment of the address.

Example

G01X[ROUND[#1]].

If #1 is 1.4567 and the least input increment of X is 0.001, this block becomes G01 X1.457;.
In this example, the command is the same as G01 X #1; command.
Function ROUND in an address command is used mainly in the following case.

Example

Program to move incrementally by #1 and #2 only and then return to the starting point

```
N1 #1 = 1.2345;
N2 #2 = 2.3456;
N3 G01 X #1 F100;   (X moves 1.235.)
N4 X #2;            (X moves 2.346.)
N5 X–[#1+#2]; (X moves –3.58, since #1 + #2 is 3.5801.)
```

Since 1.234 + 2.346 = 3.581, the program does not return to the starting point by N5.

Assume

N5 X – [ROUND[#1] + ROUND[#2]];

It becomes equal to N5 X–[1.235 + 2.346]; and the program returns to the starting point.

5) Combination of arithmetic operations

The above arithmetic operations and functions can be combined. The order of priority in an arithmetic operation is function, multiplication arithmetic then addition arithmetic.

Example 1 #i=#j+#k*SIN [#l]

$$\#i=\underbrace{\#j+\underbrace{\#k*\underbrace{SIN\,[\#l]}_{1}}_{2}}_{3}$$

Sequence of arithmetic operations

6) Modification of arithmetic sequence using

A portion to be assigned priority in an arithmetic sequence can be enclosed in [ ].

[ ] can be nested up to five times (including [ ] used in functions).

Example 2 #i=SIN [[[#j + #k] *#l + #m] *#n] Nesting to a level of three

$$\#i=\underbrace{SIN\,[\underbrace{[\underbrace{\underbrace{[\underbrace{\#j+\#k}_{1}]*\#l}_{2}+\#m}_{3}]*\#n}_{4}]}_{5}$$

7) Precision

Always consider the precision of a custom macro function used for preparation of programs.

a) Data format

Numeric data handled by a custom macro is in a floating decimal point format as follows:

$M*2^E$

where M: 1–bit sign + 31–bit binary data
E: 1–bit sign + 7–bit binary data

b) Operational precision

An operation executed once generates the following error. These errors are accumulated with each repeated operation.

Operation format	Average error	Maximum error	Type error
a = b × c	1.55×10^{-10}	4.66×10^{-10}	Relative error $\left\lvert \frac{\varepsilon}{a} \right\rvert$
a = b / c	4.66×10^{-10}	1.86×10^{-9}	
a = b	1.24×10^{-9}	3.73×10^{-9}	
a = b + c a = b – c	2.33×10^{-10}	5.32×10^{-10}	Min $\left\lvert \frac{\varepsilon}{b} \right\rvert , \left\lvert \frac{\varepsilon}{c} \right\rvert$
a = SIN b a = COS b	5.0×10^{-9}	1.0×10^{-8}	Absolute error $\lvert \varepsilon \rvert$
a = ATAN b/c	1.8×10^{-6}	3.6×10^{-6}	

NOTE Function TAN performs SIN/COS.

8) Notes on decreased precision

a) Addition and subtraction

Note that when absolute values are used subtractively in addition or subtraction, the relative error cannot be held under 10–8. For example, suppose that the real values of #1 and #2 are as follows.

#1 = 9876543210123.456

#2 = 9876543277777.777

Performing operation #2 – #1 does not produce

#2 – #1 = 67654.321,

since the custom macro has a precision of only eight decimal digits, the values of #1 and #2 have a precision as low as approximately

#1 = 9876543200000.000

#2 = 9876543300000.000,

respectively. (The internal values differ somewhat from the above values they are binary numbers.)

Consequently,

#2 – #1 = 100000.000

which generates a large error.

b) Logical operation

EQ, NE, GT, LT, GE and LE are basically the same as addition and subtraction.

Therefore, be careful of errors. To determine whether or not #1 and #2 are equal in the above, for example,

IF [#1 EQ #2]

is not always evaluated correctly. When the error is evaluated as in

IF [ABS [#1 – #2] LT 50000]

and the difference between #1 and #2 falls within the range error, both values must be considered equal.

c) Trigonometric functions

Absolute errors occur in trigonometric functions; but, since they are not under 10–8, be careful of integration or division after using a trigonometric function.

d) FIX function The user needs to be always careful about precision also when using the FIX function for the results of an operation. For example, when the following operations are performed, #2 does not always assume 2:

N10#1 = 0.002;

N20#2 = #1*1000;

N30#3 = FIX[#2];

For some error occurs in the operation in N20 and there may be a very small deviation like #2 = 1.999999997 instead of 2.000000000. To prevent such problem, fix N30 as follows:

FIX[formula] → FIX[formula $\pm\varepsilon$]

Where, + ε when the value of the formula is positive, or - ε.ε when the value of the formula is negative, must be 0.1, .0.01, 0.001, ... or like according to the need.

17.2.6 Control command

The program flow can be controlled by using the following commands.

1) Divergence

(1) IF [Conditional expression] GOTOn

When <conditional expression> is satisfied, the next operation is executed in the block with the sequence number n in the same program.

Sequence number n can be replaced by a variable or <Formula>.

When the condition is not satisfied, the control proceeds to the next block.

IF [<conditional expression>] can also be omitted. In this case, control is unconditionally passed to block n.

The following expressions can be used for <conditional expression>.

#j EQ #k =

#j NE #k ≠

#j GT #k >

#j LT #k <

#j GE #k ≧

#j LE #k ≦

<Formula> can be used instead of #j and #k, and a variable or <Formula> can be used instead of n.

NOTE	In the block with the sequence number “n” which will be executed after a GOTO n command, the sequence number must be at the top of the block.

Branching in the reverse direction takes more time than that in the forward direction.

(2) IF [<conditional–expression>] macro–statement;

When <conditional–expression> is satisfied, the specified macro statement is executed. The number of the macro statements is limited to 1.

Example

IF [#1 EQ #2] THEN#3=0;

When #1 agrees with #2, 0 is assigned to #3.

2) Iteration

WHILE [<conditional expression>] DO m

(m = 1,2,3)

.

.

.

END m

While <conditional expression> is satisfied, blocks DO m to END m are repeatedly executed. That is, the condition of DO m block is examined and when the <conditional expression> is satisfied, control passes to the next block. But when it is not satisfied, the block after END m is executed.

WHILE [<conditional expression>] can also be omitted as with IF, and if omitted, blocks from DO m to END m are executed eternally.

WHILE [<conditional expression>] DO m and END m must be used as a pair.

Example 1 #120 = 1;

```
N1 WHILE [#120 LE 10] DO1 ;     ←  Repeated ten times

  N2 WHILE [#30 EQ1] DO2 ;

  N3 END2 ;                      ←  Repeated valiable while
                                    variable #30 =1
      .
      .
      .
#120=#120+1 ; N4
END 1 ;
```

NOTE Notes on REPEAT programming

(1) DO m must be specified before END m.

```
  :
END 1;
  :     (No)
DO 1;
  :
```

(2) DO m and END m must correspond to each other in one program.

```
  :
DO 1;
  :
DO 1;      (No)
  :
END 1;
  :
DO 1;
  :
END 1;     (No)
  :
END 1;
```

(3) The same identification number can be used many times.

```
  :
DO 1;
  :
END 1;
  :
DO 1;
  :     (Yes)
END 1;
  :
```

(4) DO statement can be nested up to three times.

```
  :
DO 1;
  :
DO 2;
  :
DO 3;
  :     (Yes)
END 3;
  :
END 2;
  :
END 1;
```

(5) DO ranges cannot be intersected.

```
  :
DO 1;
  :
DO 2;
  :
END 1;     (No)
  :
END 2;
  :
```

(6) A branch can be made from inside to outside a DO range.

```
  :
DO 1;
  :
GOTO 9000;
  :     (Yes)
END 1;
  :
N9000;
  :
```

(7) A branch cannot be made from outside to inside a DO range.

```
  :
GOTO 9000;      (No)
  :
DO 1;
  :
N9000.....;
  :
END 1;
  :
DO 1;
  :
N9000.....;
  :
END 1;
  :
GOTO 9000;      (No)
  :
```

(8) Custom macro bodies or subprograms can be called from inside a DO range. DO statements can be nested up to three times more in the custom macro body or in the subprogram.

```
  :
DO 1;
  :
G65.....;    (Yes)
  :
G66;      (Yes)
  :
G67;      (Yes)
  :
END 1;
  :
  :
DO 1;
  :
M98.....; (Yes)
  :
  :
END 1;
  :
```

17.2.7 Macro and NC statements

The following blocks are called macro statements.

i) Operation command (block including=)

ii) Control command (block including GOTO, DO or END)

iii) Macro call command (block including G65, G66.1, G661, G67 or G codes calling macro)

Blocks other than macro statements are sometimes referred to as NC statements.

The following blocks have the same features as the macro statements:

(1) A subprogram call block (a block including subprogram call command by M98, M, S, T or B code) which does not include any command address other than O, N, P and L.

(2) A block which includes M99 but does not include any command address other than O, N, P and L.

The macro statement differs from the NC statement in the following points:

i) In the normal single block mode, the single block stop does not occur.

ii) The macro statement is not regarded as the no movement block in the cutter radius C.

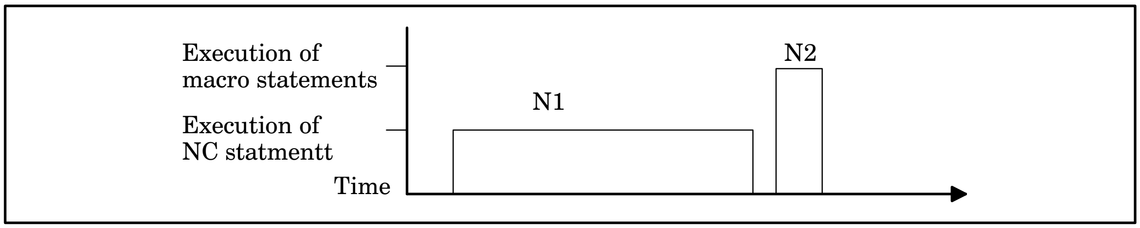
iii) The time of execution differs.

iii)is described below in further detail.

a) The macro statement existing next to the block which does not buffer the next one (block of non–buffering M code, or G31 block) is executed after that block is executed.

Example 1

N1X1000M00; Block currently executed
N2#1100 = 1; Macro statement



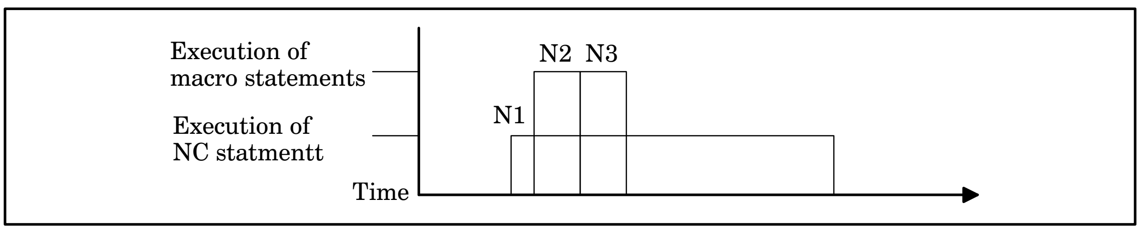
b) Macro statement existing next to the block buffering the next one.

i) When not in cutter compensation C mode

When the current block starts to be executed, the next macro statement is immediately executed. The macro statement up to the next NC statement is executed.

Example 2

N1XG01X1000; Block currently executed
N2#1100=1; Macro statement executed
N3#1=10; Macro statement executed
N4X2000; Next NC statement



ii) When in cutter compensation C mode

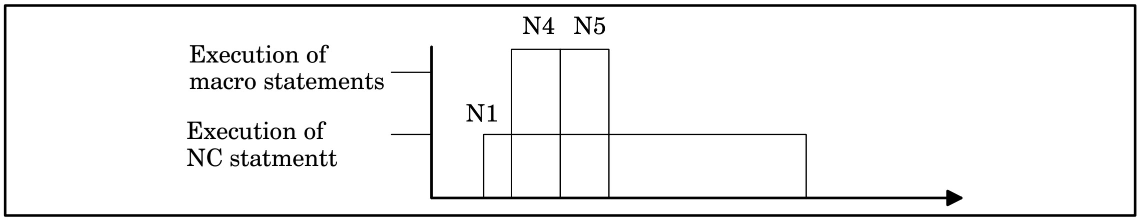
(a) When the first NC statement next to the block currently executed is not the no–movement block (block containing no movement command in the cutter compensation plan).

(1) When the second NC statement is not the no–movement block.

The macro statement after the first NC statement next to the block currently executed, is executed.

Example 3

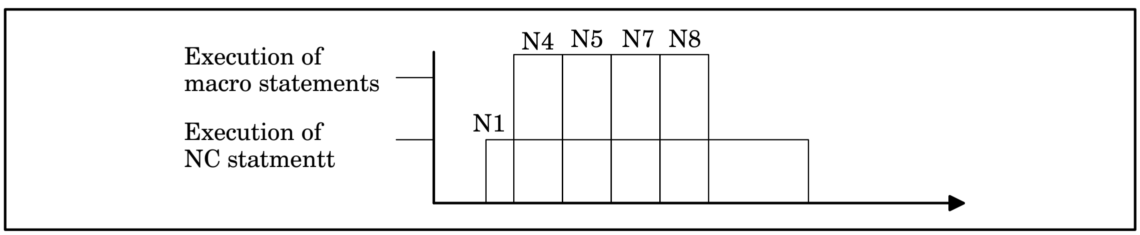
N1X1000; Block currently executed
N2#10=100; Macro statement already executed
N3Y1000; 1st NC statement
N4#1100=1; Macro statement executed
N5#1=10; Macro statement executed
N6X–1000; 2nd NC statement



(2) When the second NC statement after the block currently executed is the no–movement block, macro statements up to that and next to the second NC statement (i.e. the no–movement block) after the block currently executed are executed.

Example 4

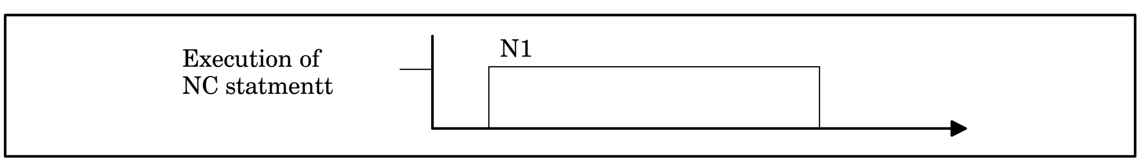
N1X1000; Block currently executed
N2#10=100; Macro statement already executed
N3Y1000; First NC statement
N4#1100=1; Macro statement executed
N5#1=10; Macro statement executed
N6Z1000; 2nd NC statement
(no–movement block)
N7#1101=1; Macro statement executed
N8#2=20; Macro statement executed
N9X–1000; 3rd NC statement



(b) When the first NC statement after the block that is currently being executed specifies no movement, subsequent macro statements are not executed simultaneously with the current block (macro statements N2, N3, N5, and N6 have already been executed in Example 5).

Example 5

N1Y1000; Block currently executed
N2#1100=1; Macro statement already executed
N3#1=10; Macro statement already executed
N4Z1000; 1st NC statement (no–movement block)
N5#1101=1; Macro statement executed
N6#2=20; Macro statement executed
N7X–1000; 2nd NC statement



17.2.8 Codes and words used in custom macro

The following code can be used in the program of the custom macro in addition to codes used in conventional programs.

1) ISO

Meaning	8	7	6	5	4		3	2	1	Character
[	○	○		○	○	∘		○	○	[
]	○	○		○	○	∘	○		○	]
#	○		○			∘		○	○	#
*	○		○		○	∘		○		*
=	○		○	○	○	∘	○		○	=
O	○	○			○	∘	○	○	○	O

2) EIA

Meaning	8 7 6 5 4 3 2 1	Character
[	Parameter(No.7010)	
]	Parameter(No.7011)	
#	Parameter(No.7012)	
*	Parameter(No.7013)	&
=	Parameter(No.7014)	

O, the same code O as in the program number, must be used. The hole pattern for [,], #, * and=in EIA code must be set as parameters (Nos. 7010–7014). However, the character with no punched hole cannot be used. Note that alphabetic codes can be used, but when used as #, they are not used in their proper sense.

Special words used in custom macro are:

OR, XOR, IF, GOTO, EQ, NE, GT, LT, GE, LE, AND, SIN, COS, TAN, ATAN, SQRT, ABS, BIN, BCD, ROUND, FIX, FUP, WHILE, DO, END, ASIN, ACOS, MOD, LN, EXP

17.2.9 Write–protecting common variables

(1) Multiple common variables 500 to 699 can be write–protected. The protection is effective during setting on the MDI, all clear, and writing in a macro program.

(2) Specify the following parameters:

7031	(Number of the first variable to be write–protected)–500
7032	Number of variables to be write protected

To write–protect variables 545 to 550, for example, specify 45 in parameter 7031 and 6 in parameter 7032.

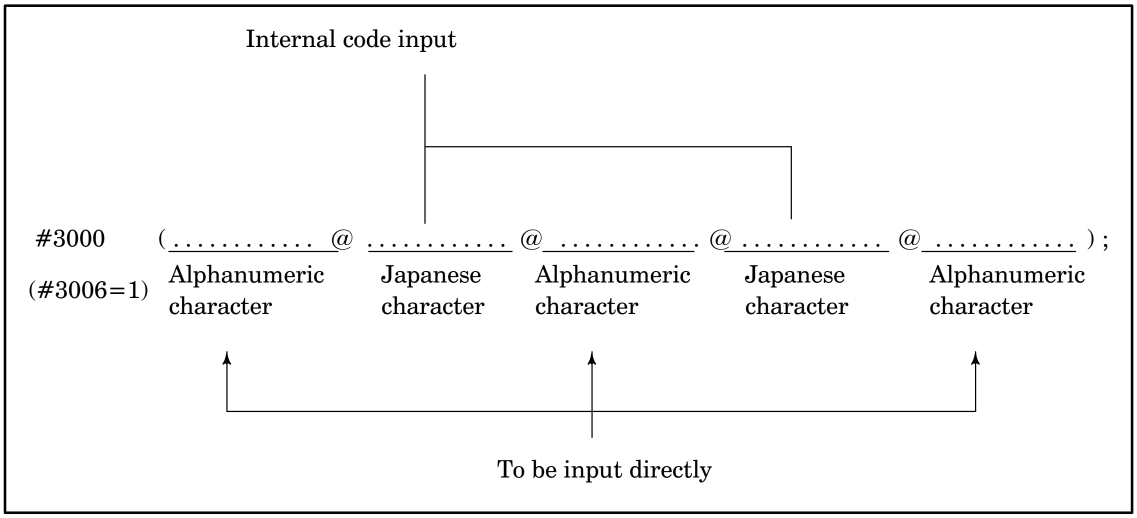
(3) If setting on the MDI or writing on the program is performed for the common variables which have been write–protected with the above parameters, PS alarm 116 occurs.

17.2.10 Displaying a macro alarm and macro message in Japanese

(1) Kanji, katakana and hiragana characters as well as alphanumeric characters and special characters can be displayed on the alarm screen and external operator message screen using system variables 3000 and 3006.

(2) To display alphanumeric and special characters, directly enter the alphanumeric and special characters.

To display Japanese characters, enclose the internal codes of the characters in @ symbols. This is called internal code input. (The internal codes of kanji and hiragana characters are represented by four hexadecimal digits. The internal code of a katakana character is represented by two hexadecimal digits.)



To display the following macro message, for example,

“ MADE IN 日本 ファナック”

Input the following:

#3006 = 1

(MADE IN @867C 8B5C 0020 CC A7 C5 AF B8 @)

日 本 フ ァ ナ ッ ク

(3) @ cannot be used as a special character.

(4) In the internal code input mode, displaying the character string stops when one of the following occurs:

(a) When a character that cannot be recognized as an internal code is specified

(b) When a control–in code is specified

The internal code input mode is terminated when the following occurs:

(c) The number of hexadecimal characters enclosed in @ symbols is an odd number.

(5) For the internal codes of kanji, hiragana, katakana characters, refer to the FANUC internal codes.

17.3 Registration of Custom Macro Body

A custom macro body is similar to a subprogram; it is registered and edited in the same way as a subprogram. The memory used to register custom macro bodies is included in the storage capacity of the CNC.

17.4 Limitations

1) Sequence number search

Sequence numbers in the custom macro body cannot be searched for.

2) Single block

A block, other than one including a macro call command, arithmetic command, or control command, can be executed as a single block stop even when in the macro.

A block including a macro call command (G65, G66, G66, G66.1, G67), is not stopped even by single block operation.

The following settings or parameter setting executes a single block stop for arithmetic and control commands. This function is used for testing a custom macro body.

2201						SB9		
0010			SBM	SB8	SB7			

When SBM = 1, the single–block stop function is executed in all the macro statements.

When SB7 = 1, the single–block stop function is executed for the macro statements in programs O7000 to O7999.

When SB8 = 1, the single–block stop function is executed for the macro statements in programs O8000 to O8999.

When SB9 = 1, the single–block stop function is executed for the macro statements in programs O9000 to O9999.-

A block in which a macro statement has executed a single block stop, is assumed to have no movement in the cutter compensation mode. Sometimes incorrect compensation is performed. (This case is handled the same as it is for a movement amount of 0 although a move has been specified.)

3) Optional block skip

/. appearing in the middle of <expression> (right–hand–side of an arithmetic formula, or in [ ]

is treated as division, and not an optional block skip.

4) Operation in EDIT mode

Setting parameters can be used to protect registered custom macro bodies and subprograms from accidental destruction.

2201								NE9
0011								NE8

Setting parameters NE8 (No. 0011#0) and NE9 (No, 2201#0) to 1 specifies that custom macro bodies or subprograms with program numbers 8000 to 9899–cannot be cancelled, and edited.

These parameters are not effective either when all the programs are deleted by turning on the power or when a single program is punched.

5) Display on the PRGRM page in a mode other than the EDIT mode

When a custom macro or subprogram is called, the called program is generally displayed on the PRGRM pages. Setting the following parameters stops the called program from being displayed on the PRGRM pages.

2201							ND9	

0011							ND8	

Setting ND8 to 1 stops custom macros or subprograms 8000 to 8999 from being displayed on the PRGRM page in a mode other than the EDIT mode. Setting ND9 to 1 stops custom macros or subprograms 9000 to 9999 from being displayed on the PRGRM page in a mode other than the EDIT mode.

6) Reset

When a cleared status is produced by a reset, all local variables and common variables #100 to #149 become <vacant>.

System variables #1000 to #1133 are not cleared.

Custom macro unit, subprogram call condition, and DO condition are cleared by reset.

7) Display in PROGRAM RESTART page

The M and T codes used for subprogram calling are not displayed like M98.

8) Feed hold

Macro statement execution stops at the end of the block by turning on feed hold. (It also stops by resetting or alarm generation).

9) Other limitations

a) Usable variables

Variables 0 to 33, 100 to 199, and 500 to 999 (restricted according to the system)

b) Valid variable values

Maximum value $\pm 10^{47}$
Minimum value $\pm 10^{-29}$

c) Constant value valid in <Formula>

Maximum value ± 99999999
Minimum value ± 0.0000001
Decimal point can be used

d) Arithmetic precision

Eight–digit decimal number

e) Macro nesting

Maximum of four

f) Iteration identification number

1 to 3

g) Nesting of [ ]
Maximum of five

h) Subprogram nesting

Eight (Subprogram call and macro call)

10) No tape operation can be executed for control commands.

17.5 External Output Commands

The following macro commands can be executed in addition to the standard custom macro commands. (These commands are called external output commands.)

a) BPRNT

b) DPRNT

c) POPEN

d) PCLOS

These commands are provided to output variable values and characters through the reader/puncher interface. Specify these commands according to the following procedure.

1) Open command: POPEN

The connection processing to external I/O units is made before executing a series of data output commands.

2) Data output command: BPRNT or DPRNT

Necessary data output commands are executed.

3) Close command: PCLOS

This command is specified after all data output commands have been completed in order to disconnect external I/O units.

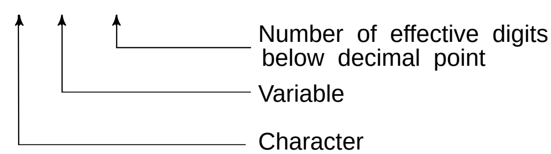
(1) Open command POPEN

POPEN;

This command is provided to connect to external I/O units, and it is specified before sending a series of data output commands. The DC2 control code is output from CNC.

(2) Data output commands BPRNT, DPRNT

(a) BPRNT [a #b [c] ………]



Characters are output and variable values are binary–output when BPRNT is given.

(i) The specified characters are output by codes set ISO.

The following characters are com–mand–able.

· Alphabetic characters (A to Z)

· Numeric characters

· Special characters (*, /, #+, –)

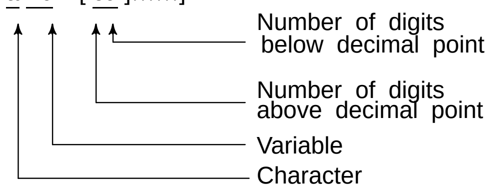
Asterisk (*) is output by the space code.

(ii) Since all variables are being stored with a decimal point, the number of effective digits below the decimal point must be specified by parenthesizing them just after specifying variables. Variable values are treated as 2–word (32–bit) data by taking the number of digits below decimal point into account, and they are output as binary data, starting with high–order bytes.

(iii) EOB code is output by ISO after the command data output.

(iv) “vacant” variable is outputtable as 0.

(b) DPRNT [a #b [cd]……]



When DPRNT is specified, characters and numeric characters and every variable value digit are output by a code set by parameter EIA (No. 0000#4).

(i) The description of the commands is same as in (i), (iii), (iv) of BPRNT command.

(ii) For outputting variable values, specify the variable number following number of digits below the decimal point by parenthesizing these values.

Variable values are output by the specified number of digits, every digit starting with a higher significant digit by a code set by parameter EIA (No. 0000#4). The decimal point is also output by a code set by parameter EIA (No. 0000#4).

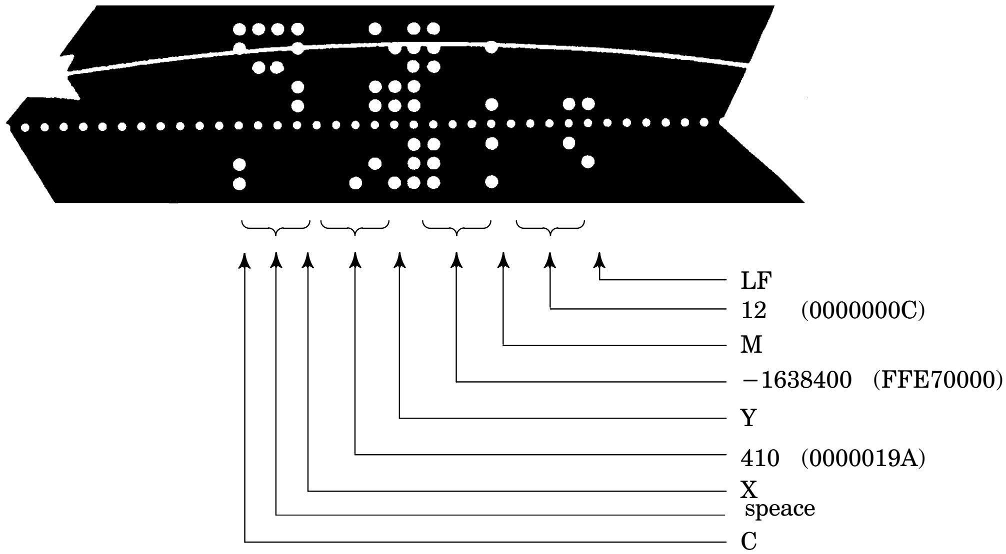
A variable value is regarded as being composed of a max. 8–digit numeric value. If the higher significant digits are 0, they are not output when parameter PRT (No. 7000#7)=1, and the space code is output when PRT=0.

Whenever the number of digits below the decimal point is other than 0, the numeric value below the decimal point is output. When the number of digits below the decimal point is 0, the decimal point is not output.

For the plus (+) code in case of a positive sign, the space code is output when parameter PRT (No. 7000#7)=0, but no code is output when PRT=1.

Example 1 BPRNT [C**X#100 [3] Y#101 [3] M#10 [0]]

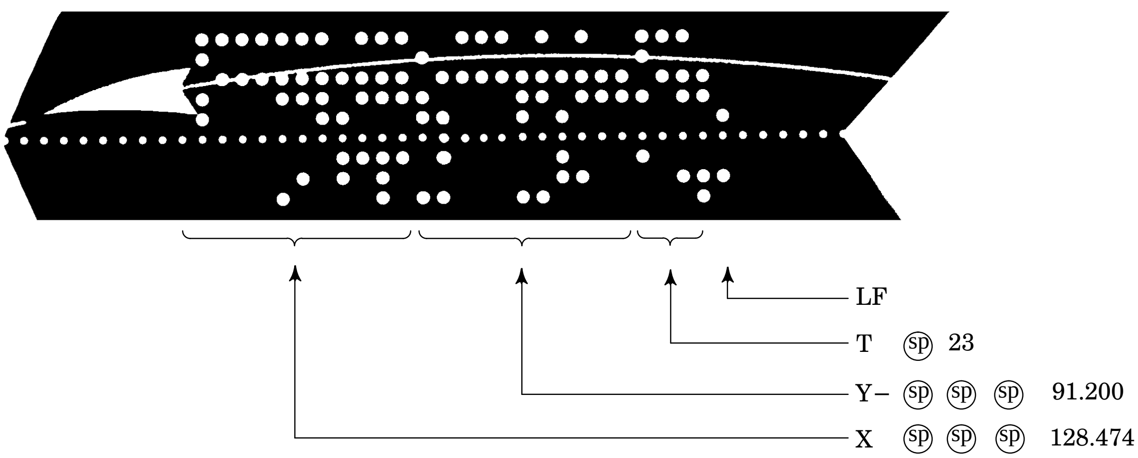
Variable value
#100=0.40956
#101=–1638.4
#10=12.34



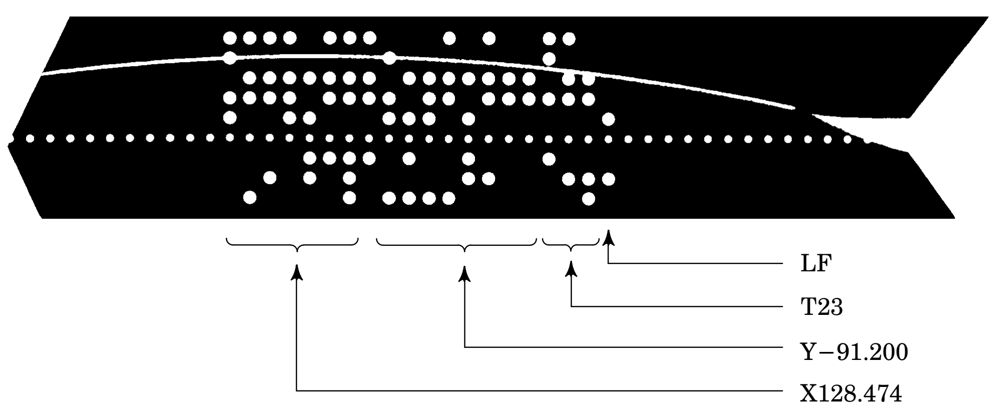
Example 2 DPRNT [X#2 [53] Y#5 [53] T#30 [20]]

Variable value
#2= 128.47398
#5=–91.2
#30=123.456

(1) Parameter PRT=0



(2) Parameter PRT=1



(3) Close command PCLOS

PCLOS ;

This command is specified when all data output commands are completed, to release the processing connection to external I/O units. DC4 control code is output from CNC.

(4) Setting required for using this function

(a) Set parameter 21 so that the output device for punch out conforms to the reader/punch interface. Do not set the parameter so that output is sent to the FANUC Cassette.

(b) Specify the data, such as the baud rate, for the reader/punch interface in any of parameters 5001 to 5162 according to the output device number specified by parameter 21.

(c) To output data with the DPRNT command, specify whether leading zeros are output as blanks.

	#7	#6	#5	#4	#3	#2	#1	#0
7000	PRT							

When the PRT DPRNT command is executed, the reading zeros
0: Are output as blanks.
1: Are not output.

(5) Cautions

(a) It is not necessary to sequentially specify open command (POPEN), data output commands (BPRNT, DPRNT) and close command (PCLOS). When the open command is specified at the start of a program, it is no longer necessary to specify the open command again until after the close command is specified.

(b) Specify the open command and close command as a pair of commands without fail.

In other words, send the close command at the end of a program. Don't specify the close command independently without the open command.

(c) A data output command in progress is stopped and subsequent data are erased by reset processing.

Accordingly, if reset processing is specified by M30, or the like, at the end of a data output program, specify the close command at the end of the program and wait until all data are output before starting M30 processing or other reset processing.

17.6 Interruption Type Custom Macro

Entering an interrupt signal (UINT) from the machine during execution of a program permits call–out of another program. This function is called the interruption type custom macro.

The interrupt command is programmed as follows:

M96P×××× : (Custom macro interrupt on)
M97 : (Custom macro interrupt off)

This function permits any execution block of the program to call out another program, thus enabling program operations according to the ever changing conditions.

<Applications>

(1) Processing for tool error detection is initiated by an external signal.

(2) Another program is inserted in a machining series without the stopping of machining operation.

(3) The current machining information is read at fixed time intervals, etc. Thus, adaptive control–like applications are possible.

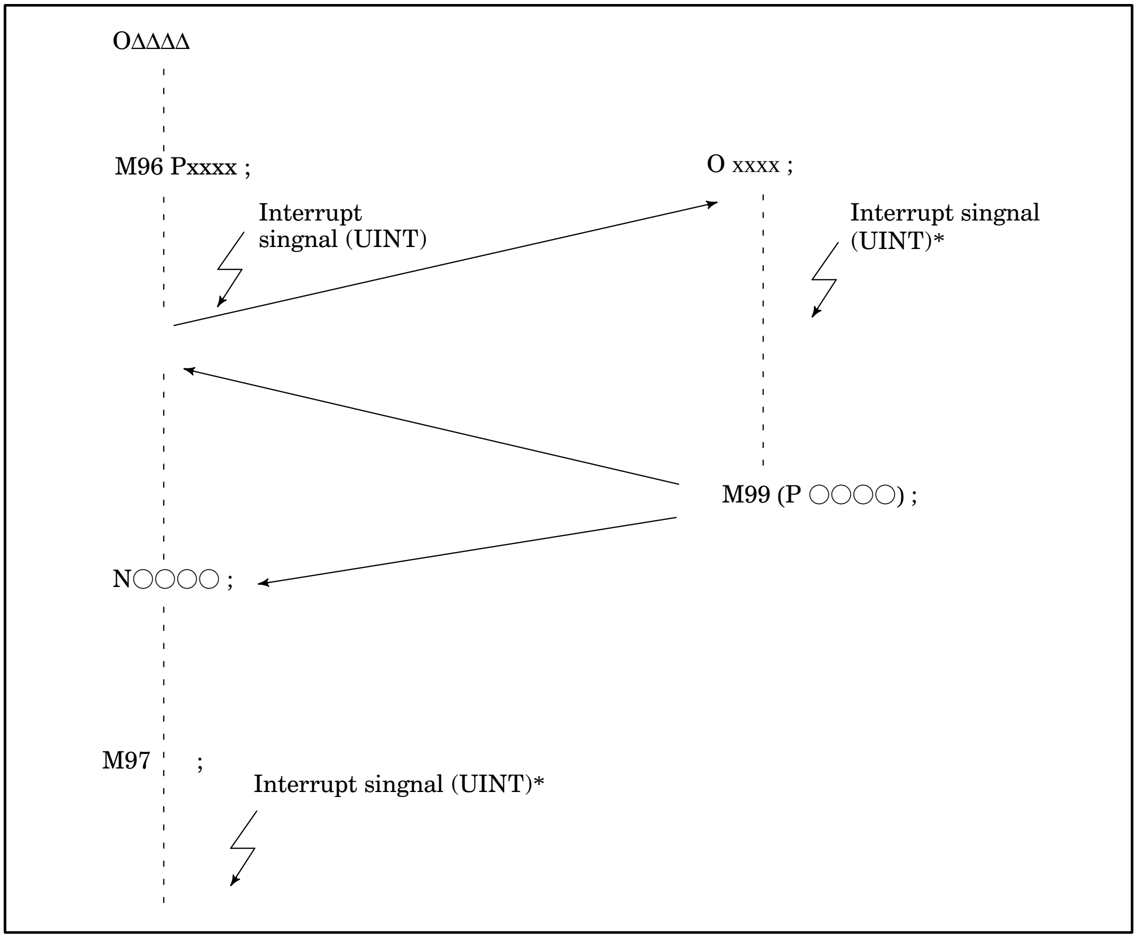


Fig. 17.6 Interrupt custom macro

The figure shows that specifying M96 P×××× in programming permits a program specified by P×××× to interrupt the current program and be executed, when the interrupt signal (UINT) is entered. The interrupt signal (UINT marked * in Fig. 17.6) is ignored during execution of the interrupt program or after M97.

17.6.1 Commanding

(1) Effective Conditions

Custom macro interruption can be used only during program execution. Therefore, it is effective when:

(a) Memory or MDI operation has been selected;

(b) The start lamp (STL) is on;

(c) A custom macro interrupt is not in process.

No custom macro interrupt can be issued in a manual operation mode such as the JOG, STEP, or HANDLE mode.

(2) Command Format

The custom macro interrupt function is, in principle, executed by making the interrupt signal effective and ineffective by M96 and M97. That is, the interrupt signal entered until M97 is commanded or the NC is reset, after M96 is commanded, initiates the custom macro interruption. On the other hand, entering the interrupt signal after M97 is commanded or after the NC is reset, does not initiate the custom macro interruption, but the interrupt signal entered until M96 is commanded is ignored.

M96 PXXXX ; Speciffies custom macro program number.

Speciffies the interrupt custom macro program number.

M97 ; Custom Macro interrupt off.

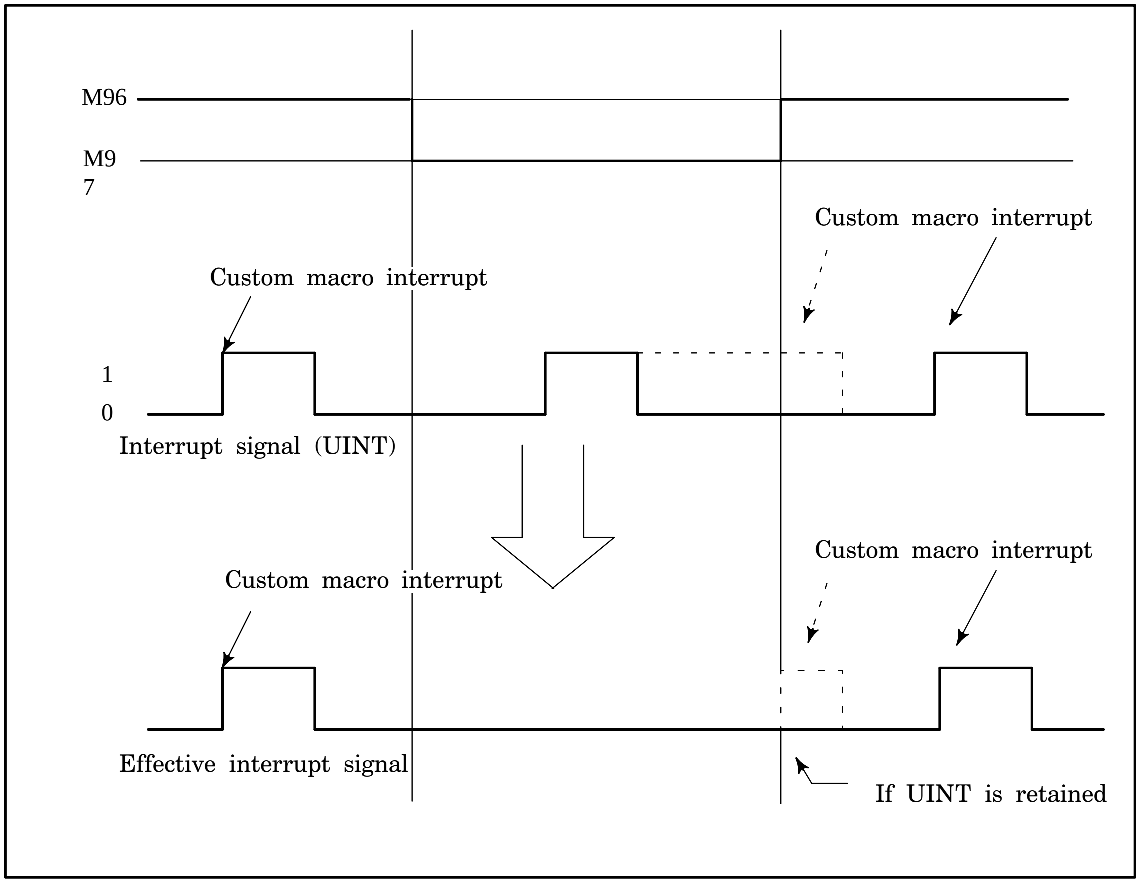


Fig. 17.6.1 Relations between M96, M97, and interrupt signal

The interrupt signal (UINT) becomes effective after M96 is entered.

If the signal is input in the M97 mode, it is ignored. But if the signal is held until M96 is specified, the custom macro interruption is immediately started by M96 specification in case of the status trigger method. In case of the edge trigger method, the custom macro interruption is not started by this specification.

17.6.2 Detailed descriptions

(1) Subprogram–type interrupt and macro–type interrupt

The custom macro interrupt method includes subprogram type and macro type. One of the two types is selected by parameter MSB (No.7002#1).

(a) Subprogram type

The interrupt program is called out as a subprogram. That is, the local variable level does not change before and after the interruption. Also, this interruption is not added to the degrees of multiple subprogram calling.

(b) Macro type

The interrupt program is called out as a custom macro. That is, the local variable level changes before and after the interruption. Also, this interruption is not added to the degrees of multiple macro calling.

A subprogram call and a custom macro call executed in the interrupt program are added to their respective degrees of multiple calling.

The execution program cannot pass arguments in the custom macro interruption, even if it is macro type.

(2) Custom macro interruption control M codes

The custom macro interruption is, in principle, controlled by M96 and M97. However, they can be used for other uses (M functions, macro M code call–out, etc.), depending on the machine manufacturer. Therefore, the parameter which can set these M codes is available MPR (No.7002#2).

When this parameter is designed to set the M codes, the following:

· Set the M code enabling custom macro interrupt for parameter 7033.

· Set the M code disabling custom macro interrupt for parameter 7034.

When parameter MPR is designed for not setting of M codes, M96 and M97 become the custom macro control M codes, regardless of the contents of Nos.7033 and 7034.
In either case, the custom macro interruption control M codes are processed internally and are not output externally.

It is not preferable, from the viewpoint of program compatibility, that M codes other than M96 and M97 control custom macro interruption.

(3) Custom macro interruption and NC command

Two types of custom macro interruption are considered: interruption performed by stopping an NC command being executed prematurely; and interruption performed after completion of that block. Therefore, the parameter MIN (No. 7002#4) which selects between the interrupts during, and after completion of, the block is available.

(a) In case of TYPE–I (Interrupt during execution of a block)

(i) Entering the interrupt signal (UINT) interrupts the movement or dwell being executed and executes the interrupt program;

(ii) If NC statements exist in the interrupt program, the interrupted block of command disappears and interrupt programs are executed. When having returned to the original program, the execution continues from the next block on;

(iii) If NC statements do not exist in the interrupt program, when having returned to the original program by M99, the execution continues from the interrupted command on.

(b) In case of TYPE–II (Interrupt after completion of a block)

(i) If the executing block is not of a cycle block in which the block is divided into several blocks like a drilling canned cycle or automatic reference position return (G28), the following are performed.

When an interruption signal (UINT) is input, the macro statements in the interrupt program are monetarily executed up to the first NC statement. The NC statement is executed after completion of a block being executed.

(ii) If the executing block is of a cycle block in which the block is divided into several blocks, the following are performad.

When an interruption signal (UINT) is input, the macro statements in the interrupt program are executed when the last move of the cycle operation has started. The NC statement is executed after completion of all the cycle operation.

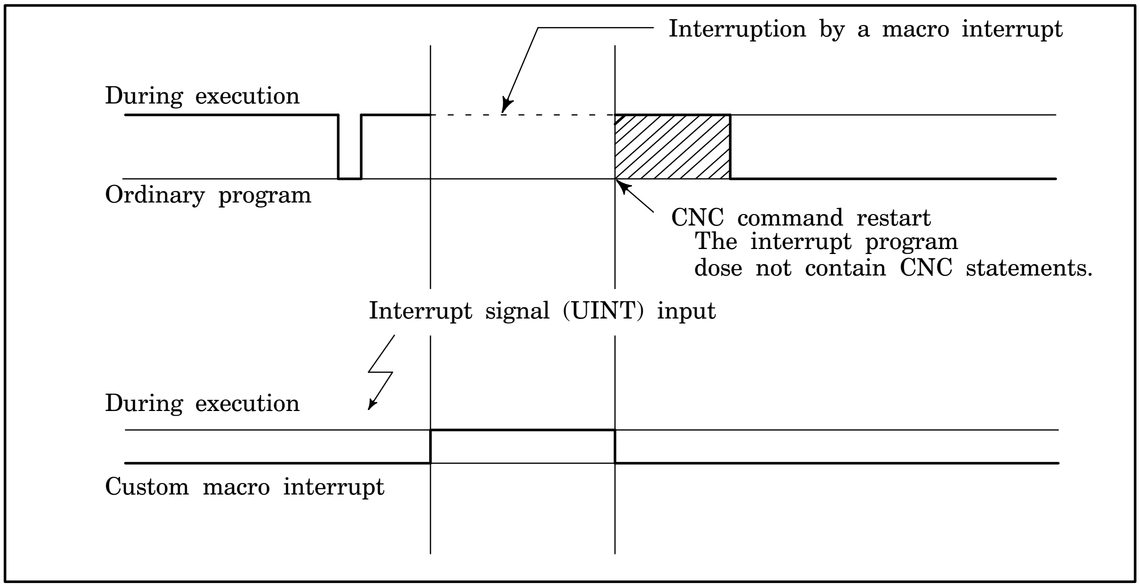


Fig. 17.6.2 (a) In Case of Type I (Interruption during execution of block)

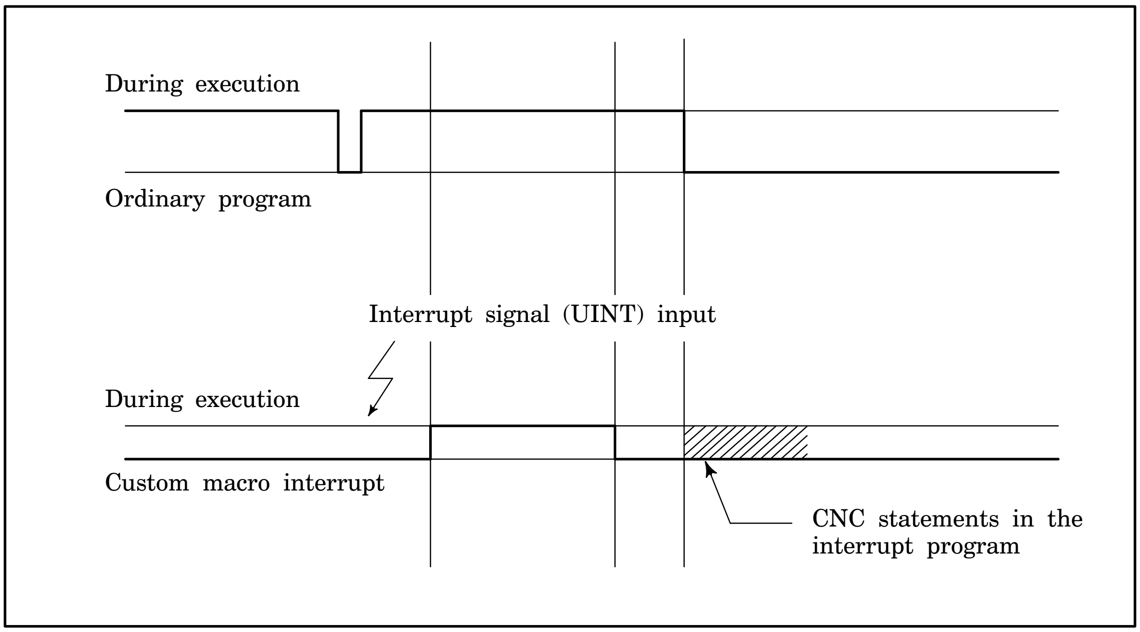


Fig. 17.6.2 (b) In Case of Type II (Interruption during execution of block)

(4) Effective/ineffective conditions of custom macro interruption

The interrupt signal becomes effective when an M96 block makes custom macro interruption effective. It becomes ineffective when a block which includes an M97 has started execution.

The interrupt signal is not effective during execution of an interrupt program. After returning from the interrupt program, when the next block to the interrupted block in the main program has been started, the interrupt signal becomes effective.

In case of Type–I, if interrupt program is made of only macro statements, the interruption becomes effective from the time when the interrupted block is started execution after returning from the interrupt program.

(5) Custom macro interrupt during execution of a block of a cycle operation

(a) Operation for Type – I

The interrupt program is executed by interrupting the block being executed.

If there is no NC statements in the interrupt program, the cycle operation is started execution after returning to the main program. If there is an NC statement, the interrupted cycle is discarded and the next block is executed.

(b) Operation for Type – II

Macro statements up to the first NC statement in the interrupt program are executed when the last move of the cycle operation has started.

The NC statement is executed after completion of the cycle operation.

(6) Acceptance of custom macro interrupt signal (UINT)

There are two methods of accepting the custom macro interrupt signal (UINT) status trigger and edge trigger. The status trigger makes the signal effective when it is on. The edge trigger makes it effective when it turns from off to on.

The parameter TSE (No. 7002#3) selects one of the two types.

When this parameter specifies the status trigger, if the interrupt signal is "1" when it has become effective, the custom macro interrupt occurs. Therefore, while the signal is "1", the interrupt program can be executed repeatedly.

When this parameter specifies the edge trigger, the interrupt signal becomes effective only when it turns on, and the interrupt program is terminated in a moment (Programs of macro statements alone, etc.). Therefore, when the status trigger is not suitable, or when the custom macro interrupt is to be performed only once through the whole program (the interrupt signal stays "1"), this type may be used.

Either type will provide the same effect in practice, except for special uses, such as noted above. (Both types take same amount of time from the input of the signal to the execution of the custom macro interrupt.)

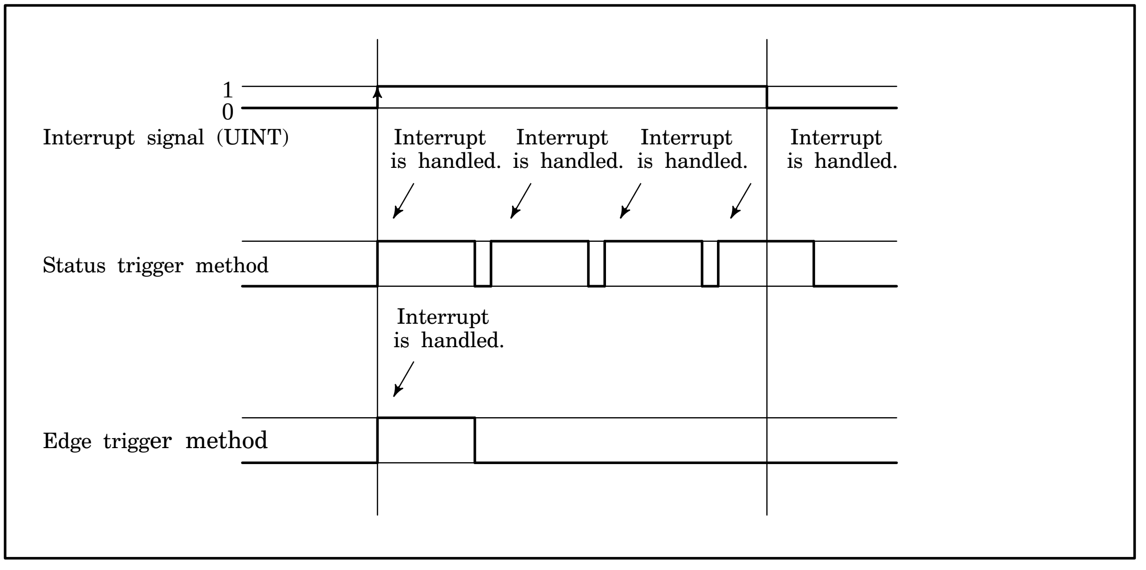


Fig. 17.6.2 (c) Custom macro interrupt signal

A custom macro interrupt is issued when the status–trigger interrupt signal is on. A custom macro interrupt is issued when the edge–trigger interrupt signal rises.

The example above shows that four interrupts are executed in the former type; only one in the latter type.

(7) Return from custom macro interrupt

M99 commands return from the custom macro interrupt to the original program. Also, address P specifies a sequence number in the program to be returned to. In this case, the corresponding program is searched from its beginning and the sequence number appearing first is returned to.

Another interrupt does not occur during execution of a custom macro interrupt program. M99 clears this situation. When M99 is specified independently, it is executed before completion of the preceding command. Therefore, the custom macro interrupt also becomes effective for the last command of the interrupt program. If this is inconvenient, control the custom macro interrupt by M96 and M97 in the program.

NOTE The block with M99 alone, which consists of addresses O, N, P, L, or/and M alone, is assumed to be the same block as the preceding block in programming. Therefore, the single block stop is not performed. In programming, – and – below provide the same effect. (They differ in whether G×× is executed or not before M99 is known.)

Gxx Xxxx ;
M99 ;
Gxx Xxxx M99;

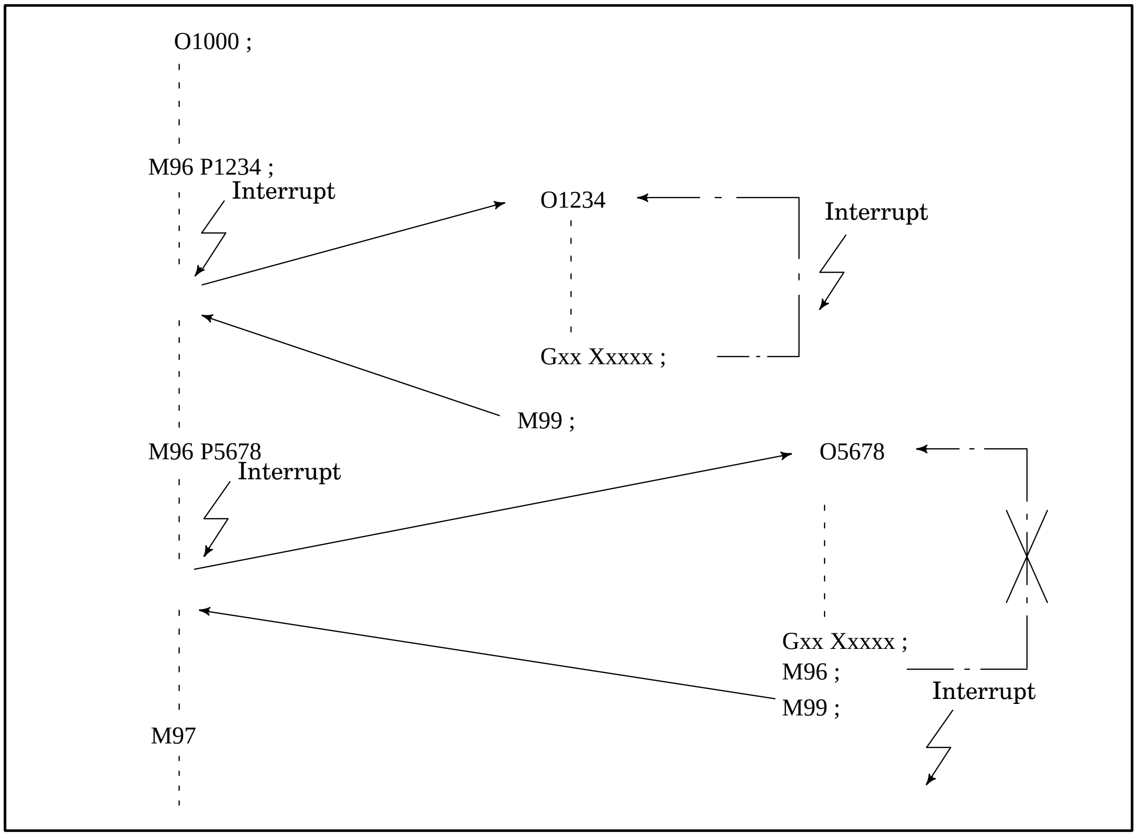


Fig. 17.6.2 (d) Return from custom macro interrupt

A custom macro interrupt does not overlap with another which is being executed. That is, if an interrupt occurs, another is prohibited automatically; if M99 is executed, the custom macro interrupt becomes effective again. Since the M99 block is executed independently before completion of the preceding block, it can also interrupt the Gxx block of O1234 as shown in Fig. 17.6.2 (d). If the signal is input, the O1234 program is executed again. On the other hand, the O5678 program is controlled by M96 and M97 and the interrupt becomes effective after it returns to O1000.

(8) Custom macro interruption and modal information

The custom macro interrupt, unlike the ordinary program call, is initiated by the interrupt signal (UINT). Therefore, it is not desirable that the original program is affected if modal information is modified in the interrupt program. Thus, even if moda

l information is modified in the interrupt program, when the original program is returned to by M99, the modal information is restored to that before interruption.

When the interrupt program returns to the original program by M99 Pxxxx, modal information can be controlled in the program. Therefore, the modal information modified in the interrupt program is taken over. (Conversely, when modal information in the original program is to be taken over, movement after return may vary depending on the modal information at the time of interruption.) Therefore, in this case:

(1) Specify modal information in the interrupt program or

(2) Re specify necessary modal information at the returned point.

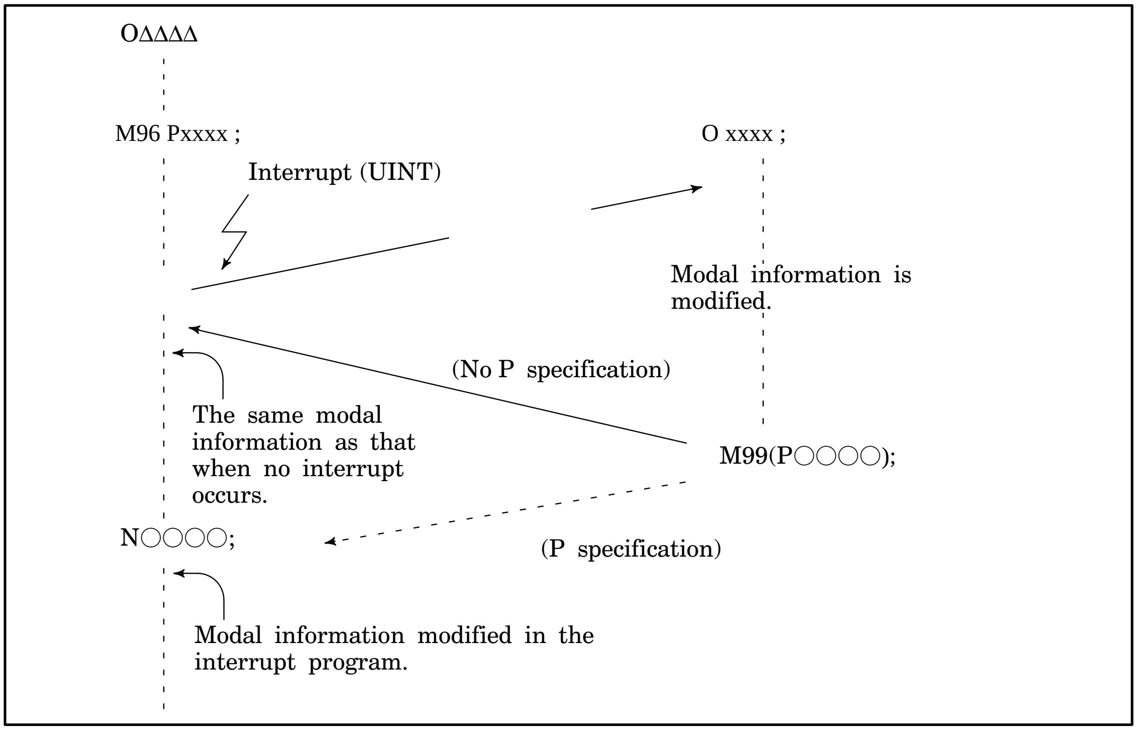


Fig. 17.6.2 (e) Custom macro interruption and modal information

(a) Modal information for return by M99

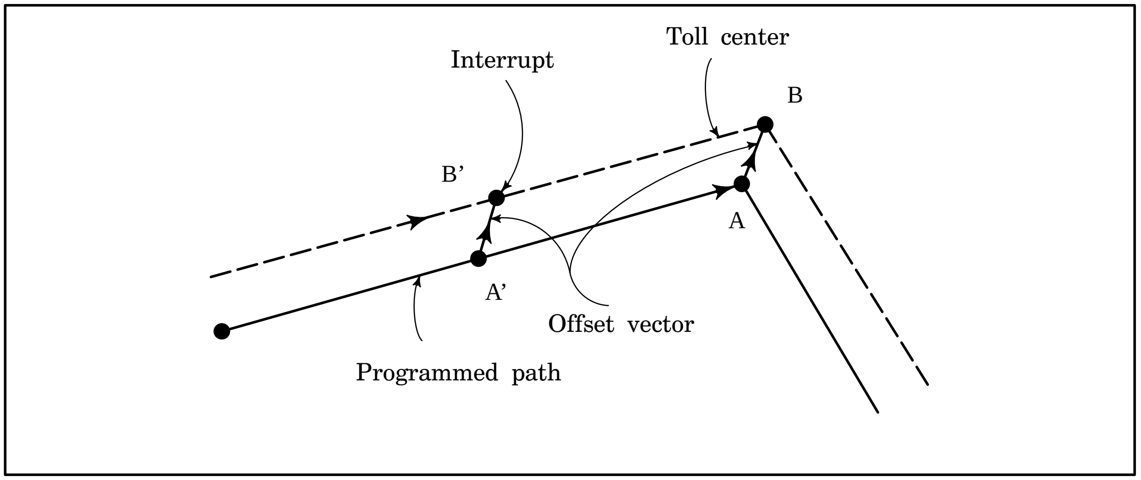
The modal information before interruption remains valid, and the modal information modified in the interrupt program is neglected.

(b) Modal information for return by M99 P○○○○

The modal information modified in the interrupt program also becomes valid after return.

Modal information of the interrupted block is stored in system variables #4001 to #4120. (These values are not changed when model information is changed in the interrupt program.)

(9) Values of system variables (positional information) in the interrupt program



(a) Coordinate of position A is stored in #5001 and later until the first NC statement is encountered.

(b) Coordinate of position A' is stored in #5001 and later after the first NC statement (without movement) is encountered.

(c) Machine coordinate and work coordinate of position B' are stored in #5021 and later and #5041 and later, respectively.

(10) Custom macro interruption and custom macro modal call

The custom macro call is in the cancelled condition (G67) when the interrupt program is called for after the interrupt signal (UINT) is input.

However, if G66 is specified in the interrupt program, it is effective.

When the execution returns from the interrupt program by M99, the modal call status returns to the one effective before interruption. If the execution returns by M99Pxxxx;, the modal call status remains as it is effective in the interrupt program.

17.6.3 Parameters related to interrupt custom macros

The parameters related to interrupt custom macros are parameters 7002, 7033, and 7034.

For these parameters, see the parameter list.

NOTE	M97 must be an unbuffered M code. Specify 97 in one of parameters 2411 to 2418.

18.FUNCTIONS FOR INCREASING THE MACHINING SPEED

18.1 High–Speed Machining Function (G10.3, G11.3, G65.3)

Prior to machining, a machining program can be preprocessed and stored in memory. When machining is performed, the program is called and executed. This speeds up the machining operation.

(1) High–speed machining data

A machining program preprocessed and converted for high–speed machining is called high–speed machining data. Multiple items of high–speed machining data can be stored in memory, and each item is identified by a data number (equivalent to a program number in a general machining program).

(2) Cluster

A group of blocks in a machining program, starting with the high–speed machining start block (G10.3) and ending with the high–speed machining end block (G11.3), is called a cluster. A single machining program may contain any number of clusters. However, there are some restrictions on the commands coded in clusters. These restrictions are explained later.

A single item of high–speed machining data can contain one or more clusters. Each cluster in the machining data is identified by an identification number (equivalent to a sequence number in a general machining program). Cluster numbers can be assigned arbitrarily. When a cluster number is registered, no duplication check is performed. Note that, however, if two or more clusters having the same identification number are registered within the same high–speed machining data, the desired cluster cannot be called correctly in later machining operation.

(3) Machining modes

High–speed machining has two modes: The registration mode and the call mode. The registration mode performs machining while creating high–speed machining data. The call mode simply machines a workpiece by calling high–speed machining data which was created in advance. The modes are switched by setting HSO (bit 6 in parameter No. 0011).

(a) Registration mode

If the following is specified in a program running in the registration mode (setting HSO = 1), all commands in that entire cluster are converted to high–speed machining data and registered in part program storage. Then, the created high–speed machining data is used to execute the commands in high–speed operation.

```
      .
      .
G10.3 L1 Pp Qp ;
(block specification 1) ;
(block specification 2) ;
      .
      .
(block specification 3) ;
G11.3
      .
      .
```

(i) G10.3 : High–speed machining start G code

(ii) L1 : Specify the registration mode.

L1 ..... Create high–speed machining data for a specified number, and register the cluster in it.

L2 ..... Add the cluster to the end of existing high–speed machining data with the specified number.

(iii) Pp : High–speed machining data number (1 to 9999). If it is omitted, the same number as the current main program is assumed.

(iv) Qp : Cluster identification number (1 to 9999)

(v) G11.3 : High–speed machining end G code

If high–speed machining data having the number specified by the G10.3 L1 block already exists, an alarm may occur, or the existing data may be erased and the new data registered. One of these responses can be selected by setting PRD (bit 0 in parameter No. 7607).

If there is no high–speed machining data having the number specified by the G10.3 L2 block, an alarm occurs.

(b) Call mode

If G10.3 L1 or G10.3 L2 is specified in a program being executed in the call mode (setting HSO = 0), the commands up to the G11.3 block are read and skipped, and a cluster of specified high–speed machining data created in advance is called for high–speed machining.

If G10.3 L3 is specified in a running program, the commands are read and skipped, and high–speed machining is performed in the same manner as explained above, regardless of the value of HSO. Therefore, the registration and call modes can be switched by changing the L command with a switch or by other means while HSO is always set to 1.

(4) Call command

By specifying the following command, one cluster of high–speed machining data can be called and executed at high–speed as many times as specified by the repetition count, regardless of the registration or call mode.

G65.3 Pp Qq L1 ;

(i) G65.3 : G code specifying a call

(ii) Pp : High–speed machining data number (1 to 9999). If it is omitted, the same number as the current main program is assumed.

(iii) Qq : Cluster identification number (1 to 9999)

(iv) L1 : Repetition count (1 to 9999). If it is omitted, L1 is assumed.

When each cluster of an original machining program is replaced with G65.3 in advance and the resultant program is executed in the registration mode, the time taken to read and skip G10.3 to G11.3 in each cluster can be eliminated, thereby greatly enhancing the efficiency of machining.

(5) Data deletion command

The following command deletes registered high–speed machining data:

G10.3 L0 Pp

(i) G10.3 L0 : Data deletion command

(ii) Pp : High–speed machining data number (1 to 9999). If it is omitted, the same number as the current main program is assumed.

A deletion command can be used only in the registration mode (HSO = 1). If it is specified in the call mode or if data for the specified number is not found, the command is ignored.

(6) Relationship between machining program registration and memory capacity

If the high–speed machining function is provided, the number of programs registered in normal part program storage is reduced to 87.5%. The length of part program storage does not vary unless high–speed machining data is created. If high–speed machining data is created, however, the storage capacity decreases by that amount of data. The reduced amount depends on the commands specified. For a block specifying only simultaneous three–axis linear movement, the storage capacity would be reduced by approximately 0.06 m per block.

WARNING 1 Note the following in the registration and call modes:
A single–step operation cannot be done.
Operation can be stopped temporarily by feed hold, but the mode cannot be switched from that status.
Operation can be stopped by reset operation, but once operation stops, it cannot be continued.
Do not perform background editing. (An alarm may occur.)
An interrupt–type custom macro cannot be used.
Machining profiles remain unchanged even when offsets are changed.
Do not change offsets in the workpiece coordinate system.
The cutting feedrate cannot be clamped by setting parameter No. 1422.
During cutting, dry run/external deceleration is set to zero.
Specification of F1 line feed is not permitted.
The function for acceleration/deceleration before interpolation is invalid.
Automatic corner override is not applied.

WARNING 2 A custom macro can be used. When it is registered, it is called and executed. During operation, the macro in the registered executable format is only executed. A continuous–state call must be specified and canceled within a cluster. In the G66 mode, neither G10.3 nor G11.3 can be specified.

NOTE 1 Each of G10.3, G11.3, and G65.3 must be specified alone in a single block.

NOTE 2 Only the following addresses can be specified in registered blocks:
G : G00, G01, G02, G03, G33, G40, G41, G42, G65, G66, G67, G90, G91
F, E
M, S, T, and the second miscellaneous function (Commands for tool life management and constant surface speed control, and commands causing a reset, such as M02 and M03, can not be specified.)
M98, M99
Axis command, I, J, K, R
O, N (During high–speed machining, the indications of O and N are not updated.)
P, L

NOTE 3 All traveled distances are indicated in the incremental mode. Even when an absolute command is specified at registration, the command results in an incremental movement at execution.

NOTE 4 Cutter compensation C must be started and canceled within a cluster. In the cutter compensation C mode, neither G10.3 nor G11.3 can be specified.

NOTE 5 Within a cluster, other high–speed machining data cannot be called by using G65.3.

18.2 Multibuffer (G05.1)

While executing a block, the CNC usually calculates the next block to convert it to an applicable data form for execution (executable form). This feature is called buffering. The multi–buffer function increases the number of blocks buffered to five. The number of blocks can be increased to 15 with an option. Consequently, even if two or more small blocks are in succession, an interruption in the pulse distribution between blocks is prevented.

NOTE 1 Macro statements do not count as blocks mentioned here. A macro statement specified immediately before NC statements to be converted to an executable form is executed when the CNC statements are converted to an executable form.

Example

```
N010 G01 X100.0 F100 ;
N020 G90 X200.0 ;
N021 #1=100.0
N022 #2=200.0
N030  X300.0 ;
N031  #3=300.0 ;
N040  X400.0 ;
N050  X500.0 ;
N060  X600.0 ;
N070  X700.0 ;
```

When the N010 block is being executed, blocks up to N060 have been buffered, and the N021, N022, and N031 macro statements have been executed. When execution of the N020 block starts, the N061 macro statement is executed, and N070 is buffered.

NOTE 2 If many small blocks are specified in succession, an interruption in pulse distribution may occur between blocks. Such an interruption can be prevented if the time for executing blocks read in advance is longer than the time required for advance reading of the following block. (Up to 15 blocks are read in advance if there is sufficient time.)

NOTE 3 The blocks mentioned above include cycles created internally in the CNC. Fifteen blocks including canned cycles and cycles created by cutter compensation C are buffered.

(1) Setting the multibuffer mode

The multibuffer mode is set on and off by using single–shot G code G05.1 as follows:

G05.1P1 ;: Multibuffer mode off

G05.1 ;: Multibuffer mode on

Do not specify addresses other than O and N in this block.

When multi–buffer mode is set on in a block, the five blocks (15 blocks when the option is used) following the block are buffered.

If the multibuffer mode is disabled, the system reads the commands in advance in the normal way.

(Sample program)

```
N1 G92 X0Y0 G01 ;
N2 G05.1 ; (Multibuffer mode on)
N3 G42 G90 X1000 Y1000 F1000 D01 ;
N4 G68 R–3000 ;
....
....
N20 G05.1P1 ; (Multibuffer mode off)
N21 G49 G40 G90 X0 Y0 M30 ;
```

Whether the multibuffer mode is set on or off at power–on or immediately after a clear operation can be selected by bit 6 in a parameter No. 2401.

(2) Restrictions on the multibuffer

Some G codes and M codes are not buffered even in the multibuffer mode. Thus, when such a G or M code is specified in a block, buffering for the following blocks is suppressed until execution of that block is completed. This occurs not only in the multibuffer mode, but also in normal automatic operation.

The following lists the G codes and M codes that suppress buffering.

Table G codes and M codes that suppress buffering

Code	Function
G10	Data setting
G10.1	PMC data setting
G11	Data setting mode cancel
G20	Imperial input (inch)
G21	Metric input (mm)
G22	Stored stroke check on
G23	Stored stroke check off
G31	Skip function
G31.1	Multistage skip function 1
G31.2	Multistage skip function 2
G31.3	Multistage skip function 3
G31.4	Multistage skip function 4
G37	Automatic tool length measurement
G28	Reference position return
G30	Return to second, third, or fourth reference position
G53	Machine coordinate system selection
M00	Program stop
M01	Optional stop
M02	End of program
M30	End of program

In addition, eight M codes to suppress buffering can be set with parameters.

NOTE 1 Note the following when using system variables 5021 to 5055 in the multibuffer mode:
When current position information is read through a system variable in the multibuffer mode, specify G53 alone in the previous block.

Example

```
#3 = 5003 ;
G00X# 24 Y# 25 ;
G53 ; (To read #5021 to #5023, stop buffering.)
G91X[10.0 – # 5021] X[20.0 – # 5022]
      Z[30.0 – # 5023]
```

Specifying G53 suppresses advance reading both in the multibuffer mode and in normal operation.

NOTE 2 Indication of a canned cycle
On the program screen or program check screen, the block being executed is normally marked by > at the left of the block. When a canned cycle with L specified is executed in the multibuffer mode, the mark > is sometimes not indicated. This only concerns the display, however, and it does not affect operation.

NOTE 3 Processing performed at buffering
The following processes performed at buffering are also performed at buffering in the multibuffer mode:

(1) Tool selection according to tool life management

(2) Input of the park signal

Example

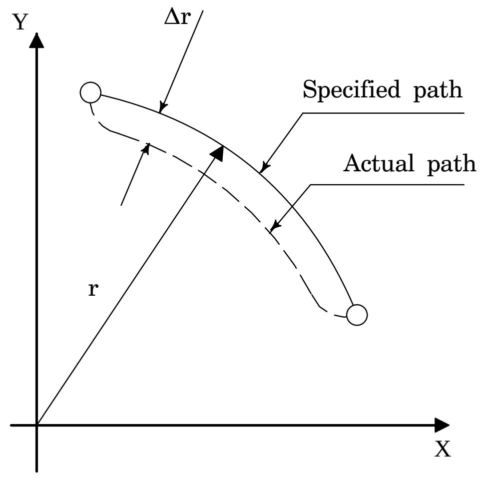
```
N1 G01G91X100.0
N2 Y100.0 ;
N3 X100.0 ;
N4 Y100.0 ;
N5 X100.0 ;
N6 M06T101 ;
```

Suppose that blocks up to N6 have been buffered during execution of the N1 block. In this case, even if the end of the useful life of group 01 is reached while the N1 to N5 blocks are being executed, the next tool is not used for N6 because N6 is already buffered.

NOTE 4 In the multibuffer mode as well as other modes, the single–block stop, feed hold stop, and restart operations are enabled.

18.3 Feedrate Clamp Based on Arc Radius

In circular interpolation, especially when high–speed arc cutting is performed, the actual tool path deviates from the specified arc in the radial direction. The approximate error is obtained by the following equation:



Δr	:	Maximum radial error (mm)
v	:	Feedrate (mm/s)
r	:	Acr radius (mm)
T1	:	Time constant of exponential acceleration/deceleration of cutting feed (s)
T2	:	Servo motor time constant (s)

$$\Delta r = \frac{1}{2}\left(T_1^2 + T_2^2\right)\frac{V^2}{r} \quad \text{.................................... } \textbf{(Equation 1)}$$

In actual machining, the radius r of the arc to be machined and the permissible error r are given. So the maximum allowable feedrate, v mm/min, is found by equation 1.

If the specified feedrate would cause an error exceeding the permissible error in the radial direction to an arc having a programmed arbitrary radius, the arc–radius–based feedrate clamp function clamps the arc cutting feedrate automatically.

If the permissible error r is set, the maximum allowable feedrate v when the arc radius is r is obtained from equation 2.

$$\Delta r = \frac{1}{2}\left(T_1^2 + T_2^2\right)\frac{V^2}{R} \quad \text{.................................... } \textbf{(Equation 2)}$$

When the arc radius is r, the maximum allowable feedrate v to set the permissible error to r is obtained from equation 1. Therefore, equations 1 and 2 give the following:

$$\frac{1}{2}\left(T_1^2 + T_2^2\right)\frac{V^2}{r} = \frac{1}{2}\left(T_1^2 + T_2^2\right)\frac{V^2}{R}$$

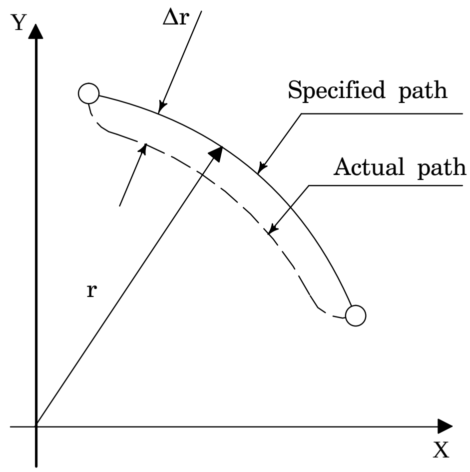
$$\therefore v = \sqrt{\frac{r}{R}}\ V \quad \text{.................................... } \textbf{(Equation 3)}$$

If a certain arc radius R and its maximum allowable feedrate V are set as parameters in advance, the maximum allowable feedrate v for an arc with programmed radius r is obtained from equation 3. If the specified feedrate exceeds feedrate v, the feedrate is automatically clamped to v.

The maximum allowable feedrate v obtained from equation 3 decreases as a smaller value is specified for the arc radius. To prevent the maximum allowable feedrate from decreasing excessively, RVMIN in parameter No. 1491 can be used so that the maximum allowable feedrate v is set to RVMIN if the obtained feedrate is lower than RVMIN.

Obviously, if a specified feedrate is less than or equal to the maximum allowable feedrate v obtained from equation 3, the arc is cut at the specified feedrate.

NOTE 1 If the cutting feed linear acceleration/deceleration function is applied, the approximate arc cutting error is found by equation 4.



Δr : Maximum radial error (mm)
v : Feedrate (mm/s)
r : Arc radius (mm)
T1 : Time constant of linear accelaration/deceleration of cutting feed (s)
T2 : Servo motor time constant (s)

$$\Delta r = \frac{1}{2}\left(\frac{T_1^2}{12} + T_2^2\right)\frac{V^2}{r} \quad \text{.................................... (Equation 4)}$$

As indicated by equation 4, equation 3 also holds in linear acceleration/deceleration after interpolation. Thus, the feedrate can be clamped continuously based on the arc radius.

NOTE 2 Equations 1, 2, and 4 indicate the approximate logical error in the NC, and do not indicate the actual error in machining.
Let the error in actual machining be rall. rall is expressed as follows:

$$\Delta rall = \Delta rNC + \Delta r\ machine \quad \text{.................................... (Equation 5)}$$

Δr machine: Error due to the machine used

ΔrNC: Error due to acceleration/deceleration time constant (T1) and servo motor time constant (T2)

Since this function is intended to keep the 1st item at right side of formula–5, to constant level, it has nothing to do with the error resulting from machine.
In addition, the formula–1, formula–2 and formula–4 are approximation formulas and the approximation accuracy may be worsened as the arc radius becomes smaller, the permissible error may not be given even if the feed rate is clamped at the maximum permissible speed (v) found by the formula–3.

18.4 ADVANCED Preview Control

Advanced preview control has been designed for high–speed, high–precision machining. This function reduces acceleration/deceleration delay and servo delay, which increase as the feedrate increases. When this function is used, the tool is moved as specified, and the machining error in circular or corner machining is reduced.

Advanced preview control function is implemented by the following functions:

1. Cook ahead acceleration/deceleration before interpolation (including advance feed–forward) (See 6.4.5)
2. Multibuffer (See 18.2)
3. Feedrate clamp by circular radius (See 18.3)
4. Linear acceleration/deceleration after interpolation (See 6.4.1)

NOTE When advanced preview control is enabled, follow–up control is disabled during servo–off.

(1) Command specification

The advance control mode is selected and canceled by the following commands:

G05.1 Q1; Selects the advance control mode.

G05.1 Q0; Cancels the advance control mode.

The MBF bit (bit 6 of parameter 2401) specifies whether advanced preview control mode is selected at power–on and in the clear state. If the system enters advanced preview control mode when the advance feed–forward function is enabled, the time constant after interpolation is switched from the value in parameter 1622 to the value in parameter 1635.

NOTE 1 If the time constant is the same, the error caused by acceleration/deceleration after interpolation as shown below:

(Error caused by bell–shaped acceleration/deceleration)
< (Error caused by linear acceleration/deceleration)
< (Error caused by exponential acceleration/deceleration)

Either exponential or linear acceleration/deceleration can be selected using parameter 1600. If parameter 1600 is not set, the time constant of linear acceleration/deceleration is applied after interpolation.

NOTE 2 Using advanced preview control function with the feed–forward control function reduces the acceleration/deceleration delay and servo delay. In advanced preview control mode, the feed–forward control function is executed according to the speed specified in the next block, which further reduces the servo delay.

18.5 High–Precision Contour Control

18.5.1 General

The high–precision contour control function allows precise high–speed machining when a free sculptured surface, such as a metal die, is machined using linear interpolation. To achieve greater speed and precision, the function calculates and controls the appropriate feedrate (automatic feedrate control) according to the machining profile.

This function includes the optional multi–buffer function for 15 blocks.

To enter the automatic feedrate control mode, specify the following:

G05.1 Q1 ;

The feedrate is automatically controlled in automatic feedrate control mode by buffering the next 15 blocks (the next 60 blocks when the optional multi–buffer function for 60 blocks is provided). The feedrate is determined by the following conditions. If the specified feedrate exceeds the value determined by the conditions, acceleration/deceleration before interpolation is performed to reach the determined feedrate.

(1) Change in speed on each axis at corners and specified allowable speed change

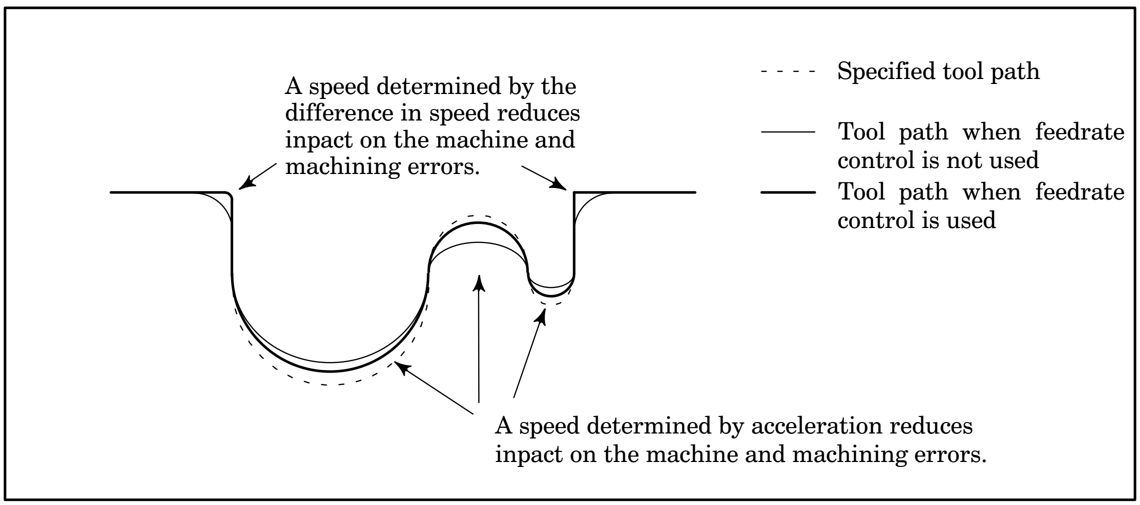
(2) Expected acceleration on each axis and specified allowable acceleration

(3) Expected variations in cutting load from movement along the Z–axis

If an appropriate feedrate is determined and acceleration/deceleration is performed according to these conditions, the impact on the machine and the machining errors liable to be produced when the direction in which the tool moves changes substantially are decreased. As a result, precise high–speed machining is enabled.

The feature of acceleration/deceleration before interpolation is used for automatic feedrate control. Since extending the time constant does not produce a machining error, machining can be done with small impact and high precision.

A specific time constant for acceleration/deceleration after interpolation is provided for the automatic feedrate control mode. By setting the time constant for the automatic feedrate control mode to a small value, the machining error due to acceleration/deceleration delay is reduced.



18.5.2 Automatic feedrate control

To enter the automatic feedrate control mode, specify the following:

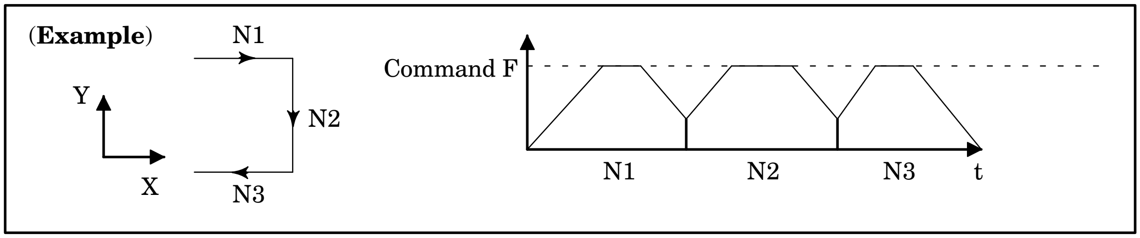
G05.1 Q1 ;

To cancel the automatic feedrate control mode, specify the following:

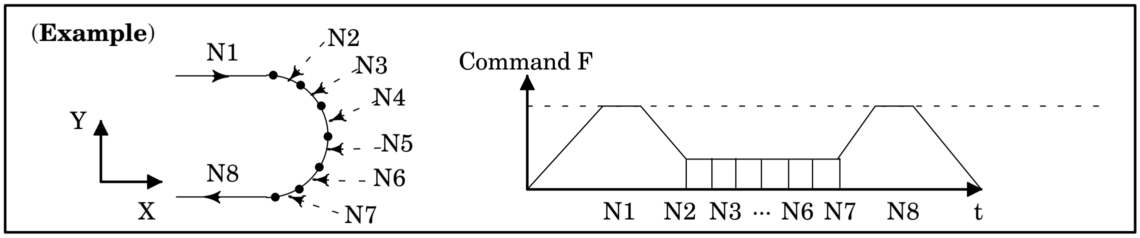
G05.1 P1 Q0 ;

In the automatic feedrate control mode, the feedrate is controlled as follows:

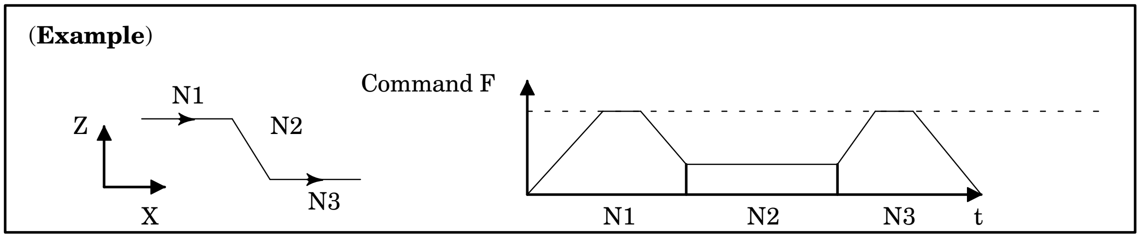
(a) The feedrate is decelerated so that, at a corner, it reaches the target feedrate determined with the feedrate difference conditions for axes at the corner.



(b) The feedrate is decelerated so that, in a block, it is equal to or less than that determined with the acceleration conditions for axes at the corners for the start and end points.



(c) The system calculates the feedrate in the block from the downward slope along the Z–axis, and the actual feedrate is decelerated to the calculated feedrate or less.



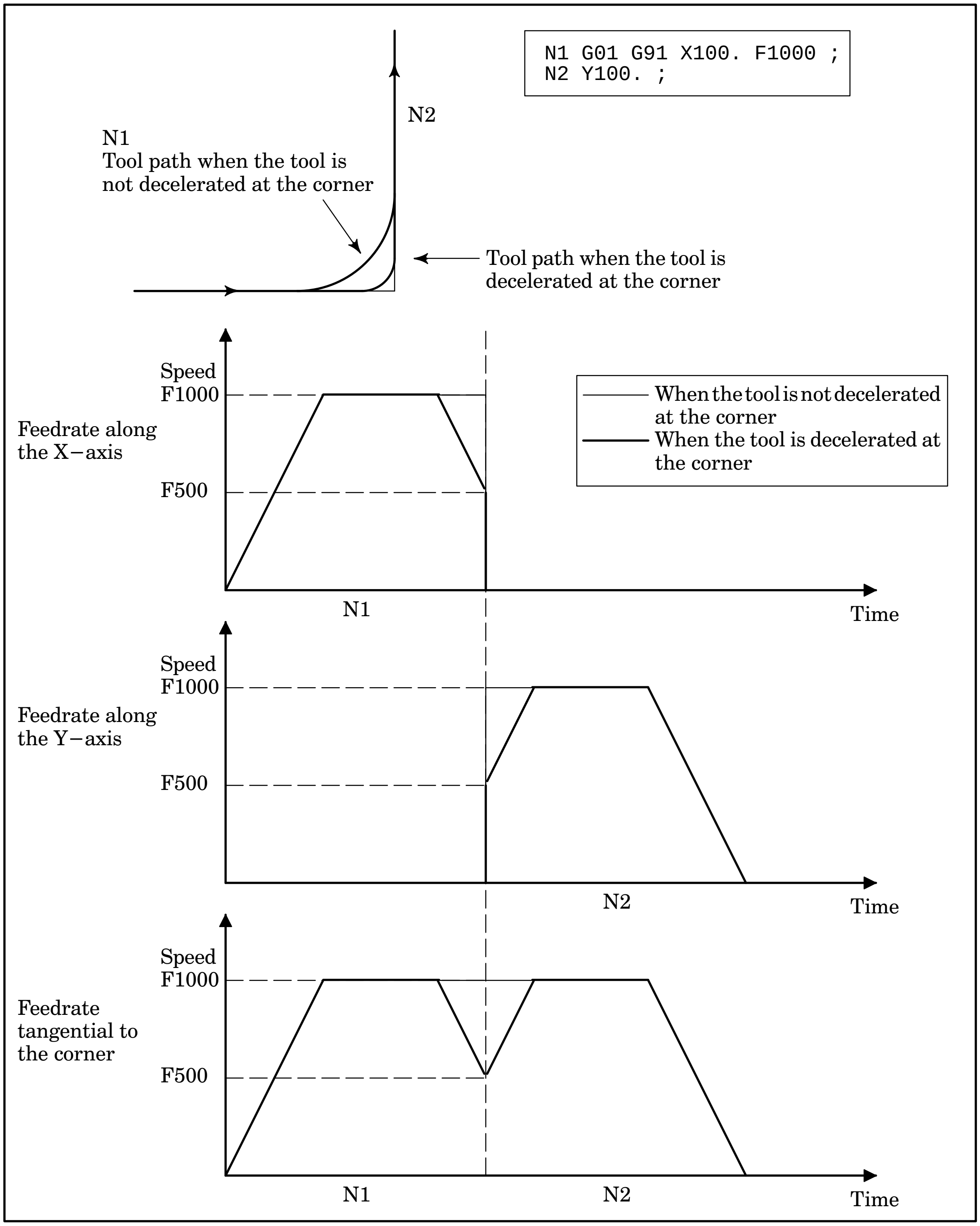
18.5.3 Deceleration at a corner

(1) Determining the feedrate with feedrate difference conditions

Feedrate at a corner is determined as follows:

Changes in feedrates along axes (Vc[X], Vc[Y], ...) are compared with Vmax specified in parameter No. 1478. Let the specified feedrate be F. If the largest Vc exceeds Vmax, let Rmax be (the largest Vc)/Vmax. Then, the target feedrate at the corner Fc is F/Rmax. The feedrate is decelerated to this value.

For example, if a tool moving at a feedrate of 1000 m/min along the X–axis changes its direction by 90 degrees to move along the Y–axis and a permissible error in the feedrate is set to 500 mm/min, the tool is decelerated as shown below:



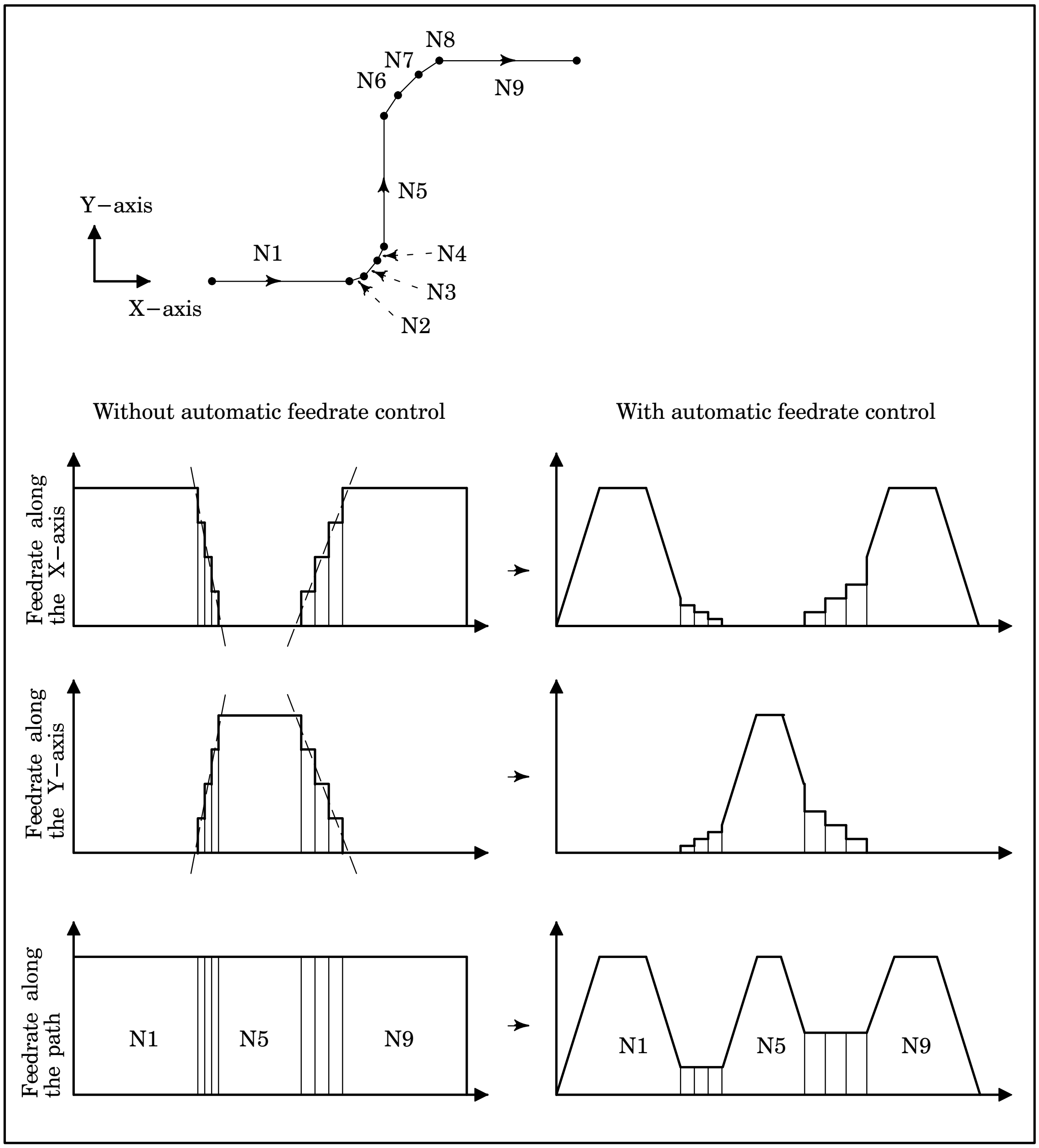
(2) Determining the feedrate based on the acceleration along each axis

When a curve consists of consecutive minute straight lines as shown below, the change in the feedrate along each axis at a corner is not large. Deceieration determined with the feedrate difference conditions is not activated. Small feedrate differences continue a large acceleration is generated along each axis as a whole. In this case, the following deceleration is applied to prevent machining errors and impact on the machine caused by too large an acceleration. The feedrate is decelerated to a target feedrate so that the acceleration calculated by the equation below is equal to or less than the specified acceleration for all axis.

$$\text{Acceleration along each axis} = \frac{\text{Change in the feedrate at a corner for each axis}}{\max = \left(\dfrac{\text{Distance traveied in the preceding block}}{F} , \dfrac{\text{Distance traveied in the next block}}{P} \right)}$$

The target feedrate is determined for each corner. The smaller of the feedrates calculated at the start and end points in a block is then used.

Example The feedrate is decelerated at cornrrs N2 to N4 and N6 to N8 because the corresponding acceleration Islopsof the dotted lines in the following feedrate graphs) is too large.

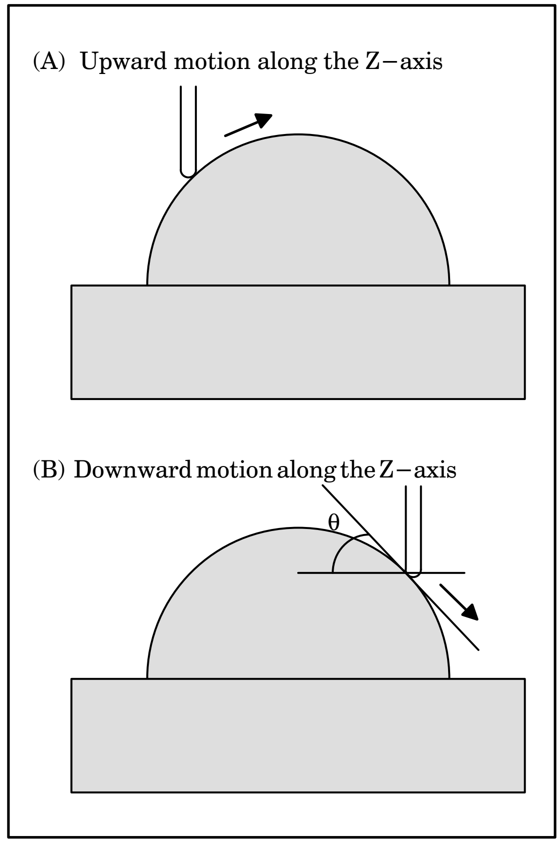


(3) Determining the feedrate based on the cutting load

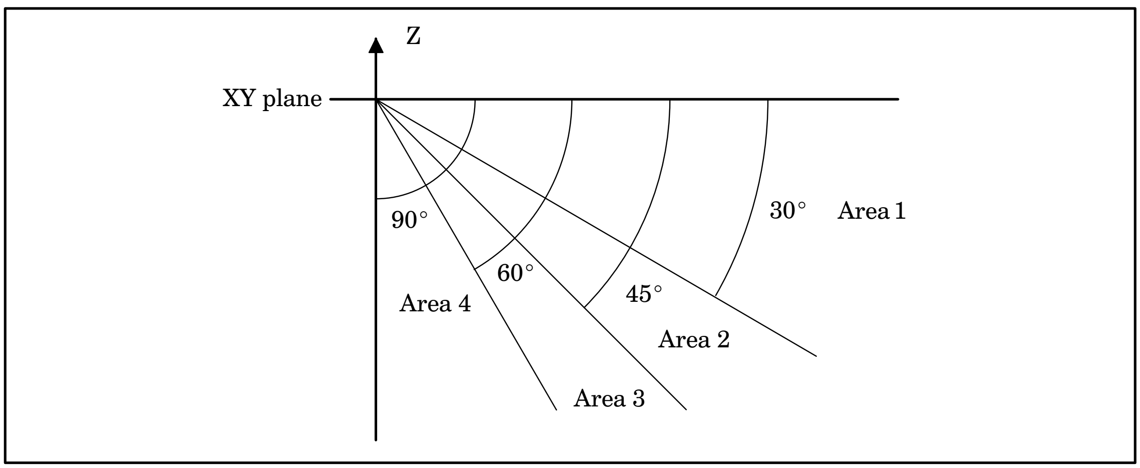
In the cutting operation shown in the figure at the right, when workpiece is machined along the Z–axis as shown in (B), the bottom of the cutter may be used for machining, which depends on the depth of cut. Consequently, a larger cutting resistance than for (A) may be generated.

For this reason, automatic feedrate control uses the direction of tool movement along the Z–axis as a condition for calculating the machining feedrate.

When the cutter moves down along the Z–axis, the downward inclination (angle of the cutter path to the XY plane) is found as shown in (B). Possible downward inclinations are divided into four areas, and an override value for each area is set with a parameter. (Area 1 has no parameter, and is always 100%.) A feedrate calculated by another feedrate control feature is multiplied by the override value for the area to which downward inclination belongs.



Area 1 $0° \leqq \theta < 30°$
Area 2 $30° \leqq \theta < 45°$
Area 3 $45° \leqq \theta < 60°$
Area 4 $60° \leqq \theta < 90°$



WARNING	The feature for determining feedrate based on the cutting load uses NC commands to identify the direction of motion along the Z–axis. This means that if the operator manually intervenes in Z–axis motion in the manual absolute mode, or if a mirror image is applied to the Z–axis, the direction of the Z–axis cannot be identified. Therefore, do not use these functions when using the feature for determining feedrate based on the cutting load.

(4) Ignoring an F code

In a block subject to automatic feedrate control, all feed commands (F commands) can be ignored by setting a parameter. Here, the feed commands include the following:

(1) Continuous–state F commands before a block subject to automatic feedrate control

(2) F commands and continuous–state F commands in a block subject to automatic feedrate control

The specified F commands and continuous–state F commands are stored internally in the CNC.

When automatic feedrate control becomes invalid within a block, the continuous state of an F command calculated by automatic feedrate control is not used as a continuous–state F command. Instead, the continuous state of an F command mentioned in (1) or (2) above is used.

(5) Other conditions for determining the feedrate

(a) If the calculated feedrate exceeds the upper limit of automatic feedrate control set by parameter No. 7567 or if it exceeds the value specified in an F command, the lower feedrate between the upper limit and the feedrate specified in the F command is selected and the feedrate clamped at that value.

(b) If the difference between the feedrate calculated by automatic feedrate control and the feedrate calculated by the previous block falls within the range of feedrate variation that can be ignored set by parameter No. 7566, the same feedrate as the previous block is used.

18.5.4 Notes

WARNING Cutting feed override and feedrate in dry run are calculated when blocks are buffered. When cutting feed override or feedrate in dry run is changed, the data is ignored for blocks already buffered.

NOTE 1 The optional multi–buffer is prepared for this function.

NOTE 2 The function for determining the feedrate from acceleration is only valid for a block containing the G01 code. In a block containing the G02 or G03 code, the equivalent result is obtained by the function for clamping the feedrate using the radius of a circle. For details, see 3.1, ”Parameters.”

NOTE 3 To use the high–precision contour control function in DNC operation, DNC, bit 0 of parameter No. 0, must be set to 1 to disable high–speed distribution.

NOTE 4 Turn off the automatic feedrate control mode when the following functions are used.

G10.3, G11.3, G65.3 : High–speed machining
G07 : Hypotherical axis interpolation
G07.1 : Cylindrical interpolation
G12.1 : Polar coordinate interpolation
G84.2, G84.3 : Rigid tapping

NOTE 5 A feedrate is determined only in the following modes. In the other modes, the specified feedrates are used.

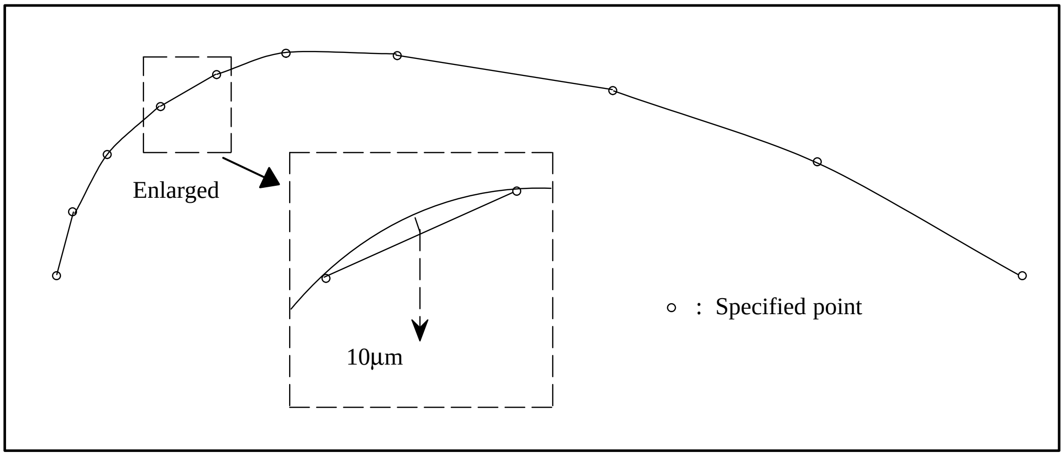
G01 : Linear interpolation
(The feedrate is only determined with the feedrate difference conditions.)
G94 : Feed per minute
G80 : Canceling canned cycles
G64 : Cutting feed

NOTE 6 Even if the machine is in the above modes, automatic feedrate control is not used when the machine moves with the following codes:

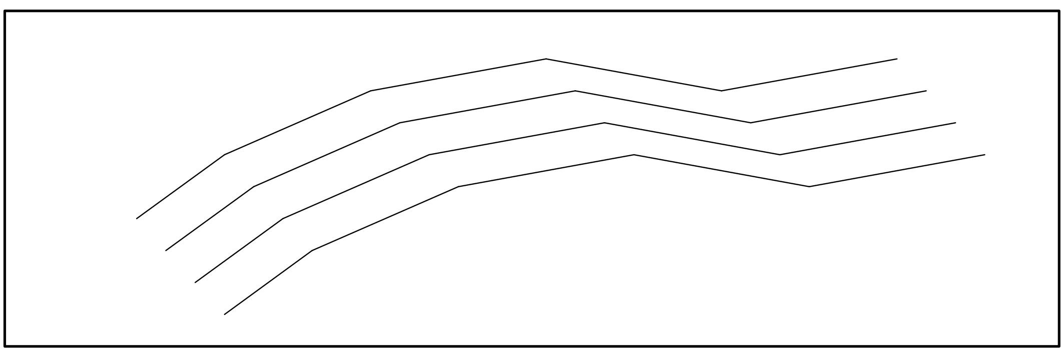
G31 : Skip function
G37 : Automatic measurement of tool length

18.6 Smooth Interpolation Function

To machine a part having sculptured surfaces, such as metal moldings used in automobiles and airplanes, a part program usually approximates the sculptured surfaces with minute line segments. As shown in the following figure, a sculptured curve is normally approximated using line segments with a tolerance of about 10µm.



When a program approximates a sculptured curve with line segments, the length of each segment differs between those portions that have mainly a small radius of curvature and those that have mainly a large radius of curvature. The length of the line segments is short in those portions having a small radius of curvature, while it is long in those portions having a large radius of curvature. The high-precision contour control of the FANUC Series 15 moves the tool along a programmed path thus enabling highly precise machining. This means that the tool movement precisely follows the line segments used to approximate a sculptured curve. This may result in a non-smooth machined curve if control is applied to machining a curve where the radius of curvature is large and changes only gradually. Although this effect is caused by high-precision machining, which precisely follows a pre- programmed path, the uneven corners that result will be judged unsatisfactory when smooth surfaces are required.



Example of uneven surfaces (polygon) resulting from machining that precisely follows the line segments.

Profile	Portions having mainly asmall radius of curvature	Portions having mainly alarge radius of curvature
Example ofmachined parts	Automobile parts	Decorative parts, such asbodyside moldings
Length of line segment	Short	Long
Resulting surfaces produced using high-precision contour-control	Smooth surface even when machin-ing is performe dexactly as specified by aprogram	Uneven surfaces may result when machining is perform-ed exactly as specified by a program

The smooth interpolation function enables high- speed, high- precision machining, as follows:

Either of two types of machining can be selected, depending on the program command.

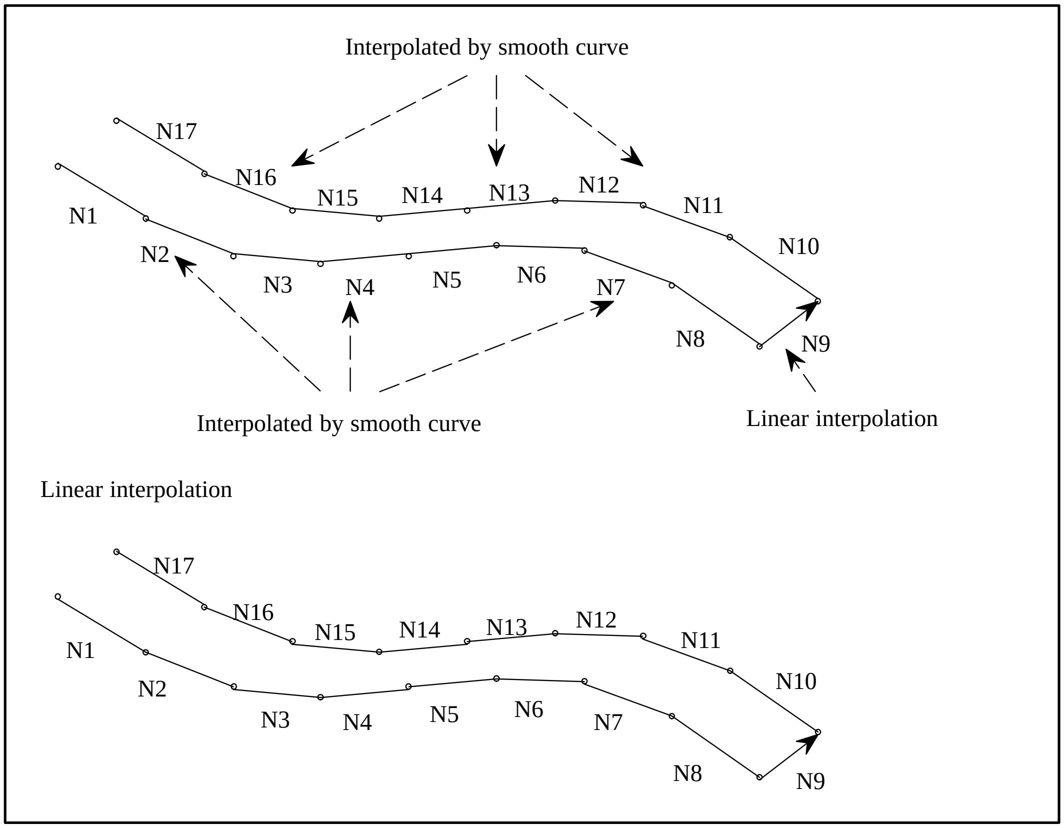
– For those portions where the accuracy of the figure is critical, such as at corners, machining is performed exactly as specified by the program command.

– For those portions having a large radius of curvature where a smooth figure must becreated, points along the machining path are interpolated with a smooth curve, calculated from the polygonal lines specified with the program command (smooth interpolation).

Use the following command to specify smooth interpolation mode:

G05. 1Q2X0Y0Z0 ; The CNC automatically selects either of the above machining types, according to the program command being used. If a block specifies a travel distance or direction which differs greatly from that in the preceding block, smooth interpolation is not performed for that block. Linear interpolation is performed exactly as specified by the program command. Programming is thus very simple.

Example



This function is optional.

18.6.1 Command Method

The command below makes the smooth interpolation mode.

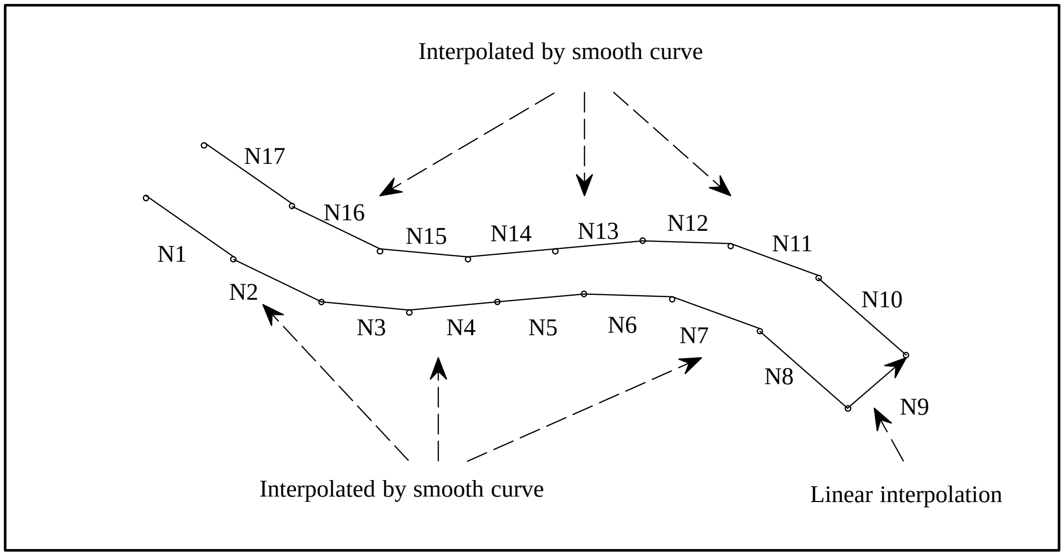
G05. 1Q2X0Y0Z0;

The command below cancels the smooth interpolation mode.

G05. 1Q0;

Example of program:

```
G91 ;                       N10 X- 1000 Z350 ;
G05.1 Q2 X0 Y0 Z0 ;         N11 X- 1000 Z175 ;
N01 G01 X1000 Z- 300 ;      N12 X- 1000 Z- 25 ;
N02 X1000 Z- 200 ;          N13 X- 1000 Z- 50 ;
N03 X1000 Z- 50 ;           N14 X- 1000 Z- 50 ;
N04 X1000 Z50 ;             N15 X- 1000 Z50 ;
N05 X1000 Z50 ;             N16 X- 1000 Z200 ;
N06 X1000 Z- 25 ;           N17 X- 1000 Z300 ;
N07 X1000 Z- 175 ;          G05. 1 Q0 ;
N08 X1000 Z- 350 ;
N09 Y1000 ;
```



(1) Conditions for performing smooth interpolation

Smooth interpolation is performed when all of the following conditions are satisfied:

1) The machining length specified in the block is shorter than the length specified with parameter No.7672.

2) The machining length is other than 0.

3) The modes are:

G01	:	Linear interpolation
G13.1	:	Polar coordinate interpolation cancel
G15	:	Polar coordinate command cancel
G40	:	Cutter compensation cancel (except for 3- dimensional tool compensation)
G64	:	Cutting mode
G80	:	Canned cycle cancel
G94	:	Feed per minute

4) Machining is specified only along the axes specified with G05.1Q2.

5) The block is judged to be unsuitable for smooth interpolation, as performed with the internal algorithm of the CNC.

If one or more of the above conditions is not satisfied, or the block contains the following commands, the block is executed without smooth interpolation. Also, the next block is checked in the same way:

(a) G04 : Dwell

G09 : Exact stop check

G31, G31.1, G31.2, G31.3 : Skip function

G37 : Tool length measurement

(b) M code that does not buffer.

(2) Three-dimensional tool compensation

Three-dimensional tool compensation can be used in smooth interpolation mode.

Three-dimensional tool compensation vectors, however, are calculated automatically.

If the interpolation mode changes from smooth interpolation to linear interpolation during three- dimensional tool compensation, the vector at the end point of the path subject to smooth interpolation is used for three-dimensional compensation in subsequent linear interpolation. If a three-dimensional compensation command is executed, however, the vector specified with the command is used.

Three-dimensional tool compensation vectors in smooth interpolation mode are calculated as follows:

(a) Three-dimensional tool compensation vector at the start point

The vector specified with the previously executed three-dimensional tool compensation command block is used. Say K is the three-dimensional tool compensation vector for command G41 IiJjKkDd;. Its components (Kx, Ky, Kz) will be as follows:

Kx = (i*r) / p

Ky = (j*r) / p,

Kz = (k*r) / p,

where r is the offset corresponding to offset number d specified with the command and $p=(i^2+ j+k)^{1/}$.

(b) Three-dimensional tool compensation vectors at intermediate points

(i) Vector: The vector is on the plane including the point, the previous point, and the subsequent point, and is perpendicular to the line passing through the previous and subsequent points.

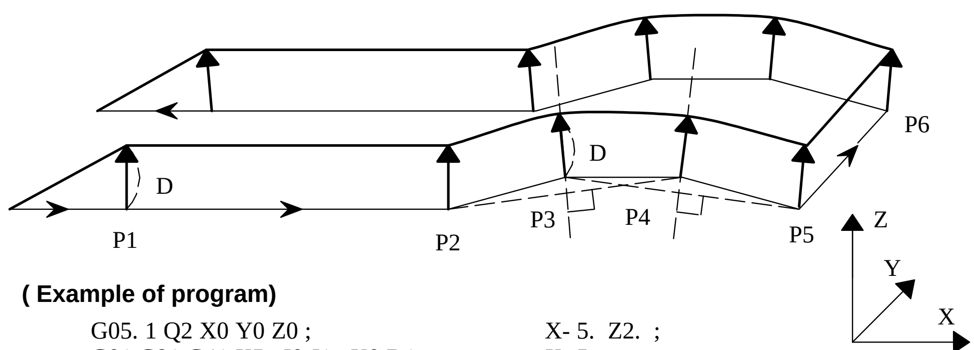
(ii) Direction: The vector points in the direction closest to that of the vector at the previous point (that is, the direction in which |θ| < 90, whereθ is the angle of the vector at the point, relative to the vector at the previous point).

(iii) Length: The length of the vector is the offset corresponding to the offset number specified with G41.

(c) Three-dimensional tool compensation vector at the end point

The same vector as that for the previous point is used.

Example



(Example of program)

```
G05. 1 Q2 X0 Y0 Z0 ;
G01 G91 G41 X5.  I0 J1.  K0 D1 ;
X15.  ;
X5.  Z2.  ;
X5.  ;
X5.  Z- 2.  ;
Y5.  ;
X- 5.  Z2.  ;
X- 5.  ;
X- 5.  Z- 2.  ;
X- 15.  ;
G40 X- 5.  ;
G05. 1 Q0 ;
```

The three- dimensional tool compensation vector specified in a previous block is used at the start point of smooth interpolation (P2). At P3 and P4, vectors are calculated based on the previous and subsequent points, as described above. At the end point of smooth interpolation (P5), the same vector as that at the previous point is used. That point is also used at subsequent points until a new vector is calculated for the next smooth interpolation.

18.6.2 Notes

NOTE 1 Smooth interpolation can be specified for the X-axis, Y-axis, Z-axis, and all axes parallel to them, up to three axes at a time.

NOTE 2 When smooth interpolation is specified for a two–way path, the machined figure may differ slightly between the forward and return paths.

18.7 High Precision Contour Control Using 64bit RISC Processor

18.7.1 High–Precision Contour Control Using RISC processor (HPCC)

Machining errors caused by the CNC include errors due to acceleration/ deceleration after interpolation. To eliminate these errors, a RISC processor is used to provide the high- speed functions listed below.

– Multiblock look- ahead acceleration/ deceleration before interpolation that prevents machining errors caused by acceleration/ deceleration
– Automatic feedrate control function based on the multiblock look- ahead capability that enables smooth acceleration/ deceleration by taking feedrate changes along each axis, and the allowable acceleration range of the machine into account.

These two functions enable smoother acceleration/ deceleration, and as a result the feed-forward coefficient can be increased. This advantage also reduces the tracking error of the servo system.

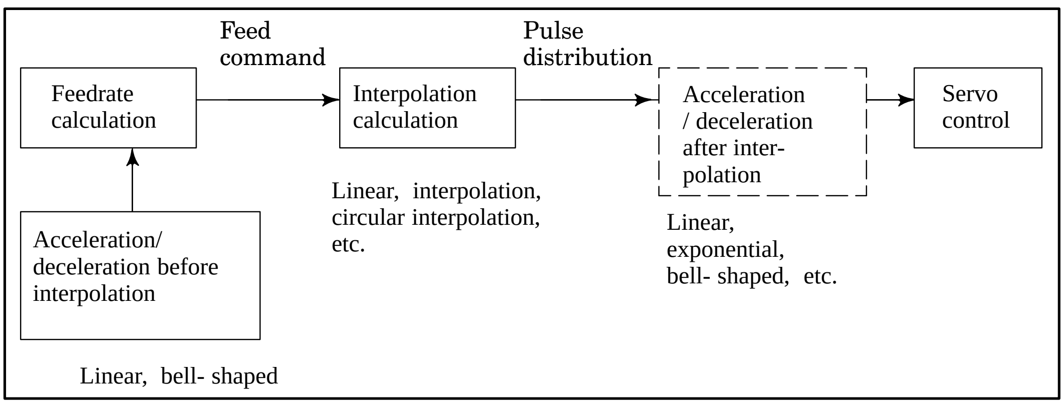
(1) Multiblock look- ahead acceleration/ deceleration before interpolation

When cutting feed per minute is specified, this function can read multiple of blocks ahead to perform acceleration/ deceleration before interpolation, that is, to apply acceleration/ deceleration to the specified feedrate.

When acceleration/ deceleration after interpolation is used, acceleration/deceleration is applied to interpolation data. Consequently, the interpolation data is changed by acceleration/deceleration.

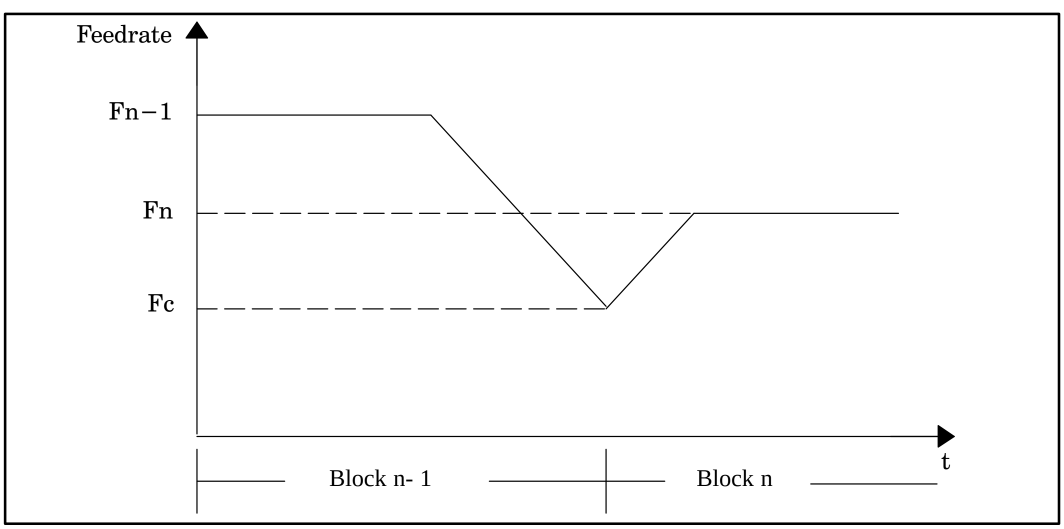
When acceleration/ deceleration before interpolation is used, however, acceleration/deceleration is applied to feedrate data before interpolation. Consequently, the interpolation data is not changed by acceleration/ deceleration.

Accordingly, interpolation data can ensure machining along a specified line or curve at all times, thus eliminating machining profile errors caused by delays in acceleration/ deceleration.



If a feedrate change along any axis is greater than the value set in a parameter at the joint (corner) between two successive blocks, a feedrate is calculated so that the feedrate difference is less than the specified value. The feedrate is automatically reduced to the calculated value at the corner.

Let Fn-1 be a feedrate specified for block n-1, Fn be a feedrate specified for Fn, and Fc be a feedrate calculated as described above. Then, the actual feedrate is as shown below:



– Switching between linear acceleration/ deceleration before interpolation and bell-shaped acceleration/ deceleration before interpolation, and maximum feedrates

The NWBL bit of parameter No. 1601 can be used to select either linear acceleration/ deceleration before interpolation or bell- shaped acceleration/deceleration before interpolation.

As indicated in the table below, the maximum specifiable feedrate depends on whether linear acceleration/ deceleration or bell- shaped acceleration/deceleration is selected.

[Maximum specifiable feedrates when one NC statement block for a 1- mm line segment is specified]

N / M	Linear acceleration/ deceleration before interpolation	Bell- shaped acceleration/ deceleration before interpolation
3 / 3	60m / min	30m / min
5 / 5	40m / min	30m / min

NOTE 1 The data above assumes that cutter compensation is not specified.
NOTE 2 The values above may change, depending on the specified value of the NC statement, NC statement length, whether DNC operation or memory operation is used, and the number of axes.

(2) Automatic Feedrate Control Function

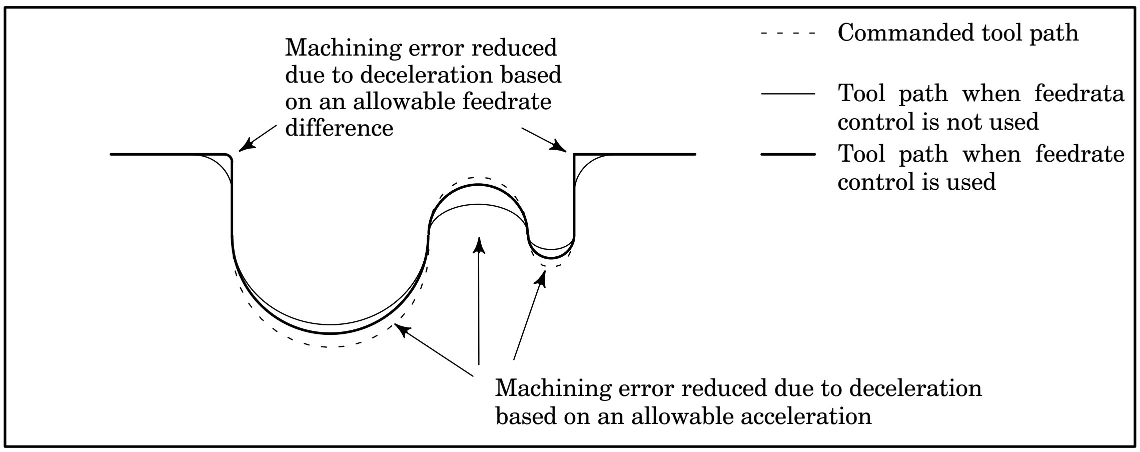
This function reads multiple blocks ahead to perform automatic feedrate control. A feedrate is determined according to the conditions below. If a commanded feedrate exceeds a calculated feedrate, acceleration/ deceleration before interpolation is used so that the calculated feedrate can be reached.

(i) Feedrate change along each axis and specified allowable feedrate difference at a corner

(ii) Estimated acceleration and specified allowable acceleration along each axis

(iii) Cutting load estimated from the direction of motion along the Z- axis

In the automatic feedrate control mode, the feedrate is automatically reduced with acceleration/ deceleration before interpolation to minimize shock to the machine.



(3) Specifications

(a) Basic functions

Name	Function
Number of controlled axes	1 to 9 axes
Maximum number of axes that can be controlled simultane-ously	9 axes
Axis name	A, B, C, U, V, W, X, Y, and Z (freely selectable)
Increment unit	0. 01, 0. 001, 0. 0001, 0. 00001mm 0. 001, 0. 0001, 0. 00001, 0. 000001 inch
Interpolation unit	0. 005, 0. 0005, 0. 00005, 0. 000005 mm 0. 0005, 0. 00005, 0. 000005, 0. 0000005 inch
Maximum programmable value	±9 digits
Positioning	Available with parameter no. 8403#1 MSU=1.
Linear interpolation	Available
Multi- quadrant circular interpolation	Available
Feed per minute	Available
Cutting feedrate clamping	Available
Feedrate override	0% to 254% in 1% steps
Workpiece coordinate system (G54–G59)	Available (not changeable in the G05P10000 mode)
Local coordinate system (G52)	Available (Not changeable in the G05P10000 mode)
Change of workpiece coordinate system (92)	Available (Not changeable in the G05P10000 mode)
Absolute/ incremental programming	Mixed use is allowed in a block.
Sequence number	5 digits
Tape code	ISO, EIA
Tape format	Word address format
Control- in/ control- out	Available
Optional block skip	Available
Arc radius R specification	Available
Automatic operation	Memory operation, tape operation
Details of tape operation	RS- 232- C
Manual absolute- on/ off	Available
Cycle start/ feed hold	Available
Dry run	Available
Feedrate override in dry run mode	0% to 655. 34% in 0. 01% steps
Single block	Available
Multi buffer	Available (Included in RISC function)
Look- ahead acceleration/deceleration before interpolation	Available (Included in RISC function)
Acceleration/ deceleration after inter-polation (linear)	Available in G05 P10000 mode (Time constant: 0 ms to 1000 ms)
Cutter compensation C	Available
Interlock	Available (all- axis interlock capability only)
Machine lock	Available (all- axis machine lock capability only)
Miscellaneous function	Available with parameter no. 8403#1 MSU=1.
2nd auxiliary function	Available with parameter no. 8403#1 MSU=1. (*1) Any address from A, B, C, U, V, W can be used by parameter no. 1030 except for addresses of axes.
Subprogram calling	Available (Calling by auxiliary function code is included.)
Subprogram calling in the external memory	Available

(b) Optional functions (☆ : The associated option is required)

Name	Functions	
2nd feedrate override	0% to 254%, in steps of 1%	☆
Remote buffer interface	Available	☆
DNC1 interface	Available (enabled only when bit 5 of parameter No.0000 is set to 1 (when high–speed pulse distribution in DNC operation is disabled))	☆
DNC2 interface	Available (enabled only when bit 5 of parameter No.0000 is set to 1 (when high–speed pulse distribution in DNC operation is disabled))	☆
Inch/metric conversion	Available (cannot be changed in G05 P10000 mode)	☆
Automatic speed control function	Available (included in the RISC function)	
Acceleration/deceleration after interpolation (bell–shaped)	Available (in G05 P10000 mode, the time constant is 0 to 500 ms.)	☆
Tool offset C	Available	☆
2nd auxiliary function	Available (enabled when bit 1 (MSU) of parameter No. 8403 is set to 1) Any of A, B, C, U, V, and W (specified with parameter No. 1030) can be used, excluding those used as an axis address.	☆
Scaling	Available (G51 and G50 can be specified only between G05 P10000 and G05P0. In scaling mode (G51), therefore, HPCC mode can neither be started nor canceled.)	☆
Coordinate system rotation	Available (G68 and G69 can be specified only between G05 P10000 and G05P0. In coordinate rotation mode (G68), therefore, HPCC mode can neither be started nor canceled.)	☆
Spindle speed fluctuation detection	Available (G25/G26 mode cannot be changed in G05 P10000 mode.)	☆
Addition of workpiece coordinate system pairs	Available (cannot be changed in G05 P10000 mode)	☆
Chopping function	Available (G81.1 and G80 can be specified only outside the range of G05 P10000 to G05P0.)	☆
Multiple M–command in one block	Available (G05 P10000 mode is automatically canceled.)	☆
2nd auxiliary function	Available (G05 P10000 mode is automatically canceled.)	☆
Simple synchronous control	Available (Switching between synchronous operation and normal operation is disabled in G05 P10000 mode.)	☆
PMC axis control	Available (Specify the axes to be controlled by the PMC using a parameter. These axes cannot be specified in G05 P10000 mode.)	☆
Stored stroke chack2	Available (G22/G23 mode cannot changed in G05 P10000 mode.)	☆
NURBS interpolation	Available	☆

(4) Condition for entering the HPCC mode

The mode in which high-precision contour control using RISC is enabled is referred to as the high-precision contour control (HPCC) mode. Specify the following G codes to start and end the HPCC mode:

G05P10000; Starts the HPCC mode.

G05P0; Ends the HPCC mode.

Command G05P10000; immediately before the block in which you want to start HPCC.
Command G05P0; where you want to end HPCC.

Note that the modal G code must be as the following table when G05P10000; is commanded. Otherwise, alarm PS630 occurs.

G code	Function
G13. 1	Polar coordinate interpolation cancel mode
G15	Polar coordinates command cancel
G40	Cutter compensation cancel/three-dimensional cutter compensation (milling series) cancel
G40. 1	Normal direction control cancel mode (milling series only)
G50	Scaling cancel
G50. 1	Programmable mirror image cancel
G64	Cutting mode
G69	Coordinate system conversion cancel
G80	Canned cycle cancel
G94	Feed per minute
G97	Constant surface speed control cancel
M97	Interrupt- type macro cancel

Example Example of a program:

```
O0001;
G91G01F1000. ;
X10. Y20. Z30. ;
G05P10000;          ←  HPCC mode on
X100. Y100. ;
G02I10. ;
G01X100. Y300. F1500. ;
      .
      .
      .
X30. Y-10. ;
G05P0;              ←  HPCC mode off
G04X5. ;
G90G00X100. Y300. ;
M02;
```

(5) Data which can be commanded

G00	:	Positioning (note)
G01	:	Linear interpolation
G02	:	Circular interpolation (clockwise)
G03	:	Circular interpolation (counterclockwise)
G17	:	Xp Yp plane selection where Xp : X-axis or a parallel axis
G18	:	Zp Xp plane selection Yp : Y-axis or a parallel axis
G19	:	Yp Zp plane selection Zp : Z-axis or a parallel axis
G38	:	Cutter compensation C vector retention
G39	:	Cutter compensation C corner rounding
G40	:	Cutter compensation cancel
G41	:	Cutter compensation (left)
G42	:	Cutter compensation (right)
G51	:	Scaling
G50	:	Scaling cancel
G68	:	Coordinate system rotation
G69	:	Coordinate system rotation cancel
G90	:	Absolute programming
G91	:	Incremental programming
Dxxx	:	D code command
Fxxxxx	:	F code command
Nxxxxx	:	Sequence number command
G05P10000	:	HPCC mode on
G05P0	:	HPCC mode off
I, J, K, R	:	I, J, K, and R commands associated with circular interpolation Data for axis motion: Data for axis motion whose name is specified in parameter No. 1020 (freely selectable from X, Y, Z, U, V, W, A, B, and C)
( )	:	Control-in/control-out command (comment command)
/ n	:	Optional block skip command (n:Number)
Mxxxx	:	Auxiliary function (note)
Sxxxx	:	Auxiliary function (note)
Txxxx	:	Auxiliary function (note)
Axxxx	:	2nd auxiliary function
Bxxxx	:	2nd auxiliary function
Cxxxx	:	2nd auxiliary function
Uxxxx	:	2nd auxiliary function
Vxxxx	:	2nd auxiliary function
Wxxxx	:	2nd auxiliary function

(Axxxx – Wxxxx:) An address set by parameter no. 1030 is used. Axis address cannot be used. (see note below) .

M98	:	Subprogram call (Auxiliary function codes can also be used for calling.)
M99	:	Return from subprogram
M198	:	External memory subprogram call (The M code to be used is specified with parameter No. 2431.)
Pxxxx	:	Scale factor used for the scaling function

– The following codes are used for NURBS interpolation:

G06.2	:	NURBS interpolation command
Pxxxx	:	Rank of NURBS curve
Rxxxx	:	Weight
Kxxxx	:	Knot

– The following code is used for smooth interpolation:

G05.1Q1	:	Smooth interpolation command

NOTE To specify a positioning or an auxiliary function in the HPCC mode, set parameter MSU no. 8403#1 to 1. If it is not set, an alarm may be generated. Also, macro call by an auxiliary function code cannot be specified irrespective of setting of parameter no. 8403#1 (MSU).

(6) Screen display during HPCC mode

– Status display

When G05P10000 is specified, the status display of automatic operation changes from STRT to HPCC (blinking). While HPCC is blinking, automatic operation is performed in the HPCC mode.

When operation is stopped by the reset function, the status display changes to RSET; when stopped by the single block function, it changes to STOP; and when stopped by the feed hold function, it changes to HOLD.

When a positioning or auxiliary function is specified with parameter no.8403#1=1, status shows ”START” even in the HPCC mode.

– Program display

During HPCC mode, a specified NC program is transferred to the buffer area prepared for the HPCC mode. When G05P10000 is commanded, the display of a program on the program screen and program check screen changes to the one for HPCC mode. That is, the previously executed block, currently executed block (preceded by the symbol >), and blocks transferred to the buffer area for HPCC mode are highlig hted on the 9” CRT screen, or displayed in green on the 14” CRT screen. During HPCC mode, the buffer area for HPCC mode can store more blocks than the screen can display, so usually the entire screen displays program blocks.

Comments enclosed in a pair of parentheses (”(”and”)”) are not displayed. When a block contains an optional block skip command, /or/n (n:1 to 9) is displayed without modification regardless of the state of the optional block skip signal. When a program is displayed on the program screen in the HPCC mode, the cursor is not displayed.

When G05P0 is specified, the program display for HPCC mode changes to the ordinary display. Immediately after returning to the ordinary display, the line for the previously executed block displays G05P10000;, and the line for the currently executed block displays the block immediately following the block specifying G05P0;.

Example of Display in the HPCC Mode (9” CRT Screen/ Program Screen)

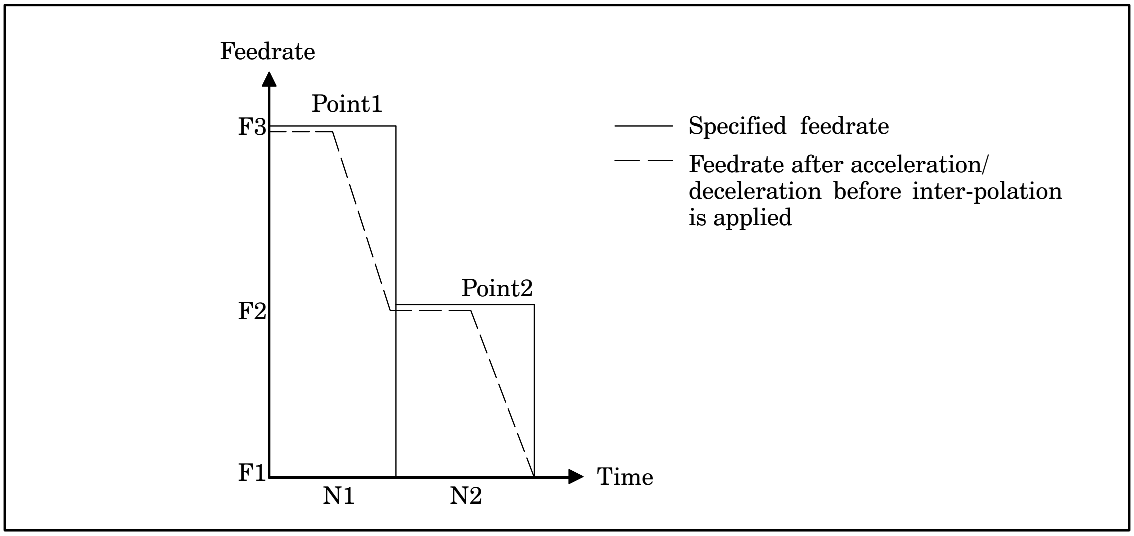
```
PROGRAM (MEMORY)                O1234 N23456

  G05 P10000 ;            → Block previously executed
 >N10 X10. Y10. Z10. ;    → Block currently executed
  N20 X10. Y10. Z10. ;
  / N30 X10. Y10. Z10.;   → The symbol/is displayed
  / 2 N40 X10. Y10. Z10. ;  regardless of the state of the
  N50 X10. Y10. Z10. ;      optional block skip signal.
  N60 X10. Y10. Z10. ;
  N70 ;                   → If a comment is present after
  N80 X10. Y10. Z10. ;      N70, it is not displayed
  N90 X10. Y10. Z10. ;
  N100 X10. Y10. Z10. ;
  N110 X10. Y10. Z10. ;
  G05 P0 ;

MEM *** HPCC **** *** ***  01:23:45 LSK

POSITON PROGRAM OFFSET PRG_CHK CHAPTER+
```

18.7.2 Multiblock Look–Ahead Linear Acceleration/ Deceleration before Interpolation

To use this function, set parameter Nos. 1630 and 1631 appropriately so that acceleration used for acceleration/ deceleration before interpolation is specified.

(1) Example of deceleration

To reach the specified feedrate of a block when the block is executed, deceleration is started in the previous block.



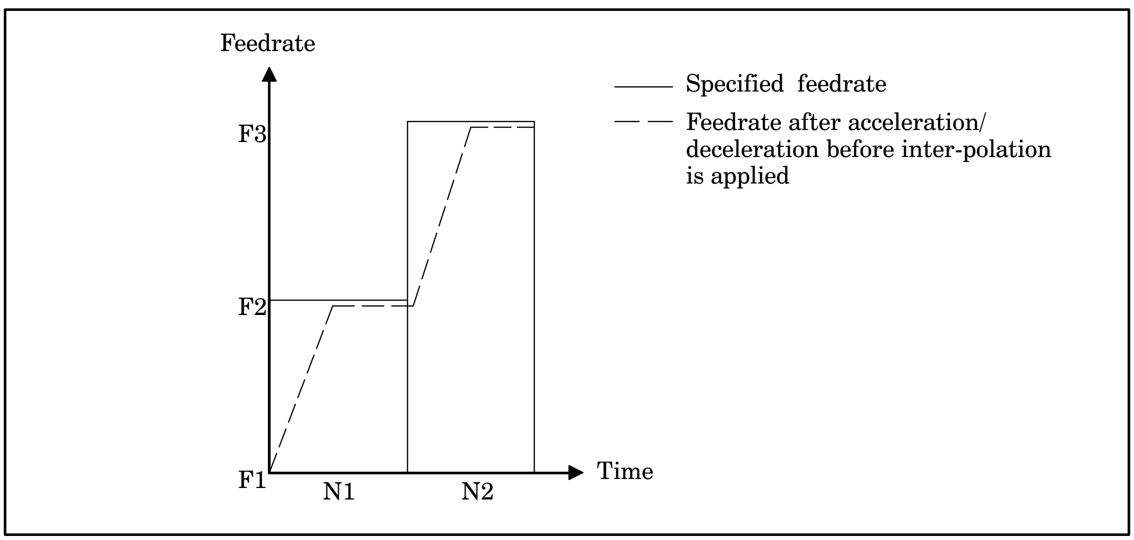
To reduce feedrate F3 to feedrate F2, deceleration must be started at point1.

To reduce feedrate F2 to feedrate F1, deceleration must be started at point2.

The tool can be decelerated over multiple blocks because multiple blocks are read in advance.

(2) Example of acceleration

Acceleration is started when the block is executed.



18.7.3 Automatic Feedrate Control Function

To use this function, set the USE bit of parameter No. 7565 to 1, and set the following parameters:

Parameter No.1478 : Allowable feedrate difference used for feedrate determination based on a corner feedrate difference

Parameter No.1601, BIP bit=1:
Enables deceleration at a corner

Parameter No.1643 : Parameter to specify an allowable acceleration for feedrate determination based on acceleration

Parameter No.7559, CTYP bit=1, CDEC bit=0

Parameter No.7561 : Initial feedrate for automatic feedrate control

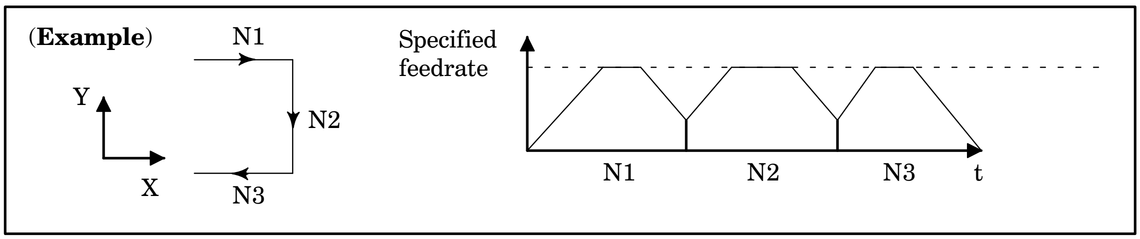
Parameter No.7567 : Maximum allowable feedrate for automatic feedrate control

For details, see the description of the parameters.

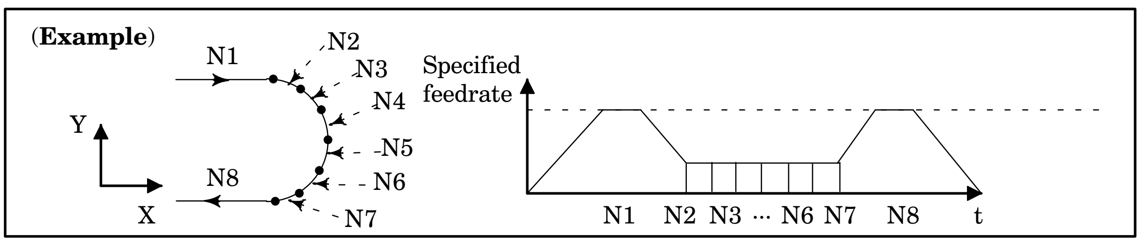
(1) Conditions for Feedrate Control

In the automatic feedrate control mode, the feedrate of the tool is controlled as described below.

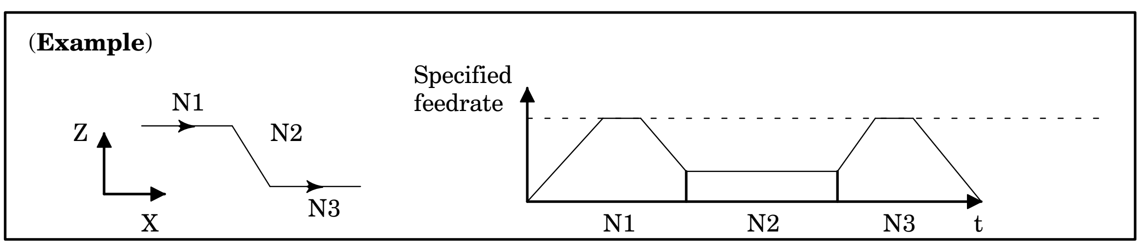
(a) The feedrate required at a corner is calculated from the specified feedrate difference at the corner along each axis, then the tool is decelerated to the calculated feedrate at the corner.



(b) The feedrate required in a block is calculated from the specified accelerationalong each axis at the start point and end point of the corner, and the tool is decelerated so that the feedrate in the block does not exceed the calculated feedrate.



(c) The feedrate required in a block is calculated from the angle of downward move-ment along the Z-axis, and the tool is decelerated so that the feedrate in the block does not exceed the calculated fee-drate.



(2) Example of feedrate determination based on a feedrate difference along each axis

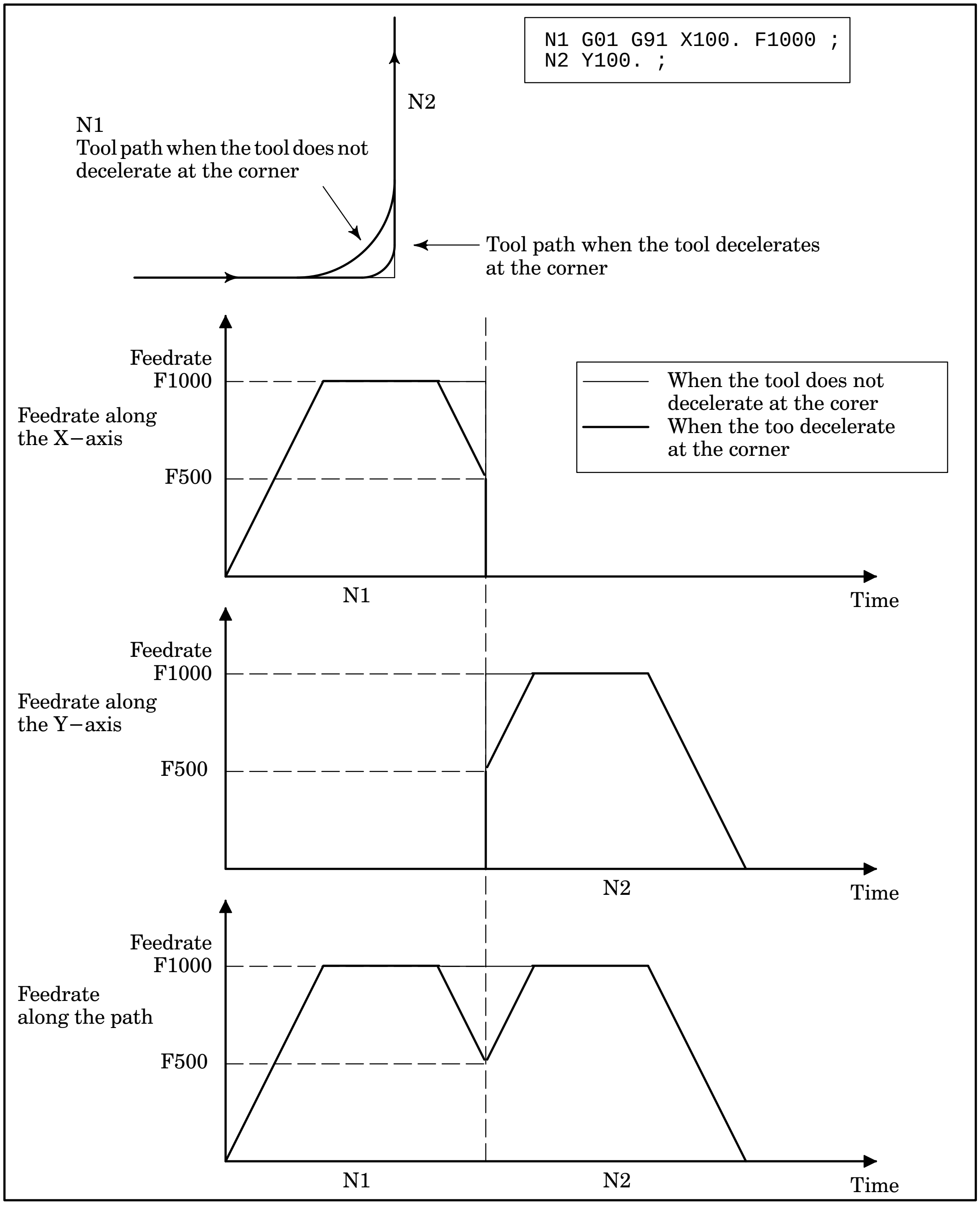
The feedrate required at a corner is calculated from the feedrate difference along each axis as described below.

When the tool is to move at the specified feedrate F, a comparison is made between a feedrate change along each axis (Vc [X], Vc [Y], . . .) and the value (Vmax) set in parameter No.1478. If Vmax is exceeded by a feedrate change along any axis, the tool is decelerated at the corner to the required feedrate Fc:

where Rmax is the largest value of R = Vc/Vmax.

$$Fc = F * \frac{1}{Rmax}$$

Suppose that the specified feedrate of the tool is 1, 000 mm/min, and the tool changes its direction of movement by 90 degrees (from along the X-axis to along the Y-axis). Suppose also that an allowable feedrate difference of 500 mm/min is set. Then, the tool decelerates as shown below.



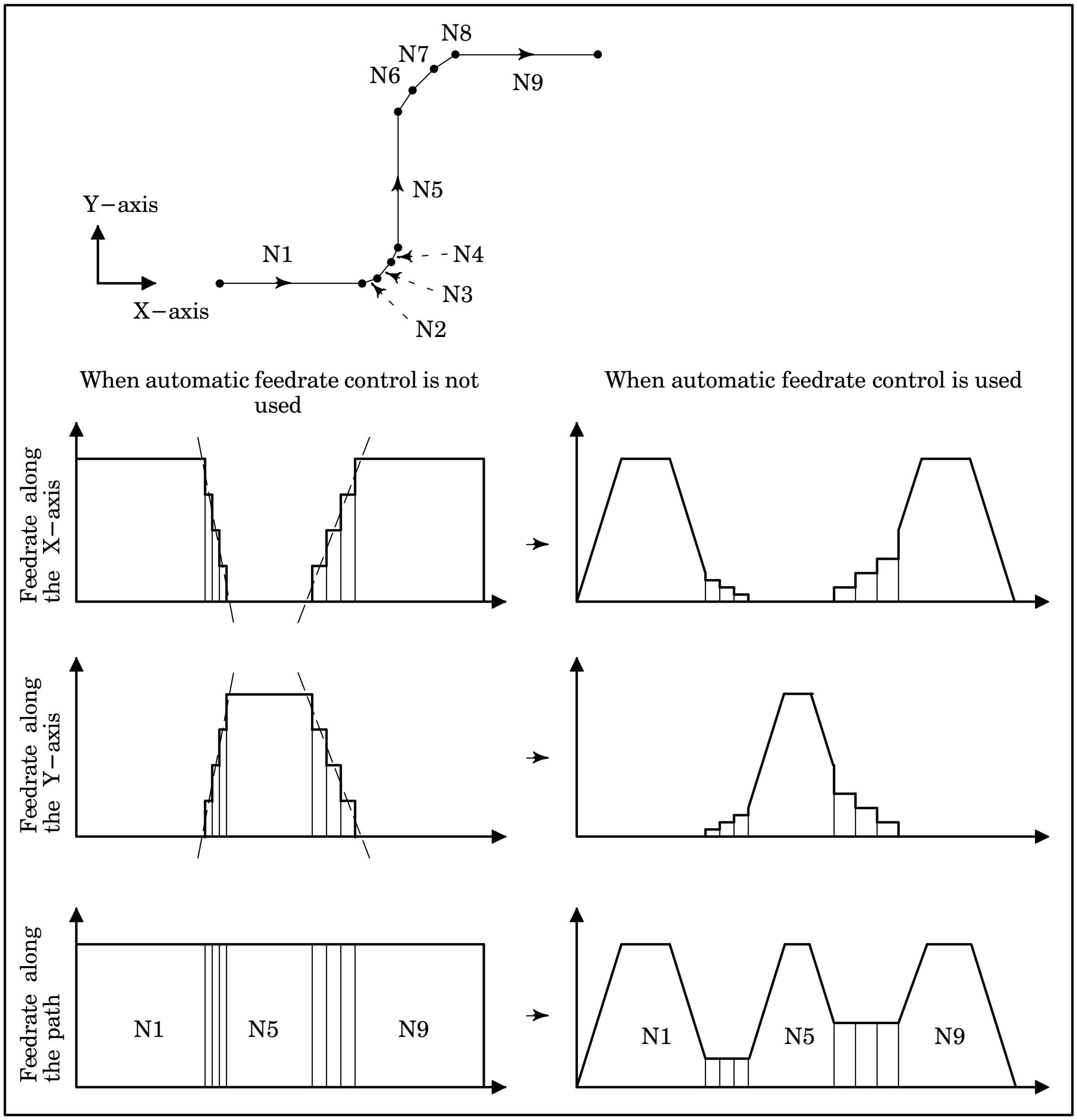
(3) Example of feedrate determination based on an acceleration along each axis

As shown below, when a curve is formed by very short successive line segments, there is no significant feedrate difference along each axis at each corner. Consequently, the tool need not be decelerated to compensate for feedrate differences. When taken as a whole, however, successive feedrate differences generate a large acceleration along each axis. In this case, the tool must be decelerated to suppress the shock to the machine and the machining error that may be caused by such excessive acceleration. The tool is decelerated to the feedrate at which the acceleration along each axis found from the formula below is equal to or less than a specified allowable value.

$$\text{acceleration along each axis} = \frac{\text{Feedrate difference along each axis at a corner}}{\max\left(\frac{\text{Travel distance in the previous block}}{F}, \frac{\text{Travel distance in next blick}}{F}\right)}$$

The deceleration feedrate that is required for each corner is calculated. The tool is decelerated to either the feedrate calculated at the start point or end point of each block, whichever is smaller.

Example In the example below, the tool is accelerated too quickly from N2 to N4 and from N6 to N8 (as indicated by the dashed-line slopes in the feed-rate graphs below) when automatic feedrate control is not used. So, the tool is decelerated.



(4) Feedrate determination based on an allowable acceleration during circular interpolation

When a circular interpolation block of feed per minute is commanded and the CIR bit of parameter No.1601 is set to 1, the feedrate of the tool is automatically determined so that the acceleration along each axis does not exceed the allowable acceleration calculated from the time constant specified in parameter No.1643. During circular interpolation, the tool is controlled so that it always moves along the path at the specified feedrate. At this time, the total acceleration of the tool made of an acceleration along each axis is calculated as follows:

$$\text{Acceleration} = \frac{F^2}{R}$$

F : Feedrate
R : Arc radius

A feedrate is calculated as shown below so that the total acceleration does not exceed the smaller of the allowable accelerations along the two axes involved incircular interpolation. If a specified feedrate is greater than the calculated feedrate, the tool is decelerated to the calculated feedrate.

$$\frac{F^2}{R} = \min(\alpha_x, \alpha_y)$$

α_x, α_y : Allowable accelerations along the X–axis and Y–axis

$$F = \sqrt{R \times \min(\alpha_x, \alpha_y)}$$

(5) Example of feedrate determination based on cutting load

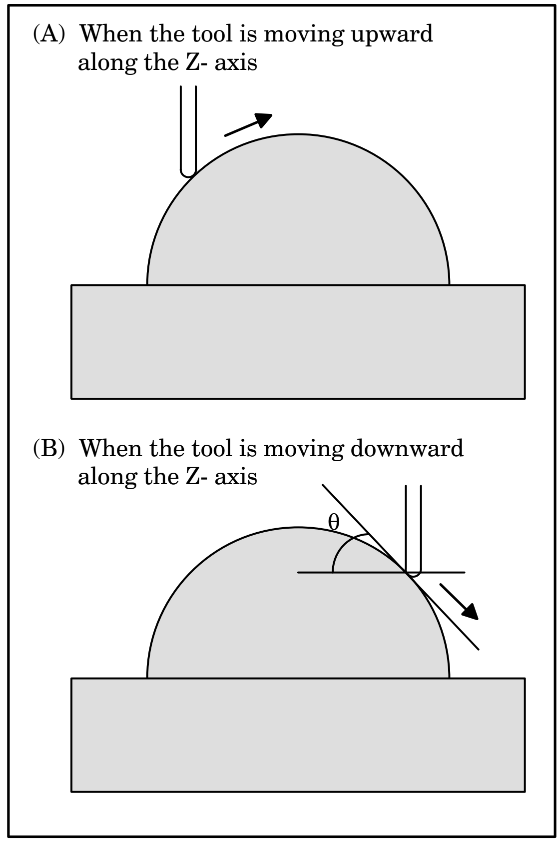
This function can be used when the ZAG bit of parameter No.7565 is set to 1. Cutting the workpiece with the bottom of the cutter ((B) at right) usually involves a greater cutting resistance than cutting the workpiece with the side of the cutter ((A) at right). Therefore, for (B), the tool needs to be decelerated.

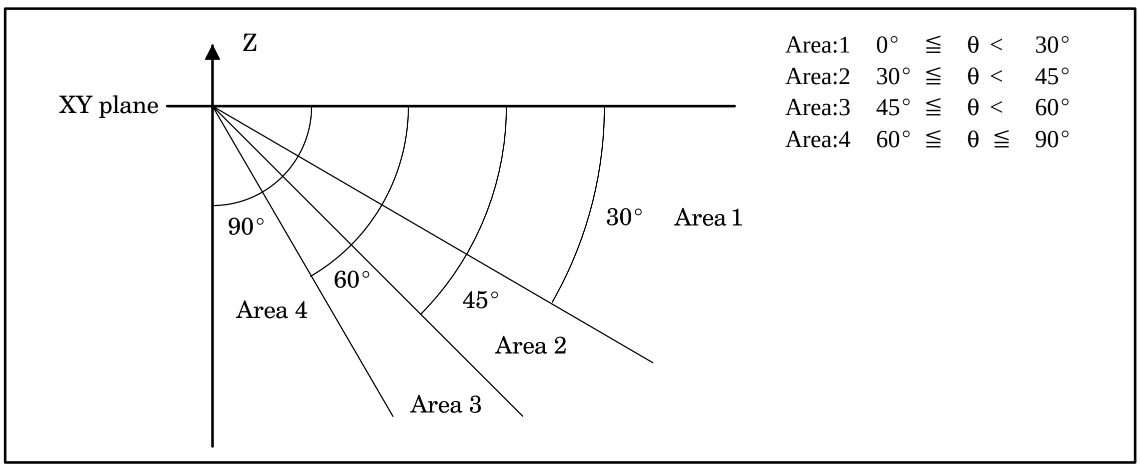
To calculate the decelerated feedrate which is required, the automatic feedrate control function uses the angle of downward movement of the tool along the Z-axis.

When the tool is moving downward along the Z-axis, the angle (θ) of downward movement formed by the XY plane and cutter path is as shown by (B). The angle of downwad movement is divided into four areas, and an override value for each area is specified in a parameter as follows:

Area 2: Parameter No.7591
Area 3: Parameter No.7592
Area 4: Parameter No.7593

No override parameter is provided for area 1; the override value for area 1 is always 100%. A feedrate determined with a separate feedrate control function is multiplied by the override value specified for the area to which the angle (θ) of downward
movement belongs.





NOTE The feedrate determination function based on cutting load uses an NC command to find the direction of movement along the Z-axis. This means that the direction of movement along the Z-axis cannot be found if the movement along the Z-axis is subject to manual intervention with manual absolute set to on, or the mirror image function is used with the Z-axis. So, never use these functions when using the feedrate determination function based on cutting load.

(6) Ignoring F code commands

In a block where the automatic feedrate control function is enabled, all feed commands (F commands) can be ignored by setting the NOF bit of parameter No.7565. The feed commands which are ignored are:

(1) Modal F command specified before a block where the automatic feedrate control function is enabled

(2) Modal F command and F command commanded in a block where the automatic feedrate control function is enabled

Note, however, that specified F commands and modal F commands are stored in the CNC. This means that in a block where the automatic feedrate control function is disabled, the modal F command of (1) or (2) is used instead of the F command calculated by the automatic feedrate control function.

(7) Other examples of feedrate determination conditions

(1) If a calculated feedrate exceeds the maximum allowable feedrate for automatic feedrate control specified in parameter No. 7567 or an F command, the feedrate is clamped to the maximum allowable feedrate or F command, whichever is smaller.

(8) Feedrate fluctuation clamp

This function adds the following control to automatic feedrate control in high–precision contour control mode (G05 P10000):

If the fluctuation of the specified feedrate in a block falls within a specified range, relative to the feedrate for the previous block, the feedrate for that block is clamped to that for the previous block. As an exception, if the block specifies a feedrate of 0, the feedrate is unconditionally changed to 0.

This function enables stable feedrate control even if automatic feedrate control causes minor fluctuation in the feedrate.

Example If the feedrate fluctuation clamp ratio is set to 10%, the feedrate is changed only when its fluctuation exceeds 10%.

N1	F=1000	[mm/min]			⇒	F=1000	[mm/min]	
N2	F=1500	[mm/min]	Fluctuation ratio	50%	⇒	F=1500	[mm/min]	Changed
N3	F=1600	[mm/min]	Fluctuation ratio	7%	⇒	F=1500	[mm/min]	Not changed
N4	F=1650	[mm/min]	Fluctuation ratio	10%	⇒	F=1500	[mm/min]	Not changed
N5	F=1400	[mm/min]	Fluctuation ratio	7%	⇒	F=1500	[mm/min]	Not changed
N6	F=2000	[mm/min]	Fluctuation ratio	33%	⇒	F=2000	[mm/min]	Changed

The above F commands indicate the feedrates that are determined internally by the automatic feedrate control function when an F command greater than 2000 (mm/min) is actually specified in the program.

Every feedrate that is actually specified by an F command in the program is always valid even if it falls within the feedrate fluctuation clamp range.

(If the F command in the N5 block is actually specified in the program, and not generated internally, the feedrate becomes F = 1400 (mm/min).)

18.7.4 Multiblock Look–Ahead Bell–Shaped Acceleration/ Deceleration before Interpolation

To use this function, set the NWBL bit of parameter No.1601 to 1, and set the following parameters:

Parameter No.1630 : Parameter 1 for setting an acceleration used for acceleration/deceleration before interpolation
Parameter No.1631 : Parameter 2 for setting an acceleration used for acceleration/deceleration before interpolation
Parameter No.7614, DST bit=1, BLK bit=0
Parameter No.8416 : Time needed to reach maximum allowable acceleration
For details, see the parameter manual.

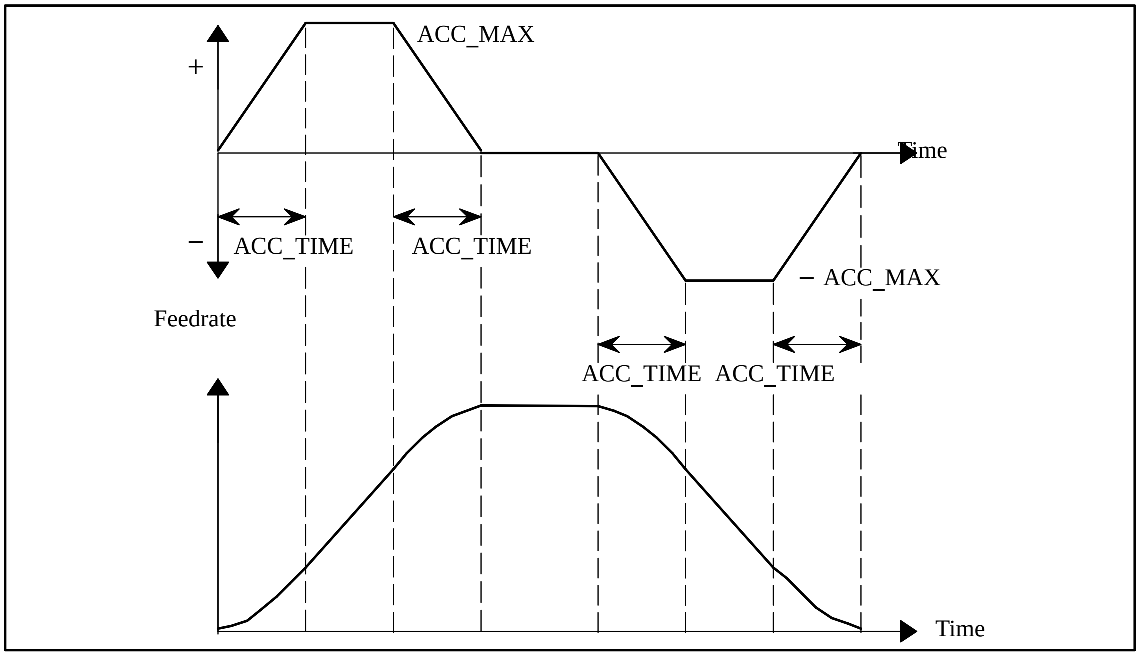
(1) Description

The look-ahead bell- shaped acceleration/deceleration before interpolation controls acceleration as described below.

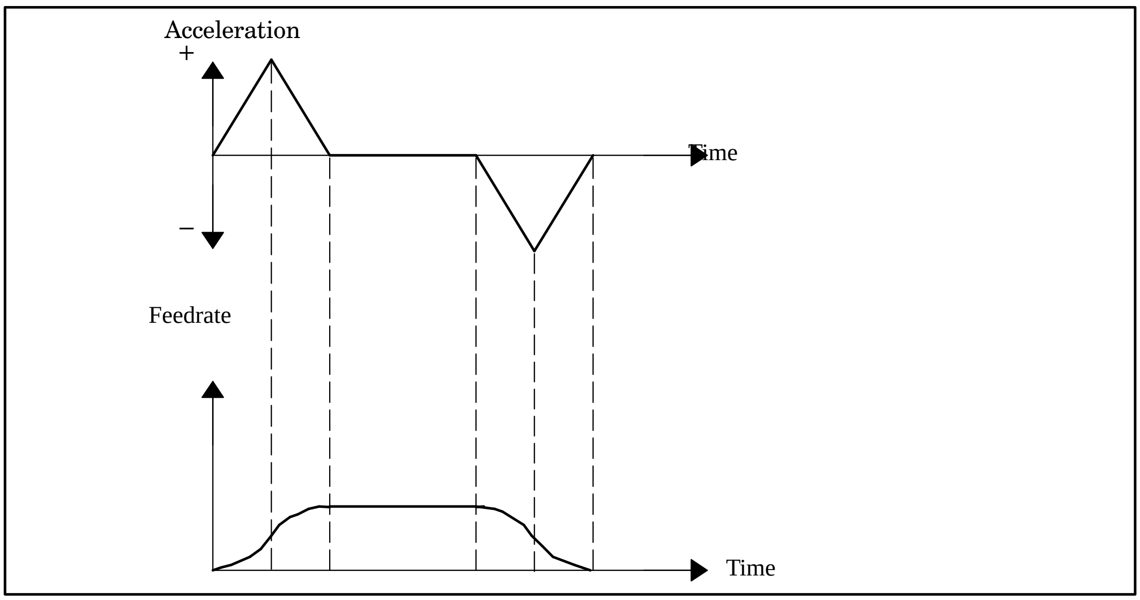
$$\text{Maximum acceleration ACC\_MAX} = \frac{\text{Parameter No.1630 [mm/min, inch/min]}}{\text{Parameter No.1631 [msec]}}$$

Time needed to reach maximum acceleration:ACC_TIME=Setting in parameter No.8416 [msec]

(1) When maximum acceleration is reached



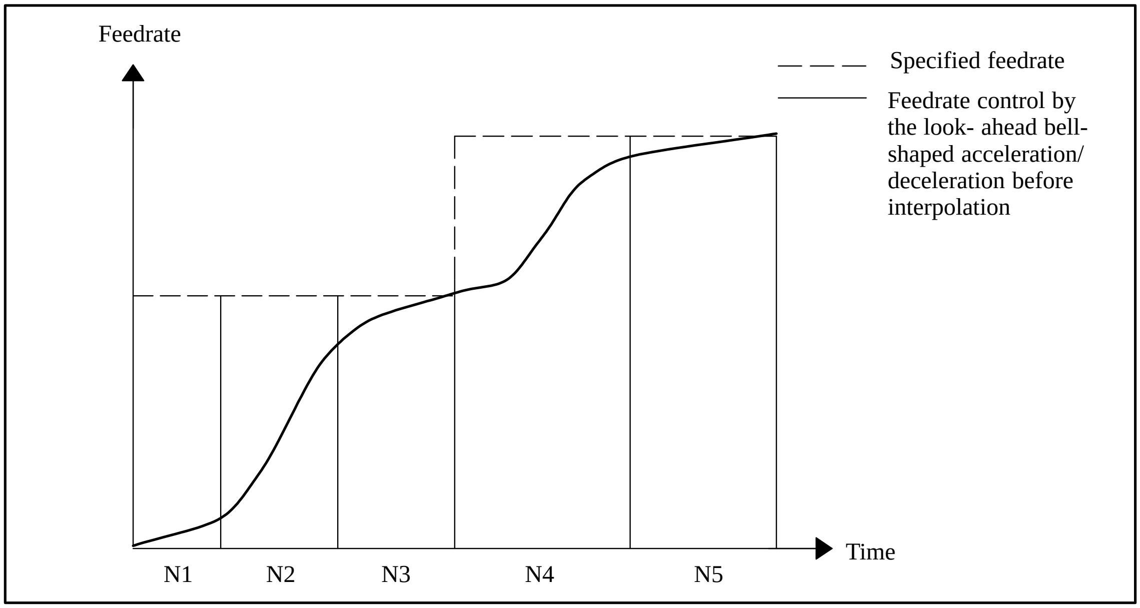
(2) When maximum acceleration is not reached



(2) Acceleration

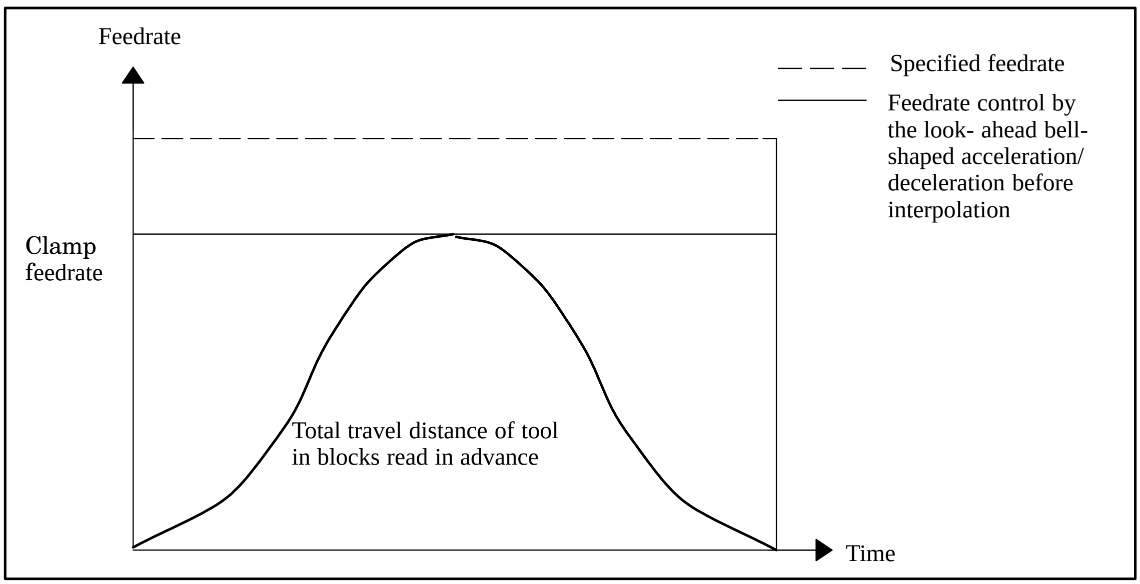
The tool is accelerated to a specified feedrate, starting at the beginning of a block.

The tool can be accelerated over multiple blocks.



(a) Feedrate clamping based on the total travel distance of the tool in lookahead blocks

When the distance required to decelerate the tool from a specified feedrate is less than the total travel distance of the tool in the blocks read in advance, the feedrate is automatically clamped to a feedrate which decelerates the tool over the total travel distance.

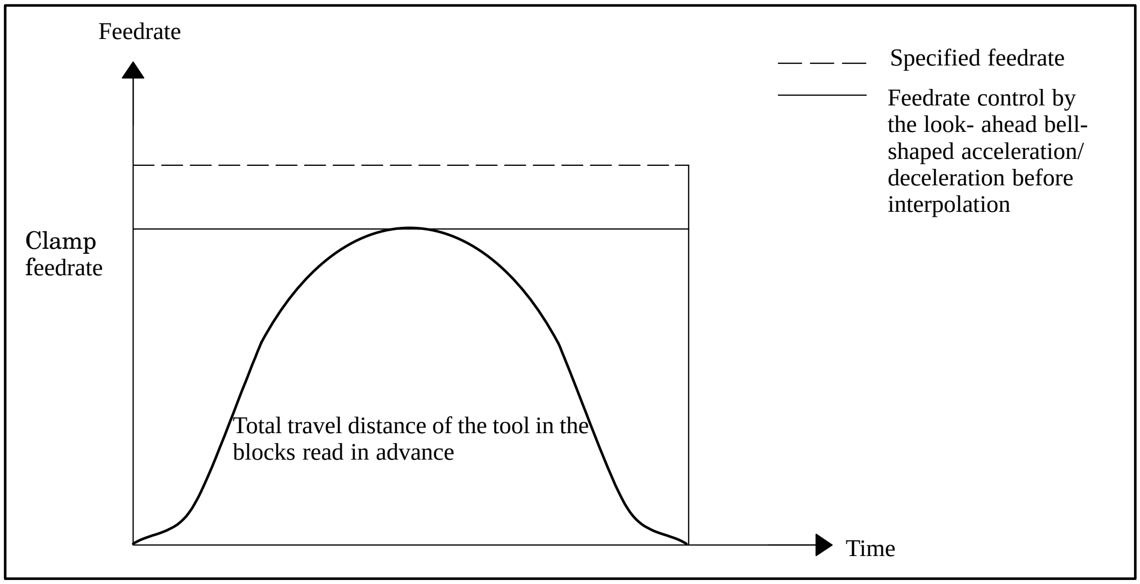


When several blocks, each specifying a short travel distance, are specified in succession, the following situation can occur:

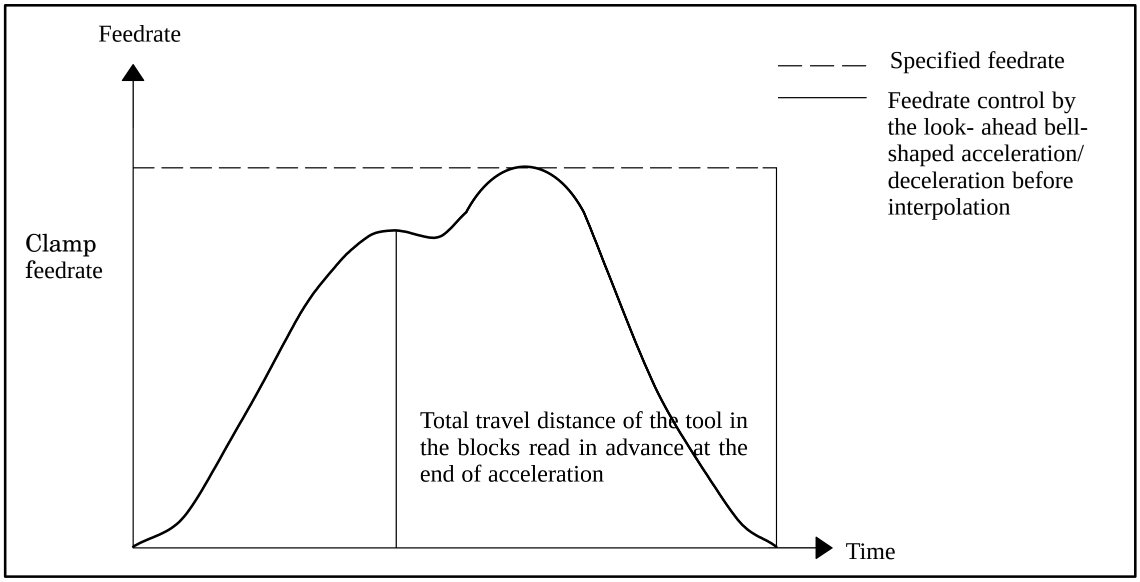
The total travel distance of the tool in the blocks read in advance at the start of acceleration is less than the distance required to decelerate the tool from a specified feedrate, but the total travel distance of the tool in the blocks read in advance at the end of acceleration is greater than the distance required to decelerate the tool from a specified feedrate.

In such a case, the tool is once accelerated and clamped to the feedrate obtained based on the total travel distance of the tool in the blocks read in advance, then the tool is accelerated to a specified target feedrate.

(i) At the start of acceleration



(ii) At the end of acceleration

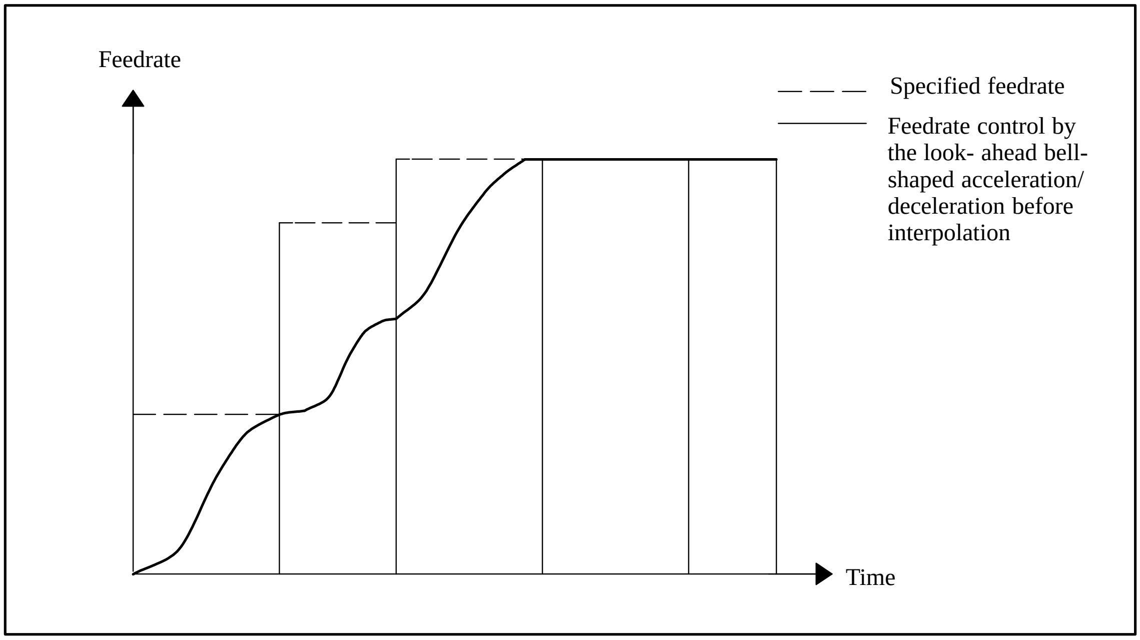


(b) Feedrate command and feedrate

If an F command is changed by, for example, an other F command, the corner deceleration function, or the automatic feedrate determination function, the look-ahead bell-shaped acceleration/ deceleration before interpolation treats the changed feedrate as a new target feedrate, and restarts acceleration/ deceleration.

Each time an F command is changed, bell-shaped acceleration/deceleration is performed.

Bell- shaped acceleration/deceleration is performed each time a different feedrate command is specified, before example, in a program containing successive blocks specifying a short travel distance.



(c) When the feed hold function is used during acceleration

When the feed hold function is used during acceleration, feedrate control becomes as described below.

(i) While applying constant or increasing acceleration

Starting at the point where the feed hold function is specified, first the acceleration is gradually decreased to 0, then the feedrate of the tool is gradually decreased to 0. Thus, the feed hold function does not immediately decrease the feedrate of the tool; it can increase the feedrate for a time before decreasing the feedrate.

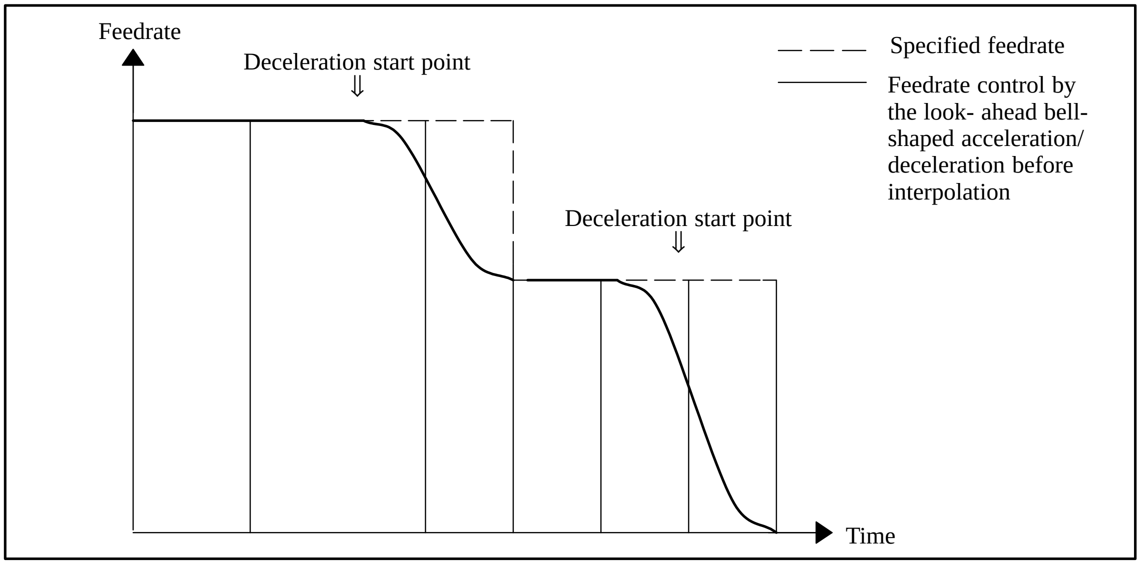
(ii) While applying decreasing acceleration

First, the acceleration is gradually decreased to 0, then the feedrate is gradually decreased to 0.

(3) Deceleration

The tool is decelerated to the specified feedrate of a block, starting at the previous block.

The tool can be decelerated over multiple blocks.

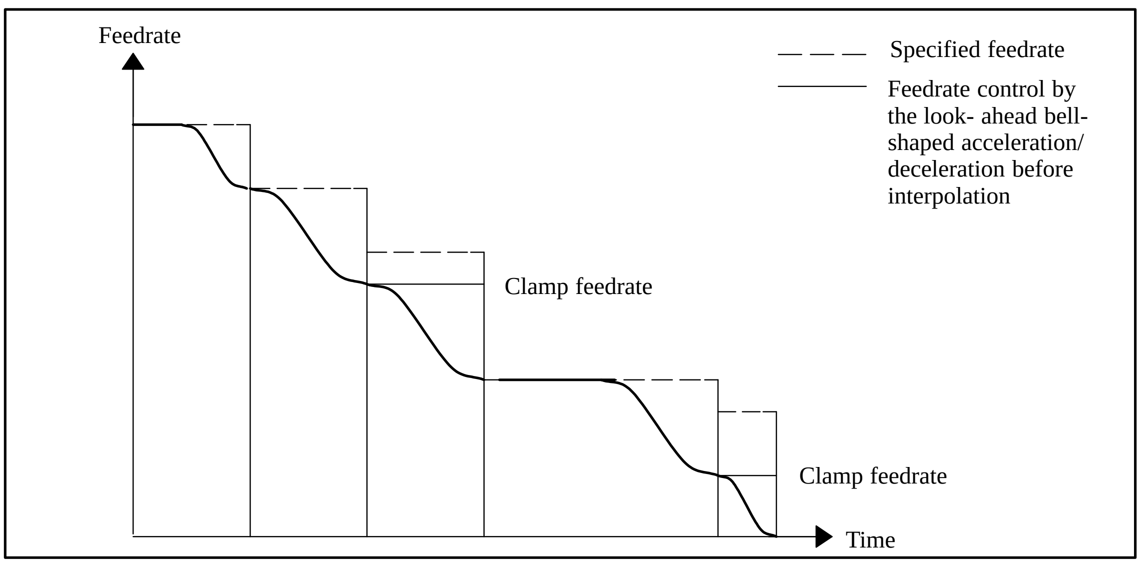


(a) Feedrate command and feedrate

If an F command is changed by, for example, another F command, the corner deceleration function, or the automatic feedrate determination function, the look-ahead bell- shaped acceleration/deceleration before interpolation function trea ts the changed feedrate as a new target feedrate, and restarts acceleration/deceleration.

Each time an F command is changed, bell- shaped acceleration/ deceleration is performed.

When the distance required to decelerate the tool from a specified feedrate is longer than the total travel distance of the tool in the blocks read in advance, the feedrate is automatically clamped as in the case of acceleration.



(b) Deceleration based on distance

The deceleration of the tool is started when the total travel distance of the tool in the blocks read in advance is less than the distance required to decelerate the tool from the current feedrate.

When the total travel distance of the tool in the blocks read in advance increases at the end of deceleration, the tool is accelerated.

When blocks specifying a short travel distance are specified in succession, the tool may be decelerated, then accelerated, then decelerated, and so on, resulting in an unstable feedrate. In such a case, specify a smaller feedrate.

(c) When the feed hold function is used during deceleration

When the feed hold function is used during deceleration, feedrate control becomes as described below.

(i) While applying constant or increasing deceleration

The point where the deceleration starts being decreased to 0 is shifted from the point usually used (i. e., used when feed hold is not applied) so that the feedrate of the tool is gradually decreased to 0.

(ii) When applying decreasing deceleration

The deceleration is gradually reduced to 0, then the feedrate is decreased to 0.

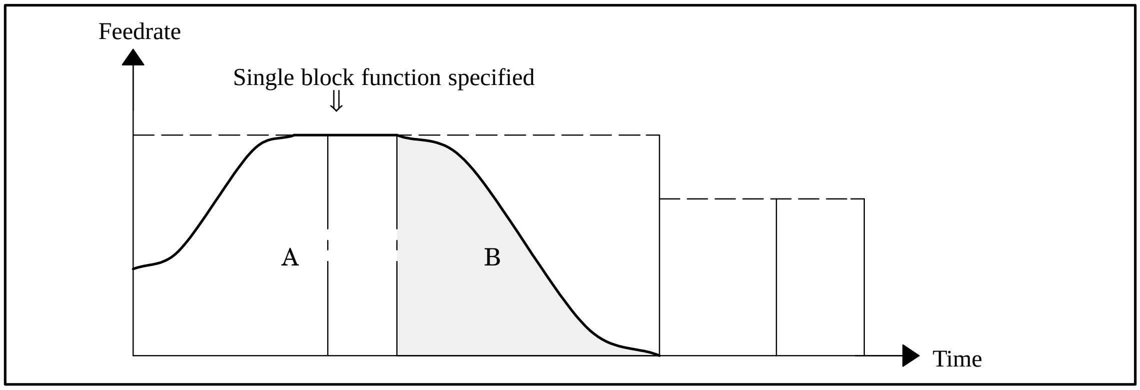
(4) Single Block Function

When the single block function is specified while the look-ahead bell- shaped acceleration/ deceleration before interpolation is used, control is exercised as described below.

(a) While the tool is being accelerated or decelerated when the single block function is specified

(i) A + B ≦ Remaining travel distance of the tool in the block being executed when the single block function is specified

The tool is gradually decelerated so that the feedrate is 0 when the execution of the block that was being executed when the single block function was specified is completed.



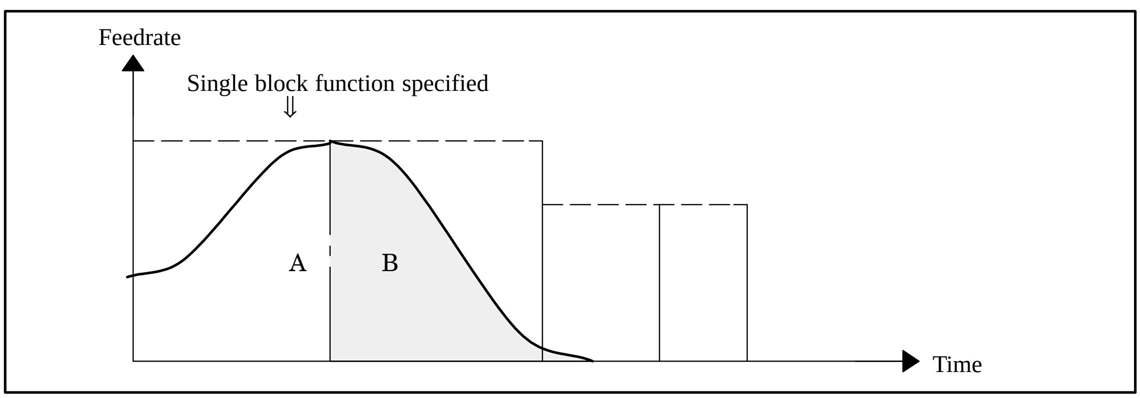
A: Distance traveled until the tool reaches the specified feedrate with the current acceleration/ deceleration

B: Distance traveled until the feedrate becomes 0.

(ii) A + B > Remaining travel distance of the tool in the block being executed when the single block function is specified

The tool may be decelerated over multiple blocks until it stops.

The tool is stopped as described later in 2. 4. 4. 3.



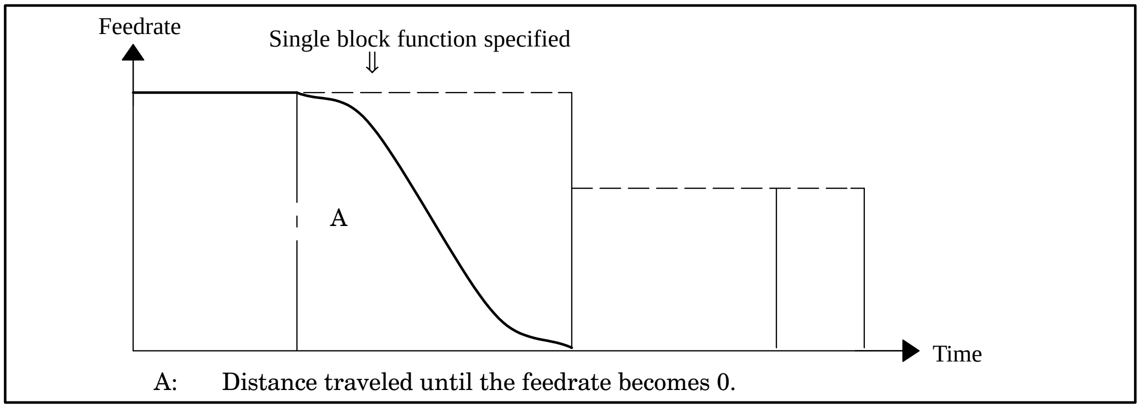
A: Distance traveled until the tool reaches the specified feedrate with the current acceleration/ deceleration

B: Distance traveled until the feedrate becomes 0.

(b) While the tool is not being accelerated or decelerated when the single block function is specified

(i) A ≦ Remaining travel distance of the tool in the block being executed when the single block function is specified

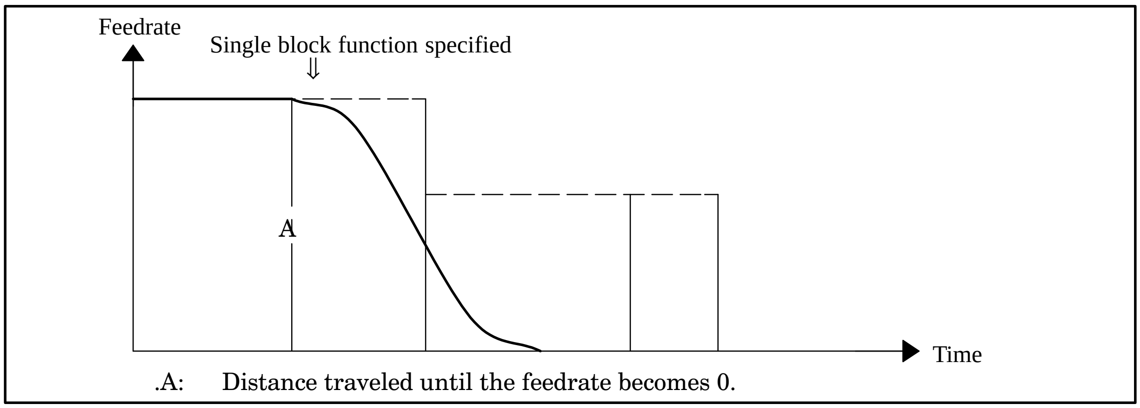
The tool is gradually decelerated so that the feedrate is 0 when the execution of the block that was being executed when the single block function was specified is completed.



(ii) A > Remaining travel distance of the tool in the block being executed when the single block function is specified

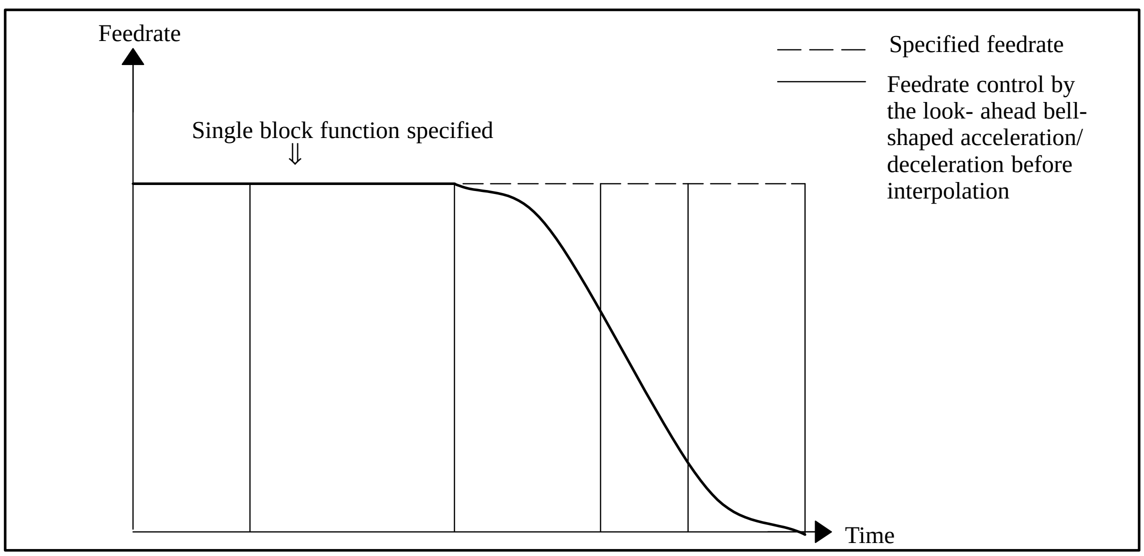
The tool may be decelerated over multiple blocks until it stops.

The tool is stopped as described later in 2. 4. 4. 3.



(c) How the tool stops when it may be decelerated over multiple blocks

The tool is decelerated (or accelerated) over multiple blocks until the feedrate becomes 0.



NOTE 1 Depending on the stop point and the remaining blocks, two or more acceleration/ deceleration operations may take place.

NOTE 2 When the single block function is specified, an acceleration/ deceleration curve recalculation is required while the tool is moving along an axis. So, the tool is not necessarily decelerated over a minimum number of blocks before it stops.

(5) Dry run/ feedrate override

When a change in the specification of the dry run function or feedrate override function changes a specified feedrate (feedrate change due to an external cause) while the look-ahead bell-shaped acceleration/deceleration before interpolation is being used, control is exercised as described below.

(a) While the tool is being accelerated or decelerated when the specification of the dry run function or feedrate override function is changed

After the current acceleration/ deceleration operation brings the tool to a specified feedrate, the tool is accelerated or decelerated to a specified feedrate.

(b) While the tool is not being accelerated or decelerated when the specification of the dry run function or feedrate override function is changed

The tool is accelerated or decelerated from the current feedrate to a specified feedrate.

(c) Notes

- When the specification of the dry run function or feedrate override function is changed, the acceleration/ deceleration curve needs to be recalculated while the tool is moving along an axis. For this reason, there is a slight delay before a feedrate change is actually started.
- When the specification of the dry run function or feedrate override function is changed, the tool may be decelerated below a specified feedrate and then accelerated, depending on the remaining travel distance, current feedrate, and target feedrate.

18.7.5 NURBS Interpolation

(1) Specification

Recently, as a means to express curves and curved surfaces for mold, NURBS(Non uniform rational B–Spline) has been widely adopted on the ground of CAD to design molds of Automobile and Aircraft. This feature enables to command the expression of NURBS curves directly to CNC. By this feature, it is not necessary to approximate NURBS curve with small lines, and the following merits are obtained.

1) There is no error which is created by the approximation with small lines for designed NURBS curve.
2) The length of the CNC part–program shortens compared with the CNC part–program by small consecutive lines.
3) The pauses, which may occur when small blocks are executed at high speed, are not generated.
4) The high speed transmission from the Host computer to CNC is not necessary.

In this feature, CAM has to create NURBS curve by adding the tool offset of the tool radius and the tool length to the NURBS expression, which is output from CAD, and has to command 3 parameters, the control points, the weight, and the knots, which define NURBS curve. Here, the tool offset of the tool length is not necessarily needed.

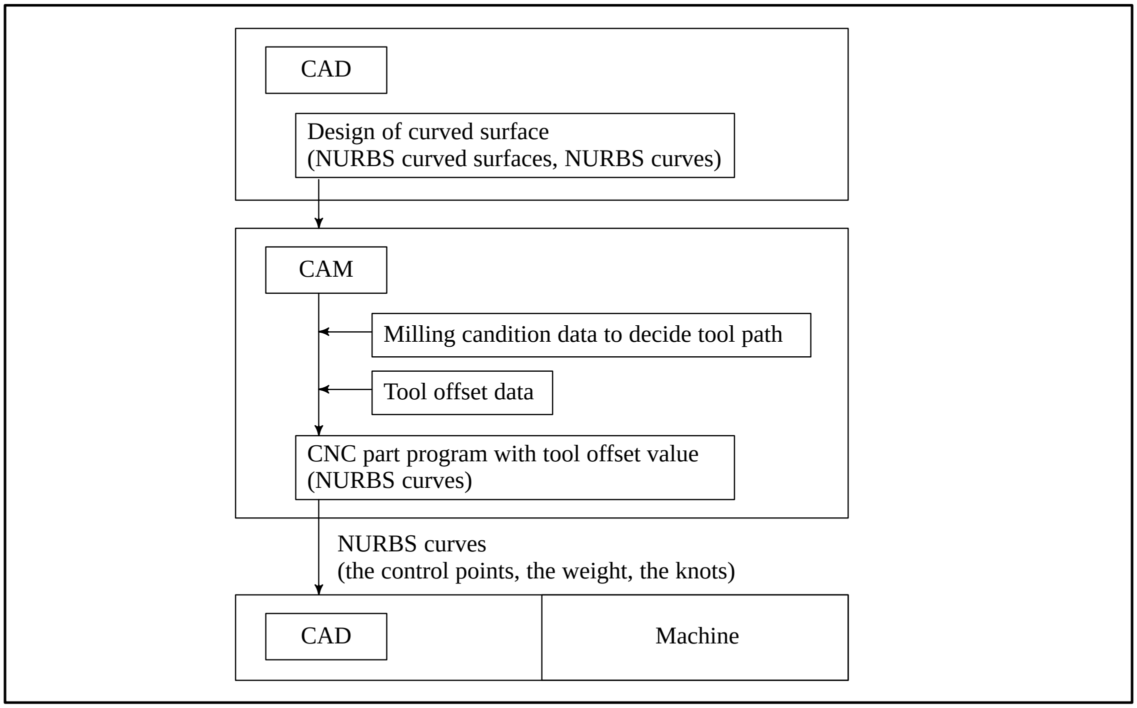


Fig. 18.7.5 (a) CNC part program with NURBS curves

The NURBS interpolation has to be commanded during HPCC(High Precision Contouring Control : G05 P10000 to G5 P0) mode. While FANUC Series 15–MB executes the NURBS interpolation, it automatically controls the feedrate so that the acceleration of each axis does not exceed the permissible acceleration of the machine and the shock does not occur on the machine.

(2) Command format

The NURBS interpolation has to be commanded with the following format.

```
G05 P10000 ; (HPCC mode ON)
...
G06.2 [P_]   K_ X_ Y_ Z_ [R_] [F_] ;
   K_ X_ Y_ Z_ [R_] ;
   K_ X_ Y_ Z_ [R_] ;
   K_ X_ Y_ Z_ [R_] ;
   ...
   K_ X_ Y_ Z_ [R_] ;
   K_ ;
   ...
   K_ ;
G01 ...
...
G05 P0 ; (HPCC mode OFF)
```

Here, each code has the following meaning.

G06.2	: NURBS interpolation mode ON
P_	: the order of NURBS curve
X_ Y_ Z_	: the control point
R_	: the weight
K_	: the knot
F_	: feedrate

G06.2 is a modal G–code of the group 01. Therefore, when a G–code(G00,G01,G02,G03...) of the group 01 except G06.2 is commanded, the NURBS interpolation ends. And the NURBS interpolation has to be canceled before HPCC mode is canceled.

The order of NURBS can be specified with an address word of P. The command must be specified at the first block. When it is not specified, the order is regarded as 4. The effective value of P are as follows:

P2 : NURBS curve of the order=2
P3 : NURBS curve of the order=3
P4 : NURBS curve of the order=4 (default)

Here, “order” means “*k*” described in “3.Definition of NURBS”. For example, when the order is 4, the rank in the definition of NURBS is 3, in other words, the NURBS expression is expressed by t^3 , t^2 , t^1 and constant.

The maximum number of axes for the NURBS interpolation is 3. The axes for the NURBS interpolation should be commanded at the first block. Any axis which is not specified at the first block can not be commanded until the NURBS interpolation ends.

The weight code is the weight for the control point in the same block. When the weight code is not specified, the weight is regarded as 1.0.

The number of the knot codes should be [The number of the control points + the order]. At the blocks from the first control point to the last control point, the knots have to be specified with the control points. After the block of the last control point, the rest knots, whose number is the same with the order, have to be specified without any other codes. And, the first control point must be the starting point of the NURBS curves and the last control point must be the ending point. Therefore, the leading knots and the trailing knots, whose number is the same with the order, must have the same value (the multi–knot). When the absolute position where the NURBS interpolation starts is different from the first control point, an alarm occurs. (In the case of incremental commands, G91 G06.2 X0 Y0 Z0 K_ must be specified.)

During the NURBS interpolation, any other commands than the NURBS interpolation (ex. Auxiliary function etc.) can not be commanded.

Example

```
G05 P10000;
G90;
...
G06.2 K0.   X0.     Z0.;
      K0.   X300.   Z100.;
      K0.   X700.   Z100.;
      K0.   X1300.  Z–100.;
      K0.5  X1700.  Z–100.;
      K0.5  X2000.  Z0.;
      K1.0;
      K1.0;
      K1.0;
      K1.0;
G01   Y0.5;
G06.2 K0.   X2000.  Z0.;
      K0.   X1700.  Z–100.;
      K0.   X1300.  Z–100.;
      K0.   X700.   Z100.;
      K0.5  X300.   Z100.;
      K0.5  X0.     Z0.;
      K1.0;
      K1.0;
      K1.0;
      K1.0;
G01   Y1.0;
G06.2 ...
...
G01 ...
G05P0;
```

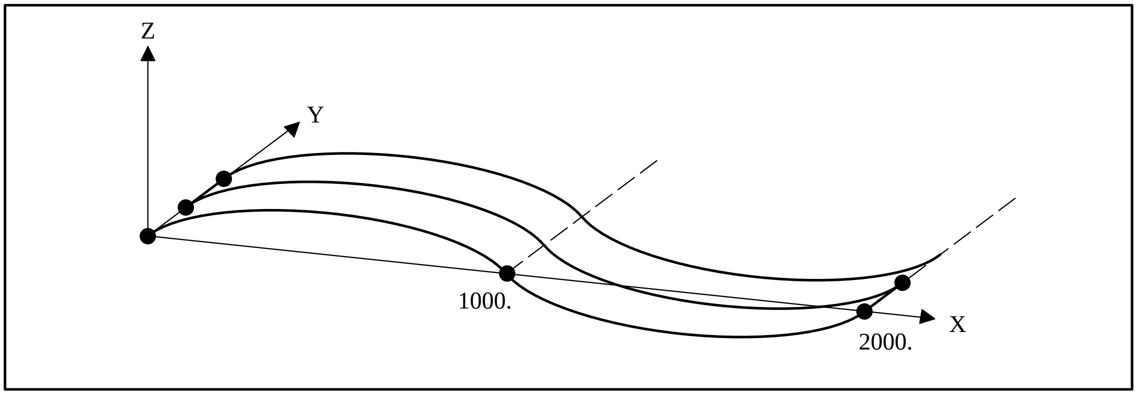


Fig. 18.7.5 (b) Example

(3) Definition of NURBS

The variables are defined as follows:

- k : the order
- $\boldsymbol{P}_i$: the control point
- W_i : the weight
- X_i : the knot $(X_i \leq X_{i+1})$
  the knot vector $[X_0, X_1, \ldots, X_m]$ $(m=n+k)$
- t : the parameter of B–Spline

Spline basis function ***N(t)*** is expressed by de Boor–Cox as follows:

$$N_{i,1}(t) = \begin{cases} 1 & (X_i \leq t \leq X_{i+1}) \\ 0 & (t < X_i,\ X_{i+1} < t) \end{cases}$$

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

Here, the NURBS curve ***P(t)*** is expressed as follows:

$$\boldsymbol{P}(t) = \frac{\sum_{i=0}^{n} N_{i,k}(t)\, W_i\, \boldsymbol{P}_i}{\sum_{i=0}^{n} N_{i,k}(t)\, W_i} \qquad (X_0 \leq t \leq X_m)$$

(4) Notes

NOTE 1	An alarm occurs when the manual interruption is executed with the manual absolute signal ON.
NOTE 2	The cutter radius compensation can not be used with the NURBS interpolation. When the NURBS interpolation starts, the cutter radius compensation must have been canceled.

18.7.6 NOTES

(1) Functions disabled in HPCC mode

The following functions cannot be used in high–precision contour control (HPCC) mode using a RISC processor:

Basic functions

Exact stop, exact stop mode
Tapping mode
Dwell
Reference position return (G27, G28, G29)
Machine coordinate system (G53)
Calculation type decimal point input
Input unit 10 times multiply
Tool length measurement
Workpiece coordinate system preset (G92.1)
Diameter programming
Feed per revolution without position coder
Retrace program editing
Full circle cutting
Programmable diameter/radius specification switching function
Follow–up for individual axis
Transverse inhibit limit
Automatic corner deceleration
Feedrate cramp by circular radius
Backlash compensation for eath rapid traverse and cutiing feed

Optional functions

J808	Single direction positioning
J810	Hypothetical axis interpolation
J811	Thread cutting, Feed per revolution
J815	Polar coordinate system interpolation
J816	Cylindrical interpolation
J813	Circular thread cutting
J823	Automatic corner override
J825	2nd reference position return
J826	3rd and 4th reference position return
J828	Programmable data input
J829	Polar coordinate command
J835	Constant surface speed control
J836	Spindle positioning
J726	Cs contour control function
J841	Tool life management
J843	Tool pairs for tool life management : 512 pairs
J689	Tool pairs for tool life management : 1024 pairs
J846	Canned cycle
J852	Optional chamfering/corner R
J853	Programmable mirror image
J857	Tool offset
J858	Cutter compensation B
J861	3–dimensional cutter compensation
J875	Skip function
J876	Multi–step skip function
J877	High–speed skip signal input
J878	Automatic tool length measurement
J882	High–speed measuring position reach signal input
J883	Custom macro
J889	Interruption type custom macro
J897	Feed stop
J901	Sequence number comparison and stop
J903	Block restart
J907	Manual handle interruption
J908	Simultaneous operation by automatic and manual
J941	Stroke check before move
J942	External deceleration
J943	Moving axis signal

J970 Index table indexing
J622 Retrace function
J987 High–speed machining function
J993 Circular thread cutting B
J999 Exponential function interpolation
J601 Axis switching
J612 Programmable parameter input
J616 Tool offset used by tool number
J620 Normal direction control
J648 Rigid tapping
J650 Figure copying
J653 Override playback
J657 Tool retract & recover
J664 Twin table control
J681 Floating reference position return
J690 Involute interpolation
J692 Spline interpolation
J611 Tool length compensation in tool axis direction
J618 Functions for hobbing machine
J621 Functions for gas cutting machine
J623 Automatic exact stop check
J626 Torch swing
J627 In–acceleration/deceleration signal
J625 Gentle curve cutting
J651 3–dimensional coordinate system conversion
J675 Error detect
J884 Least input increment E
J720 Spline interpolation B
J748 Multiple rotary axis control
J691 Wheel wear compensation
J729 Rotary table dynamic fixture offset
J638 Active block cancel
J646 Axis name expansion
J820 Speed control at cutting point
J892 Gentle normal direction control
J874 Manual interruption during 3–dimensional coordinate system conversion
J927 Helical involute interpolation
J602 Plane conversion
J734 Multi–teaching function
J669 Electrical gear box for two axes
J686 Spiral or conical interpolation
J974 3–dimensional circular interpolation
J636 EGB axis skip function
J649 Electric gear box automatic phase synchronization
J659 Tool length compensation in specification direction
J893 Parallel axes control
J896 Operation check
J739 Synchronous or tandem control

(2) Multiblock Look- Ahead Acceleration/ Deceleration Before Interpolation

1) When several successive blocks, each specifying a very short travel distance, cause a small acceleration used for acceleration/ deceleration before interpolation, the feedrate of the tool may not reach a specified feedrate.

(3) Automatic Feedrate Control Function

1) When 0 is specified in parameter No. 7567 which specifies the maximum allowable feedrate for automatic feedrate control, this means that the maximum feedrate is 0. Therefore the alarm PS187 (feed zero) is issued even if a feedrate more than 0 is specified. So, be sure to set the parameter to an appropriate value other than 0.

2) If the override value is changed when the automatic feedrate control function is enabled, the feedrate determined by the automatic feedrate control function is overridden.

(4) Notes on Operation

1) If a code other than those listed in Section 2.1.2 is specified in the HPCC mode, an alarm is issued.

   To specify codes other than the specifiable codes, exit from the HPCC modeby specifying G05P0. If a subprogram is called in HPCC mode, it is executed in HPCC mode. When calling a macro, first exit from HPCC mode.

Sample program:

Main program

```
O0001;
G00X100. Y200. ;
G05P10000;          HPCC-ON
G91G01X100Y200Z300F2000;
X200Y300Z400;
X300Y400Z500;
X400Y500Z600;
X300Y400Z500;

      .

M98P0002;   SUB PROGRAM

X500Y400Z300;
X600Y500Z400;
      .
      .
X50Y100Z150;
G05P0;              HPCC-OFF
G04X3. ;
M30;
```

Subprogram

```
O0002;
G00X50. Y50. ;
G05P10000;          HPCC-ON
G02I20. F3000;
G01X100. ;
G03I80. ;
G01X- 50. ;
G02I100. F5000;
      .
      .
G01X200. Y300. F1500;
X50. Y100. Z150. ;

M99;
```

(Executed in HPCC mode)

2) A block specifying G05P10000 does not allow single block stop operation.
3) The secondary override function cannot be used even in the HPCC mode if the associated option is not specified.
4) External deceleration, F1 digit feed, and automatic corner override cannot be used.
5) In the HPCC mode, none of the following can be performed; the MDI operation, changing parameters in the MDI mode, and editing a program in the EDIT mode.
6) In the HPCC mode, none of the interlock function and machine lock function can be activated for each axis separately.(See the description of parameter No.1009.)
7) In the HPCC mode, the block start interlock signal and cutting block start interlock signal are disabled.
8) In the HPCC mode, manual handle interrupt operation and simultaneous manual/automatic operation cannot be performed. When performing these operations, stop the movement of the tool with the single block function or feed hold function beforehand.
9) In the HPCC mode, never switch a mirror image based on the external mirror (DI signal) and setting.
10) In the HPCC mode data cannot be input in the fixed-point format (specified when bit 1 of parameter No.2400 is set to 1).
11) Set bit 5 (DNC) of parameter No.0 to 0 to specify the HPCC mode in DNC operation using remote buffers. If this parameter is set to 1, the maximum feedrate of the tool along an axis cannot exceed that in memory operation.
12) Programs in the HPCC mode do not allow block restart operation.
13) Parallel axis control cannot be used.
14) In the HPCC mode, even if the option for axis name extension is specified, none of I, J, K, and E can be specified as axis addresses. The specifiable addresses are X, Y, Z, U, V, W, A, B, and C only.
15) An axis cannot be detached in HPCC mode.

(5) Notes on Cutter Compensation C

When the option for cutter compensation C is specified, cutter compensation C is enabled even in the HPCC mode. Operation in the offset mode is the same as when the HPCC mode is not set. For details, refer to the ”FANUC Series 15-MB Operator’s Manual (B-62084E).”

When using the cutter compensation C function in the HPCC mode, note the items below.

1) Be sure to code a G41/G42-G40 pair within a G05P10000-G05P0 pair. This means that the HPCC mode cannot be started or cancelled in the cutter compensation (G41/ G42) mode. If such an attempt is made, alarm PS630 is issued.

(Example of correct programming)

```
    .
G05 P10000 ;        ─────────────────────────┐
    .                                        │
G41 X _ Y_ D01 ;    ─┐                       │
    .                │ Cutter compensation   │
    .                │ (G41) mode            │
    .                │                       │
G40 X _ Y_ ;        ─┘                       │
    .                                        │ HPCC mode
    .                                        │
G42 X _ Y_ D02 ;    ─┐                       │
    .                │ Cutter compensation   │
    .                │ (G42) mode            │
    .                │                       │
G40 X _ Y_ ;        ─┘                       │
    .                                        │
G05 P0 ;            ─────────────────────────┘
    .
```

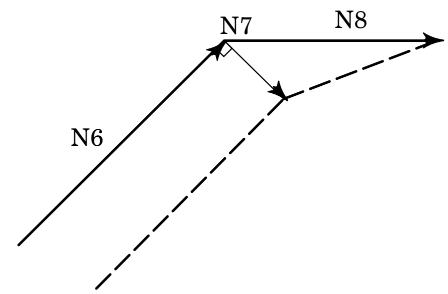
(Example of incorrect programming (1))

```
    .
    .
G41 X _ Y_ D01 ;
    .
G05 P10000 ;          ──> If the start of the HPCC mode is specified in the cutter
                          compensation mode, alarm PS630 is issued.
```

(Example of incorrect programming (1))

```
    .
    .
G05 P10000 ;
    .
G41 X _ Y_ D01 ;
    .
    .
    .
G05 P0 ;              ──> If the cancellation of the HPCC mode is specified in
                          the cutter compensation mode, alarm PS630 is
                          issued.
```

2) When a block not specifying the movement of the tool specifies G40 (cutter compensation cancel), a vector which has the same magnitude as the offset and which is perpendicular to the direction of movement specified in the previous block is generated. This means that the cutter compensation mode is cancelled with the vector retained. This retained vector is cancelled with the next move command.



If the cutter compensation mode is canceled with a vector retained, and the cutter compensation cancel command is immediately followed not by a move command but by an HPCC mode cancel command, alarm PS633 is issued.

```
      .
      .
N6 G91 X100.  Z100.  ;
N7 G40 ;
N8 G05 P0 ;                   ──> Alarm PS633 is issued.
      .
      .
```

3) When the HPCC mode is set, an offset change made in cutter compensation C becomes valid in the next block specifying a D code. So, the EV0 bit of parameter No6000 has no effect.

4) Specify the corner rounding code (G39) for the outside of a curve. If this code is specified for the inside of a curve, an excessive cut can occur.

(6) Notes no Positioning and auxiliary function command

By setting parameter no.8403#1 (MSU) to 1, positioning and auxiliary functi on (including 2nd auxiliary function) can be used.

Note the following when performing the those commands.

1) Subprogram call and macro call by G code and auxiliary function code cannot be specified. If specified, PS634 alarm may be generated.

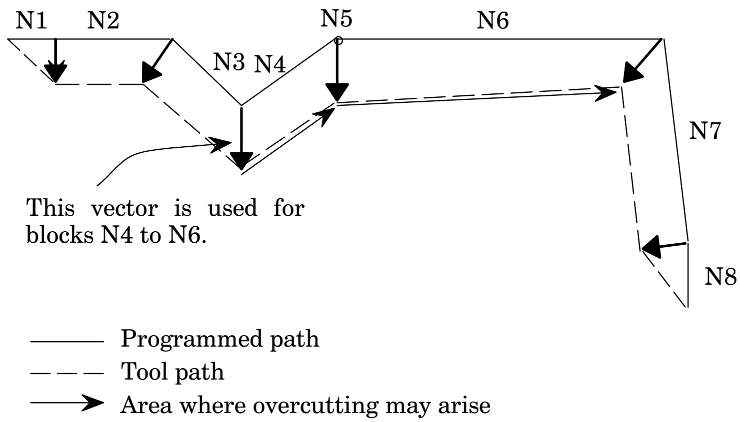
2) In a block where a positioning command or an auxiliary function is specified, cutter compensation (G41, G42) and cancel command (G40) cannot be specified simultaneously. Also G38 (vector retention) nor G39 (corner arc) cannot be specified. If specified, PS634 alarm will be produced.

3) When G00, M, S, T or B code is specified during cutter compensation, the cutter compensation vector created at the end of the block and the vector created at the end of the preceding block are the same as the vector created at the block one before the preceding block (the last vector when plural vectors exist).

Therefore, if G00, M, S, T or B is specified during cutter compensation in the HPCC mode, the workpiece may be overcut different from its operation other than in the HPCC mode. In this case, because PS272 (overcutting alarm) would not be informed, take special care in programming.

When parameter no.8403#7 (SG0) is set to 1 (G00 is replaced with G01 and the RISC processor lets the tool positioning), correct vector is created at the end of the G00 block, no overcutting will be generated. Refer to 4) below in this case.

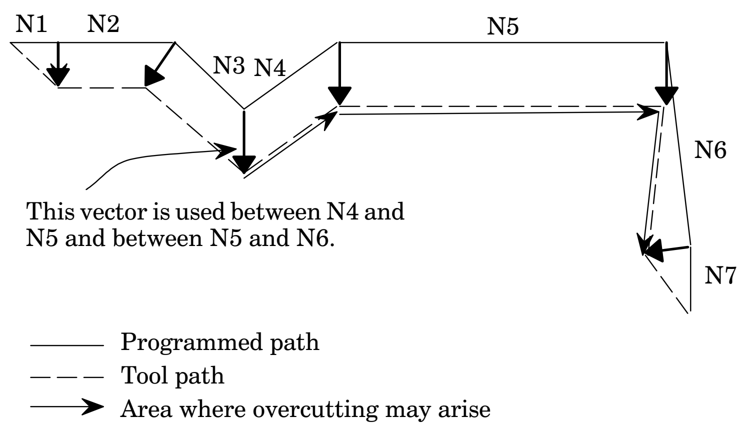
Example 3-1: When an auxiliary function is executed (Offset value D1=10mm)




```
O0001;
G92 G90 X- 10. Y20. ;
G05 P10000;
N1  G01  G42  X0  D1
F1000;
N2 X20. ;
N3 X40. Y0;
N4 X60. Y20. ;
N5 M01;
N6 X120. ;
N7 X130.  Y- 20. ;
N8 G40 Y- 30. ;
G05 P0;
M30
```

The starting point of N6 is determined by the vectors prepared by N3 and N4. CNC waites for FIN signal at the block of N5.

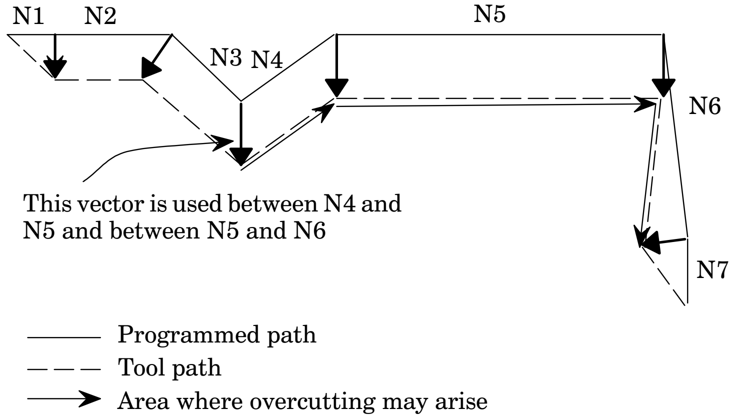
Example 3-2: When positioning is executed (Offset value D1=10mm)




```
O0002;
G92 G90 X- 10. Y20. ;
G05 P10000;
N1  G01  G42  X0  D1
F1000;
N2 X20. ;
N3 X40. Y0;
N4 X60. Y20. ;
N5 G00 X120. ;
N6 G01 X130.  Y- 20. ;
N7 G40 Y- 30. ;
G05 P0;
M30
```

The starting point of N5 is determined by the vectors at N3 and N4.
When parameter no. 8403#7 (SG0) is set to 1, the vector between N4 and N5 is determined correctly.

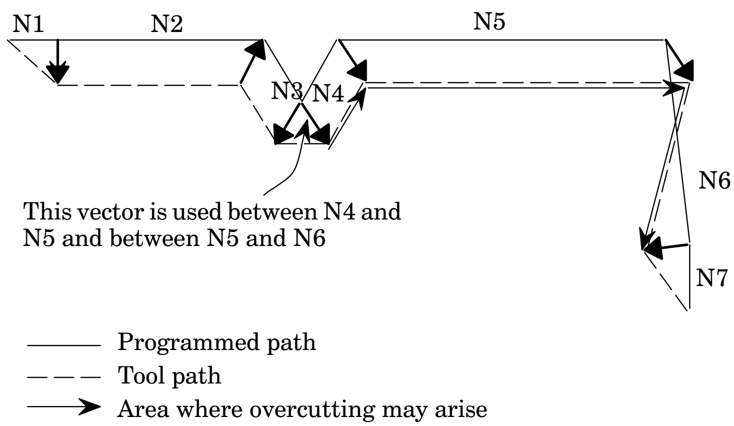
Example 3-3: When positioning and auxiliary function are specified simultaneously (Offset D1=10mm)




```
O0002;
G92 G90 X- 10. Y20. ;
G05 P10000;
N1  G01  G42  X0  D1
F1000;
N2 X20. ;
N3 X40. Y0;
N4 X60. Y20. ;
N5 G00 X120.  M01;
N6 G01 X130.  Y- 20. ;
N7 G40 Y- 30. ;
G05 P0;
M30
```

The movement is the same as example 3-2. The control waites for FIN at N5.

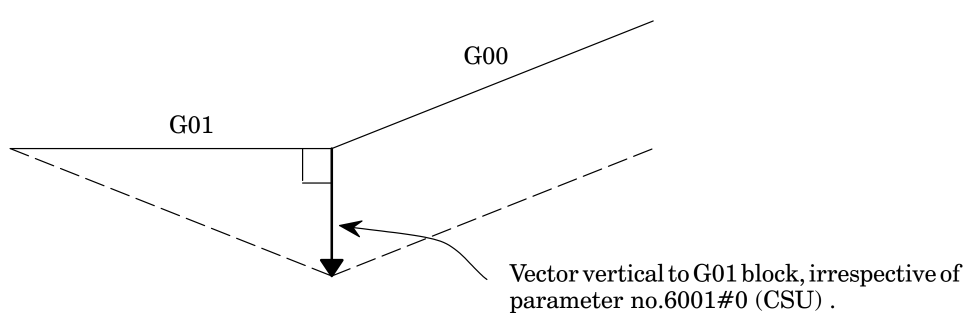
Example 3-4: When plural vectors are produced at the end of second block before the positioning block (Offset value D1=10mm)




```
O0002;
G92 G90 X- 10. Y20. ;
G05 P10000;
N1  G01  G42  X0  D1
F1000;
N2 X40. ;
N3 X50. Y0;
N4 X60. Y20. ;
N5 G00 X120. ;
N6 G01 X130.  Y- 20. ;
N7 G40 Y- 30. ;
G05 P0;
M30
```

When the start- up block is G01 and the next block is G00, the start- up operation is always type B (vector vetical to the start up block), irrespective of parameter no.6001#0 (CSU).

Example 3-5: When the start up block is G01 and the next block is G00



4) When G00 is specifed with parameter no. 8403#7 (SG0) =1, note the following points:

– Because G00 is replaced with G01 and executed, even if 2 axes is specified, the tool moves at a rate set to parameter no.8481.

Example When parameter no. 8481=1000 mm/ min and the following command is specified,

G00 X100. Y100.;

the feedrate does not become F1414, but F1000.

– Rapid traverse override is invalid and cutting feed override becomes effective.

– Time constant for acceleration/ deceleration after interpolation is of cutting feed.

– Linear or bell- shaped acceleration/ deceleration before interpolation is available.

– In- position check is not performed.

– Positioning is of linear interpolation type.

5) An M, S, T, or B code cannot be specified in a block which contains a cutting command (such as G01, G02, or G03).

M, S, T, and B codes must be specified separately from cutting commands.

(Bad coding)	G05 P1000 ; . . G90 G01 X100. Y100. M14 ; X200. Y300.; . . G05 P0 ;	Cannot be specified together.
(Good coding)	G05 P1000 ; . . M14 ; G90 G01 X100. Y100. ; X200. Y300. ; . . G05 P0 ;	Specified separately.

(7) Notes on inverse time feed

The following notes apply to HPCC mode when using a RISC processor. For general information, see the description of inverse time **(G93).**

1) G93 command

G93 (inverse time feed) and G94 (feed per minute) commands can be specified regardless of whether HPCC mode using a RISC processor (between G05P10000 and G05P0) is currently selected. Modal information for G93 and G94 remains valid even after HPCC mode using a RISC processor is canceled.

		Modal
O1000;		
G01 G91		G94
G93;		G93
G05 P10000;	HPCC mode	G93
X100. F30.;	HPCC mode	G93
G94;	HPCC mode	G94
X100. F3000;	HPCC mode	G94
G05 P0;	HPCC mode	G94
X100. F3000.;		G94
M30;		G94

2) High–precision contour control and inverse time command

When using an inverse time command together with high–precision contour control, set the high–precision contour control parameter as follows. This is the standard setting for high–precision contour control.

Data No.	#7	#6	#5	#4	#3	#2	#1	#0
7559							CTYP	CDEC

Data type : Bit

CDEC : To use high–precision contour control, set this bit to 0.

CTYP : To use high–precision contour control, set this bit to 1.

Usually, F commands are not modal in inverse time command mode. The feedrate must, therefore, be specified in every block. When the following parameters are set accordingly, however, no alarm is issued even if an F command is not specified in a block in inverse time command mode. Instead, the maximum feedrate for high–precision contour control is assumed as the feedrate for that block.

Data No.	#7	#6	#5	#4	#3	#2	#1	#0
7565	NOF							USE

USE : High–precision contour control is:
0 : Disabled.
1 : Enabled.

NOF : In blocks for which high–precision contour control is enabled, F commands are:
0 : Enabled.
1 : Ignored. The maximum feedrate for high–precision contour control is used instead.

When both the above parameters are set to 1, if a block specified in inverse time mode does not contain an F command, an alarm is not issued but the maximum feedrate for high–precision contour control is used.

3) Specifying F0 causes a no–feedrate–command alarm (PS187) to be issued.

4) This function (inverse time feed in HPCC mode using a RISC processor) is optional.

(8) Notes on the scaling and coordinate system rotation functions

The following notes apply to HPCC mode when using a RISC processor. For general information, see the descriptions of the **scaling (G50, G51)** and **coordinate system rotation (G68, G69)**.

1) The command for entering high–precision contour control mode can be specified only in G50 (scaling cancel)/G69 (coordinate system rotation cancel) mode. Specifying this command in G51/G68 mode causes an alarm.

2) Once scaling or coordinate system rotation has been turned on in high–precision contour control mode, it must be turned off before high–precision contour control mode is canceled.

Example

```
G05 P10000 ;
    :
G51 ...  ;
    :              Scaling mode      High–precision contour control mode
G50 ;
    :
G05 P0 ;
```

Specifying a command for exiting from high–precision contour control mode in G51/G68 mode causes an alarm.

3) Only the following G codes can be specified in G51/G68 mode:

G01/G02/G03
G17/G18/G19 (disabled in G68 mode)
G40/G41/G42/G38/G39
G50/G51
G68/G69
G90/G91
G00

To use a G00 command, set the SG0 bit of parameter No. 8403 to 1. The specified G00 command is, however, executed as a simple G00 command.

4) To use an auxiliary function command (M/S/T/B) in G51/G68 mode, set the MSU bit of parameter No. 8403 to 1. A block which contains an auxiliary function command (M/S/T/B) in G51/G68 mode must not contain any other commands. Otherwise, an alarm is issued.

5) If a scaling center is not specified in a G51 command in G51 mode, the current position becomes the scaling center. This differs from the processing performed when high–precision contour control mode is turned off (G05 P0 state).

6) In a G68 block, do not specify a command other than plane selection and the specification of the rotation center coordinates or rotation angle.

7) A G69 block must not contain other commands.

(9) Notes on the chopping function

The following notes apply to HPCC mode when using a RISC processor. For general information, see the description of the chopping function in this manual or that in the FANUC Series 15–MODEL B BMI Interface Connection Manual.

1) If any of the following operations (commands) is performed (executed) during chopping, the chopping axis operates in either of two ways depending on the operation (command), as follows:

(a) Stops chopping, moves to point R, then restarts chopping.

– G05P10000 or G05P0 command
– Single block stop
– Single block start
– Feed hold stop
– Restart after feed hold stop
– Reset
– Interlock for an axis other than the chopping axis
– All–axis interlock during automatic operation

The chopping axis does not, however, move to point R if the chopping override is set to 0 or if chopping operation has been stopped by driving the chopping stop signal low.

(b) Immediately terminates chopping.

– All–axis interlock
– Interlock for the chopping axis

2) A G81.1 command can be specified only before entering HPCC mode. A chopping cancel command (G80) can be specified only after canceling HPCC mode. Specifying a G81.1 command in HPCC mode causes a PS alarm.

Example

```
O0001 ;
   :
   :
G81.1 Rxx Zxx Qxx ;   ← Start chopping.
G05.1 P10000 ;        ← Start HPCC mode.
   :
   :
G05 P0 ;              ← Cancel HPCC mode.
G80 ;                 ← Cancel chopping.
   :
```

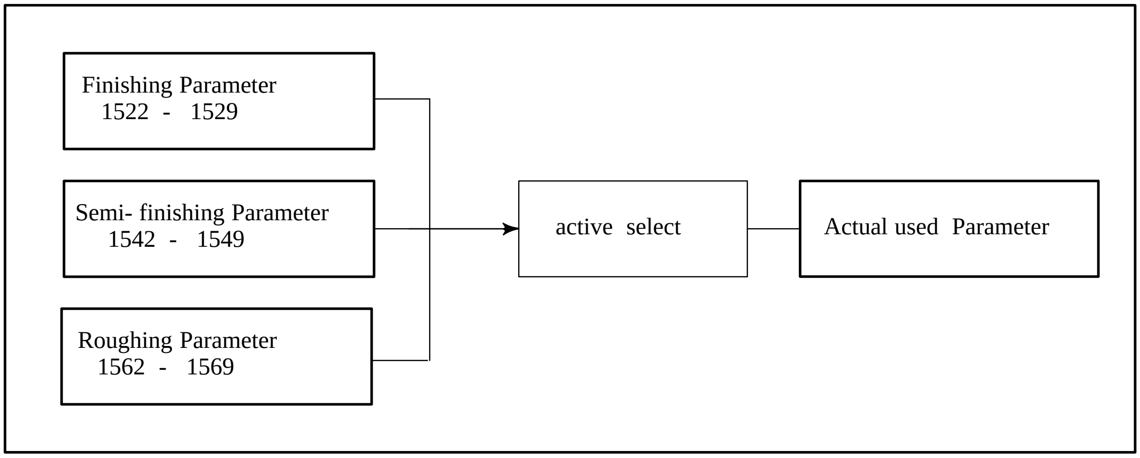
(10) Notes on PMC axis control

The following notes apply to HPCC mode when using a RISC processor. For general information, see the description of PMC axis control in the FANUC Series 15–MODEL B BMI Interface Connection Manual.

1) To apply PMC axis control in HPCC mode when using a RISC processor, specify the axes to be controlled by the RISC processor and those to be controlled by the PMC in advance, using a parameter (No. 1006 bit 6).

18.8 Maching Type in HPCC Screen Programming

The high-speed, high-precision machining setting screen supports parameter setting for three types of machining: finishing, semi-finishing and roughing. One of these three types can be selected in MDI mode.



When one of the three machining types in selected, the values set for the corresponding parameters are set in the parameters used for actual machining. When finishing is selected, for example, the values of the following finishing parameters are written into the parameters to be used for actual machining.

	Finishing Parameter	Actual used Parameter
T- CONST FOR BIPL	1523	1631
T- CONST FOR AIPL	1522	1635
CORNER FEED	1524	1478
CLAMP BY ACC	1525	1643
RADIUS	1526	1492
MAX FEED	1527	1490
MIN FEED	1528	1491
FEED FOWARD	1529	1985 (3488) (Note1)

This function allows one of the three types of machining to be selected by using a program command during actual operation, in addition to the selection possible in MDI mode. The parameters corresponding to the selected machining type are also displayed on the high- speed, high-precision machining screen.

(Command methods)

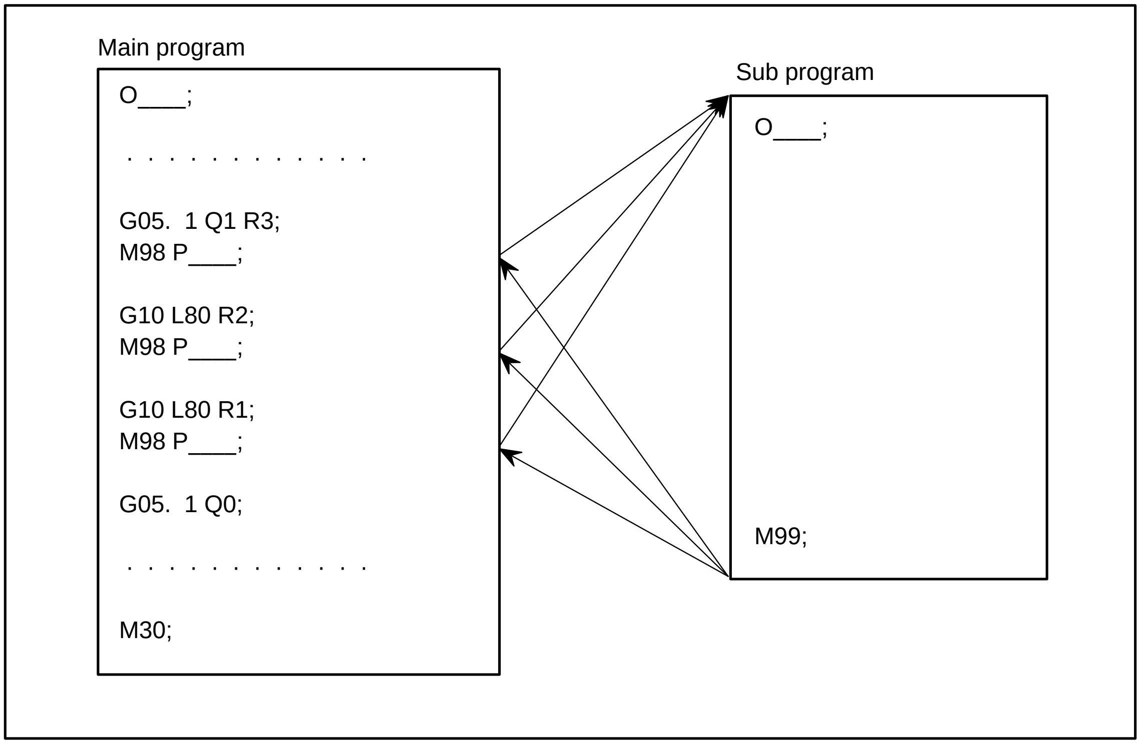
(1) If you want to set the mode to HPCC mode, and to select the pattern parameters,

G05.1 Q1 R* ;
- R1 : the transfer of finishing parameter
- R2 : the transfer of semi- finishing parameter
- R3 : the transfer of roghing parameter

(2) If you want to select the pattern parameters,

G10 L80 R* ;
- R1 : the transfer of finishing parameter
- R2 : the transfer of semi- finishing parameter
- R3 : the transfer of roghing parameter

Example



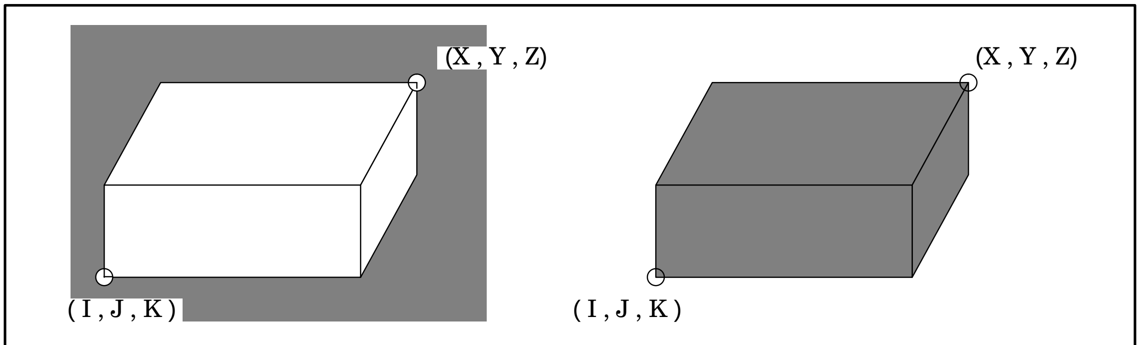
NOTE 1 Advanced feed- forward factor can be transferred to the spindle by setting TSP, bit 0 of Parameter No.8403.

NOTE 2 If R command other than R1, R2 or R3 is specified, alarm PS302 (ILLEGAL DATA NUMBER) is issued.

NOTE 3 Do not specify G05.1 Q1 R* ; in G05.1 mode. Use only G10 L80 R* ; in G05.1 mode. Alternatively, cancel G05.1 mode, then specify G05.1 Q1 R* ;.

19.STROKE CHECK (G22, G23)

The movable range of a tool can be set to one of the following areas. The hatched portions are areas in which the tool is prevented from moving.

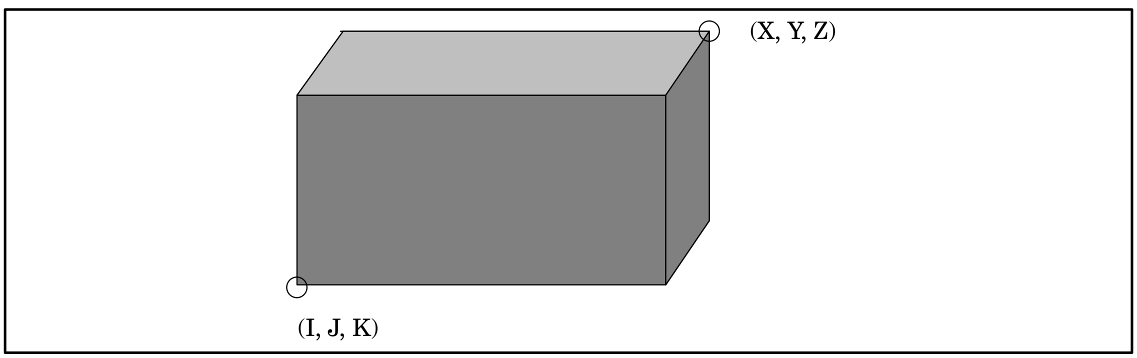


When the prohibited zone is outside When the prohibited zone is inside

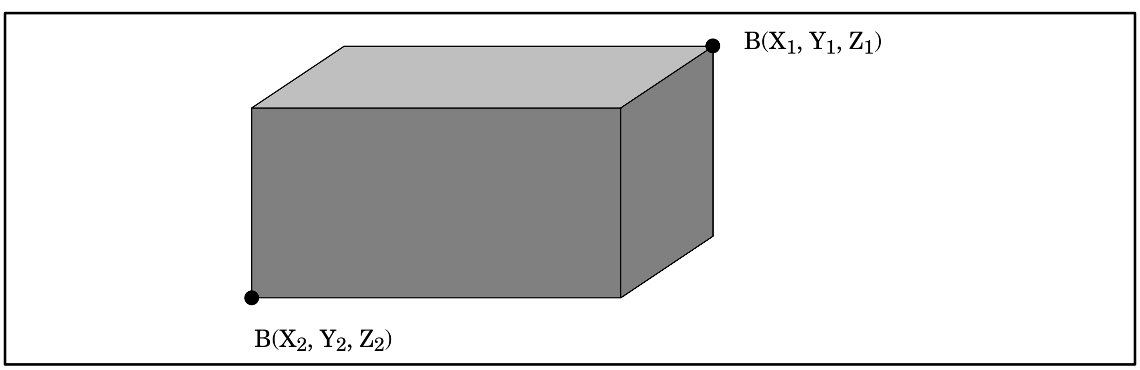
Stored stroke limit 1: A boundary is set by a parameter. The area beyond the set boundary is the prohibited zone. In general, once the boundary is set by the machine tool builder, it should not be changed. It is set at the maximum stroke of the machine.

Stored stroke limit 2: A boundary is set by a parameter or program. Either the area beyond or before the boundary is the prohibited zone. Whether the inside or outside is the prohibited zone is specified by a parameter. G22 disables the tool from entering the prohibited zone of limit 2. G23 enables the tool to enter the prohibited zone of limit 2. Setting and changing the boundary is programmed using the following command:

G22 X____Y____Z____I____J____K____;

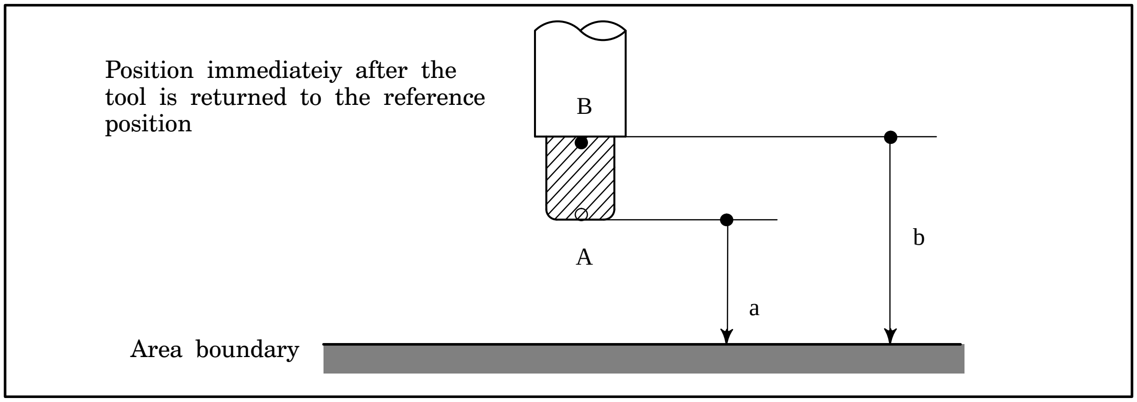


X > I , Y > J , Z > K
X – I > 2000 (least command increment)
Y – J > 2000 (least command increment)
Z – K > 2000 (least command increment)



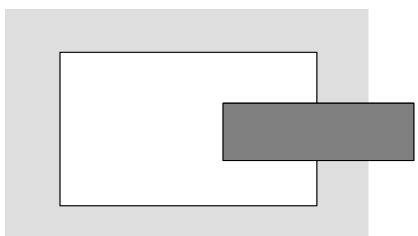
$X_1 > X_2$, $Y_1 > Y_2$, $Z_1 > Z_2$
$X_1 – X_2$ > 2000 (least command increment)
$Y_1 – Y_2$ > 2000 (least command increment)
$Z_1 – Z_2$ > 2000 (least command increment)

Values set by the parameters X, Y, Z, I, J, and K are set in the least command increment (output unit) according to the machine coordinate system where the reference position is assumed to be 0. The values programmed with G22, i.e., X, Y, Z, I, J, and K are set in the least command increment (input unit) according to the machine coordinate system where the reference position is assumed to be 0. The programmed values are converted to the least command input form and replaced by the values set by the parameters. The method for measuring X, Y, Z, I, J, and K differs depending on which point on a tool or tool holder is to be checked to see if it intrudes into the prohibited zone.



If the system is to check whether point A is in the prohibited zone, set a. If point B is to be checked, set b.

If the tip of a tool, such as point A, is to be checked while the tool length varies from tool to tool, the setting should be made for the longest of the tools used. This ensures safety and eliminates the need to change the setting when tools are changed. It is also possible to set an overlapping area.



WARNING When the power is turned on, it is necessary to return to the reference position manually or by G28 before the limits become valid.

NOTE 1 If the reference position is found to be in the prohibited zone defined by limits when return to the reference position has validated the stored stroke limit after power–on, an alarm occurs immediately. (Stored stroke limit 2 is available in the G22 mode only.)
If G23 is switched to G22 when the tool has entered the prohibited zone, an alarm occurs when movement starts in the next block.

NOTE 2 If the tool cannot go out of the prohibited zone in NOTE 1, press the emergency stop button, release it, then specify G23. In this case, if an incorrect clearance or set value is found, correct it, then repeat the steps from reference position return.

NOTE 3 Axes which are not provided with the reference position return function have no prohibited zone. Thus, no alarm occurs for such axes.

NOTE 4 If two points specified for setting a prohibited zone are identical, the following is assumed:
When the prohibited zone is outside……: The entire area is assumed as the prohibited zone.
When the prohibited zone is inside (for the G22 mode)……: The entire area is assumed as the tool movable area.

NOTE 5 Unnecessary limits must be set outside the machine stroke range.

NOTE 6 If intrusion into a prohibited zone causes an alarm, movement is allowed only in the direction opposite that in which the tool was moving when the alarm occurred.

NOTE 7 Even if two points are specified in the erroneous order, a rectangular prism with these two points defining opposite corners is formed as the limit range.

NOTE 8 G22–; and G23; must be specified individually in a block without any other codes.

NOTE 9 Stored stroke limit 2 is not available for additional axes.

19.1 Validating the Stored Stroke Limits (Series 10 type)

There are two modes for validating the stored stroke limits. In the valid mode, the stroke limits become valid when the power is turned on. In the invalid mode, it is necessary to manually return to the reference position or execute G28 to validate the stroke limits.

The valid or invalid mode is set by the BZR bit in parameter No. 5200.

(1) Setting

(a) Setting the valid mode

Set the BZR bit of parameter No. 5200 to 1.

As soon as the power is turned on, the stored stroke limits are checked.

(b) Setting the invalid mode

There are two methods for setting the invalid mode.

1) Set the BZR bit in parameter No. 5200. to 0.

2) Turn on the power while holding down both the P and CANCEL keys. The invalid mode is then entered even when the BZR bit in parameter No. 5200 is set to 1.

In this mode, the stored stroke limits are not checked until return to the reference position is performed manually or by G28 after the power is turned on.

WARNING In the valid mode, the stored stroke limits are checked based on the machine coordinates stored as parameters immediately before the power is turned off. If the tool moves immediately after the power is turned off, these parameters are not updated. As a result, when the power is turned on again, the stored stroke limits are checked with positions shifted by the amount of movement, and so the check is not correct.

NOTE 1 Changing the setting of the BZR bit in parameter No. 5200 does not change the modes immediately. The system enters the new mode after the power is turned off then on again.

NOTE 2 This function is unavailable in multiaxis systems.

20.AXIS CONTROL

20.1 Axis Interchange

Either settings or a signal sent from the machine can change the correspondence between the machine axes and the X, Y, and Z addresses in the move command that is input from memory, from tape, or in the MDI mode.

For example, this function enables the program specified on the coordinate system shown in Fig. 20.1(1) to mount workpieces as shown in Fig. 20.1(2).

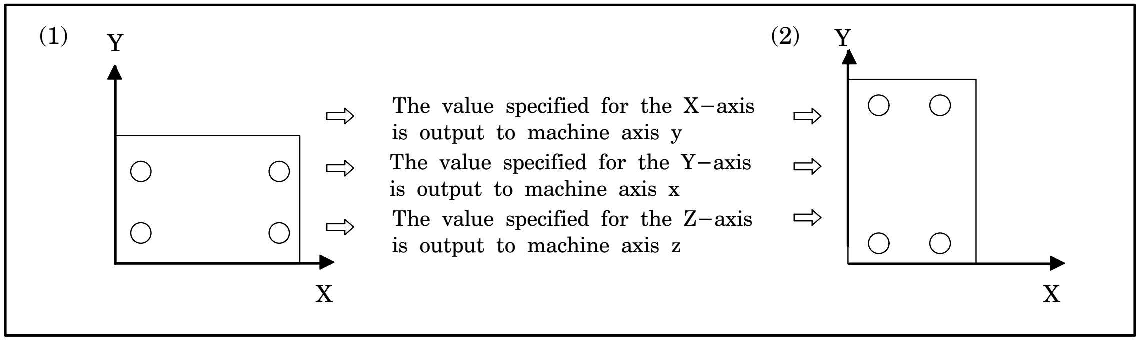


Fig. 20.1

This function allows six patterns of axis interchange. The table below shows the correspondence between program addresses X, Y, and Z and machine axes x, y, and z.

Axis interchange number	Program address		
	X	Y	Z
0(*1)	x	y	z
1	x	z	y
2	y	x	z
3	y	z	x
4	z	x	y
5	z	y	x

* 1 The axes are not interchanged.

The axes are not interchanged in manual operation.

The axes are not interchanged when the following commands are executed: Commands specifying movement to a specific machine position, commands concerning the machine coordinate system, and commands specifying a coordinate system, such as:

(i) Automatic return to the reference position (or the middle point): G28, G30

(ii) Return from the reference position (or the middle point)(NO TAG): G29

(iii) Stored stroke limit: G23

(iv) Specification of a coordinate system: G92

(v) Specification of an offset value: G10

(vi) Others

NOTE 1 Bit 0 of parameter No. 1001 specifies whether the axes are interchanged.
The result of axis interchange is displayed as the current position on the CRT screen.

Example
Start point : X0, Y0
Move command : G90X100.Y200.;
Axis interchange : The value specified for the X–axis is output to machine axis y.
The value specified for the Y–axis is output to machine axis x.
Current position displayed on the CRT screen after movement (absolute, relative, machine):
X200.0
Y100.0

The axis interchange number is specified by the data input in the MDI mode or the signal sent from the machine.

Setting data

Data number	Data
1049	Axis interchange number (0 to 5)

When either the input data or the signal sent from the machine sets the axis interchange number to zero, the axes are interchanged according to the non–zero interchange number. When neither interchange number is zero, the axes are interchanged according to the number specified by the signal sent from the machine. If axis interchange is not required, set both axis interchange numbers to zero.

A parameter can specify that the drilling axis used in a canned cycle or tool length compensation or the axis to be subjected to tool length compensation is always the Z–axis. If the axes are interchanged so that program address Z corresponds to machine axis x or y, the drilling axis or the axis of tool length compensation is also changed.

NOTE 2 The macro system variable corresponding to the new axis after axis interchange is used to read data.

Example When the X– and Y–axes are interchanged
Variable #5022 is used to read the machine coordinate after the X10.; command is executed.

20.2 Twin Table Control

The input signal sent from the machine can switch the operation mode of two or more specified axes: Synchronous, independent, or normal operation.

When a machine has two tables that are independently driven with different control axes, the following operation can be executed. The later part of this section describes the operation of the machine having two tables that are driven by the Y– and V–axes independently. If the actual machine has different axis names, read the Y– and V–axes as the actual axis names.

(1) Synchronous operation

This mode is used to machine a large workpiece that requires two tables at a time. When a move command is specified for one of the two axes, the other axis is synchronized with it. The axis to which the move command can be specified is called the master axis. The axis which is synchronized with the master axis is called the slave axis. When the Y–axis is used as the master axis and the V–axis as the slave axis, the Y– and V–axes operate in synchronization, as specified by command Yyyyy entered for the master axis.

In this synchronous operation, the move command specified for the master axis is simultaneously provided for the servo motors of the master and slave axes. The system does not carry out continuous detection of the difference between the two servo motors or compensate for the difference by adjusting the servo motor of the slave axis. It does not detect the synchronization error alarm.

The synchronous operation can be executed in automatic operation, continuous manual feed, manual handle feed, and incremental feed. It cannot be executed in manual return to the reference position.

(2) Independent operation

This mode is used to machine a small workpiece on either of the two tables. The move command specified for the master axis can determine the movement along the master or slave axis.

(i) Command Yyyyy specified for the master determines the movement along the Y–axis.

(ii) Command Yyyyy specified for the master axis determines the movement along the V–axis.

The command specified for the master axis can be used even when only the slave axis is operated. An identical command can be used, irrespective of the table on which the workpiece is placed.

Independent operation can be executed in automatic operation. When it is executed in manual operation, the resulting operation becomes the same as the normal operation described below.

(3) Normal operation

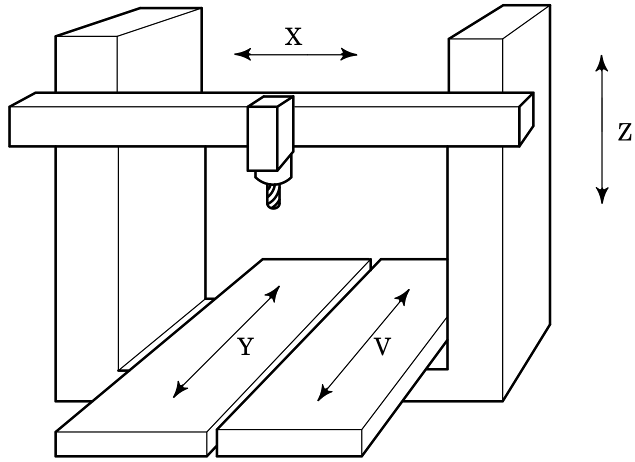
This mode is used to machine different workpieces on different tables. As in usual CNC operation, movements along the master and slave axes are specified by individual axis addresses, Y and V. The movements along the master and slave axes can be specified in an identical block.

(i) Command Yyyyy specified for the master axis determines movement along the Y–axis as usual.

(ii) Command Vvvvv specified for the slave axis determines movement along the V–axis as usual.

(iii) Command YyyyyVvvvv determines simultaneous movement along the Y– and V–axes as usual.

When the normal operation is executed in automatic or manual operation, the resulting operation becomes the same as the normal CNC operation.



WARNING 1 When the command for automatic return to the reference position (G28) or second, third, or fourth reference position return (G30) is issued in the synchronous operation mode, the tool returns to the reference position along the Y–axis and an identical movement is made along the V–axis. If the movement along the V–axis returns the tool to the reference position, the signal indicating that the reference position return is completed is set on for both the Y–and V–axes. However, it is recommended to specify the G28 and G30 commands in the normal operation mode.

WARNING 2 Manual return to the reference position must be executed in the normal operation mode.

NOTE 1 When the command for checking automatic return to the reference position (G27) is specified in the synchronous operation mode, identical movements are made along the Y– and V–axes. If the movements cause the tools to return to the reference position along both the Y– and V–axes, the signal indicating that reference position return is completed is set on for both axes. If they do not, an alarm occurs. However, it is recommended to specify the G27 command in the normal operation mode.

NOTE 2 The G28, G30, and G27 commands can be specified only when the tool is manually returned to the reference position after power is turned on.

NOTE 3 When the command to specify the origin of the workpiece coordinate system (G92) or of the local coordinate system (G52) or another command which does not move the tool along an axis is specified as command Yyyyy for the master axis, the axis to which the command applies depends on the control mode. When the axes are controlled in synchronization, the command is applied to the Y–axis. When the slave axis is independently controlled, the command is applied to the V–axis.

NOTE 4 When the slave axis is independently controlled, the external deceleration, interlock, and machine lock signals of only the slave axis are validated. When the axes are controlled in synchronization, the identical signals of only the master axis are validated and those of the slave axis are ignored.

NOTE 5 Parameter 7702 specifies the master and slave axes.

NOTE 6 Pitch error compensation and backlash compensation are executed for the master and slave axes separately.

NOTE 7 During the synchronous operation, the manual absolute switch (ABS) must be on (1). If it is off, movement along the slave axis may not be correctly controlled.

NOTE 8 Be sure to specify a plane selection code (G17, G18 or G19) immediately after switching to synchronous operation, single operation, or normal operation.

NOTE 9 When G17 (plane selection) is to be specified for a twin–table controlled axis, use one of the following formats:

(1) In a block that requires plane selection, specify G17 together with a move command.
In any block (G02, G03, G41, or G42) that requires G17, specify G17 and the movement end point together.

(2) Specify a twin–table controlled axis in the block specifying G17, as follows:
G17 X0; (when the twin–table controlled axes are X–U)

NOTE 10 Tool retraction and return function (see Section 5.6.4 of III) and twin table control
When the X–, Y–, Z–, V–, and W–axes are set as the controlled axes, and a master–to–slave relationship exists between the Y– and V–axes and Z– and W–axes during twin table control, the following operations can be performed:

Twin–table mode	When retraction displacement is set in G10.6	When retraction is performed manually
Synchronization mode	G10.6 X_ Y_ Z_ Sets the retraction displacement for the master axis.	The tool can be retracted and returned for both the master and slave axes.
Slave mode	G10.6 X_ V_ W_ If the retraction displacement is set for the slave axis only, master axis setting is unnecessary(*2).	Since slave mode is not supported for manual operation in the twin table system, the same operation as that performed in normal mode takes place. (All SYNC signals are off.)
Normal mode	An axis can be specified for the retraction displacement set in G10.6.	Normal tool retraction and return is performed.

* 1 Suppose that the retraction displacement in G10.6 is specified in normal mode.
* 2 The axis along which the tool is to be retracted in slave mode is that axis specified in G10.6. Therefore, if G10.6 Y10. ; is specified, for example, the tool is retracted along the Y–axis. In this case, the tool is not retracted along the V–axis.

20.3 Simple Synchronous Control

This function is exactly the same as twin table control, except for the following points:

(1) Unlike twin table control, this function does not allow independent operation. It only allows synchronous operation and normal operation. The move command specified for the master axis cannot move the tool along only the slave axis.

(2) This function allows manual return to the reference position in synchronous operation. If this is executed, however, the movements along the master and slave axes are synchronized with each other until they are decelerated. After that, the grid detection for the axes is executed independently of each other.

Simple synchronous control is the same as section 20.2 twin table control, except for the points described above. For details, see the section on twin table control.

20.4 Upgrades to Simple Synchronous Control Function

20.4.1 Synchronization error check

The machine coordinates of the master and slave axes are compared. If an error exceeding the value set with parameter No. 7723 is detected between the master and slave axis coordinates, alarm OT513 is issued and the machine is stopped.

This function always monitors the coordinates of the master and slave axes, thus preventing the machine from being damaged if the synchronous control selection signal (SYNCn) is turned off by mistake during simple synchronous control.

WARNING Synchronization error check can be performed only for linear axes. It cannot be performed for rotary axes. (Therefore, set 0 for parameter No. 7723.)

NOTE 1 Synchronization error check is not performed while the tool is moving at FL speed during manual reference position return.

NOTE 2 To perform synchronization error check, set an identical reference position for both the master and slave axes.

NOTE 3 When synchronization error check is not to be performed, set 0 for parameter No. 7723.

NOTE 4 To release alarm OT513, first set parameter No. 7723 to a value greater than the current value, then reset the system. Then, perform a manual operation, such as manual handle operation, to set the machine coordinates of the master and slave axes to the same values, then restore parameter No. 7723 to its original value.

NOTE 5 If alarm OT513 is detected during synchronous operation, always turn off the synchronous control selection signal (SYNCn) before releasing the alarm.

NOTE 6 The diameter/radius specification and increment system for the master and slave axes to be synchronized must be the same.

20.4.2 Synchronization alignment

Upon the release of the emergency stop, servo–off, or servo alarm state, the error between the master and slave axis coordinates is checked, after which compensation pulses are output to the slave axis to adjust its machine coordinates to the same values as those of the master axis.

For a motor having an absolute position detector, the above operation is also performed upon power–on.

This function eliminates any error between the coordinates of the master and slave axes. If the amount of compensation required for synchronization alignment exceeds the value set with parameter No. 7724, however, synchronization alignment is not performed but alarm OT513 is issued.

NOTE 1 Synchronization alignment can be performed only for linear axes.
It cannot be performed for rotary axes. (Therefore, set 0 for bit 0 of parameter No. 1010.)

NOTE 2 To perform synchronization alignment, set 1 for bit 0 of parameter No. 1010. Specify synchronization alignment for the slave axis.

NOTE 3 The synchronization alignment function is enabled only after the reference position has been established. Synchronization alignment is not performed before reference position return, even if bit 0 of parameter No. 1010 is set to 1.

NOTE 4 When synchronous control is applied to two or more pairs of axes, alarm OT513 for any pair of axes disables synchronization alignment for all pairs of axes.

NOTE 5 Synchronization alignment is not performed upon the release of alarm OT513. In this case, perform positioning in the same way as for synchronization check.

NOTE 6 The diameter/radius specification and increment system for the master and slave axes to be synchronized must be the same.

20.5 Chopping Function

To grind the side of a workpiece (contour grinding), continuously move the axis with a grinding wheel up and down while executing the contour program with another axis. This operation is called chopping. The G81.1 command specifies chopping and the G80 command cancels it.

Command format

G81.1 Z––– Q––– R––– F–––;

Z: Position of top dead center (The position can be specified for an axis other than the Z–axis. Specify the address corresponding to the axis name.)
Q: Distance from the top dead center to the bottom dead center (Specify an incremental value.)
R: Distance from the top dead center to point R (Specify an incremental value.)
F: Chopping feedrate

Chopping follows the steps described below:

(1) The tool is positioned at point R at the rapid traverse feedrate.

(2) The tool is moved to the bottom dead center at the feedrate specified by F.

(3) The tool is returned to the top dead center at the feedrate specified by F.

(4) The tool travels between the top dead center and bottom dead center at the chopping feedrate specified by F. The chopping feedrate can be overridden by pressing a switch on the machine operator's panel.

Chopping is continued even when the mode is switched to the manual mode or the automatic operation is halted by the feed hold function.

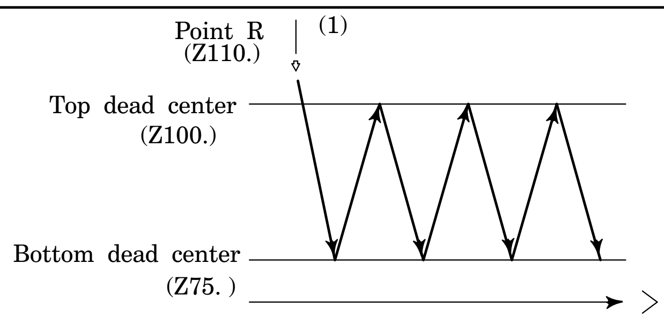
In the chopping mode, the move command for the chopping axis or the command of a canned cycle cannot be specified.

When G80 is specified, or when a reset, emergency stop, or servo alarm occurs, chopping is stopped and the tool is returned to point R.

Example G90G81.1Z100.Q–25.R10. F3000;
G01X——Y——F1200;
X——Y——;
......
......

G80

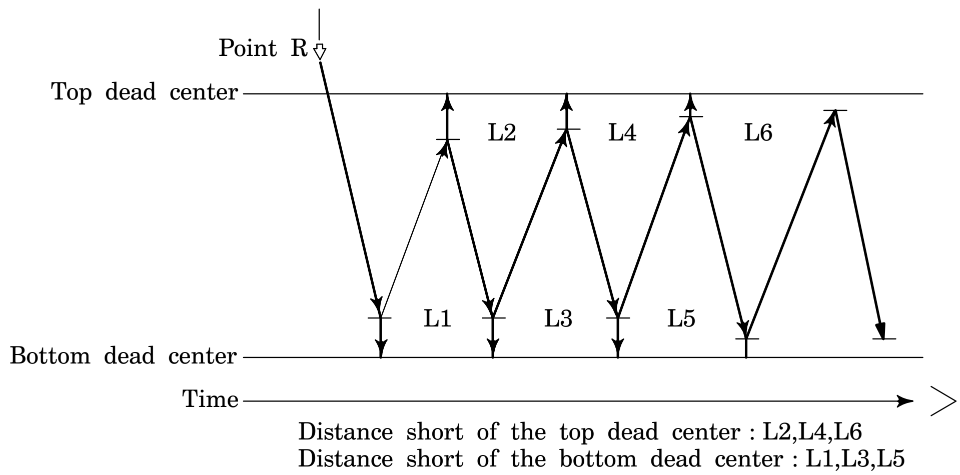
Point R = 100, Top dead center = 100, Bottom dead center = 75,
Chopping feedrate = 3000 mm/min, Feedrate on the XY plane = 1200 mm/min



WARNING	Buffering (advance reading) is suppressed for the chopping (G81.1) and chopping cancel (G80) commands. Note, therefore, that specifying chopping in cutter compensation mode may vary the way in which offset vectors are created.

NOTE 1 When the grinding axis is moved quickly in the chopping mode, the servo and acceleration/deceleration delays prevent the tool from reaching the points specified in the program. The control unit measures the differences between the points specified in the program and actual tool positions and automatically compensates for the differences. Adding the difference to the distance between the top dead center and the bottom dead center specified in the command, the control unit executes the chopping command so that the number of chopping actions per unit time equals the number specified in the program.

Compensation is started when the difference between the distances short of the top dead center and the bottom dead center becomes smaller than or equal to the value specified in parameter 1881. After compensation is started, the movement along the chopping axis exceeds the range specified by the top dead center and the bottom dead center and the chopping feedrate is gradually increased. When the differences between the actual tool positions and the specified tool positions become smaller than or equal to the value specified in parameter 1827, the compensation is stopped. The feedrate determined at that moment is retained and used during subsequent chopping. The compensation can be stopped also by the cancel command (G80), by activating the chopping stop switch, or by changing the override value.



Compensation is started when: | L3 – L2 | < Parameter 1881.
Compensation is stopped when:| L6 | < Effective area (parameter 1827). The feedrate detemined at that moment is retained and used during subsequent chopping.

NOTE 2 If the move command is issued for the chopping axis in the chopping (G81.1) mode, P/S alarm No. 472 occurs. If another G81.1 command is specified in the G81.1 mode, P/S alarm No. 472 occurs. To change the top dead center, bottom dead center, or point R, specify G80 to cancel the chopping mode, then specify another G81.1 command.

NOTE 3 The Q and R values of the G81.1 command are specified in the unit of the basic axis specified in parameter 1031.

NOTE 4 When the chopping stop switch on the machine operator's panel is activated, chopping is temporarily stopped. The table below lists the operations and command to stop chopping, the chopping stop positions, and the operations after chopping is stopped.

Operation or command	Stop position	Operation after chopping is stopped
G80	Point R	The chopping mode is canceled.
Emergency stop	Point where the operation is executed	The chopping mode is canceled.
Servo alarm	Point where the operation is executed	The chopping mode is canceled.
Reset	Point R	The chopping mode is canceled.
Chopping stop switch	Point R	The chopping mode becomes effective again after the chopping stop switch is turned off.

20.6 Parallel Operation

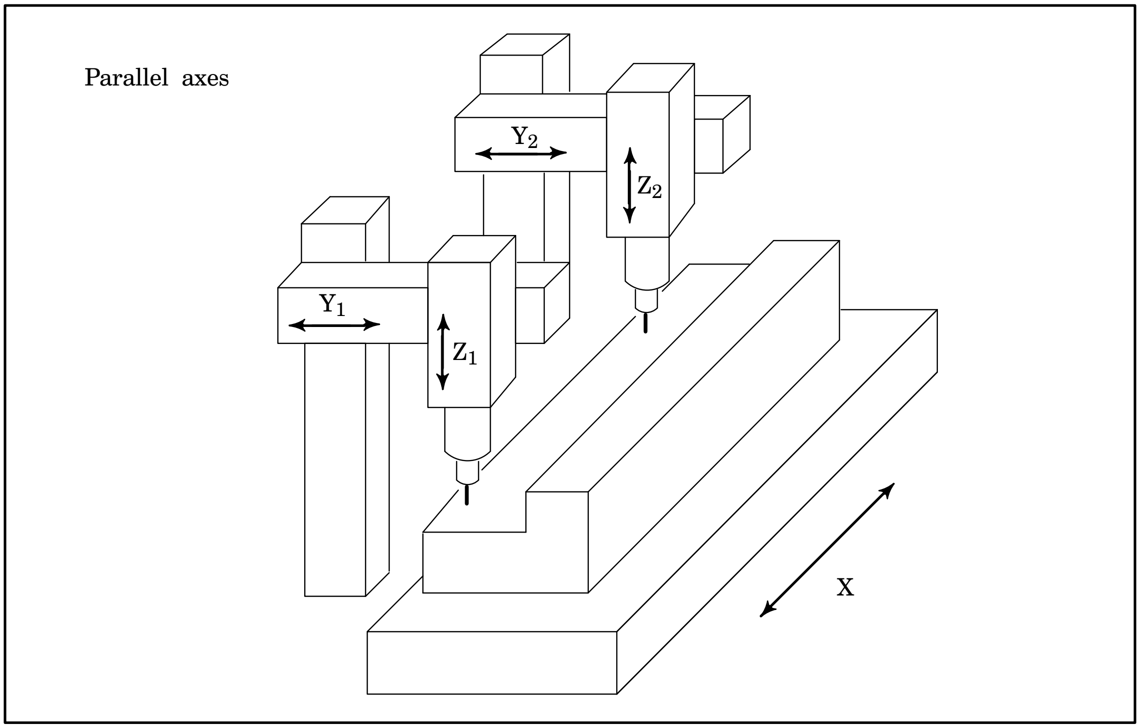
20.6.1 Parallel axis control

When a machine having two or more heads or tables is used to simultaneously machine two or more identical workpieces, parallel operation is executed. In parallel operation, the move command specified for a programmed axis simultaneously controls two or more control axes having the same name.

Either the move command specified for one programmed axis or the command of one address simultaneously controls multiple axes in parallel. These axes are called parallel axes.

This function can be used when the automatic or MDI operation is executed or when the manual numerical command is specified. This function cannot be used in normal manual operation. If this function is attempted in normal manual operation, the control axes are operated independently.

In parallel operation, the control axes represented by a single programmed axis are generally operated in a specified manner. When an external signal is used, one or more of the parallel axes can be selected and operated in a different manner (parking).



In the figure shown above, Y1 and Y2 are parallel axes and are operated in parallel by the command of a single address, address Y.

Z1 and Z2 are also parallel axes and are operated in parallel by the command of a single address, address Z.

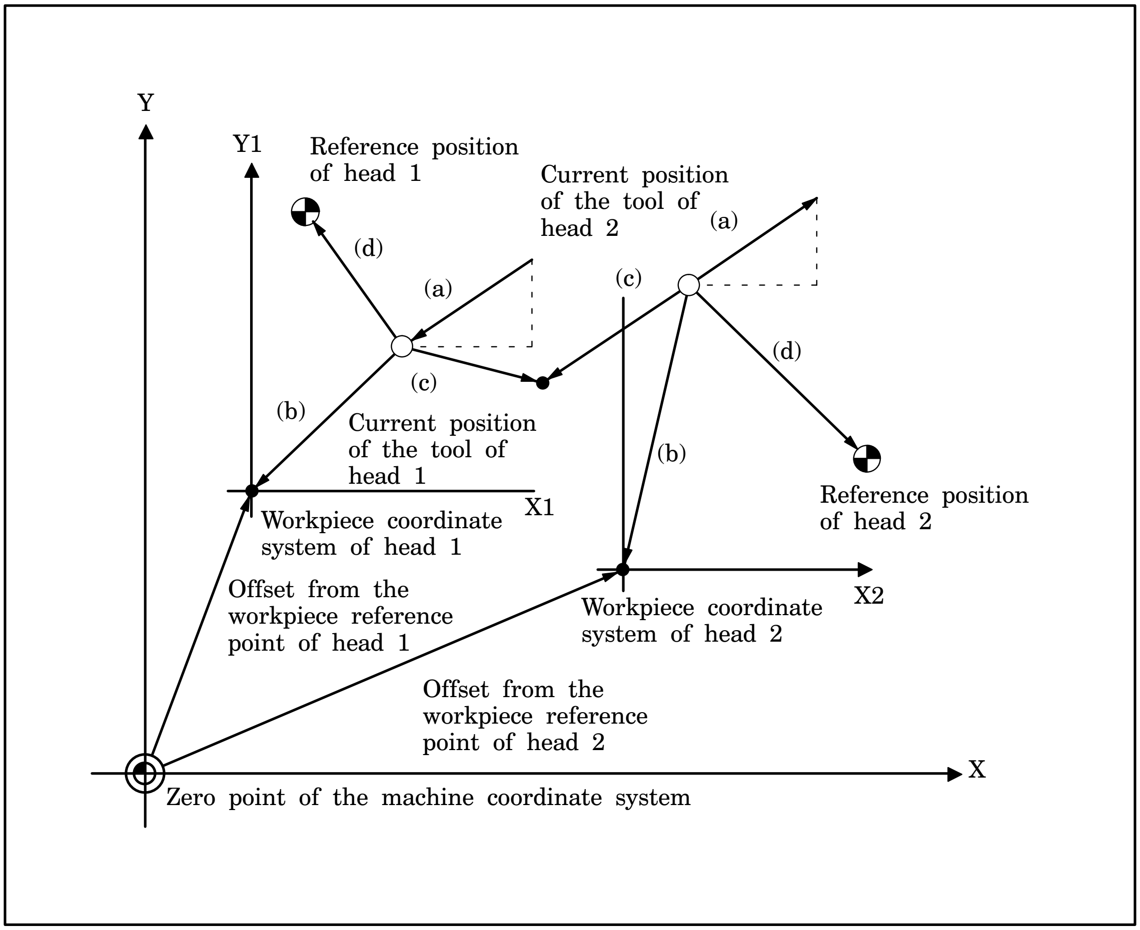
An external input signal selects one of the following two operations for each parallel axis:

- Normal (parking off): The axis is operated as specified by the command.
- Parking (parking on): The command is ignored and the axis is not operated.

20.6.2 Selection of the coordinate system in parallel axes

An individual offset from the workpiece reference point can be specified for each of the control axes represented by a single programmed axis. The coordinate systems of the control axes can be programmed independently.

When a machine has two heads and each head has two control axes X and Y, for example, the tools are operated as shown below.



(a) Incremental move command

Example G91 X__ Y__;

(b) Absolute move command

Example G90 X0 Y0;

(c) Move command on the machine coordinate system

The tool is moved to the point determined by the axes of the machine coordinate system.

Example G90 G53 X__ Y__;

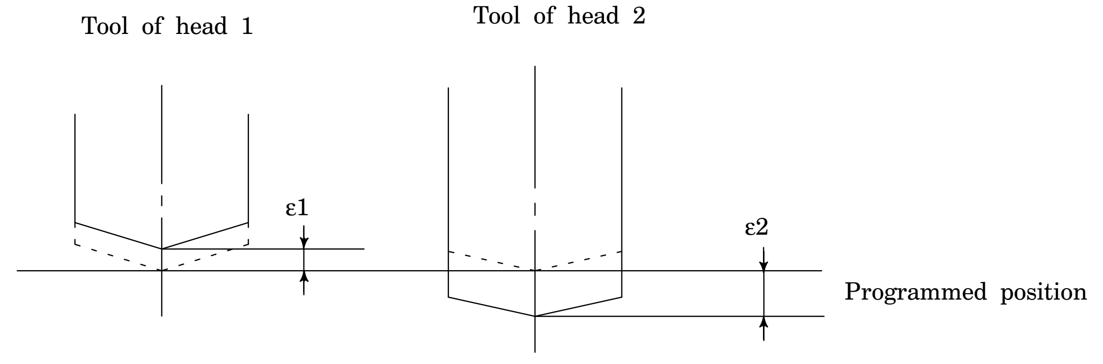
(d) Automatic return to the reference position (G27 to G30)

The tools are moved to the reference positions of the control axes, which are individually specified in the parameters.

Example G91 G28 X0 Y0;

20.6.3 Tool length compensation and tool offset in parallel axes

Tool length compensation can be executed for tools of individual axes when an H code number and the difference between the H code number and the corresponding offset data number, or a bias value, are specified for each axis in the parameter. In the same manner, tool offset can be executed.



Head	Offset number	Bias value	Offset data number	Offset value
Head1	07	10	17	ε1
Head2	07	20	27	ε1

For example, when the third and fourth axes are called as Z1 and Z2 and handled as parallel axes, specify 10 as the bias value of the Z1–axis and 20 as that of the Z2–axis in parameter No. 6021. Then enter G43H07; as the command of tool length compensation. Tool length compensation for the tool of head 1 is executed with the offset value obtained from offset data number 17 (07 + 10). The tool length compensation for the tool of head 2 is executed with the offset value obtained from offset data number 27 (07 + 20).

NOTE 1 The offset value corresponding to offset number 00, or H00, is always 0 and does not depend on the bias value.

NOTE 2 When the number of sums of the H code number and the bias value specified in a parameter exceeds the specified number, an alarm occurs.

NOTE 3 The bias value of tool offset is specified in parameter 6020. The bias value of tool length compensation is specified in parameter No. 6021.

20.6.4 Important matters concerning parallel operation

NOTE 1 The traveling distance along a parallel axis depends on whether the incremental or absolute command is specified.

· When the incremental command is specified

Rapid traverse, linear interpolation:

The traveling distances along the parallel axes are identical.

Circular interpolation, helical interpolation:

The traveling distances along the parallel axes are identical. Two or more identical arcs can be simultaneously drawn.

· When the absolute command is specified

Rapid traverse, linear interpolation :

The end point is specified by the command. If the parallel axes have different start points, the traveling distances along the axes are also different.

Circular interpolation, helical interpolation :

The pulse distribution is executed with the data of the axis having the smallest number of all control axes. An identical number of pulses is output for the other parallel axes. The end points of the axes are not necessarily identical.

NOTE 2 In linear interpolation, the feedrate of parallel operation is calculated from the data of the axis along which the largest movement is made of all control axes. The data of the other axes is not reflected in the calculation of the feedrate.

Example

Command	G01 G90 X10.0Y20.0F500;
Coordinate of the start point	X1 : 0.0
	X2 : 5.0
	Y : 0.0

Feedrate along each axis

Fx1 = 500 × 10/SQRT(10vv 2 + 20vv2)
Fx2 = 500 × 5/SQRT(10vv 2 + 20vv 2)
Fy = 500 × 20/SQRT(10vv 2 + 20vv 2)

NOTE 3 Tool length compensation and tool offset can be executed for an individual axis.

NOTE 4 In manual operation, the axes are operated independently of each another. The manual numerical command must be specified as in automatic operation.

NOTE 5 Parking signal

· When the parking signal is set off, the tool is moved along the axis.

· When the parking signal is set on, the move command for the axis is ignored. The G92 or G10 command for an axis is ignored when the parking signal is set on for the axis.

· The parking signal is processed according to the state when the tape for automatic operation is read. The parking signal does not affect the operation immediately after it is set on or off.

NOTE 6 The input or output signal specified for each axis such as overtravel or interlock is always applied to the corresponding axis even when it is a parallel axis.

NOTE 7 Cutter compensation cannot be specified for individual parallel axes. An identical offset is applied to all parallel axes.

NOTE 8 Axes are handled as parallel axes when identical names are specified in parameter No. 1020. When parallel axes are displayed on the CRT screen, they can be identified by subscripts specified in parameter No. 1021.

NOTE 9 When circular interpolation or cutter compensation is executed according to the data in parameter 1022, the coordinate axis to which each specified axis is parallel must be specified. An identical coordinate axis must be specified for the parallel axes.

20.6.5 Parallel axis control and external signal

External signals can stop (park) and reverse (mirror image) the operation of the parallel axis.

(a) Parking signal

The parking signal can stop individual parallel axes represented by a single programmed axis. Commands specified for the axis in the parking state are all ignored. It is not possible to place all parallel axes represented by the single programmed axis in the parking state during operation. If this is attempted, P/S alarm No. 180 occurs.

The parking signal must be set on or off in the clear state or an equivalent state (where the commands of offset, coordinate system rotation, and scaling are all canceled).

(b) External mirror image

The mirror image function can be executed for individual parallel axes represented by a single programmed axis.

These external signals can be usually set on or off by pressing a switch on the machine operator's panel or by specifying an M code. For details, refer to the manual supplied by the machine tool builder.

20.7 Roll–Over Function for a Rotation Axis

The roll–over function for a rotation axis prevents a coordinate overflow for the corresponding rotation axis.

When the roll–over function for a rotation axis is executed, each absolute coordinate is kept within the range of 0 to 359.999 degrees.

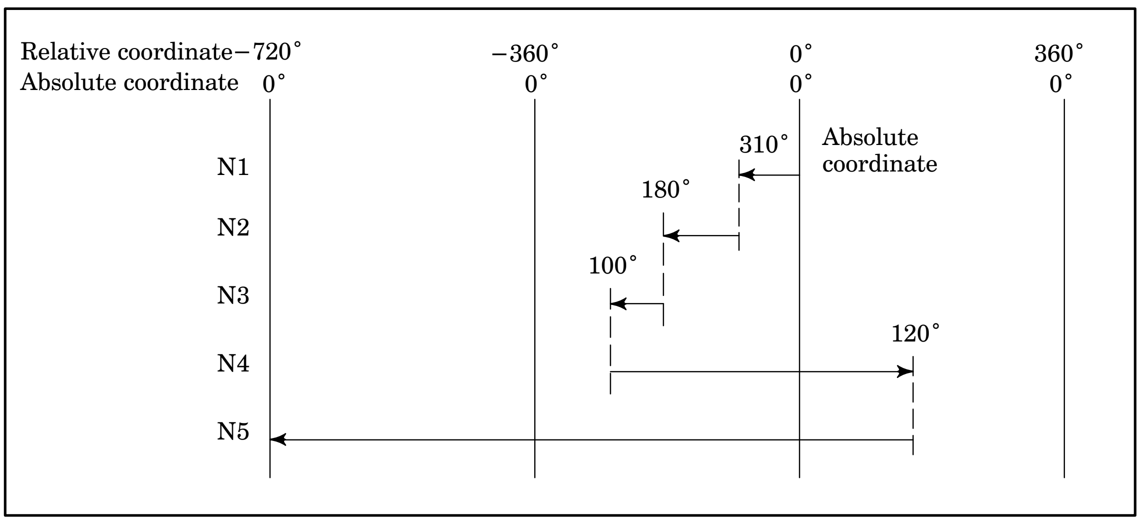
In the incremental mode, a specified value directly indicates an angular displacement. In the absolute mode, the specified value is converted to the remainder obtained by dividing the specified value by 360 degrees. The difference between the converted value and the current value indicates the angular displacement. The movement by angular displacement is always made in the shorter direction. That is, if the difference between the converted value and the current value is greater than 180 degrees, the movement to the specified position is made in the opposite direction. If the difference is 180 degrees, the movement is made in the normal direction.

(1) Command Specification

When the roll–over function is used for rotation axis A with the following program, the following movements are made about the rotation axis:

Example

	Remaining angular displacement	Absolute coordinate
G90 A0 ; ···		0°
N1 G90 A–50. ; ···	–50°	310°
N2 G90 A540. ; ···	–130°	180°
N3 G91 A–620. ; ···	–80°	100°
N4 G91 A380. ; ···	+380°	120°
N5 G91 A–840. ; ···	–840°	0°



(2) Notes

(a) This function is valid only for a rotation axis.

(b) The RDAx bit of parameter 1008 and RDA2x bit of parameter 1009 are used to enable or disable the roll–over function for a rotation axis.

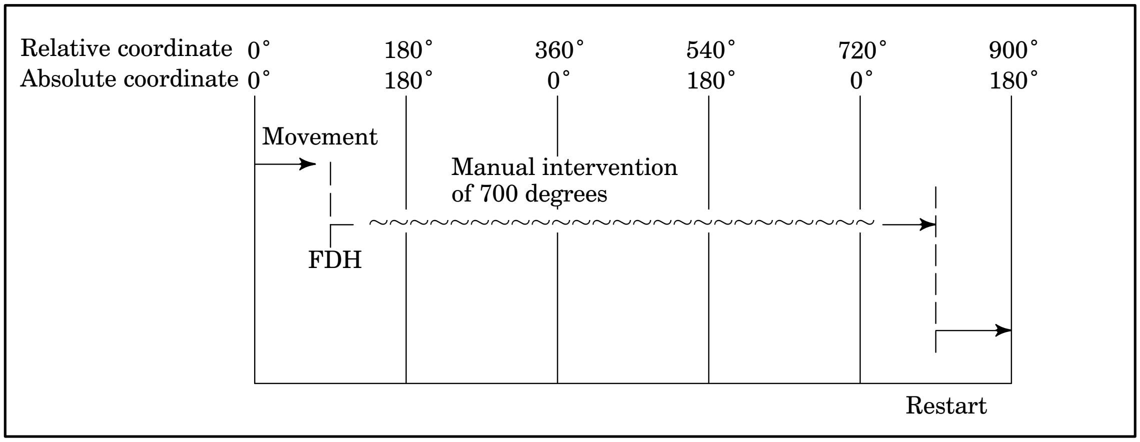
The ROPS bit of parameter 1002 specifies whether which is used, the RDAx or RDA2x bit.

Parameter 1002	ROPS=0	ROPS=1
Bit to enable or disable the roll–over function for a rotation axis	RDAx	RDA2x
Alarm issued when parameter is changed	Yes	No

(c) When this function is executed, an angle specified in the absolute mode is converted to the remainder obtained by dividing the specified value by 360 degrees. The difference between the converted value and the current value indicates the angular displacement. The movement angular displacement is always made in the shorter direction.

(d) When this function is used in the absolute mode, if the manual absolute switch is turned on to make a manual intervention during automatic operation the manual intervention is converted to the remainder obtained by dividing the intervention by 360 degrees. Then, a movement is made in the shorter direction.

Example G90 A0 ;
G90 A180. ;···After a manual intervention of 700 degrees is made, the movement is restarted.



(e) When this function is executed, the index table indexing function or rotary control function cannot be used.

20.8 Multiple Rotary Control Axis Function

The multiple rotary control axis function rotates a rotary table when the corresponding rotary control axis is specified in the PROT bit in parameter No. 1008.

When the incremental command is specified, the value specified in the command indicates the angular displacement.

When the absolute command is specified, the RSRV bit in parameter No. 1007 specifies either of the following operations:

(1) The NC unit rounds down the value specified in the absolute command (G90) to a value indicating a full turn or less. The difference between the rounded value and the current position is the angular displacement. When the RINC bit in parameter No. 1007 is specified, short–cut control can be executed. When the difference between the rounded value and the current position indicates more than a half turn, the short–cut control function allows reverse rotation up to the specified position.

(2) The sign of the specified value indicates the direction of rotation. (The plus sign (+) indicates counterclockwise rotation and the minus sign (–) clockwise rotation). The absolute value of the specified value indicates the destination.

This function can be executed while another axis control is executed.

The current position on the workpiece coordinate system is rounded to a value indicating a full turn or less and displayed on the screen.

20.8.1 Command procedure

When the rotary axis is called the B–axis, the following sample programs control the rotary axis as shown below:

(1) When the two bits RSRV and RINC in parameter No. 1007 are both set to 0, the NC unit rounds the specified value down to a value indicating a full turn or less. The difference between the rounded value and the current position is the angular displacement.

Example

G90B0 ; …Movement to the 0–degree position
G90B380. ; …Rotation by 20 degrees in the positive direction. The destination is the 20–degree position.
G90B–90. ; …Rotation by 250 degrees in the positive direction. The destination is the 270–degree position.
G90B60. ; …Rotation by 210 degrees in the negative direction. The destination is the 60–degree position.

(2) When the RINC bit in parameter No. 1007 is set to 1, the same program as shown in (1) above causes short–cut rotation.

Example

G90B0 ; …Movement to the 0–degree position
G90B380. ; …Rotation by 20 degrees in the positive direction. The destination is the 20–degree position.
G90B–90. ; …Rotation by 110 degrees in the negative direction. The destination is the 270–degree position.
G90B60. ; …Rotation by 150 degrees in the positive direction. The destination is the 60–degree position.

(3) When the RSRV bit in parameter No. 1007 is set to 1 and the RINC bit in parameter No. 1007 is set to 0, the sign of the specified value determines the direction of rotation and the absolute value of the specified value indicates the destination.

Example

G90B0 ; …Movement to the 0–degree position
G90B380. ; …Rotation by 20 degrees in the positive direction. The destination is the 20–degree position.
G90B–90. ; …Rotation by 290 degrees in the negative direction. The destination is the 90–degree position.G90B60.0; Rotation by 330 degrees in the positive direction. The destination is the 60–degree position.

20.8.2 Note

(1) This function is validated when the PROT bit in parameter No. 1008 is set only for the rotation axis. After the data of the parameter is changed, the power must be turned off.

(2) The following parameters are used for the multiple rotary control axis function. Set the following bits of parameter No. 1006 to 1.

ROTx (#0): Specifies whether the inch/metric mode must be switched for the rotation axis.

ROSx (#1): Specifies whether the machine coordinate system for stroke check and automatic return to the reference position uses the linear axis or rotation axis.

ROPx (#2): Specifies whether the machine coordinate system for stored pitch error compensation uses the linear axis or rotation axis.

Specify 0 (rotation axis) for the axis subjected to rotary control in parameter No. 1022.

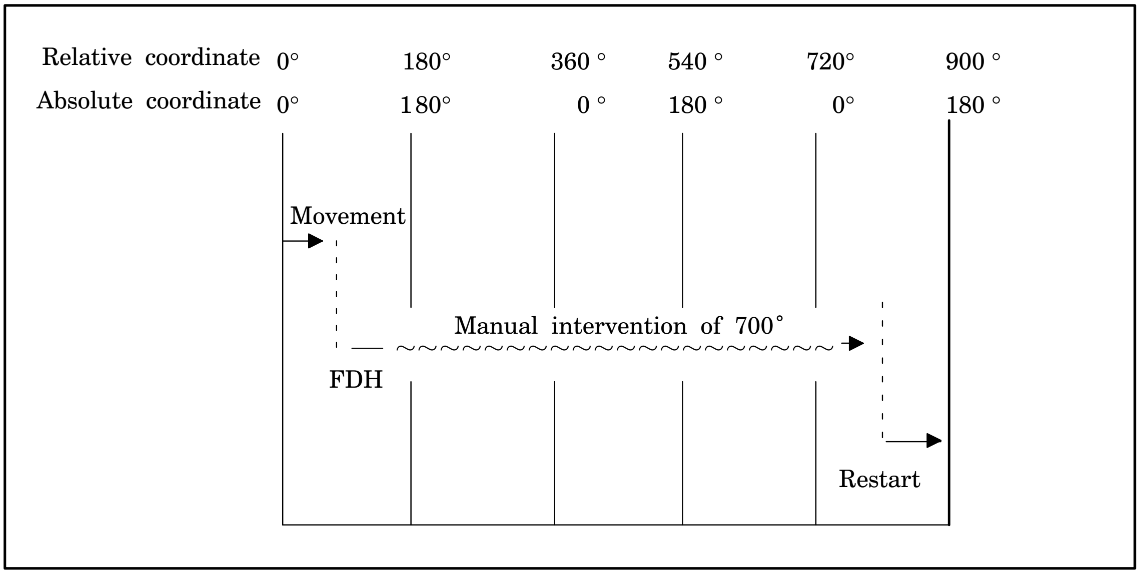
Specify the traveling distance per rotation of the rotation axis in parameter No. 1260.

Set SVFx of data 1802 to 1.

(3) If the manual absolute switch is set on to carry out manual intervention while this function is executed in automatic operation in the absolute mode, the amount of manual intervention is rounded down to a value indicating a full turn or less and the manual intervention is offset according to the rounded value.

Example

G90 B0 ;
G90 B180. ; ... Manual intervention of 700 degrees is carried out during traveling, then operation is resumed.



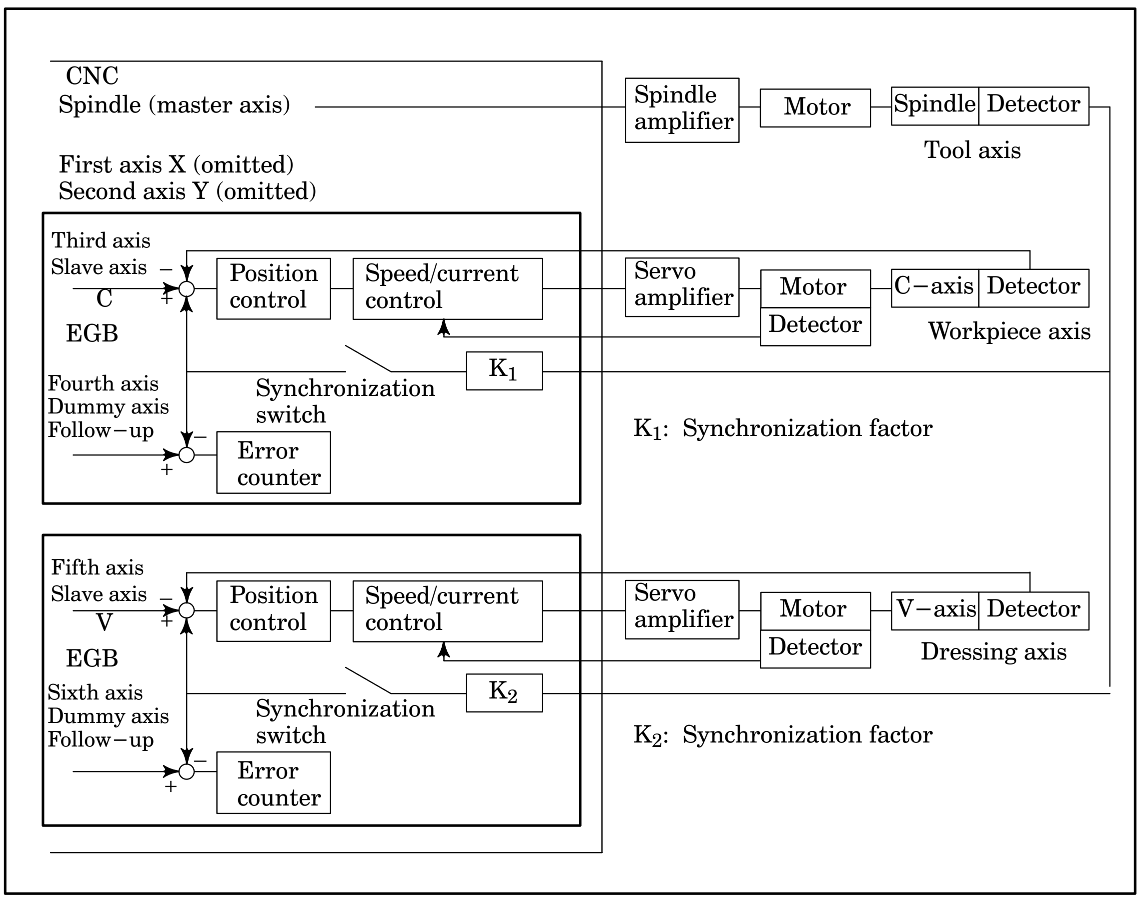
20.9 Two–Axis Electronic Gear Box

This function rotates a workpiece in synchronization with a rotating tool, or moves a tool in synchronization with a rotating workpiece to produce high–precision gears, screws, and so forth. A desired synchronization ratio can be programmed. This function can implement an electronic gear box (EGB) that enables the user to reprogram the synchronization ratio between a workpiece and tool.

When the two–axis electronic gear box option is selected, up to two groups of axes can be specified for synchronization. This means that on a gear grinder, for example, the user can use one axis to rotate a workpiece in synchronization with the tool, and can use the other axis to move the dressing axis in synchronization with the tool.

20.9.1 Example of controlled axis configuration (Gear grinder using the two–axis electronic gear box)

Spindle	:	EGB master axis: Tool axis
First axis	:	X
Second axis	:	Y
Third axis	:	C–axis (EGB slave axis: Workpiece axis)
Fourth axis	:	C–axis (EGB dummy axis: Not usable as an ordinary controlled axis)
Fifth axis	:	V–axis (EGB slave axis: Dressing axis)
Sixth axis	:	V–axis (EGB dummy axis: Not usable as an ordinary controlled axis)



20.9.2 Command specification

(1) Start of synchronization

Synchronization is started by specifying the ratio of the amount of slave axis movement to the amount of master axis movement using the following command:

$$\mathbf{G81.5}\ \underbrace{\left\{\begin{matrix}\mathbf{T}\boldsymbol{t}\\ \mathbf{P}\boldsymbol{p}\\ \alpha\boldsymbol{i}\end{matrix}\right\}}_{\text{Amount of master axis movement}}\ \underbrace{\left\{\begin{matrix}\beta\boldsymbol{j} & \\ \beta\mathbf{0} & \mathbf{L}\boldsymbol{l}\end{matrix}\right\}}_{\text{Amount of slave axis movement}}\ ;$$

Specification using G81.5 is allowed only when the two–axis electronic gear box option is selected

The amount of master axis movement is to be specified using one of the methods described below.

(a) Master axis speed

T $\underline{t}$: Specify the master axis speed in t (1 < t < 1000).

(b) Number of pulses for the master axis

P $\underline{p}$: Specify the number of pulses for the master axis in p (1 < p < 99999999). Specify this number with four pulses equaling one A/B phase cycle.

(c) Amount of master axis movement

α $\underline{i}$: Specify the address of the master axis in α .

Specify the amount of master axis movement in i with the least command increment. (The range of valid settings is the same as for movement along an ordinary axis.)

In specifying the start of synchronization, method (3) above for specifying the amount of master axis movement can be used only when the master axis is an NC controlled axis, and the feedback signal from the pulse coder used for master axis position control is also used as the pulse signal for slave axis synchronization. In this case, however, an alarm is issued if an axis that is not specified in parameter 5998 (for specifying which NC controlled axis to use as the master axis for a slave axis) is specified as the master axis.

Method (3) above cannot be used if the master axis is the spindle. Method (3) can also not be used if the feedback signal from a pulse coder other than the one used for master axis position control is used as the pulse signal for slave axis synchronization; this restriction applies even when the master axis is an NC controlled axis. In these cases, method (1) or method (2) must be used.

The amount of slave axis movement is to be specified using one of the methods described below.

(a) Amount of slave axis movement

β $\underline{j}$: Specify the address of the slave axis in β .

Specify the amount of slave axis movement in i with the least command increment. (The range of valid settings is the same as for movement along an ordinary axis.)

When $j = 0$ is specified, use of method (2) below is assumed. In this case, an alarm is issued if L is not specified.

(b) Slave axis speed

$\underline{\beta}$ 0 L $\underline{\pm l}$:

Specify the address of the slave axis in β .

Specify the slave axis speed in l (1 $\leqq$ l $\leqq$ 21).

(2) End of synchronization

(a) Axis–by–axis cancellation of synchronization using a command

The command below cancels synchronization.

G80.5 $\underline{\beta 0}$;

Specification using G80.5 is allowed only when the two–axis electronic gear box option is selected.

In β , specify the address of a slave axis. The synchronization of the slave axis specified in β is canceled. One block allows the synchronization of only one axis to be canceled. When β 0 is not specified, the synchronization of all synchronized axes is canceled. The absolute coordinates are updated according to the amount of synchronous movement. For a rotation axis, the amount of synchronous movement (modulus with respect to 360 degrees) is added to the absolute coordinates.

(b) Cancellation of synchronization by reset

The synchronization of an axis is canceled by a reset when the RSH bit of parameter 7612 is set to 0 (cancel the synchronization of the axis by a reset). Whether the absolute coordinates are updated depends on the manual absolute signal present at that time.

(c) Other causes of synchronization cancellation

Synchronization is automatically canceled when:

(a) Emergency stop is applied

(b) A servo alarm is issued

(c) An overtravel (hard/soft) alarm is issued for the EGB axis

(d) A PW000 alarm (requiring power–off) is issued

(e) A PC alarm or IO alarm is issued

20.9.3 Command specification for hobbing machines

To use this function with a hobbing machine, specify a command in the format below. Note, however, that this command specification method is not usable when an optional canned cycle is used.

When the two–axis electronic gear box option is not selected, only this command specification method for hobbing machines can be used to start synchronization.

When the two–axis electronic gear box option is selected, parameter 5995 determines which EGB axis is to be synchronized by this command specification.

(1) Start of synchronization

The command below starts the synchronization between the spindle and C–axis.

G81 T––– L––– Q––– P––– ;

T : Number of teeth (Range of valid settings: 1 to 1000)

L : Number of hob threads (Range of valid settings: –21 to +21 excluding 0).
Specify the direction of rotation about the C–axis with the sign of L:

– When the sign of L is positive, rotation about the C–axis is in the forward (+) direction.

– When the sign of L is negative, rotation about the C–axis is in the reverse (–) direction.

Q : Module or diametral pitch
Specify a module for metric input.

(Unit: 0.00001 mm, Range of valid settings: 0.1 to 25.0)

Specify a diametral pitch for inch input.

(Unit: 0.0001 inch^{-1}, Range of valid settings: 0.1 to 25.0)

P : Gear helix angle
(Unit: 0.0001 degree, Range of valid settings: –90.0 degrees to +90 degrees)

Specify P and Q when applying helical gear compensation. Specify a value for P and Q without using a decimal point.

When the synchronization mode is set by specifying G81, the EGB synchronization switch is closed, the in–synchronization signal SYNMOD (F153.5) goes high, and the synchronization between the master axis and workpiece rotation axis (C–axis) starts. This differs from the function for other types of hobbing machines, where synchronization starts with a single–rotation signal after G81.

During synchronization, the ratio of the rotation about the C–axis to the rotation of the spindle is made equal to the following ratio:

T (number of teeth): L (hob threads)

If another G81 command is specified during synchronization without canceling C–axis synchronization, the synchronization factor is updated by T and L if specified. If only P and Q are specified and T and L are not, helical gear compensation is applied with the synchronizing factor remaining unchanged. Thus, a helical gear and spur gear can be machined in succession.

(2) End of synchronization

The command below cancels synchronization.

G80 ;

The synchronization of all synchronized axes is canceled. The absolute coordinates are updated according to the amount of synchronous movement. For a rotation axis, the amount of synchronous movement (modulus with respect to 360 degrees) is added to the absolute coordinates. In a G80 block, no address other than O or N can be specified.

(3) Compensation for helical gears

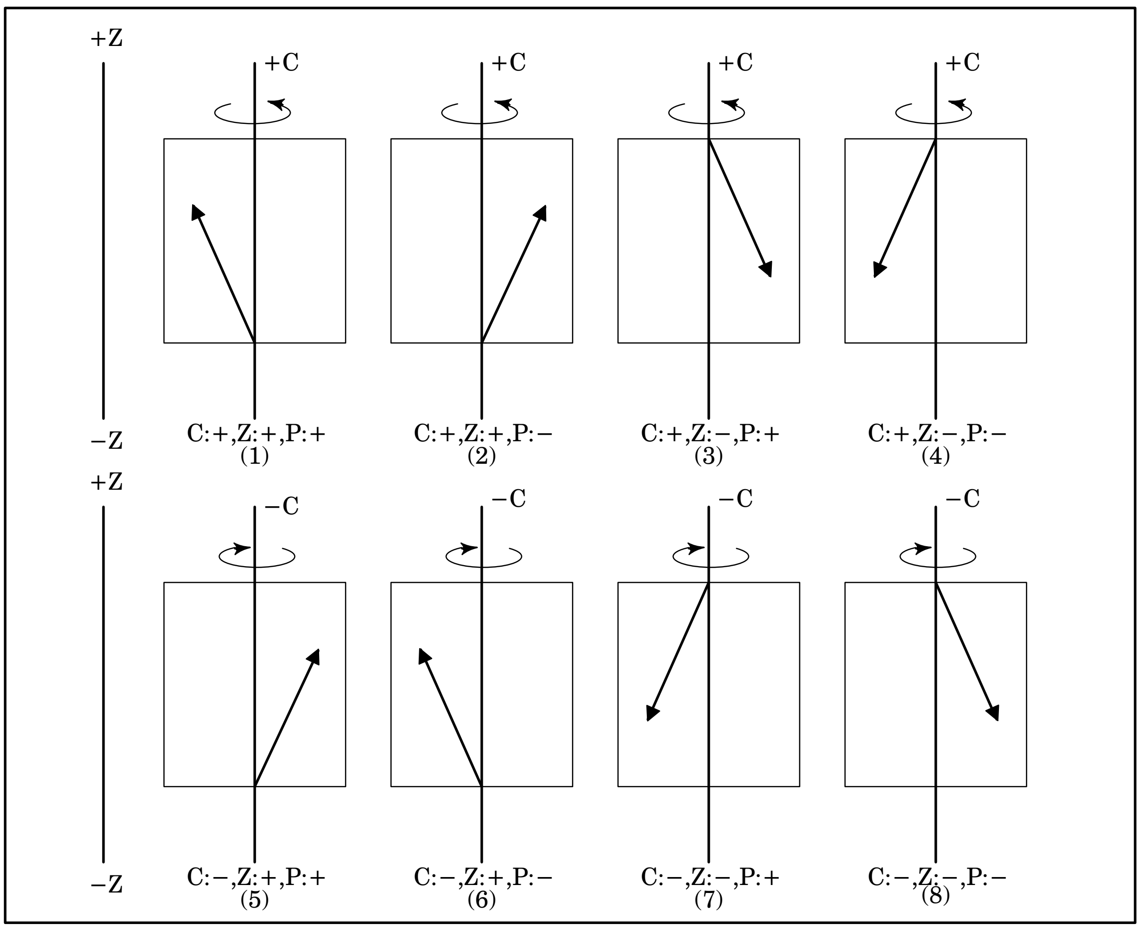
For a helical gear, a compensation movement along the C–axis is made with respect to movement along the Z–axis according to the helix angle of the gear.
Helical gear compensation is applied by adding compensation pulses calculated from the expression below to the C–axis synchronized with the spindle.

$$\text{Compensation angle} = \frac{Z \times \text{SIN}(P)}{\pi \times T \times Q} \times 360 \qquad \text{(for metric input)}$$

$$\text{Compensation angle} = \frac{Z \times Q \times \text{SIN}(P)}{\pi \times T} \times 360 \qquad \text{(for inch input)}$$

Compensation angle	:	Absolute value (degrees) with a sign
Z	:	Amount of Z–axis movement after G81 (mm or inch). All movements along the Z–axis including automatic and manual movements.
P	:	Gear helix angle (plus or minus value in degrees)
π	:	Ratio of the circumference of a circle to its diameter
T	:	Number of teeth
Q	:	Module (mm) or diametral pitch (1/inch)

The values of P, T, and Q are those specified in a G81 block.



20.9.4 Sample programs

Among the sample programs below, those that use G81.5 and G80.5 are allowed only when the two–axis electronic gear box option is selected.

(1) When the master axis is the spindle, and the slave axis is the C–axis

(a) G81.5 T10 C0 L1 ;

Synchronization between the master axis and C–axis is started at the ratio of one rotation about the C–axis to ten rotations about the master axis.

(b) G81.5 T10 C0 L–1 ;

Synchronization between the master axis and C–axis is started at the ratio of one rotation about the C–axis to ten rotations about the master axis. In this case, however, the direction of rotation is opposite to that of (a) above.

(c) G81.5 T1 C3.26 ;

Synchronization between the master axis and C–axis is started at the ratio of a 3.26–degree rotation about the C–axis per one rotation about the master axis.

(d) G81.5 P10000 C–0.214 ;

Synchronization between the master axis and C–axis is started at the ratio of a –0.214–degree rotation about the C–axis to 10,000 feedback pulses from the pulse coder of the master axis.

(2) When the master axis is the spindle, the slave axis is the V–axis (linear axis), and inch/metric conversion is performed

(a) For a millimeter machine and metric input

G81.5 T1 V1.0 ;

Synchronization between the master axis and V–axis is started at the ratio of a 1.00 mm movement along the V–axis per rotation about the master axis.

(b) For a millimeter machine and inch input

G81.5 T1 V1.0 ;

Synchronization between the master axis and V–axis is started at the ratio of a 1.0 inch movement (25.4 mm) along the V–axis per rotation about the master axis.

(3) When the master axis is the Y–axis (linear axis), the slave axis is the V–axis (linear axis)

(a) For a millimeter machine and metric input

G81.5 Y1.0 V0.25 ;

Synchronization between the Y–axis and V–axis is started at the ratio of a 0.25 mm movement along the V–axis to a 1.00 mm movement along the Y–axis.

(b) For a millimeter machine and inch input

G81.5 Y1.0 V0.25 ;

Synchronization between the Y–axis and V–axis is started at the ratio of a 0.25 inch movement along the V–axis to a 1.0 inch movement along the Y–axis.

(4) When two groups of axes are synchronized simultaneously

Based on the controlled axis configuration described in Section 20.9.1, the sample program below synchronizes the spindle with the V–axis while the spindle is synchronized with the C–axis.

O0100 ;	
....	
N01 Mxx ;	Performs spindle orientation.
N02 G00 G90 V... ;	Positions the dressing axis.
N03 G00 G90 C... ;	Positions the C–axis.
N04 G81.5 P12000 V0.5 ;	Starts synchronization at the ratio of a 0.5 mm movement along the V–axis every 12,000 pulses on the spindle.
N05 G81.5 T10 C0 L1 ;	Starts synchronization at the ratio of one rotation about the C–axis to ten spindle rotations.
N06 Myy S300 ;	Rotates the spindle.
N07 G01 G91 V... F... ;	Starts dressing.
.....	
.....	
N08 Mzz ;	Stops the spindle.
N09 G80.5 V0 ;	Cancels V–axis synchronization.
N10 Myy S300 ;	Rotates the spindle.
N11 G01 G90 X... Y... F... ;	Makes movements for grinding.
.....	
.....	
N12 Mzz ;	Stops the spindle.
N13 G80.5 C0 ;	Cancels C–axis synchronization.

V–axis synchronization mode

C–axis synchronization mode

Thus, the synchronizations of two groups can be started and canceled independently of each other.

(5) Command specification for hobbing machines

Based on the controlled axis configuration described in Section 20.9.1, the sample program below sets the C–axis (in parameter 5995) for starting synchronization with the spindle according to the command specification method for hobbing machines.

O1234 ;	
.....	
.....	
N01 G81 T20 L1 ;	Starts synchronization with the spindle and C–axis at the ratio of a 1/20 rotation about the C–axis to one spindle rotation.
N02 Mxx S300 ;	Rotates the spindle at 300 rpm.
N03 X... F... ;	Makes a movement along the X–axis (for cutting).
N04 Y... F... ;	Makes a movement along the Y–axis (for grinding). Axes such as the C–axis, X–axis, and Y–axis can be specified as required.
N05 X... F... ;	Makes a movement along the X–axis (for retraction).
N06 Mzz ;	Stops the spindle.
M07 G80 ;	Cancels the synchronization between the spindle and C–axis.

20.9.5 Synchronization ratio specification range

The programmed ratio (synchronization ratio) of a movement along the slave axis to a movement along the master axis is converted to a detection unit ratio inside the NC. If such converted data (detection unit ratio) exceeds a certain allowable data range in the NC, synchronization cannot be established correctly, and an alarm is issued.

Even when a programmed master axis movement and a programmed slave axis movement are within specifiable ranges, a detection unit ratio obtained by conversion can exceed the allowable range, thus resulting in an alarm.

Let K be a synchronization ratio. The internal data corresponding K is the amount of slave axis movement (Kn) represented in the detection unit divided by the amount of master axis movement (Kd) represented in the detection unit; this fraction is represented as Kn/Kd (reduced to its lowest terms) as indicated below.

$$K = \frac{Kn}{Kd} = \frac{\text{Amount of slave axis movement represented in the detection unit}}{\text{Amount of master axis movement represented in the detection unit}}$$

Kn and Kd must lie within the following ranges:

$$-2147483648 \leqq Kn \leqq 2147483647$$
$$1 \leqq Kd \leqq 65535$$

When Kn or Kd exceeds its allowable range above, an alarm is issued.

In conversion to the detection unit, when the CMR (command multiplication: parameter 1820) is a fraction or when inch/millimeter conversion is used, the fraction is directly converted without modification so that no error can occur in the conversion of specified amounts of movement.

During conversion, the amount of movement is multiplied by 254/100 for inch input on a millimeter machine, and 100/254 for metric input on an inch machine. Thus, Kn and Kd can become large numbers. If a synchronization ratio cannot be reduced to its lowest terms, an alarm condition is likely to occur.

In the examples below, those sample programs that include G81.5 and G80.5 can be used only when the two–axis electronic gear box option is selected.

Example 1 Based on the controlled axis configuration described in Section 20.9.1, suppose that the spindle and V–axis are as follows:

Spindle pulse coder	:	72000 pulses/rev (4 pulses for one A/B phase cycle)
C–axis least command increment	:	0.001 degree
C–axis CMR	:	5
V–axis least command increment	:	0.001 mm
V–axis CMR	:	5

Then, the C–axis detection unit is 0.0002 degree. The V–axis detection unit is 0.0002 mm. In this case, the synchronization ratio (Kn, Kd) is related with a command as indicated below. Here, let Pm and Ps be the amounts of movements represented in the detection unit on the master axis and slave axis specified in a synchronization start command, respectively.

(1) When the master axis is the spindle, and the slave axis is the C–axis

(a) Command: G81.5 T10 C0 L1 ;

Operation : Synchronization between the spindle and C–axis is started at the ratio of one rotation about the C–axis to ten spindle rotations.

Pm : (Number of pulses per spindle rotation) $\times$ 10 rotations $\rightarrow$ 72000 $\times$ 10

Ps : (Amount of movement per rotation about the C–axis) $\times$ CMR $\times$ (one rotation) $\rightarrow$ 360000 $\times$ 5 $\times$ 1

$$\frac{Kn}{Kd} = \frac{360000*5*1}{72000*10} = \frac{5}{2}$$

(b) Command: G81.5 T10 C0 L–1 ;

Operation : Synchronization between the spindle and C–axis is started at the ratio of one rotation about the C–axis to ten spindle rotations. In this case, however, the direction of rotation is opposite to that of (a) above.

Pm : (Number of pulses per spindle rotation) $\times$ 10 revolutions $\rightarrow$ 72000 x 10

Ps : (Amount of movement per rotation about the C–axis) $\times$ CMR $\times$ (one rotation) $\rightarrow$ –360000 $\times$ 5 $\times$ 1

$$\frac{Kn}{Kd} = \frac{-360000*5*1}{72000*10} = \frac{-5}{2}$$

(c) Command: G81.5 T1 C3.263 ;

Operation : Synchronization between the spindle and C–axis is started at the ratio of a 3.263–degree rotation about the C–axis to one spindle rotation.

Pm : (Number of pulses per spindle rotation) × 1 rotation → 72000 × 1

Ps : (Amount of C–axis movement) × CMR → 3263 × 5

$$\frac{Kn}{Kd} = \frac{3263*5*}{72000*1} = \frac{3263}{14400}$$

In this sample program, when T1 is specified for the master axis, the synchronization ratio (fraction) of the CMR of the C–axis to the denominator Kd can always be reduced to lowest terms, thus Kd falls in the allowable range. So, the specifiable range of C is as follows:

–99999999 < c < 99999999

(d) Command: G81.5 T10 C3.263 ;

Operation : Synchronization between the spindle and C–axis is started at the ratio of a 3.263–degree rotation about the C–axis to ten spindle rotations.

Pm : (Number of pulses per spindle rotation) × 10 rotations → 72000 × 10

Ps : (Amount of the C–axis movement) × CMR → 3263 × 5

$$\frac{Kn}{Kd} = \frac{3263*5*}{72000*10} = \frac{3263}{144000}$$

In this case, an alarm is issued because Kd exceeds the specifiable range.

(e) Command: G81.5 P10000 C–0.214 ;

Operation : Synchronization between the spindle and C–axis is started at the ratio of a –0.214–degree rotation of the C–axis to 10,000 feedback pulses from the pulse coder of the spindle.

Pm : (Specified number of feedback pulses from the pulse coder of the spindle) → 10000

Ps : (Amount of C–axis movement) × CMR → –214 × 5

$$\frac{Kn}{Kd} = \frac{-214*5}{10000} = \frac{-107}{1000}$$

(2) When the master axis is the spindle, the slave axis is the V–axis (linear axis), and inch/metric conversion is performed

(a) (For a millimeter machine and metric input

Command : G81.5 T1 V1.0 ;

Operation : Synchronization between the spindle and V–axis is started at the ratio of a 1.00 mm movement along the V–axis per spindle rotation.

Pm : (Number of pulses per spindle rotation) × 1 rotation → 72000 × 1

Ps : (Amount of V–axis movement) × CMR → 1000 × 5

$$\frac{Kn}{Kd} = \frac{1000*5}{72000} = \frac{5}{72}$$

(b) For a millimeter machine and inch input

Command : G81.5 T1 V1.0 ;

Operation : Synchronization between the spindle and V–axis is started at the ratio of a 1.0 inch move ment (25.4 mm) along the V–axis per spindle rotation.

Pm : (Number of pulses per spindle rotation) × 1 revolution → 72000 × 1

Ps : (Amount of V–axis movement) × CMR × 254 ÷ 100 → 10000 × 5 × 254 ÷ 100

$$\frac{Kn}{Kd} = \frac{1000*5*254}{72000*100} = \frac{127}{72}$$

(c) For a millimeter machine and inch input

Command : G81.5 T1 V0.0013 ;

Operation : Synchronization between the spindle and V–axis is started at the ratio of a 0.0013 inch (0.03302 mm) movement along the V–axis per spindle rotation.

Pm : (Number of pulses per spindle rotation) $\times$ 1 rotation $\rightarrow$ 72000 $\times$ 1

Ps : (Amount of V–axis movement) $\times$ CMR $\times$ 254 $\div$ 100 $\rightarrow$ 13 $\times$ 5 $\times$ 254 $\div$ 100

$$\frac{Kn}{Kd} = \frac{13*5*254}{72000*100} = \frac{1651}{720000}$$

In this case, an alarm is issued because Kd exceeds the specifiable range.

(3) When the master axis is the Z–axis (linear axis), the slave axis is the V–axis (linear axis)

(a) For a millimeter machine and metric input

Command : G81.5 Z1.0 V0.25 ;

Operation : Synchronization between the Z–axis and V–axis is started at the ratio of a 0.25 mm movement along the V–axis to a 1.00 mm movement along the Z–axis.

Pm : (Amount of Z–axis movement) $\times$ CMR $\rightarrow$ 1000 $\times$ 5

Ps : (Amount of V–axis movement) $\times$ CMR $\rightarrow$ 250 $\times$ 5

$$\frac{Kn}{Kd} = \frac{250}{1000} = \frac{1}{4}$$

(b) For a millimeter machine and inch input

Command : G81.5 Z1.0 V0.25 ;

Operation : Synchronization between the Z–axis and V–axis is started at the ratio of a 0.25 inch movement along the V–axis to a 1.0 inch movement along the Z–axis.

Pm : (Amount of Z–axis movement) $\times$ CMR $\times$ 254 $\div$ 100 $\rightarrow$ 10000 $\times$ 5 $\times$ 254 $\div$ 100

Ps : (Amount of V–axis movement) $\times$ CMR $\times$ 254 $\div$ 100 $\rightarrow$ 2500 $\times$ 5 $\times$ 254 $\div$ 100

$$\frac{Kn}{Kd} = \frac{2500*5*254*100}{10000*254*100} = \frac{1}{4}$$

Example 2 Based on the controlled axis configuration described in Section 20.9.1, suppose that the spindle and V–axis are as follows:

Spindle pulse coder	:	72000 pulses/rotation (4 pulses for one A/B phase cycle)
C–axis least command increment	:	0.001 degree
C–axis CMR	:	1/2
V–axis least command increment	:	0.001 mm
V–axis CMR	:	1/2

Then, the C–axis detection unit is 0.002 degree. The V–axis detection unit is 0.002 mm. In this case, the synchronization ratio (Kn, Kd) is related with a command as indicated below. Here, let Pm and Ps be the amounts of movements represented in the detection unit for the master axis and slave axis specified in a synchronization start command, respectively.

(1) When the master axis is the spindle, and the slave axis is the C–axis

(a) Command: G81.5 T1 C3.263 ;

Operation : Synchronization between the spindle and C–axis is started at the ratio of a 3.263–degree rotation about the C–axis per spindle rotation.

Pm : (Number of pulses per spindle rotation) $\times$ 1 rotation $\rightarrow$ 72000 $\times$ 1

Ps : (Amount of C–axis movement) $\times$ CMR $\rightarrow$ 3263 $\times$ 1 $\div$ 2

$$\frac{Kn}{Kd} = \frac{3263*1}{72000*2} = \frac{3263}{14400}$$

In this case, an alarm is issued because Kd exceeds the specifiable range.

(b) Command: G81.5 T1 C3.26 ;

Operation : Synchronization between the spindle and C–axis is started at the ratio of a 3.26–degree rotation about the C–axis per spindle rotation.

Pm : (Number of pulses per spindle rotation) × 1 revolution → 72000 × 1

Ps : (Amount of C–axis movement) × CMR → 3260 × 1 ÷ 2

$$\frac{Kn}{Kd} = \frac{3260*1}{72000*2} = \frac{163}{7200}$$

In this case, the ratio (fraction) of the amounts of movement an be reduced to lowest terms, so no alarm is issued. As indicated in (a) and (b) above, one command causes an alarm, and another very similar command causes no alarm, depending on whether the fraction of specified values can be reduced to lowest terms.

20.9.6 Retract function

In the automatic operation mode (MEM, MDI, etc.) or manual operation mode (H, JOG, etc.), a retract operation can be performed along the axis specified in parameter 1006 by the amount specified in parameter 7796 when the retract signal, RTRCT, goes high; a rising edge is detected. Upon completion of retract operation, the retract completion signal, RTRCTF, is output.

NOTE 1 Retract operation is performed at the feedrate specified in parameter 7795. In this case, the feedrate override capability is disabled.
NOTE 2 Feed hold cannot be applied to retraction.
NOTE 3 When the retract signal goes high during automatic operation, retract operation is performed, and automatic operation is stopped.
NOTE 4 Retract operation is performed by linear rapid traverse.

(1) Notes

NOTE 1 With a sampling frequency of 1 ms, a feedback pulse signal is read from the master axis, the number of synchronization pulses for the slave axis is calculated from the synchronization factor K, and the calculated value is used for slave axis position control.
NOTE 2 During synchronization, the slave axis and other axes can be subjected to manual handle interruption.
NOTE 3 During synchronization, movements for the slave axis and other axes can be specified by programming. Note, however, that a move command must be specified in incremental mode.
NOTE 4 The maximum feedrates along the master axis and slave axis depend on the position detector used.
NOTE 5 In the synchronization mode, G27, G28, G29, G30, or G53 cannot be specified.
NOTE 6 When the two–axis electronic gear box option is used, the method of specifying start of synchronization for the master axis based on the amount of movement is usable only when the master axis is an NC controlled axis, and the feedback signal from the pulse coder used for master axis position control is also used as the pulse signal for slave axis synchronization. In this case, however, an alarm is issued if an axis that is not specified in slave axis setting parameter 5998 (for specifying which NC controlled axis to use as the master axis for a slave axis) is specified as the master axis.
The method of specification for the master axis based on the amount of movement cannot be used if the feedback signal from a pulse coder other than the one used for master axis position control is used as the pulse signal for slave axis synchronization; this restriction applies even when the master axis is an NC controlled axis. In this case, the method of specifying the speed of the master axis or the method of specifying the number of pulses must be used.
NOTE 7 Controlled axis removal is not allowed for the EGB axis.
NOTE 8 In the synchronization mode, the inch/metric conversion commands (G20, G21, G70, and G71) cannot be specified.
NOTE 9 In the synchronization mode, only the machine coordinates are updated. The amount of synchronous movement (modulus with respect to 360 degrees) is added to the absolute coordinates at the time of synchronization cancellation.
NOTE 10 When a command for hobbing machines is specitied, threading and feed per rotation are performed based on the speed of a synchronized slave axis.
NOTE 11 Do not apply a machine lock to the EGB axis.

20.9.7 Skip function for EGB axis

This function provides capability that skip function and high speed skip function can be commanded for EGB (Electronic gear box) axis in synchroni-zation mode of EGB.

The features of this function are as follows.

1) The block with this functin is not interrupted until the skip signal input has been counted to the commanded times.
2) The movement by synchronization of EGB axis is not stopped by skip signal input.
3) The value of machine coordinate is stored in the commanded variable of custom macro when the skip signal is input. And the total number of times of skip signal inputs is stored in other commanded variable.

Please refer to the manual A- 57696 XFANUC Sereis 15/ 150- MA Specifications of Electronic gear boxY concerning the function of EGB.

(1) Command format

G81 T_ L_ ; (EGB mode on)

G31.8 G91 α0 P_ Q_ R_ ; (EGB skip command)

α : EGB axis.

P : The top number of custom macro variables in which the value of machine coordinate is set when skip signal is input.

Q : The total times of skip signal input during execution of the block with G31.8.

(range of command value : 1 - 200) .

R : The number of custom macro variable in which the total number of times of skip signal inputs is set. This data is usually the same with the data of the variable specified by Q.Therefore this is not necessarily specified. Specify R when the total number of times of skip signal inputs shoud be confirmed.

G31. 8 is an one- shot G code.

After the execution of G31. 8, values of machine coordinate which are gotten at every time of skip signal input are set in custom macro variables. The numbers of valiables are used from the top number commanded by P to the nunmber added with the amount of times commanded by Q.

And the total times of skip signal input is set in custom macro variable whose number is commanded by R.

Example

G81 T200 L2; (EGB mode on)

X__ ;

Z__ ;

G31. 8G91A0P500Q200R1; (EGB skip command)

After 200 times of skip signal inputs, 200 skip positions of A axis corresponding to each skip signal input are set in the custom macro variables whose numbers are from 500 to 699. And the times of skip signal input is set in the custom macro variable whose number is 1 and whose value is 200.

(2) Notes

NOTE 1 In this block, only one axis should be commanded for EGB axis.If no axis is commanded or, 2 or more axes are commanded, an alarm (PS150) will occur.

NOTE 2 If P is not commanded in this block, an alarm (PS150) will occur.

NOTE 3 If R is not commanded in this block, the times of skip signal input is not set in the custom macro variable.

NOTE 4 The numbers of custom macro variables commanded by P or R should be specified within usable numbers. If the number is specified without usable number, an alarm (PS114) will occur. And when variables become lacking, an alarm (PS11) will occur.

20.9.8 Electronic gear box automatic phase synchronization

(1) Acceleration/deceleration type

(a) Specification method

1) Starting synchronization

When synchronization is started, the workpiece axis speed is accelerated according to the acceleration rate set in the parameters (data Nos. 1758 and 1759).

Once the synchronization speed is attained, the synchronization signal (F153.5) is turned on.

G81 T— L— R1 ;
T : Number of gear teeth (valid range: 1 to 1000)
L : Number of hob threads (valid range: –21 to +21, except 0)
When L is a positive value : The C–axis turns in the positive direction (+)
When L is a negative value : The C–axis turns in the negative direction (–)

2) Canceling synchronization

The synchronization signal (F153.5) is turned off, and deceleration starts according to the acceleration rate set in the parameters (data Nos. 1758 and 1759).

Once the speed falls to 0, synchronization is canceled.

G80 R1 ;

(b) Sample program

M03 ;	Clockwise spindle rotation command
G81 T_ L_R1 ;	Synchronization start command
G00 X_ ;	Positions the workpiece at the machining position.

Machining in the synchronous state

G00 X_ ;	Retract the workpiece from the tool.
G80 R1 ;	Synchronization cancel command

(c) Notes

NOTE 1 Acceleration and deceleration are performed linearly.

NOTE 2 When synchronization is canceled automatically as a result of any of the following, deceleration is performed before the cancellation of synchronization:
(a) Reset
(b) PW000 (POWER MUST BE OFF)
(c) PC alarm or IO alarm

(2) Acceleration/deceleration and automatic phase synchronization

(a) Specification method

1) Starting synchronization

Once the synchronization speed is attained, phase synchronization is performed automatically. In phase synchronization, the workpiece axis is moved so that the workpiece axis position at the time of synchronization start command issue matches the one–rotation signal position of the spindle.

When phase synchronization is attained, the synchronization signal (F153.5) is turned on.

G81 T— L— R2 ;
T : Number of gear teeth (valid range: 1 to 1000)
L : Number of hob threads (valid range: –21 to +21, except 0)
When L is a positive value : The C–axis turns in the positive direction (+)
When L is a negative value : The C–axis turns in the negative direction (–)

2) Canceling synchronization

The synchronization signal (F153.5) is turned off, and deceleration starts according to the acceleration rate set in the parameters (data Nos. 1758 and 1759).

Once the speed falls to 0, synchronization is canceled.

G80 R2 ;

(b) Sample program

M03 ;	
G00 G90 C_ ;	C–axis positioning
G81 T_ L_ R2 ;	Synchronization start command
G00 X_ ;	Positions the workpiece at the machining position.

Machining in the synchronous state

G00 X_ ;	Retract the workpiece from the tool.
G80 R2 ;	Synchronization cancel command

(c) Warning and Notes

WARNING The speed and direction of rotation in automatic phase synchronization are set using parameters (data Nos. 5984 and 7712).
In phase synchronization, rapid traverse linear acceleration/deceleration is performed.
The speed in the workpiece axis is determined by superimposing the automatic phase synchronization speed on the speed synchronism with the spindle rotation.
The workpiece axis speed is controlled by clamping the phase synchronization speed so that the workpiece axis speed does not exceed the rapid traverse rate, as set with the corresponding parameter (data No. 1420).
If the speed synchronism with the spindle rotation is higher than the parameter–set rapid traverse rate (data No. 1420), automatic phase synchronization fails, causing a PS alarm (No. 586) to be issued.

NOTE 1 The one–rotation signal used for automatic phase synchronization is not generated by the spindle position coder. Instead, it is generated from a separate pulse coder which is mounted on the spindle to provide EGB feedback. Therefore, the orientation position obtained using the one–rotation signal of the spindle position coder does not match the reference position used for G81R2 automatic phase synchronization for the workpiece axis.

The one–rotation signal of the separate pulse coder must be turned on each time the spindle rotates through one turn.

NOTE 2 In automatic phase synchronization, the reference position used for phase synchronization for the workpiece axis can be shifted from the one–rotation signal position by parameter setting (data No. 5985).

NOTE 3 When a synchronization command is next issued while the synchronous state is set, automatic phase synchronization causes workpiece axis motion such that the spindle one–rotation signal matches the same workpiece axis position as that when the previous G81R2 synchronization start command was executed.

NOTE 4 In automatic phase synchronization, the workpiece axis is moved in the parameter–set phase synchronization move direction to the phase position nearest to the current position.

NOTE 5 The acceleration/deceleration performed at synchronization start and cancellation is linear.

NOTE 6 Acceleration/deceleration and automatic phase synchronization can be performed by setting the appropriate parameters (data No. 7712, EGBACC). In this case, the R2 command need not be specified in either block G81 or G80.

NOTE 7 When synchronization is canceled automatically as a result of any of the following, deceleration is performed, after which synchronization is canceled:

(a) Reset
(b) PW000 (POWER MUST BE OFF)
(c) PC alarm or IO alarm

NOTE 8 Set the flexible feed gear value for the dummy axis to 1/1.

NOTE 9 Set the parameter for the travel distance per rotation of the rotary axes (No. 1260) for the workpiece axis and dummy axis.

21.CNC COMMAND TO PMC (G10.1)

The G10.1 command transfers the data to the PMC.

Format

G10.1 P__R__L__;

The data transferred with addresses P, R, and L can be read in the read procedure of the PMC. For details, refer to the manual supplied by the machine tool builder.

22.TOOL RETRACTION AND RECOVERY

This function retracts the tool from the workpiece if the tool is broken during machining or when the machining status must be checked. It also efficiently returns the tool to resume machining.

The G10.6IP—; command specifies the retraction axis and the amount of retraction for the tool retraction and recovery function.

Example G10.6G91X10.Y5.;

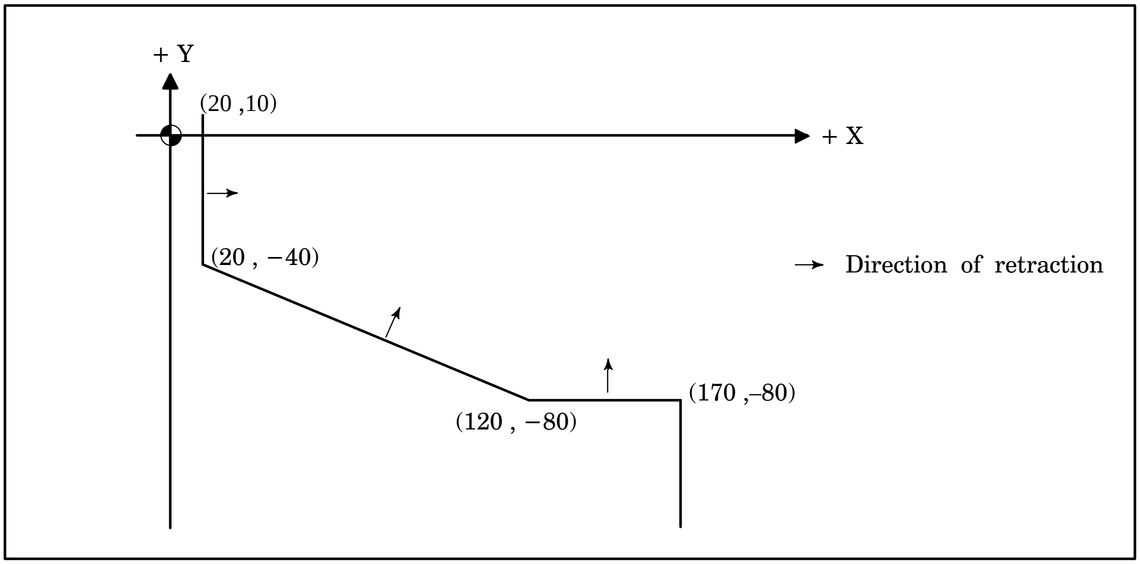
In the G10.6 command, the amount of retraction can be specified with an absolute or incremental value. When an absolute value is specified, the tool is retracted to the specified position on the workpiece coordinate system.

When the TRI (#0) bit in parameter No. 7614 is specified, the command is always processed as the incremental command. Even an absolute command is processed as the incremental command.

The retraction axis and the amount of retraction specified by the command continue to be valid until they are changed by another G10.6IP —; command. The G10.6IP —; command is a continuous–state command.

The retraction axis and the amount of retraction are canceled when the G10.6; command is specified or when a reset is executed.

Example G00X20. Z10.;
G10.6G91X10. ; (Retraction by +10.0 mm along the X–axis)
G01Y–50.F2 ;
G10.6X10.Y5. : (Retraction by +10.0 mm along the X–axis and 5.0 mm along the Y–axis)
G01X100.Y –40 ;
G10.6Y5. ; (Retraction by 5.0 mm along the Y–axis)
G01X50. ;
G10.6X10. ; (Retraction by +10.0 mm along the X–axis)
G01Y–50. ;
G10.6 ; (Retractions along all axes are canceled.)



NOTE	For the operation of the tool retraction and recovery function, refer to Section 5.6.3 of Part III.

APPENDIX

APPENDIX A. FUNCTION AND TAPE FORMAT LIST

The symbols in the list represent the following.

IP ______ : X ______ Y ______ Z ______ A …______

As seen above, the format consists of a combination of arbitary axis addresses among X, Y, Z, A, B, C, U, V, and W

x : First basic axis (X usually)

y : Second basic axis (Y usually)

z : Third basic axis (Z usually)

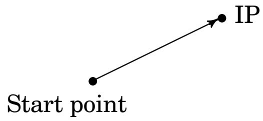
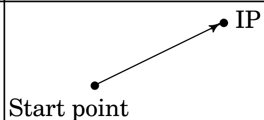
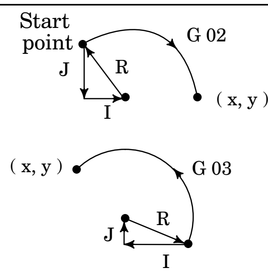
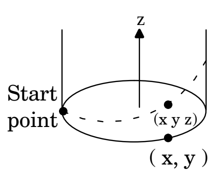
α : One of arbitary addressed

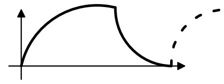
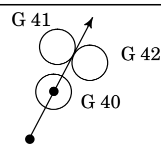
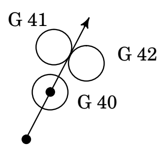
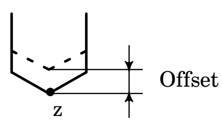
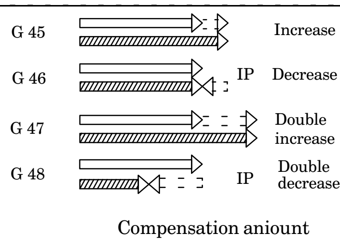
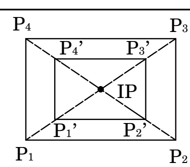
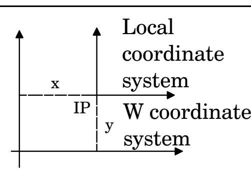
β : One of arbitary addressed

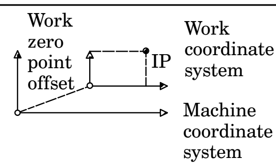
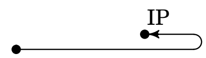
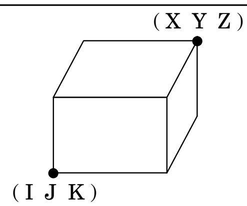
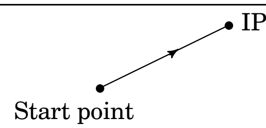
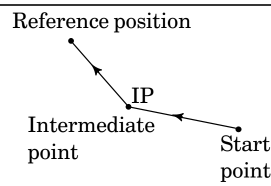
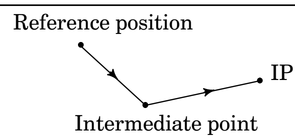
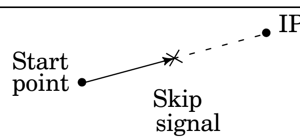
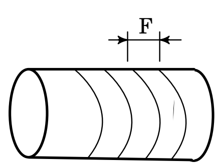
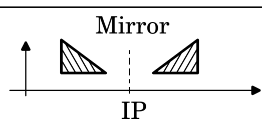
Xp : X axis or its parallel axis

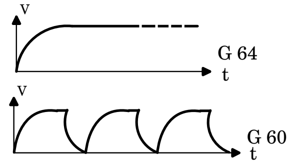
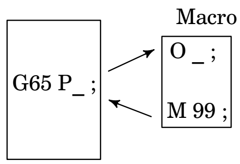
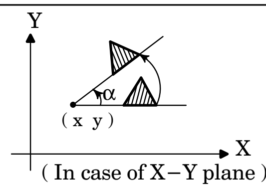
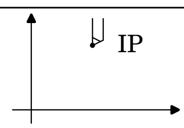
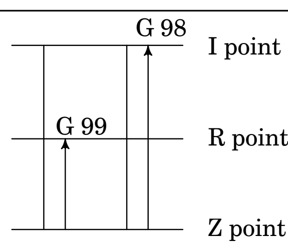
Yp : Y axis or its parallel axis

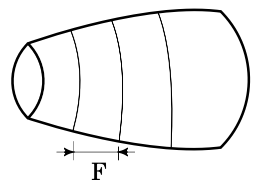
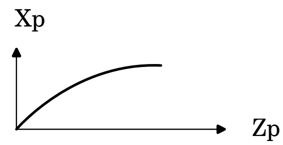
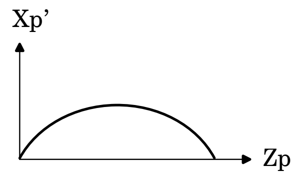
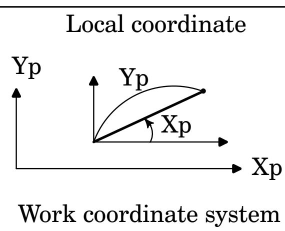
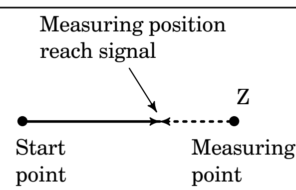
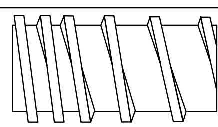
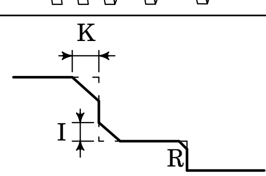
Zp : Z axis or its parallel axis

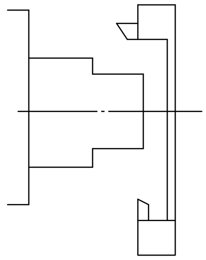
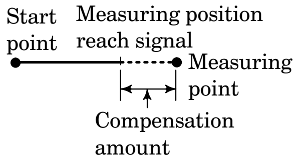
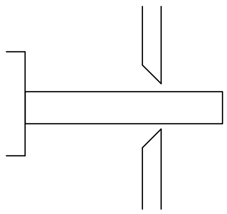
Functions	Illustrations	Tape format
Positioning (G00)		G00 IP___ ;
Linear interpolation (G01)		G01 IP___ F ___ ;
Circular interpolation (G02, G03)	(In case of X–Y plane)	G17 $\begin{Bmatrix} G02 \\ G03 \end{Bmatrix}$ Xp___ Yp___ $\begin{Bmatrix} R\_\_\_ \\ I\_\_\_ \; J\_\_\_ \end{Bmatrix}$ F___ ; G18 $\begin{Bmatrix} G02 \\ G03 \end{Bmatrix}$ Xp___ Zp___ $\begin{Bmatrix} R\_\_\_ \\ I\_\_\_ \; K\_\_\_ \end{Bmatrix}$ F___ ; G19 $\begin{Bmatrix} G02 \\ G03 \end{Bmatrix}$ Yp___ Zp___ $\begin{Bmatrix} R\_\_\_ \\ J\_\_\_ \; K\_\_\_ \end{Bmatrix}$ F___ ;
Helical interpolation (G02, G03) (only 15–M)	(In case X–Y plane, G03)	G17 $\begin{Bmatrix} G02 \\ G03 \end{Bmatrix}$ Xp___ Yp___ $\begin{Bmatrix} R\_\_\_ \\ I\_\_\_ \; J\_\_\_ \end{Bmatrix}$ α___ F___ ; G18 $\begin{Bmatrix} G02 \\ G03 \end{Bmatrix}$ Xp___ Zp___ $\begin{Bmatrix} R\_\_\_ \\ I\_\_\_ \; K\_\_\_ \end{Bmatrix}$ α___ F___ ; G19 $\begin{Bmatrix} G02 \\ G03 \end{Bmatrix}$ Yp___ Zp___ $\begin{Bmatrix} R\_\_\_ \\ J\_\_\_ \; K\_\_\_ \end{Bmatrix}$ α___ F___ ; α: Any axis other than circular interpolation axes.
Dwell (G04)		Per second dwell G04 $\begin{Bmatrix} X\_\_\_ \\ P\_\_\_ \end{Bmatrix}$; Per revolution dwell G95 G04 $\begin{Bmatrix} X\_\_\_ \\ P\_\_\_ \end{Bmatrix}$;

Functions	Illustrations	Tape format
Exact stop (G09)		G09 {G01 G02 G03} _____ ;
Change of offset value by program (G10) (15–T, 15–TT)		Geometry offset amount G10 L10 P___ X___ Z___ Y___ R___ Q___ ; Wear offset amount G10 L11 P___ X___ Z___ Y___ R___ Q___ ; Wear zero point offset amount G10 L2 P___ IP___ ;
Change of offset value by program (G10) (15–M)		Geometry offset amount G10 L10 P___ R___ ; Wear offset amount G10 L11 P___ R___ ; Wear zero point offset amount G10 L2 P___ ___ ;
Tool nose radius compensation (G10, G41, G42) (15–T, 15–TT)		{G17 G18 G19} {G40 G41 G42} IP__ ;
Cutter compensation (G40–G42) (15–M)		{G17 G18 G19} {G40 G41 G42} IP__ D___ ; D: Tool offset No.
Tool offset (15–T, 15–TT)		IP___ T___ ;
Tool length compensation (G43, G44, G49) (15–M)		{G43 G44} α__ H __ ; {G43 G44} H __ ; H:Tool offset No. G49; … Cancel
Tool offset (G45–G48) (15–M only)		{G45 G46 G47 G48} IP__ D __ ; H: Tool offset number
Scaling (G50, G51) (15–M only)		G51 IP___ P___ ; P : Scaling magnification G50; … Cancel
Setting of local coordinate system (G52)		G52 IP___ ;

Functions	Illustrations	Tape format
Command in machine coordinate system (G53)		G53 IP___ ;
Inch/metric conversion (G20, G21)		Inch input G20; Metric input G21;
Selection of work coordinate system (G54 – G59)		{G54 : G59} IP___ ;
Single direction positioning (G60) (15–M only)		G60 IP___ ;
Stored stroke check (G22, G23)		G22 X___ Y___ Z___ I___ J___ K___ : G23; … Cancel
Reference position return check (G27)		G27 IP__ ;
Reference position return (G28) 2nd, 3rd, 4th reference point return (G30)		G28 IP___ ; G30 P {2 3 4} IP___ ;
Reference from reference position (G29)		G29 IP__ :
Skip function (G31) Multiple skip function (G31.1 – G31.3)		{G30 G31.1 G31.2 G31.3} IP__ F __ ;
Thread cutting (G33)		Even lead thread cutting G33 IP___ F___ Q___ : Q : Thread cutting start point shift angle Inch thread cutting G33 IP___ E___ Q___ ; E : Threads per inch
Programmable mirror image (G50.1, G51.1)		G51.1 IP___ ; G50.1; … Cancel

Functions	Illustrations	Tape format
Cutting mode/Exact stop mode, Tapping mode, Automatic corner override mode		G64 ___ ; Cutting mode G60 ___ ; Exact stop mode
Custom macro (G65, G66, G66.1, G67)		One–shot call G65 P___ <Argument assignment>; P : Program No. Modal call {G66 / G66.1} P___ <Argument assignment>; G67; … Cancel
Coordinate system rotation (G68, G69) (15–M only)		G68 {G17 Xp __ Yp __ / G18 Zp __ Xp __ / G19 Yp __ Zp __} R α ; G69 ; … Cancel
Canned cycles for drilling (G83.1; G84–1, G86.1, G80 – G89) (15–T, 15–TT)	See Operator's manual "Canned cycle for drilling".	G80; … Cancel {G83.1 / G84.1 / G86.1 / G81 / : / G89} x__ y__ z__ P__ Q__ R__ F__ L__ ;
Canned cycles for drilling (G73, G74, G76, G80 – G89) (15–M)	See "Canned cycle".	G80; … Cancel {G73 / G74 / G76 / G81 / : / G89} x__ y__ z__ P__ Q__ R__ F__ L__ ;
Absolute/incremental programming (G90/G91)		G90 ______ ; Absolute G91 ______ ; Incremental G90 ___ G91___ ; Combined use
Change of work coordinate (G92)		G92 IP___ ;
Inverse time/Per–minute feed/Per–revolution feed (G93, G94, G95)	1/min mm/min inch/min mm/rev inch/rev	G93___ F___ ; Inverse time G94___ F___ ; Feed per minute G95___ F___ ; Feed per revolution
Initial point return/R Point R point return (G98, G99)		G98 ____________ ; G99 ____________ ;

Functions	Illustrations	Tape format
Constant surface speed control	m/min or feet/min	G96 S __ ; G97; … Cancel
Circular threading (G35, G36) (15–T, 15–TT only)		G17 {G35 / G36} Xp__ Yp__ I__ J__ F__ ; G18 {G35 / G36} Xp__ Zp__ I__ K__ F__ ; G19 {G35 / G36} Yp__ Zp__ J__ K__ F__ ;
Hypothetical axis interpolation (G07) (15–T, 15–TT)		G07 0; G17 Xp___ Yp___ G02 G18 Zp___ Xp___ ; G03 G19 Yp___ Zp___ G07α1; α; Hypothetical axis
Hypothetical axis interpolation (G07) (15–M)		G07 α0; {G17 / G18 / G19} {G02 / G02} Xp___ Yp___ Zp___ ; G07 α1; α:Hypothetical axis
Canned cycle for turning (G71 – G79) (15–T, 15–TT only)	Refer to item of "canned cycle for turning".	N___ G70___ Q___ ; {G71 / G72} P__ Q__ U__ W__ D__ F__ S__ ; G73 P__ Q__ I__ K__ U__ W__ D__ F__ S__ T__ ; {G74 / G75} X__ Z__ I__ K __ F__ D__ ; G76 X__ Z__ I__ K__ D__ F__ A__ P__ ; {G77 / G78} X__ Z__ I__ F__ ; G79 X__ Z__ K__ F__ ;
Polar coordinate (G15, G16) (15–M only)		G17 G16 Xp___ Yp____ ; G18 G16 Zp___ Xp____ ; G19 G16 Yp___ Zp____ ; G15; … Cancel
Tool length measurement (G37) (15–M only)		G37 Z____ ;
Variable lead thread cutting (G34) (15–T, 15–TT only)		G34 IP___ F___ K___ ;
Chamfering corner R (15–T, 15–TT only)		X___ {K___ / R___} ; Z___ {I___ / R___} ;

Functions	Illustrations	Tape format
Mirror image for double turret (G68, G69) (15–TT only)		G68 : Mirror image for double turret ON G69 : Mirror image cancel
Automatic tool compensation (15–T, 15–TT only)		$\left\{\begin{array}{l}\text{G37.1}\\\text{G37.2}\\\text{G37.3}\end{array}\right\}\ \alpha\_\_\ ;$
Balance cut (G68, G69) (15–TT only)		G68 : Balance cut mode G69 : Balance cut mode cancel

APPENDIX B. RANGE OF COMMAND VALUE

Table 1 (a) Linear axis
(in case of metric thread for feed screw and metric input)

	Increment system				
	IS–A	IS–B	IS–C	IS–D	IS–E
Least input increment	0.01 mm	0.01 mm or *1 0.001 mm	0.001 mm or *1 0.0001 mm	0.0001 mm or 0.00001 mm	0.00001 mm or 0.000001 mm
Least command increment	0.01 mm	0.001 mm	0.0001 mm	0.00001 mm	0.000001 mm
Interpolation unit	0.005 mm	0.0005 mm	0.00005 mm	0.000005 mm	0.0000005 mm
Max. programmable dimension *2	±9999999.99 mm	±999999.999 mm	±99999.9999 mm	±9999.99999 mm	±999.999999 mm
Max. rapid traverse *4	2400000 mm/min	24000 mm/min	100000 mm/min	10000 mm/min	1000 mm/min
Feedrate range *4	0.0001 – 2400000 mm/min	0.0001 – 240000 mm/min	0.0001 – 100000 mm/min	0.00001 – 10000 mm/min	0.000001 – 1000 mm/min
Incremental feed *5	0.01, 0.1, 1, 10, 100, 1000 mm/step	0.001, 0.01, 0.1, 1, 10, 100 mm/step	0.0001, 0.001, 0.01, 0.1, 1, 10 mm/step	0.00001, 0.0001, 0.001, 0.01, 0.1, 1.0 mm/step	0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1 mm/step
Tool compensation	0 – ±999.99 mm	0 – ±999.999 mm	0 – ±999.9999 mm	0 – ±9999.99999 mm	0 – ±999.999999 mm
Backlash compensation *6	0 – ±9999 pulses	0 – ±9999 pulses	0 – ±9999 pulses	0 – ±9999 pulses	0 – ±9999 pulses
Dwell time *3	0 – 99999.999 sec	0 – 99999.999 sec	0 – 9999.9999 sec	0 – 9999.99999sec	0 – 999.999999sec

Table 1 (b) Linear axis
(in case of metric threads for feed screw and inch input)

		Increment system				
		IS–A	IS–B	IS–C	IS–D	IS–E
Least input increment		0.001 inch	0.001 inch or *1 0.0001 inch	0.0001 inch or *1 0.00001 inch	0.00001 inch or 0.000001 inch	0.000001 inch or 0.0000001 inch
Least command increment		0.01 mm	0.001 mm	0.0001 mm	0.00001 mm	0.000001 mm
Interpolation unit		0.0005 inches	0.00005 inches	0.000005 inches	0.0000005 inches	0.00000005 inches
Max. programmable dimension	*2	±393700.787 inches	±39370.0787 inches	±3937.00787 inches	±393.700787 inches	±39.3700787 inches
Max. rapid traverse	*4	2400000 mm/min	240000 mm/min	100000 mm/min	10000 mm/min	1000 mm/min
Feedrate range	*4	0.00001 – 96000 inch/min	0.00001 – 9600 inch/min	0.00001 – 4000 inch/min	0.00001 – 400 inch/min	0.000001 – 40 inch/min
Incremental feed	*5	0.001, 0.01, 0.1, 1, 10, 100 inch/step	0.0001, 0.001, 0.01, 0.1, 1, 10 inch/step	0.000001, 0.0001, 0.001, 0.01, 0.1, 1 inch/step	0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1 inch/min	0.0000001, 0.000001, 0.00001, 0.0001, 0.001, 0.01 inch/min
Tool compensation		0 – ±99.999 inches	0 – ±99.9999 inches	0 – ±99.99999 inches	0 – ±999.999999 inches	0 – ±99.9999999 inches
Backlash compensation	*6	0 – ±9999 pulses	0 – ±9999 pulses	0 – ±9999 pulses	0 – ±9999 pulses	0 – ±9999 pulses
Dwell time	*3	0 – 99999.999 sec	0 – 99,999.999 sec	0 – 999.99999 sec	0 – 9999.99999 sec	0 – 999.999999 sec

Table 1 (c) Linear axis
(in case of inch threads for feed screw and inch input)

	Increment system				
	IS–A	IS–B	IS–C	IS–D	IS–E
Least input increment	0.001 inch	0.001 inch or *1 0.0001 inch	0.0001 inch or *1 0.00001 inch	0.00001 inch or 0.000001 inch	0.000001 inch or 0.0000001 inch
Least command increment	0.001 mm	0.0001 mm	0.00001 mm	0.000001 inch	0.0000001 inch
Interpolation unit	0.0005 inches	0.00005 inches	0.000005 inches	0.0000005 inches	0.00000005 inches
Max. programmable dimension *2	±999999.999 inches	±99999.9999 inches	±9999.99999 inches	±999.999999 inches	±99.9999999 inches
Max. rapid traverse *4	240000 inch/min	24000 inch/min	10000 inch/min	1000 inch/min	100 inch/min
Feedrate range *4	0.00001 – 240000 inch/min	0.00001 – 24000 inch/min	0.00001 – 10000 inch/min	0.000001 – 1000 inch/min	0.0000001 – 100 inch/min
Incremental feed *5	0.001, 0.01, 0.1, 1, 10, 100 inch/step	0.0001, 0.001, 0.01, 0.1, 1, 10 inch/step	0.00001, 0.0001, 0.001, 0.01, 0.1, 1 inch/step	0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1 inch/min	0.0000001, 0.000001,0.00001, 0.0001, 0.001, 0.01 inch/min
Tool compensation *6	0 – ±99.999 inches	0 – ±99.9999 inches	0 – ±99.99999 inches	0 – 999.999999 inches	0 – ±99.9999999 inches
Backlash compensation	0 – ±9999 pulses	0 – ±9999 pulses	0 – ±9999 pulses	0 – 9999 pulses	0 – ±9999 pulses
Dwell time *3	0 – 99999.999 sec	0 – 9999.9999 sec	0 – 999.99999 sec	0 – 9999.99999 sec	0 – 999.999999 sec

Table 1 (d) Linear axis
(in case of inch thread for feed screw and metric input)

	Increment system				
	IS–A	IS–B	IS–C	IS–D	IS–E
Least input increment	0.01 mm	0.01 mm or *1 0.001 mm	0.001 mm or *1 0.0001 mm	0.0001 mm or 0.00001 mm	0.00001 mm or 0.000001 mm
Least command increment	0.001 inch	0.0001 inch	0.00001 inch	0.000001 inch	0.0000001 mm
Interpolation unit	0.005 mm	0.0005 mm	0.00005 mm	0.000005 mm	0.0000005 mm
Max. programmable dimension *2	±9999999.99 mm	±999999.999 mm	±99999.9999 mm	±9999.99999 mm	±999.999999 mm
Max. rapid traverse *4	240000 inch/min	24000 inch/min	10000 inch/min	1000 inch/min	100 inch/min
Feedrate range *4	0.0001 – 2400000 mm/min	0.0001 – 240000 mm/min	0.0001 – 100000 mm/min	0.000001 – 10000 mm/min	0.0000001 – 1000 mm/min
Incremental feed *5	0.01, 0.1, 1, 10, 100, 1000 mm/step	0.001, 0.01, 0.1, 1, 10, 100 mm/step	0.0001, 0.001, 0.01, 0.1, 1, 10 mm/step	0.00001, 0.0001, 0.001, 0.01, 0.1, 1.0 mm/step	0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1 mm/step
Tool compensation	0 – ±999.99 mm	0 – ±999.999 mm	0–±999.9999 mm	0 – ±9999.99999 mm	0 – ±999.999999 inches
Backlash compensation *6	0 – ±9999 pulses	0 – ±9999 pulses	0 – ±9999 pulses	0 – 9999 pulses	0 – ±9999 pulses
Dwell time *3	0 – 999999.99 sec	0 – 99999.999 sec	0 – 9999.9999 sec	0 – 9999.99999 sec	0 – 999.999999 sec

Table 1 (e) Rotary axis

	Increment system				
	IS–A	IS–B	IS–C	IS–D	IS–E
Least input increment	0.01 deg	0.01 deg or *1 0.001 deg	0.001 deg or *1 0.0001 deg	0.0001 deg or 0.00001 deg	0.00001 deg or 0.000001 deg
Least command increment	0.01 deg	0.001 deg	0.0001 deg	0.00001 deg	0.000001 deg
Interpolation unit	0.005 deg	0.0005 deg	0.00005 deg	0.000005 deg	0.0000005 deg
Max. programmable dimension *2	±9999999.99 deg	±999999.999 deg	±99999.9999 deg	±9999.99999 deg	±999.999999 deg
Max. rapid traverse *4	2400000 deg/min	240000 deg/min	100000 deg/min	10000 deg/min	1000 deg/min
Feedrate range *4	0.0001 – 2400000 deg/min	0.0001 – 240000 deg/min	0.0001 – 100000 deg/min	0.0001 – 10000 deg/min	0.0001 – 1000 deg/min
Incremental feed *5	0.01, 0.1, 1, 10, 100, 1000 deg/step	0.001, 0.01, 0.1, 1, 10, 100 deg/step	0.0001, 0.001, 0.01, 0.1, 1, 10 deg/step	0.00001, 0.0001, 0.001, 0.01, 0.1, 1.0 deg/step	0.000001, 0.00001, 0.0001, 0.001, 0.01, 0.1 deg/step
Tool compensation	0 – ±999.99 deg	0 – ±999.999 deg	0 – ±999.9999 deg	0 – ±9999.99999 deg	0 – ±999.999999 deg
Backlash compensation *6	0 – ±9999 pulses	0 – ±9999 pulses	0 – ±9999 pulses	0 – 9999 pulses	0 – ±9999 pulses
Dwell time *3	0 – 999999.99sec	0 – 99999.999sec	0 – 9999.9999sec	0 – 9999.99999sec	0 – 999.999999sec

* 1 Selected by parameters for each axis.

Certain functions are not applicable for axes with different increment systems (e.g. circular interpolation, tool nose radius compensation, etc.)

* 2 When given commands for axes of different increment system in the same block, limitations are set by the smallest value.

* 3 Will depend on the unit system of the axis on address X.

* 4 The feed rate range shown above are limitations depending on CNC interpolation capacity. When regarded as a whole system, limitations, depending on the servo system, must also be considered.

* 5 In case of BMI interface, incremental feed amount can be specified by setting amount (parameter setting.)

* 6 The unit of backlash compensation is detection unit.

* 7 For a system using more than one increment system, the smallest of the least input increments for that system is used.

APPENDIX C. NOMOGRAPHS

1. Incorrect Threaded Length

The leads of a thread are generally incorrect in $\delta 1$, and $\delta 2$, as shown in Fig. 1, due to automatic acceleration and deceleration.

Therefore, distance allowances must be made to the extent of $\delta 1$, and $\delta 2$, in a program.

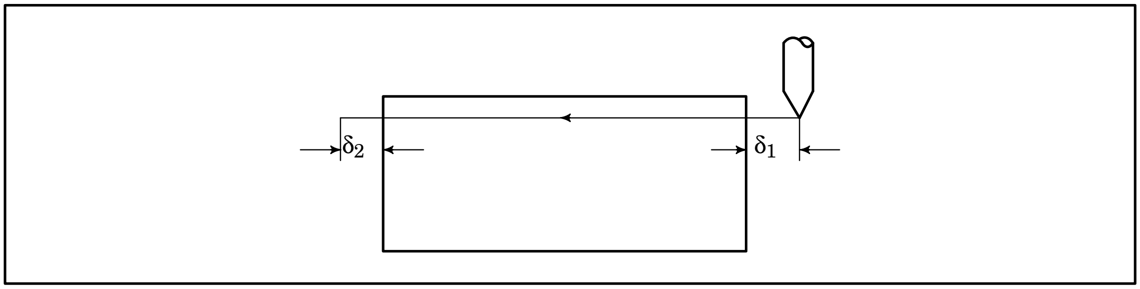


Fig. 1 Incorrect threaded length

(1) $\delta 2$ is determined by cutting speed V (mm/sec) and time constant (T1) of the servo system as shown below.

$\delta 2 = T1 \cdot V(mm)$ .... (1)

where T1 is in sec., and V is in mm/sec.

V is determined by thread lead L and spindle speed R as shown below.

V = R(rpm).L(mm)/60

L(mm) .... Lead of thread

R(rpm) .... Spindle–speed

Time constant T1 (sec.) of the servo system: Usually 0.033 sec.

(2) How $\delta 1$ is determined

The value of $\delta 1$ is determined by the cutting speed V in thread cutting, time constant T1 for the servo system, and thread accuracy “a” as shown below.

$\delta 1 = \{t - T1 + T1 \exp(-\frac{t}{T1})\} V$ (pules) .... (2)

$a = \exp(-\frac{t}{T1})$ .... (3)

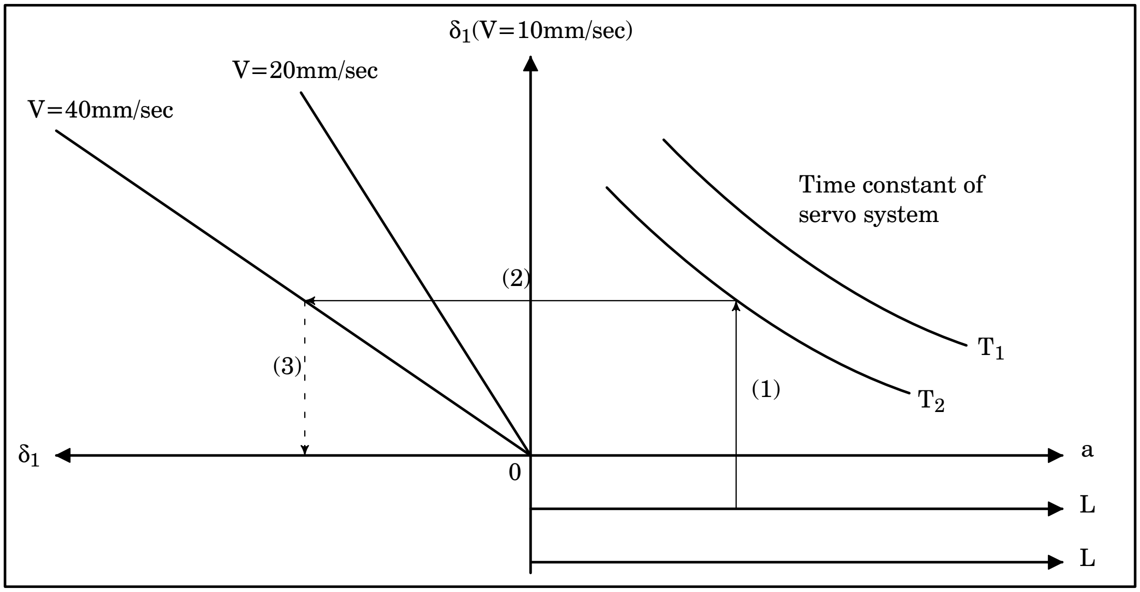
The lead at the beginning of thread cutting is shorter than the specified lead, L, and the following lead error is ΔL.

Then,

$a = \frac{\Delta L}{L}$

When the value of “a” is determined, the time lapse, t, can be calculated by formula (3) until the thread accuracy is attained. The time “t” is substituted in (2) to determine $\delta 1$. Constants V, T1 and T2 are determined in the same way as they are for $\delta 2$.

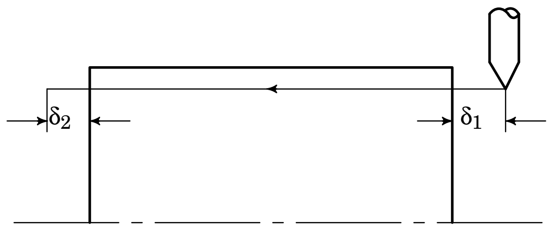
Since the calculation of $\delta 1$ is rather complex, a nomography is provided on the following pages. Instructions on how to read the appropriate $\delta 1$ value on the nomography is shown below.



First specify the class and the lead of a thread. The thread accuracy, a, will be obtained at (1), and depending on the time constant of the cutting feed acceleration/deceleration, the δ1 value when V = 10 mm/sec will be obtained at (2). Then, depending on the speed of thread cutting, the approach distance δ1 will be obtained at (3) for other speeds than 10 mm/sec.

NOTE The equation of δ1,δ2 is used when the acceleration/deceleration time constant for cutting feed is 0.

2. Simple Calculation of Incorrect Thread Length



L: Thread lead (mm)
R: Spindle speed (rpm)

(1) $\delta_2 = \dfrac{L \times R}{1800*}$ (mm)

(2) $\delta_1 = \dfrac{L \times R}{1800*}$ (–1–1na) $= \delta_2$ (–1–1na)

*When time constant T of the servo system is 0.033 sec.

An “a” indicates the error allowances. And the “–1 –1na” follows as below.

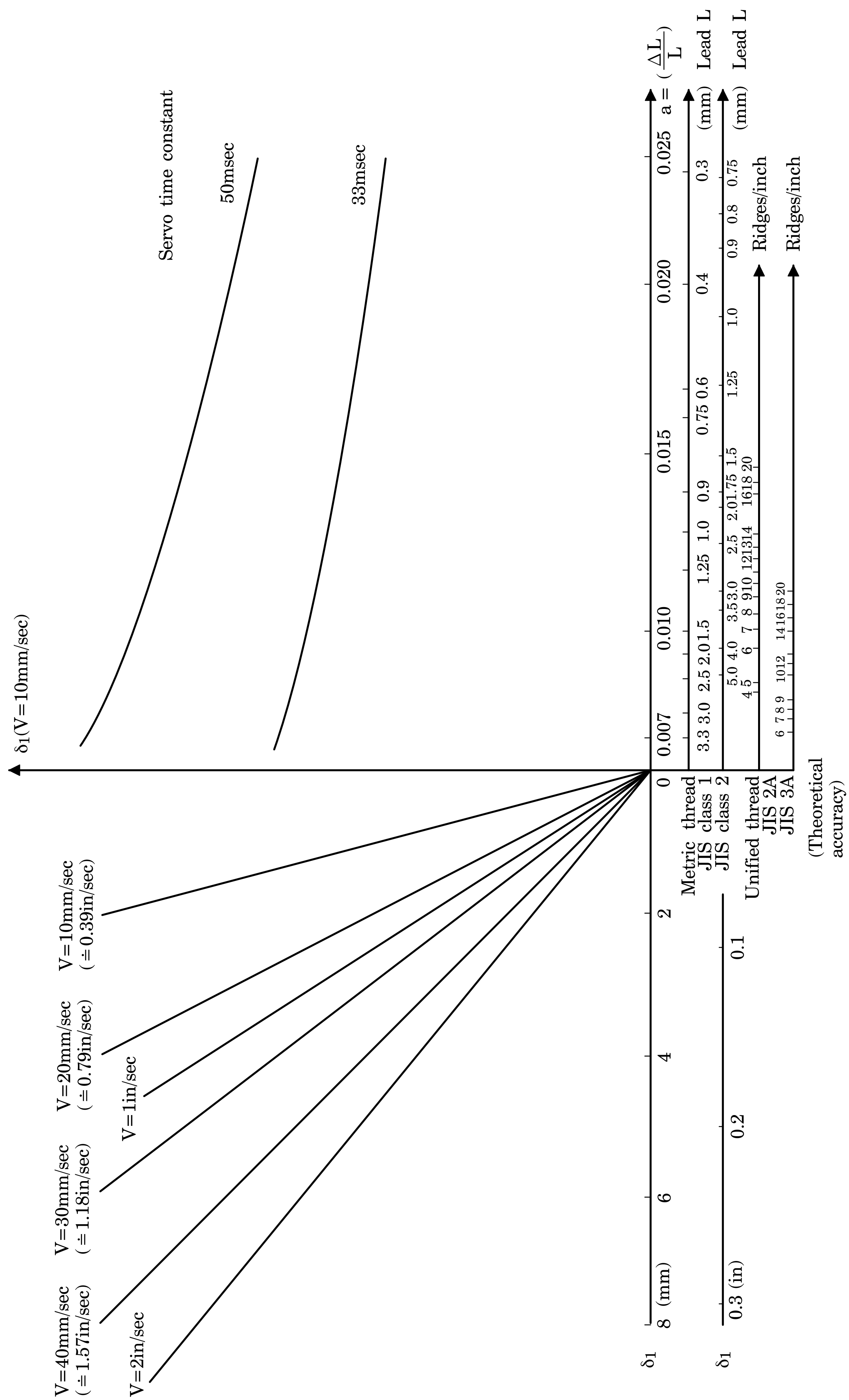
a	–1 –1na
0.005	4.298
0.01	3.605
0.015	3.200
0.02	2.912

(Example)

R = 350 rpm
L = 1 mm
A = 0.01

$\delta 2 = \dfrac{350 \times 1}{1800} = 0.194$ mm

$\delta 1 = 2 \times 3.605 = 0.701$ mm


Servo time constant
50msec
33msec
δ1(V=10mm/sec)
V=10mm/sec (≑0.39in/sec)
V=20mm/sec (≑0.79in/sec)
V=1in/sec
V=30mm/sec (≑1.18in/sec)
V=40mm/sec (≑1.57in/sec)
V=2in/sec
δ1 0 2 4 6 8 (mm)
δ1 0.1 0.2 0.3 (in)
a = (ΔL/L)
0.007 0.010 0.015 0.020 0.025
Lead L (mm)
Metric thread JIS class 1
JIS class 2
Unified thread JIS 2A
JIS 3A
Ridges/inch
Ridges/inch
(Theoretical accuracy)

3. Tool Path at Corner

(1) Description

When servo system delay (by exponential acceleration/deceleration at cutting or caused by the motor when a servo motor is used) is accompanied by cornering, a slight deviation is produced between the tool path (tool center path) and the programmed path as shown in Figure 2 (a).

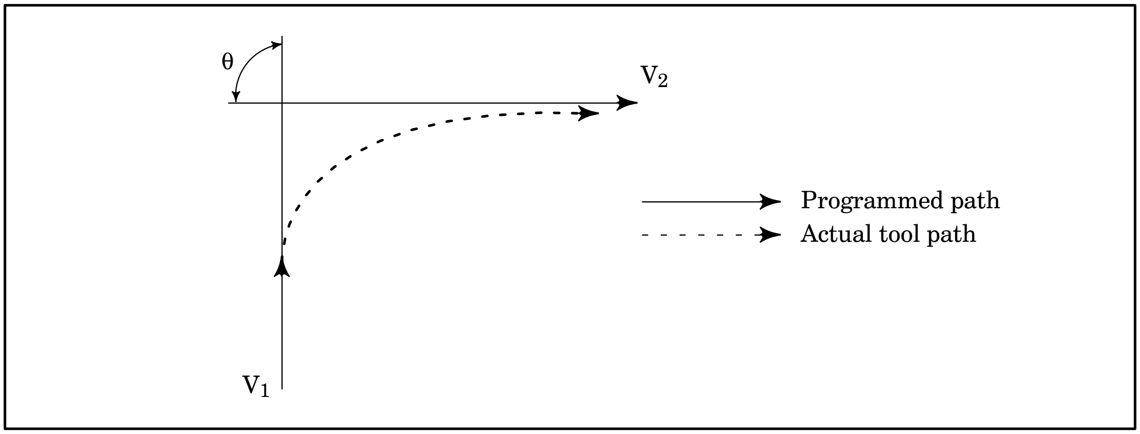


Fig. 2 (a) Tool path at cornering

This tool path is determined by the following parameters:

(a) Feedrate (V1, V2)

(b) Corner angle (θ)

(c) Exponential acceleration/deceleration time constant (T1) at cutting.

(d) Loop gain of positioning system

(e) Buffer

The above parameters are used to theoretically analyze the tool path.

When programming, the above items must be considered and programming must be performed carefully so the shape of the workpiece is within the desired precision. If the theoretical precision is out of the desired one, a dwell command shall be specified so that the next block will not be read until the commanded feedrate becomes zero.

(2) Analysis

The tool path shown in Fig. 2 (b) is analyzed based on the following conditions:

(a) Feedrate is constant at the block before and after cornering.

(b) The controller has a buffer register.

(The error differs with the reading speed of the tape reader, number of characters of the next block, etc.)

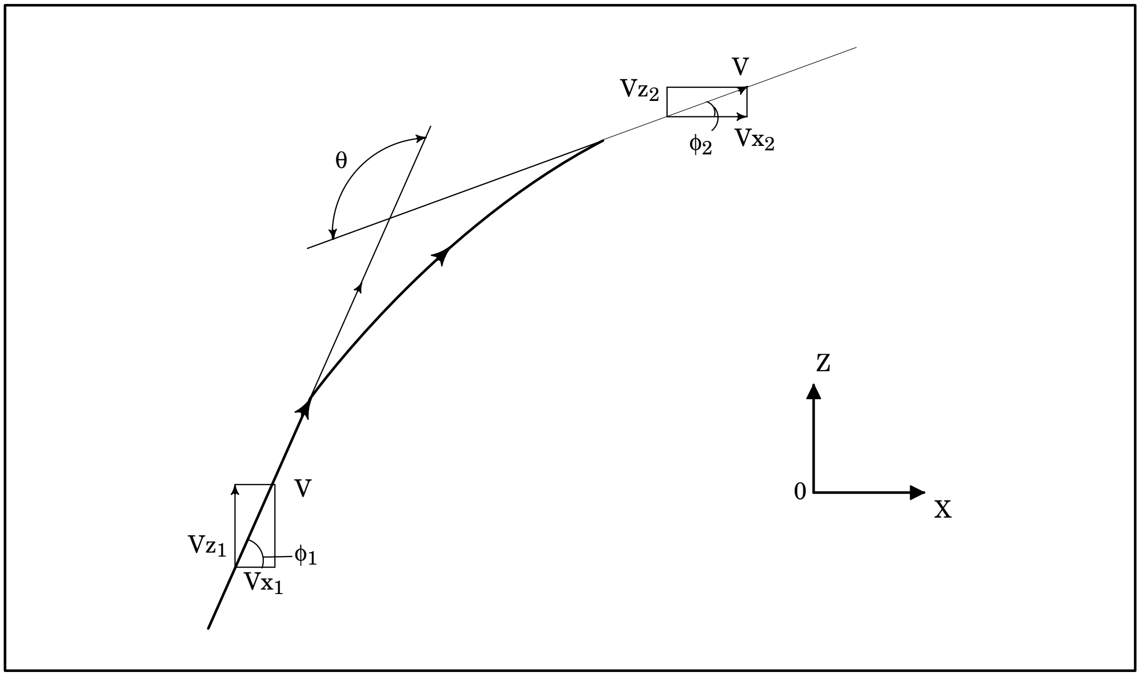


Fig. 2 (b) Command

(Conditions expressions)

$Vx_1 = Vx \cos\phi_1$

$Vz_1 = Vx \sin \phi_1$

$Vx_2 = Vx \cos\phi_2$

$Vz_2 = Vx \sin \phi_2$

$\pi - (\phi_1 - \phi_2) = \theta$

(Description of symbol)

V : Feedrate at the block before and after cornering

Vx_1 : X–axis component of feedrate of preceding block

Vz_1 : Z–axis component of feedrate of preceding block

Vx_2 : X–axis component of feedrate of following block

Vz_2 : Z–axis component of feedrate of following block

θ : Corner angle

ϕ_1 : Angle formed by commanded path direction of preceding block and X–axis

ϕ_2 : Angle formed by commanded path direction of following block and X–axis

Initial value calculation

The initial value at which cornering begins, that is, the X and Z coordinates at the end of command of the controller is determined by the feedrate and the time constant of the servo motor.

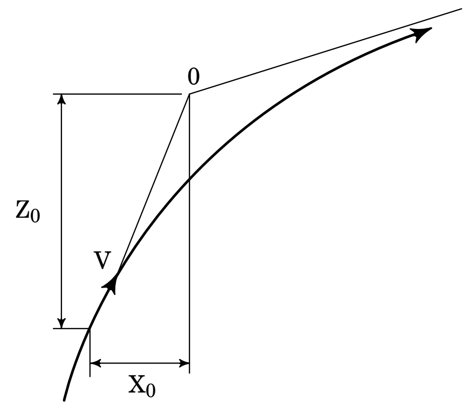


Fig. 2 (c) Initial value

$X_0 = Vx_1\ (T1 + T2)$ ······················ (1)

$Z_0 = Vz_1\ (T1 + T2)$ ······················ (2)

T1: Exponential acceleration/deceleration time constant.

T2: Time constant of positioning system (Inverse of position Loop gain)

Tool path analysis

The equations below represent the feedrates for the corner section in X–axis direction and Z–axis direction.

$$Vx(t) = (Vx_2 - Vx_1)\left(1 - \frac{1}{T1 + T2}\left(T1 \cdot \exp.\left(-\frac{t}{T1}\right) - T2 \cdot \exp\left(-\frac{t}{T2}\right)\right)\right) + Vx_1$$

$$= \frac{Vx_1 - Vx_2}{T_1 - T_2}\left\{T_1 \cdot \exp\left(-\frac{t}{T_1}\right) - T_2 \exp\left(-\frac{t}{T_2}\right)\right\} + Vx_2 \cdots\cdots \quad (3)$$

$$Vz(t) = \frac{Vz_1 - Vz_2}{T1 - T2}\left(T1 \cdot \exp\left(-\frac{t}{T1}\right) - T2 \cdot \exp\left(-\frac{t}{T2}\right)\right) + Vz_2 \cdots\cdots \quad (4)$$

Therefore, the coordinates of the tool path at time t are calculated from the following equations:

$$X(t) = \int_0^t Vx(t)dt - X_0$$

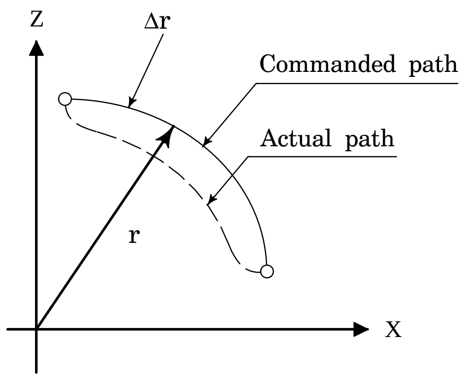
$$= \frac{Vx_2 - Vx_1}{T1 - T2}\left(T_1^{\,2} \cdot \exp\left(-\frac{t}{T1}\right) - T_2^{\,2} \cdot \exp\left(-\frac{t}{T2}\right)\right) - Vx_2\ (T1 + T2 - t) \quad \cdots\ (5)$$

$$Z(t) = \int_0^t Vz(t)dt - Z_0$$

$$= \frac{Vz_2 - Vz_1}{T1 - T2}\left(T_1^{\,2} \exp\left(-\frac{t}{T1}\right) - T_2^{\,2} \cdot \exp\left(-\frac{t}{T2}\right)\right) - Vz_2\ (T1 + T2 - t) \quad \cdots\ (6)$$

4. Radius Direction Error at Circular Cutting

When a servo motor is used, the positioning system causes a delay between the input axis and output axis. Since the tool advances along the commanded segment, an error is not produced in linear interpolation, but when especially high speed cutting is performed in circular interpolation, an error is produced in the radius direction. This error can be obtained as follows:



Δr : Maximum radius error (mm)
v : Feedrate (mm/sec)
r : Circle radius (mm)
T1 : Exponential acceleration/deceleration time constant (sec) at cutting
T2 : Positioning system time constant (sec)

$$\Delta r = \frac{1}{2}\left(T_1^{\ 2} + T_2^{\ 2} \right) \times \frac{V_2}{r} \quad \cdots\cdots \ (1)$$

Since the machining radius r (mm) and allowable error Δr (mm) of the workpiece is given in actual machining, the allowable limit feedrate v (mm/sec) is determined by equation (1).

Since the acceleration/deceleration time constant at cutting which is set by this equipment varies with the machine tool, refer to the machine tool builder’s instruction manual.

APPENDIX D. CODE USED IN PROGRAM

ISO code										EIA code										Meaning	
Character	8	7	6	5	4		3	2	1	Character	8	7	6	5	4		3	2	1		
0			○	○						0			○			●					Numeral 0
1	○		○	○		●			○	1						●			○		Numeral 1
2	○		○	○		●		○		2						●		○			Numeral 2
3			○	○		●		○	○	3				○		●		○	○		Numeral 3
4	○		○	○		●	○			4						●	○				Numeral 4
5			○	○			○		○	5				○		●	○		○		Numeral 5
6			○	○		●	○	○		6				○		●	○	○			Numeral 6
7	○		○	○		●	○	○	○	7						●	○	○	○		Numeral 7
8	○		○	○	○	●				8					○	●					Numeral 8
9			○	○	○	●			○	9				○	○	●			○		Numeral 9
A		○				●			○	a		○	○			●			○		Address A
B		○				●		○		b		○	○			●		○			Address B
C	○	○				●		○	○	c		○	○	○		●		○	○		Address C
D		○				●	○			d		○	○			●	○				Address D
E	○	○				●	○		○	e		○	○	○		●	○		○		Address E
F	○	○				●	○	○		f		○	○	○		●	○	○			Address F
G		○				●	○	○	○	g			○	○		●	○	○	○		Address G
H		○			○	●				h		○	○		○	●					Address H
I	○	○			○	●			○	i		○	○	○	○	●			○		Address I
J	○	○			○	●		○		j		○		○		●			○		Address J
K		○			○	●		○	○	k		○		○		●		○			Address K
L	○	○			○	●	○			l		○				●		○	○		Address L
M		○			○	●	○		○	m		○		○		●	○				Address M
N		○			○	●	○	○		n		○				●	○		○		Address N
O	○	○			○	●	○	○	○	o		○				●	○	○			Address O
P		○		○		●				p		○		○		●	○	○	○		Address P
Q	○	○		○		●			○	q		○		○	○	●					Address Q
R	○	○		○		●		○		r		○			○	●			○		Address R
S		○		○		●		○	○	s			○	○		●		○			Address S
T	○	○		○		●	○			t			○			●		○	○		Address T
U		○		○		●	○		○	u			○	○		●	○				Address U
V		○		○		●	○	○		v			○			●	○		○		Address V
W	○	○		○		●	○	○	○	w			○			●	○	○			Address W
X	○	○		○	○	●				x			○	○		●	○	○	○		Address X
Y		○		○	○	●			○	y			○	○	○	●					Address Y
Z		○		○	○	●		○		z			○		○	●			○		Address Z
DEL	○	○	○	○	○	●	○	○	○	Del		○	○	○	○	●	○	○	○	×	Delete (cancel an error punch).
NUL						●				Blank						●				×	Not punched. Can not be used in significant section in EIA code.
BS	○				○	●				BS			○		○	●		○		*	Back space
HT					○	●			○	Tab			○	○	○	●	○	○		*	Tabulator
LF or NL					○	●		○		CR or EOB	○					●				?	End of block
CR	○				○	●	○		○											*	Carriage return
SP	○		○			●				SP				○		●				*	Space
%	○		○			●	○		○	ER					○	●		○	○		Absolute rewind stop
(			○		○	●				(2–4–5)				○	○	●		○			Control out(a comment is started)
)	○		○		○	●			○	(2–4–7)		○			○	●		○		?	Control in(the end of comment)
+			○		○	●		○	○	+		○	○	○		●				*	Positive sign
–			○		○	●	○		○	–		○				●					Negative sign
:			○	○	○	●		○													Colon
/	○		○		○	●	○	○	○	/			○	○		●			○		Optional block skip
.			○		○	●	○	○		.		○	○		○	●		○	○		Period (A decimal point)
#	○		○			●		○	○											?	Sharp
$			○			●	○													*	Dollar sign
&	○		○			●	○	○		&					○	●	○	○		?	Ampersand
▼			○			●	○	○	○											*	Apostrophe
*	○		○		○	●		○												?	Asterisk
,	○		○		○	●	○			,			○	○	○	●		○	○	?	Comma
;	○		○	○	○	●		○	○											*	Semicolon
<			○	○	○	●	○													*	Left angle bracket
=	○		○	○	○	●	○		○											?	Equal
>	○		○	○	○	●	○	○												*	Right angle bracket
?			○	○	○	●	○	○	○											?	Question mark
@	○	○				●														?	Commercial at mark
"			○			●		○												*	Quotation
[	○	○		○	○	●		○	○											?	Left brace
]	○	○		○	○	●	○		○											?	Right brace

NOTE 1 * : Codes with an asterisk that are entered in a comment area are read into memory. When entered in a significant data area, these codes are ignored.

NOTE 2 x : Codes with an x are ignored.

NOTE 3 ? : Codes with a question mark are ignored when entered in a significant data area that comes before the program number. These codes are read into memory when entered in any other significant data area or comment area. However, when the custom macro option is used, the following codes are read into memory even when entered in a significant data area that comes before the program number.

When ISO codes are used: #, &, *, =, [and]

When EIA codes are used: (codes set by parameter)

NOTE 4 Codes not shown in this table are ignored as long as their parity is correct.

NOTE 5 Codes with incorrect parity cause a TH alarm to be generated (except when found in a comment area, in which case they are ignored.)

NOTE 6 When EIA codes are used, the code with all eight holes punched is treated as a special case and does not generate a parity alarm.

APPENDIX E. TABLE OF KANJI AND HIRAGANA CODES

Internal code (hexadecimal 4 digits)

ア	ぁ	6421	あ	6422	阿	7024	哀	7025	愛	7026	挨	7027	逢	7029	悪	702D
	握	702E	旭	7030	圧	7035	斡	7036	扱	7037	宛	7038	安	7042	暗	7045
	案	7046	闇	7047	鞍	7048										
イ	ぃ	6423	い	6424	以	704A	伊	704B	位	704C	依	704D	偉	704E	囲	704F
	委	7051	威	7052	意	7055	慰	7056	易	7057	椅	7058	為	7059	異	705B
	移	705C	維	705D	緯	705E	胃	705F	萎	7060	衣	7061	違	7063	遺	7064
	医	7065	井	7066	域	7068	育	7069	一	706C	稲	7070	印	7075	員	7077
	因	7078	引	707A	飲	707B	院	7121	陰	7122	隠	7123				
ウ	ぅ	6425	う	6426	右	7126	宇	7127	羽	7129	雨	712B	渦	7132	嘘	7133
	唄	7134	浦	313A	瓜	713B	噂	713D	運	713F	雲	7140				
エ	ぇ	6427	え	6424	営	7144	影	7146	映	7147	栄	7149	永	714A	泳	714B
	洩	714C	英	7151	衛	7152	鋭	7154	液	7155	益	7157	駅	7158	越	715B
	閲	715C	円	715F	園	7160	宴	7163	延	7164	援	7167	沿	7168	演	7169
	炎	716A	煙	716C	縁	716F	遠	7173	鉛	7174	塩	7176				
オ	ぉ	6429	お	642A	汚	7178	凹	717A	央	717B	奥	717C	往	717D	応	717E
	押	7221	横	7223	欧	7224	王	7226	黄	722B	岡	722C	沖	722D	億	722F
	屋	7230	憶	7231	臆	7232	牡	7234	乙	7235	恩	7238	温	7239	穏	723A
	音	723B														

カ	か	6428	が	642C	下	723C	化	723D	仮	723E	何	723F	価	7241	佳	7242
	加	7243	可	7244	夏	7246	家	7248	科	724A	暇	724B	果	724C	架	724D
	歌	724E	河	724F	火	7250	稼	7254	箇	7255	花	7256	荷	7259	華	725A
	菓	725B	課	725D	貨	725F	過	7261	我	7266	牙	7267	画	7268	芽	726A
	賀	726C	雅	726D	介	7270	会	7271	解	7272	回	7273	壊	7275	廻	7276
	快	7277	怪	7278	懐	727B	拐	727D	改	727E	械	7323	海	7324	灰	7325
	界	7326	皆	7327	絵	7328	開	732B	階	732C	貝	732D	劾	732E	外	7330
	害	7332	慨	7334	概	7335	涯	7336	街	7339	該	733A	垣	7340	各	7346
	拡	7348	格	734A	核	734B	殻	734C	獲	734D	確	734E	穫	734F	覚	7350
	角	7351	較	7353	郭	7354	閣	7355	隔	7356	革	7357	学	7358	楽	735A
	額	735B	掛	735D	笠	735E	潟	7363	割	7364	括	7367	活	7368	渇	7369
	滑	736A	株	7374	刈	7422	乾	7425	冠	7427	寒	7428	刊	7429	勧	742B
	巻	742C	喚	742D	完	7430	官	7431	寛	7432	干	7433	幹	7434	患	7435
	感	7436	慣	7437	換	7439	敢	743A	歓	743F	汗	7440	漢	7441	環	7444
	甘	7445	監	7446	看	7447	管	7449	簡	744A	緩	744B	缶	744C	肝	744E
	観	7451	貫	7453	還	7454	鑑	7455	間	7456	閑	7457	関	7458	陥	7459
	韓	745A	館	745B	丸	745D	含	745E	岸	745F	眼	7463	岩	7464	顔	7469
	願	746A														
キ	き	642D	ぎ	642E	企	746B	危	746D	喜	746E	器	746F	基	7470	奇	7471
	寄	7473	岐	7474	希	7475	幾	7476	揮	7478	机	7479	旗	747A	既	747B
	期	747C	棄	747E	機	7521	帰	7522	毅	7523	気	7524	汽	7525	祈	7527
	季	7528	稀	7529	徽	752B	規	752C	記	752D	貴	752E	起	752F	軌	7530
	輝	7531	騎	7533	鬼	7534	偽	7536	戯	753A	技	753B	擬	753C	欺	753D
	犠	753E	疑	753F	義	7541	議	7544	菊	7546	喫	754A	詰	754D	却	7551
	客	7552	脚	7553	逆	7555	丘	7556	久	7557	休	7559	及	755A	吸	755B
	宮	755C	弓	755D	急	755E	救	755F	求	7561	泣	7563	球	7565	究	7566
	窮	7567	級	7569	糾	756A	給	756B	旧	756C	牛	756D	去	756E	居	756F
	巨	7570	拒	7571	拠	7572	挙	7573	虚	7575	許	7576	距	7577	漁	7579
	魚	757B	亨	757C	享	757D	京	757E	供	7621	競	7625	共	7626	協	7628
	叫	762B	境	762D	強	762F	恐	7632	挟	7634	教	7635	橋	7636	況	7637
	狂	7638	狭	7639	胸	763B	脅	763C	興	763D	郷	763F	鏡	7640	響	7641
	驚	7643	仰	7644	凝	7645	業	7648	局	7649	曲	764A	極	764B	玉	764C
	勤	7650	均	7651	巾	7652	錦	7653	琴	7657	禁	7658	筋	765A	緊	765B
	近	7661	金	7662	銀	7664										

ク	く	642F	ぐ	6430	九	7665	句	7667	区	7668	矩	766B	苦	766C	駆	766E
	具	7671	愚	7672	空	7675	偶	7676	遇	7678	隅	7679	屑	767D	屈	767E
	掘	7721	靴	7724	熊	7727	繰	772B	君	772F	訓	7731	群	7732	軍	7733
	郡	7734														
ケ	け	6431	げ	6432	係	7738	傾	7739	刑	773A	兄	773B	啓	773C	型	773F
	契	7740	形	7741	径	7742	慶	7744	憩	7746	掲	7747	携	7748	敬	7749
	景	774A	系	774F	経	7750	継	7751	茎	7754	計	7757	警	7759	軽	775A
	芸	775D	迎	775E	劇	7760	撃	7762	激	7763	隙	7764	桁	7765	傑	7766
	欠	7767	決	7768	潔	7769	穴	776A	結	776B	血	776C	月	776E	件	776F
	倹	7770	健	7772	兼	7773	券	7774	剣	7775	圏	7777	堅	7778	嫌	7779
	建	777A	憲	777B	懸	777C	拳	777D	検	7821	権	7822	犬	7824	献	7825
	研	7826	絹	7828	県	7829	肩	782A	見	782B	謙	782C	軒	782E	鍵	7830
	険	7831	験	7833	元	7835	原	7836	厳	7837	幻	7838	弦	7839	減	783A
	源	783B	現	783D	言	7840	限	7842								
コ	こ	6433	ご	6434	個	7844	古	7845	呼	7846	固	7847	己	784A	庫	784B
	弧	784C	戸	784D	故	784E	湖	7850	狐	7851	誇	7858	雇	785B	顧	785C
	五	785E	互	785F	午	7861	娯	7864	後	7865	御	7866	語	786C	誤	786D
	護	786E	交	7872	侯	7874	候	7875	光	7877	公	7878	功	7879	効	787A
	勾	787B	厚	787C	口	787D	向	787E	喉	7922	坑	7923	好	7925	孔	7926
	孝	7927	工	7929	巧	792A	幸	792C	広	792D	康	792F	弘	7930	抗	7933
	拘	7934	控	7935	攻	7936	更	7939	校	793B	構	793D	江	793E	洪	793F
	港	7941	溝	7942	甲	7943	硬	7945	稿	7946	紅	7948	絞	794A	綱	794B
	耕	794C	考	794D	肯	794E	航	7952	荒	7953	行	7954	衡	7955	講	7956
	貢	7957	購	7958	郊	7959	鉱	795B	鋼	795D	降	795F	項	7960	香	7961
	高	7962	剛	7964	号	7966	合	7967	克	796E	刻	796F	告	7970	国	7971
	穀	7972	酷	7973	黒	7975	腰	7978	骨	797C	込	797E	此	7A21	頃	7A22
	今	7A23	困	7A24	婚	7A27	根	7A2C	混	7A2E						
サ	さ	6435	ざ	6436	左	7A38	差	7A39	査	7A3A	砂	7A3D	鎖	7A3F	座	7A42
	挫	7A43	債	7A44	催	7A45	再	7A46	最	7A47	妻	7A4A	彩	7A4C	才	7A4D
	採	7A4E	栽	7A4F	済	7A51	災	7A52	砕	7A55	祭	7A57	細	7A59	菜	7A5A
	裁	7A5B	載	7A5C	際	7A5D	剤	7A5E	在	7A5F	材	7A60	罪	7A61	財	7A62
	坂	7A64	阪	7A65	咲	7A69	崎	7A6A	作	7A6E	削	7A6F	昨	7A72	柵	7A74
	策	7A76	索	7A77	錯	7A78	桜	7A79	冊	7A7D	刷	7A7E	察	7B21	拶	7B22
	撮	7B23	擦	7B24	札	7B25	殺	7B26	雑	7B28	皿	7B2E	三	7B30	傘	7B31
	参	7B32	山	7B33	撒	7B35	散	7B36	産	7B3A	算	7B3B	讃	7B3E	賛	7B3F
	酸	7B40	残	7B44												

シ	し	6437	じ	6438	仕	7B45	伺	7B47	使	7B48	刺	7B49	史	7B4B	四	7B4D
	士	7B4E	始	7B4F	姉	7B50	姿	7B51	子	7B52	市	7B54	師	7B55	志	7B56
	思	7B57	指	7B58	支	7B59	施	7B5C	旨	7B5D	枝	7B5E	止	7B5F	死	7B60
	私	7B64	糸	7B65	紙	7B66	紫	7B67	脂	7B69	至	7B6A	視	7B6B	詞	7B6C
	詩	7B6D	試	7B6E	誌	7B6F	資	7B71	歯	7B75	事	7B76	似	7B77	字	7B7A
	寺	7B7B	持	7B7D	時	7B7E	次	7C21	治	7C23	磁	7C27	示	7C28	耳	7C2A
	自	7C2B	辞	7C2D	式	7C30	識	7C31	軸	7C34	七	7C37	失	7C3A	室	7C3C
	湿	7C3E	質	7C41	実	7C42	芝	7C47	縞	7C4A	写	7C4C	射	7C4D	捨	7C4E
	斜	7C50	煮	7C51	社	7C52	者	7C54	謝	7C55	車	7C56	借	7C5A	尺	7C5C
	釈	7C61	若	7C63	弱	7C65	主	7C67	取	7C68	守	7C69	手	7C6A	殊	7C6C
	狩	7C6D	種	7C6F	趣	7C71	酒	7C72	首	7C73	受	7C75	寿	7C77	授	7C78
	樹	7C79	需	7C7B	収	7C7D	周	7C7E	就	7D22	修	7D24	秀	7D28	秋	7D29
	終	7D2A	習	7D2C	臭	7D2D	舟	7D2E	衆	7D30	襲	7D31	蹴	7D33	週	7D35
	集	7D38	住	7D3B	充	7D3C	十	7D3D	従	7D3E	柔	7D40	渋	7D42	縦	7D44
	重	7D45	宿	7D49	祝	7D4B	縮	7D4C	熟	7D4F	出	7D50	術	7D51	述	7D52
	春	7D55	瞬	7D56	準	7D60	盾	7D62	純	7D63	巡	7D64	順	7D67	処	7D68
	初	7D69	所	7D6A	暑	7D6B	緒	7D6F	署	7D70	書	7D71	諸	7D74	助	7D75
	叙	7D76	女	7D77	序	7D78	除	7D7C	傷	7D7D	勝	7D21	商	7E26	唱	7E27
	奨	7E29	将	7E2D	小	7E2E	少	7E2F	尚	7E30	床	7E32	承	7E35	招	7E37
	掌	7E38	昇	7E3A	昭	7E3C	消	7E43	渉	7E44	焼	7E46	焦	7E47	照	7E48
	省	7E4A	称	7E4E	章	7E4F	笑	7E50	紹	7E52	衝	7E57	訟	7E59	証	7E5A
	詳	7E5C	象	7E5D	賞	7E5E	鐘	7E62	障	7E63	上	7E65	乗	7E68	剰	7E6A
	城	7E6B	場	7E6C	壌	7E6D	常	7E6F	情	7E70	条	7E72	浄	7E74	状	7E75
	蒸	7E78	錠	7E7B	飾	7E7E	植	7F22	織	7F25	職	7F26	色	7F27	触	7F28
	食	7F29	伸	7F2D	信	7F2E	侵	7F2F	唇	7F30	寝	7F32	審	7F33	心	7F34
	振	7F36	新	7F37	森	7F39	浸	7F3B	深	7F3C	申	7F3D	真	7F3F	神	7F40
	紳	7F42	芯	7F44	親	7F46	診	7F47	身	7F48	辛	7F49	進	7F4A	針	7F4B
	震	7F4C	人	7F4D	刃	7F4F	尽	7F54	陣	7F58						
ス	す	6439	ず	643A	須	7F5C	酢	7F5D	図	7F5E	吹	7F61	垂	7F62	推	7F64
	水	7F65	粋	7F68	遂	7F6B	酔	7F6C	錐	7F6D	数	7F74	据	7F78	杉	7F79
	裾	7F7E	澄	8021	寸	8023										

セ	せ	643B	ぜ	643C	世	8024	瀬	8025	是	8027	制	8029	勢	802A	征	802C
	性	802D	成	802E	政	802F	整	8030	星	8031	晴	8032	正	8035	清	8036
	生	8038	盛	8039	精	803A	聖	803B	声	803C	製	803D	西	803E	誠	803F
	誓	8040	請	8041	青	8044	静	8045	税	8047	席	804A	昔	804E	析	804F
	石	8050	積	8051	籍	8052	績	8053	責	8055	赤	8056	跡	8057	切	805A
	接	805C	折	805E	設	805F	節	8061	説	8062	雪	8063	絶	8064	舌	8065
	先	8068	千	8069	占	806A	宣	806B	専	806C	尖	806D	川	806E	戦	806F
	扇	8070	栓	8072	泉	8074	浅	8075	洗	8076	染	8077	潜	8078	旋	807B
	線	807E	繊	8121	船	8125	選	812A	銑	812D	鮮	812F	前	8130	善	8131
	漸	8132	然	8133	全	8134	繕	8136								
ソ	そ	643D	ぞ	643E	遡	813A	礎	8143	粗	8146	素	8147	組	8148	訴	814A
	阻	814B	創	814F	双	8150	倉	8152	奏	8155	層	8158	想	815B	捜	815C
	掃	815D	挿	815E	操	8160	早	8161	巣	8163	争	8168	相	816A	窓	816B
	総	816D	草	8170	装	8175	走	8176	送	8177	騒	817B	像	817C	増	817D
	臓	8221	蔵	8222	贈	8223	造	8224	促	8225	側	8226	則	8227	即	8228
	息	8229	束	822B	測	822C	足	822D	速	822E	俗	822F	属	8230	族	8232
	続	8233	卒	8234	其	8236	揃	8237	存	8238	尊	823A	損	823B	村	823C
タ	た	643F	だ	6440	他	823E	多	823F	太	8240	詑	8242	堕	8244	妥	8245
	惰	8246	打	8247	体	824E	対	8250	耐	8251	帯	8253	待	8254	怠	8255
	態	8256	戴	8257	替	8258	滞	825A	袋	825E	貸	825F	退	8260	隊	8262
	代	8265	台	8266	大	8267	第	8268	題	826A	滝	826C	卓	826E	宅	8270
	択	8272	拓	8273	濯	8275	託	8277	濁	8279	諾	827A	叩	8231	達	8323
	奪	8325	脱	8326	棚	832A	谷	832B	誰	832F	単	8331	嘆	8332	担	8334
	探	8335	旦	8336	淡	8338	炭	833A	短	833B	端	833C	誕	8342	団	8344
	弾	8346	断	8347	暖	8348	段	834A	男	834B	談	834C				
チ	ち	6441	ぢ	6442	値	834D	知	834E	地	834F	恥	8351	池	8353	置	8356
	致	8357	遅	8359	馳	835A	築	835B	畜	835C	竹	835D	筑	835E	秩	8361
	茶	8363	着	8365	中	8366	仲	8367	宙	8368	忠	8369	抽	836A	昼	836B
	柱	836C	注	836D	虫	836E	鋳	8372	駐	8373	貯	8379	丁	837A	兆	837B
	帳	8422	庁	8423	張	8425	彫	8426	徴	8427	懲	8428	挑	8429	朝	842B
	町	842E	脹	8431	腸	8432	調	8434	超	8436	跳	8437	長	8439	頂	843A
	鳥	843B	直	843E	沈	8440	珍	8441	賃	8442						
ツ	っ	6443	つ	6444	づ	6445	墜	8446	追	8449	痛	844B	通	844C	塚	844D
	爪	845E	吊	845F	釣	8460										

テ	て	6446	で	6447	低	8463	停	8464	定	846A	底	846C	庭	846D	廷	846E
	抵	8471	提	8473	程	8478	締	8479	訂	847B	釘	8523	泥	8525	摘	8526
	敵	8528	滴	8529	的	852A	笛	852B	適	852C	撤	8531	鉄	8534	典	8535
	天	8537	展	8538	店	8539	添	853A	貼	853D	転	853E	点	8540	伝	8541
	殿	8542	田	8544	電	8545										
ト	と	6448	ど	6449	吐	8547	塗	8549	徒	854C	登	8550	途	8553	都	8554
	砥	8556	努	8558	度	8559	土	855A	怒	855C	倒	855D	党	855E	冬	855F
	凍	8560	刀	8561	島	8567	投	856A	東	856C	盗	8570	湯	8572	灯	8574
	当	8576	等	8579	答	857A	筒	857B	糖	857C	統	857D	到	857E	藤	8623
	討	8624	踏	8627	逃	8628	透	8629	陶	862B	頭	862C	闘	862E	働	862F
	動	8630	同	8631	堂	8632	導	8633	胴	8639	道	863B	銅	863C	峠	863D
	得	8640	徳	8641	特	8643	督	8644	毒	8647	独	8648	読	8649	凸	864C
	突	864D	届	864F	曇	865E	鈍	865F								
ナ	な	644A	内	8662	謎	8666	鍋	8669	馴	866B	縄	866C	南	866E	軟	8670
	難	8671														
ニ	に	644B	二	8673	匂	8677	肉	8679	日	867C	乳	867D	入	867E	尿	8722
	任	8724	認	8727												
ヌ	ぬ	644C														
ネ	ね	644D	熱	872E	年	872F	念	8730	撚	8733	粘	8734				
ノ	の	644E	悩	873A	濃	873B	納	873C	能	873D	脳	873E	農	8740		
ハ	は	644F	ば	6450	ぱ	6451	把	8744	覇	8746	波	8748	派	8749	破	874B
	馬	874F	廃	8751	拝	8752	排	8753	敗	8754	杯	8755	背	8758	肺	8759
	配	875B	倍	875C	媒	875E	買	8763	売	8764	博	876E	拍	876F	泊	8771
	白	8772	舶	8775	薄	8776	爆	877A	縛	877B	麦	877E	箱	8822	肌	8829
	畑	882A	八	882C	発	882F	髪	8831	罰	8833	抜	8834	閥	8836	伴	883C
	判	883D	半	883E	反	883F	搬	8842	板	8844	汎	8846	版	8847	犯	8848
	班	8849	繁	884B	般	884C	販	884E	範	884F	飯	8853	番	8856	盤	8857
ヒ	ひ	6452	び	6453	ぴ	6454	否	885D	彼	8860	悲	8861	扉	8862	批	8863
	比	8866	泌	8867	疲	8868	皮	8869	秘	886B	肥	886E	被	886F	費	8871
	避	8872	非	8873	飛	8874	備	8877	尾	8878	微	8879	美	887E	鼻	8921
	匹	8924	菱	8929	必	892C	筆	892E	百	8934	俵	8936	標	8938	氷	8939
	票	893C	表	893D	評	893E	描	8941	病	8942	秒	8943	品	894A	浜	894D
	貧	894F	敏	8952												

フ	ふ	6455	ぶ	6456	ぷ	6457	不	8954	付	8955	夫	8957	婦	8958	富	8959
	布	895B	府	895C	怖	895D	敷	895F	普	8961	浮	8962	父	8963	符	8964
	腐	8965	負	8969	武	8970	舞	8971	部	8974	封	8975	風	8977	伏	897A
	副	897B	復	897C	幅	897D	服	897E	福	8A21	腹	8A22	複	8A23	払	8A27
	沸	8A28	仏	8A29	物	8A2A	分	8A2C	噴	8A2E	憤	8A30	奮	8A33	粉	8A34
	紛	8A36	文	8A38	聞	8A39										
ヘ	へ	6458	べ	6459	ぺ	645A	丙	8A3A	併	8A3B	兵	8A3C	弊	8A3E	平	8A3F
	柄	8A41	並	8A42	閉	8A44	米	8A46	頁	8A47	壁	8A49	癖	8A4A	別	8A4C
	偏	8A50	変	8A51	片	8A52	編	8A54	辺	8A55	返	8A56	便	8A58	勉	8A59
	弁	8A5B														
ホ	ほ	645B	ぼ	645C	ぽ	645D	保	8A5D	捕	8A61	歩	8A62	補	8A64	募	8A67
	墓	8A68	慕	8A69	母	8A6C	簿	8A6D	倣	8A6F	包	8A71	報	8A73	宝	8A75
	崩	8A78	捧	8A7B	放	8A7C	方	8A7D	法	8B21	泡	8B22	縫	8B25	胞	8B26
	芳	8B27	訪	8B2C	豊	8B2D	飽	8B30	乏	8B33	亡	8B34	傍	8B35	剖	8B36
	妨	8B38	帽	8B39	忘	8B3A	忙	8B3B	房	8B3C	暴	8B3D	望	8B3E	棒	8B40
	紡	8B42	肪	8B43	膨	8B44	防	8B49	北	8B4C	僕	8B4D	撲	8B50	釦	8B55
	没	8B57	本	8B5C	翻	8B5D										
マ	ま	645E	摩	8B60	磨	8B61	魔	8B62	枚	8B67	毎	8B68	幕	8B6B	膜	8B6C
	末	8B76	迄	8B78	万	8B7C	満	8B7E								
ミ	み	645F	味	8C23	未	8C24	魅	8C25	密	8C29	脈	8C2E	妙	8C2F	民	8C31
ム	む	6460	務	8C33	夢	8C34	無	8C35	矛	8C37	霧	8C38				
メ	め	6461	名	8C3E	命	8C3F	明	8C40	盟	8C41	迷	8C42	鳴	8C44	滅	8C47
	免	8C48	綿	8C4A	面	8C4C										
モ	も	6462	模	8C4F	茂	8C50	毛	8C53	盲	8C55	網	8C56	耗	8C57	木	8C5A
	黙	8C5B	目	8C5C	戻	8C61	問	8C64	紋	8C66	門	8C67				
ヤ	ゃ	6463	や	6464	冶	8C6A	夜	8C6B	野	8C6E	矢	8C70	役	8C72	約	8C73
	薬	8C74	訳	8C75	躍	8C76										
ユ	ゅ	6465	ゆ	6466	油	8C7D	諭	8D21	輸	8D22	優	8D25	勇	8D26	友	8D27
	有	8D2D	由	8D33	誘	8D36	遊	8D37	郵	8D39	融	8D3B				
ヨ	ょ	6467	よ	6468	予	8D3D	余	8D3E	与	8D3F	誉	8D40	預	8D42	幼	8D44
	容	8D46	揚	8D48	揺	8D49	曜	8D4B	様	8D4D	洋	8D4E	溶	8D4F	用	8D51
	葉	8D55	要	8D57	踊	8D59	陽	8D5B	養	8D5C	抑	8D5E	浴	8D61	翼	8D63
ラ	ら	6469	螺	8D66	裸	8D67	来	8D68	頼	8D6A	雷	8D6B	絡	8D6D	落	8D6E
	乱	8D70	卵	8D71	欄	8D73	覧	8D77								

リ		り	646A	利	8D78	理	8D7D	裏	8E22	里	8E24	離	8E25	陸	8E26	律	8E27
		率	8E28	立	8E29	略	8E2C	流	8E2E	留	8E31	粒	8E33	隆	8E34	慮	8E38
		旅	8E39	虜	8E3A	了	8E3B	両	8E3E	寮	8E40	料	8E41	療	8E45	稜	8E47
		良	8E49	量	8E4C	領	8E4E	力	8E4F	緑	8E50	林	8E53	臨	8E57	輪	8E58
		隣	8E59														
ル		る	646B	塁	8E5D	涙	8E5E	累	8E5F	類	8E60						
レ		れ	646C	令	8E61	例	8E63	冷	8E64	励	8E65	礼	8E69	鈴	8E6B	隷	8E6C
		霊	8E6E	暦	8E71	歴	8E72	列	8E73	劣	8E74	烈	8E75	裂	8E76	恋	8E78
		練	8E7D	連	8F22												
ロ		ろ	646D	路	8F29	労	8F2B	浪	8F32	漏	8F33	老	8F37	郎	8F3A	六	8F3B
		録	8F3F	論	8F40												
ワ		ゎ	646E	わ	646F	和	8F42	話	8F43	歪	8F44	脇	8F46	惑	8F47	枠	8F48
		詫	8F4D	湾	8F51	腕	8F53										
ヲ		を	6472														
ン		ん	6473														
Special symbols		α	6641	β	6642	→	6F40	↗	6F41	↑	6F42	↖	6F43	←	6F44	↙	6F45
		↓	6F46	↘	6F47	↻	6F48	↺	6F49	⌒	6F4A	◠	6F4B	▮	6F4C	▽	6F50
		▽▽	6F51	▽▽▽	6F52	▽▽▽▽	6F53										

APPENDIX F. ERROR CODE TABLE

Number	Message displayed on CRT	Contents	Remarks
PS 003	TOO MANY DIGITS	Data entered with more digits than permitted. (See APPENDIX 2, ”Table of Maximum Command Values.”)	
PS 006	ILLEGAL USE OF NEGATIVE VALUE	A minus sign (–) was specified at an address where no minus sign may be specified, or two minus signs were specified.	
PS 007	ILLEGAL USE OF DECIMAL POINT	A decimal point (.) was specified at an address where no decimal point may be specified, or two decimal points were specified.	
PS 010	IMPROPER G–CODE	An illegal G code was specified. This alarm is also generated when a G code for an option that has not been added is specified.	
PS 011	IMPROPER NC–ADDRESS	An illegal address was specified. A command was specified for a slave axis during synchronous operation or independent operation of a slave axis.	
PS 012	INVALID BREAK POINT OF WORD	NC word(s) are delimited incorrectly.	
PS 013	ILLEGAL PROGRAM NO. POSITION	Address 0 or N is specified in an illegal location (i.e. after a macro statement, etc.)	
PS 014	ILLEGAL PROGRAM NO. FORMAT	Address 0 or N is not followed by a number.	
PS 015	TOO MANY WORDS IN ONE BLOCK	The number of words in a block exceeds the maximum.	
PS 016	EOB NOT FOUND	EOB (End of Block) code is missing at the end of a program input in MDI mode.	
PS 017	ILLEGAL MODE FOR GOTO/WHILE/DO	A GOTO statement or WHILE–DO statement was found in the main program in MDI or tape mode.	
PS 058	S–COMMAND OUT OF RANGE	The specified spindle speed exceeded the maximum number of revolutions allowed for the spindle motor (parameter No. 5619). (When SAL of parameter No. 5601 is 1)	
PS 059	COMMAND IN BUFFERING MODE	Manual intervention compensation request signal MIGET was input for a block containing no M code without buffering.	

T: Only for the Series 15–T and 15–TT used with lathes

NOTE PSxxx indicates alarms related to programs or settings in the foreground. Background alarms corresponding to PSxxx are displayed in the format BGxxx.
Similarly, background alarms corresponding to RSxxx are also displayed in the format BGxxx.

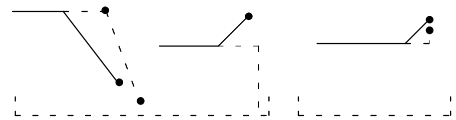
Number	Message displayed on CRT	Contents	Remarks
PS 060	SEQUENCE NUMBER NOT FOUND	The specified sequence No. was not found during sequence number search. The sequence No. specified as the jump destination in GOTOn or M99Pn was not found.	
PS 061	NO P, Q COMMAND AT G70–G72	Neither P nor Q was specified in a block of multiple repetitive canned cycles G70 to G73 for turning. (P: Sequence No. of the start cycle block Q: Sequence No. of the end cycle block)	T
PS 062	ILL COMMAND IN G70–G76	· The value specified by address D in a G71, G72, or G73 block is zero or negative. · The value specified by address K or D in a G76 block is zero or negative, or the value specified by address A is out of bounds (valid range: $0° \leqq A° \leqq 120°$). · The value specified by I, K, or D in a G74 or G75 block is negative. · X is specified even though the value specified by I in a G74 or G75 block is zero, or Z is specified even though the value specified by K is zero.	T
PS 063	P, Q BLOCK NOT FOUND	The blocks indicated by the sequence Nos. specified in P or Q in a block of multiple repetitive canned cycles G70 to G73 for turning were not found.	T
PS 064	SHAPE PROGRAM NOT MONOTONOUS	The change in movement along the Z–axis in the G71 finishing program, or that along the X–axis in the G72 finishing program, is not monotonous.	T
PS 065	ILL COMMAND IN P–BLOCK	Group 01 G code G00 or G01 has not been specified in the blocks indicated by P or Q in the blocks of multiple repetitive canned cycles G70 to G72 for turning.	
PS 066	ILL COMMAND IN PROGRAM	A group 01 G code other than G00, G01, G02, or G03, a single–shot G codes other than G04, or G65, G66, G67, M98, or M99 was specified in one of the blocks specified by the sequence numbers specified in P and Q of canned cycle G70, G71, or G72.	T
PS 067	G70–G73 IN FORBIDDEN MODE	Multiple respective canned cycle G70, G71, G72, or G73 is specified in a mode other than memory mode.	T
PS 069	ILL COMMAND IN Q–BLOCK	Chamfering or cornering is specified in a block indicated by the sequence No. specified in Q in a block of multiple repetitive canned cycles G70 to G73 for turning.	T
PS 076	PROGRAM NOT FOUND	The program indicated by the program No. called by M98, G65, G66, G66.1, or a G, M, or T code could not be found.	
PS 077	PROGRAM IN USE	An attempt was made in the foreground to execute a program being edited in the background.	
PS 079	PROGRAM FILE NO NOT FOUND	The specified program could not be found. This alarm occurs when a program specified by M198 is not registered in external memory. This alarm also occurs when a remote buffer is being used and an attempt is made to execute a program by specifying its program No. M198.	
PS 090	DUPLICATE NC, MACRO STATEMENT	An NC statement and macro statement were specified in the same block.	
PS 091	DUPLICATE SUB–CALL WORD	More than one subprogram call instruction was specified in the same block.	
PS 092	DUPLICATE MACRO–CALL WORD	More than one macro call instruction was specified in the same block.	
PS 093	DUPLICATE NC–WORD &M99	An address other than O, N, P, or L was specified in the same block as M99 during the macro continuous–state call state (G76).	
PS 094	G CANNOT BE ARGUMENT	Address G was used as the argument of a macro.	
PS 095	TOO MANY TYPE–2 ARGUMENTS	More than 10 sets of type–II arguments were specified for custom macros.	

Number	Message displayed on CRT	Contents	Remarks
PS 100	CANCEL WITHOUT MODAL CALL	Call mode cancel (G67) was specified even though macro continuous–state call mode (G66) was not in effect.	
PS 110	OVERFLOW: INTEGER	An integer went out of range during arithmetic calculations (valid range: -2^{31} to $2^{31}-1$)	
PS 111	OVERFLOW: FLOATING	A binary floating point number went out of range during arithmetic calculations.	
PS 112	DIVISION BY ZERO	An attempt was made to divide by zero in a custom macro.	
PS 114	VARIABLE NO. OUT OF RANGE	An illegal variable No. was specified in a custom macro.	
PS 115	READ PROTECTED VARIABLE	An attempt was made in a custom macro to use on the right side of an expression a variable that can only be used on the left side of an expression.	
PS 116	WRITE PROTECTED VARIABLE	An attempt was made in a custom macro to use on the left side of an expression a variable that can only be used on the right side of an expression.	
PS 118	TOO MANY NESTED BRACKETS	Too many brackets ([ ]) were nested in a custom macro.	
PS 119	ARGUMENT VALUE OUT OF RANGE	An argument of a function in a custom macro is out of range.	
PS 121	TOO MANY SUB, MACRO NESTING	1) The total number of subprogram and macro calls exceeds the allowable range. 2) An M98 or M198 command was called by external memory or a subprogram. The program specified by M198 contains an M98 or M198 command.	
PS 122	TOO MANY MACRO NESTING	Too many macros calls were nested in a custom macro.	
PS 123	MISSING END STATEMENT	The END command corresponding to a prior DO command was missing in a custom macro.	
PS 124	MISSING DO STATEMENT	The DO command corresponding to a post END command was missing in a custom macro.	
PS 125	ILLEGAL EXPRESSION FORMAT	There is an error in the format used in an expression.	
PS 126	ILLEGAL LOOP NO.	DO and END Nos. exceed the allowable range.	
PS 128	SEQUENCE NO. OUT OF RANGE	A sequence Nos. was out of range (valid range: 1 to 9999).	
PS 131	MISSING OPEN BRACKET	The number of left brackets ([ ) is less than the number of right brackets ( ]) in a custom macro.	
PS 132	MISSING CLOSE BRACKET	The number of right brackets ( ]) is less than the number of left brackets ([ ) in a custom macro.	
PS 133	MISSING “=”	An equals sign (=) is missing in a custom macro.	
PS 134	MISSING “/”	A division sign (/) is missing in a custom macro.	
PS 135	MACRO STATEMENT FORMAT ERROR	There is an error in the format used in a macro statement in a custom macro.	
PS 136	DFA STATEMENT FORMAT ERROR	There is an error in the format used in a DFA statement in a custom macro.	
PS 137	IF STATEMENT FORMAT ERROR	There is an error in the format used in an IF statement in a custom macro.	
PS 138	WHILE STATEMENT FORMAT ERROR	There is an error in the format used in a WHILE statement in a custom macro.	
PS 139	SETVN STATEMENT FORMAT ERROR	There is an error in the format used in a SETVN statement in a custom macro.	
PS 141	ILLEGAL CHARACTER IN VAR. NAME	A SETVN statement in a custom macro contains a character that cannot be used in a variable name.	

Number	Message displayed on CRT	Contents	Remarks
PS 142	VARIABLE NAME TOO LONG	The variable name used in a SETVN statement in a custom macro is too long.	
PS 143	BPRNT/DPRNT STATE- MENT FORMAT ERROR	There is an error in the format used in a BPRNT or DPRNT statement specified for external output from a custom macro.	
PS 144	G10 FORMAT ERROR	There is an error in the G10 format.	
PS 145	G10.1 TIME OUT	The response to a G10.1 command was not received from the PMC within the specified time limit.	
PS 146	G10.1 FORMAT ERROR	There is an error in the G10.1 format.	
PS 150	G31 FORMAT ERROR	The G31.8 command contains one of the following errors : – No axis is specified. – More than one axis is specified. – P is omitted.	
PS 180	ALL PARALLEL AXES IN PARKING	All of the parallel axes for an address are parked (the parking signal is on).	
PS 181	ZERO RETURN NOT FINISHED	During automatic operation, a movement command was issued for an axis for which reference position has not been performed since power–on.	
PS 182	CIRCLE CUT IN RAPID TRAVERSE	F0 (rapid traverse in inverse feed or feed specified by an F code with one–digit number was specified during circular interpolation (G02, G03) or involute interpolation (G02.2, G03.2) mode.	
PS 183	TOO MANY SIMULTA- NEOUS CONTROL AXES	A move command was specified for more axes than can be controlled by simultaneous axis control.	
PS 184	TOO LARGE DISTANCE	Due to compensation or similar reasons, a movement distance was specified that exceeds the maximum allowable distance.	
PS 185	ZERO RETURN CHECK(G27) ERROR	The axis specified in G27 (reference position return check) has not returned to the reference position.	
PS 186	ILLEGAL PLANE SELECTION	There is an error in the plane selection command (parallel axes are being specified at the same time).	
PS 187	FEEDRATE IS 0 (F COMMAND)	The cutting feedrate has been set to 0 by an F code.	
PS 188	DRY RUN FEEDRATE IS 0	The maximum feedrate or the dry run feedrate has been set to 0 by parameter.	
PS 190	CUTTING FEEDRATE IS 0	The maximum cutting feedrate has been set to 0 by parameter.	
PS 191	RADIUS TOO LARGE	An arc was specified for which the difference in the radius at the start and end points exceeds the value set in parameter 2410.	
PS 192	ILLEGAL LEAD COMMAND (G34)	In variable lead threading, the lead compensation value set with address K is greater than the maximum allowed value (the lead is negative or greater than the maximum allowed size).	
PS 193	ILLEGAL OFFSET NUMBER	An incorrect offset No. was specified.	
PS 194	ZERO RETURN END NOT ON REF	The axis specified in G28 (automatic reference position return) was not at the reference position when positioning was completed.	
PS 195	ILLEGAL AXIS SELECTED (G96)	An incorrect value was specified in P in a G96 block or parameter 5640.	
PS 196	ILLEGAL DRILLING AXIS SELECT	An illegal axis was specified for drilling in a canned cycle for drilling.	
PS 197	OTHER AXIS ARE COMMANDED	A command was issued for another axis at the same time a command was being issued for the index table axis.	
PS 198	ILLEGAL INDEX ANGLE	A value other than an integer multiple of the degree increment used by the index table was specified.	
PS 199	ILLEGAL COMMAND IN INDEXING	An illegal command was specified for indexing using the index table.	

Number	Message displayed on CRT	Contents	Remarks
PS 200	PULSE CODER INVALID ZERO RETURN	The grid position could not be set during grid reference position return using a digital position detector, because the one–revolution signal was not received before leaving the deceleration dog.	
PS 201	ILLEGAL LEAD COMMAND (G35) G02.1/03.1	During circular threading, a command for changing the long axis was issued.	
PS 201	G02.1/03.1 FORMAT ERROR	The specified arc exceeds the range where interpolation can be performed.	
PS 205	TURRET REF. NOT FINISHED	Indexing was specified without first performing reference position return of the turret axis.	
PS 206	ILLEGAL TOOL NO.	An invalid tool No. was specified.	
PS 210	ILLEGAL WAITING M–CODE	An M command with three–digit number is issued to a tool post which then entered a waiting state while a different M command with three–digit number was issued for another tool post.	TT
PS 211	PROGRAM NUMBER ODD/ EVEN ERROR	An attempt was made to execute an even–numbered program on the first tool post or an odd–numbered program on the second tool post.	TT
PS 212	ILLEGAL BALANCE CUTTING G–CODE	A G68 command was issued to a tool post which then entered a waiting state while a G69 command was issued for another tool post, or a G69 command was issued to a tool post which then entered a waiting state while a G68 command was issued for another tool post.	TT
PS 213	ILLEGAL USE OF PS213 G12.1/G13.1	Conditions are not correct for starting or cancelling polar coordinate interpolation. · G12.1 or G13.1 was commanded in a mode other than G40. · An illegal plane was selected (error in parameter 1032 or 1033).	
PS 214	ILLEGAL USE OF G–CODE	A G code which cannot be specified in G12.1 mode was specified.	
PS 215	TOOL RETRACTION INCORRECT	The retract command was specified incorrectly (long axis was specified in threading).	
PS 217	ILLEGAL OFFSET VALUE	Illegal offset number	
PS 218	FORMAT ERROR	Illegal format	
PS 223	ILLEGAL SPINDOLE SELECT	Although spindle control is disabled (G10.7 P0), an S code is specified.	
PS 229	G10.7 FORMAT ERROR	Address P specified in the G10.7 block is not 0, 1, or 2.	
PS 270	OFFSET START/STOP IN C MODE	An attempt was made to start or cancel tool tip radius compensation during arc mode.	
PS 271	CRC: ILLEGAL PLANE	An attempt was made to change the plane while offset C was in effect.	
PS 272	CRC: INTERFERENCE	The depth of cut is too great while offset C is in effect.	
PS 280	ILLEGAL COMMAND IN SPIRAL	An illegal command was specified for spiral or conical interpolation. (R was specified.)	
PS 281	OVER TOLERANCE OF END POINT IN SPIRAL	The difference between the positions of the specified and calculated end points exceeds the permissible value.	
PS 299	CRC: NO INTERSECTION	There is no point of intersection in offset C.	
PS 300	ILLEGAL ADDRESS	An illegal address was specified while loading parameters or pitch error compensation data from a tape.	
PS 301	MISSING ADDRESS	An address was found to be missing while loading parameters or pitch error compensation data from a tape.	
PS 302	ILLEGAL DATA NUMBER	An invalid data No. was found while loading parameters or pitch error compensation data from a tape.	
PS 303	ILLEGAL AXIS NUMBER	An invalid axis No. was found while loading parameters from a tape.	

Number	Message displayed on CRT	Contents	Remarks
PS 304	TOO MANY DIGITS	Data with too many digits was found while loading parameters or pitch error compensation data from a tape.	
PS 305	DATA OUT OF RANGE	Out–of–range data was found while loading parameters or pitch error compensation data from a tape.	
PS 306	MISSING AXIS NUMBER	A parameter which requires an axis to be specified was found without an axis No. while loading parameters from a tape.	
PS 307	ILLEGAL USE OF MINUS SIGN	Data with an illegal sign was found while loading parameters or pitch error compensation data from a tape.	
PS 308	MISSING DATA	An address not followed by a numeric value was found while loading parameters or pitch error compensation data from a tape.	
PS 400	PROGRAM DOES NOT MATCH	The program in memory does not match the program stored on tape.	
PS 410	G37 INCORRECT	No axis or more than one axis was specified in a tool measurement command block.	
PS 411	G37 SPECIFIED WITH D/H CODE	A D or H code was specified in the same block as an automatic tool–length measurement command.	
PS 412	G37 OFFSET NO. UNASSIGNED	No D or H code was specified before the tool length measurement command.	
PS 413	G37 SPECIFIED WITH T–CODE	A T code was specified in the same block as a tool length measurement command.	
PS 414	G37 OFFSET NO. UNASSIGNED	No T code was specified before the tool length measurement command.	
PS 415	G37 ARRIVAL SGNL NOT ASSERTED	During tool measurement, the measurement position arrival signal went on before the machine was in the area specified by parameter ε, or the measurement position arrival signal did not go on at all.	
PS 418	SPINDLE & OTHER AXIS MOVE	Spindle and other axis are specified in the same block for spindle positioning.	
PS 419	SPINDLE NOT ZERO RETURNED	Orientation (zero position return) was not completed in spindle positioning.	
PS 421	SETTING COMMAND ERROR	Command for setting tool data (G10L70 to G11, G10L71 to G11) is specified incorrectly . (Tool number specified for tool offset.)	
PS 422	TOOL DATA NOT FOUND	The pot number, tool length compensation data, and cutter compensation data has not been set for the specified tool number (tool number specified for tool offset).	
PS 425	TOOL DATA CANNOT BE DELETED	Tool data specified by the tool selection command cannot be deleted.	
PS 426	TOO MANY ADDRESSES	More than one of I, (J), K, or R was specified in the same block as chamfering or cornering. Example) G01X——K——R ;	
PS 427	TOO MANY AXES FOR I–R MOVE	There is no single–axis movement command specified in a block in which chamfering or corner rounding is specified. Example) G01X——Z——K—— ;	
PS 428	MISMATCH AXIS WITH CNR, CHF	I was specified with axis X or K was specified with axis Z in a block in which chamfering or corner rounding is specified. Example) G01X——I—— ; G01Z——K—— ;	
PS 429	MISSING VALUE AT CNR, CHF	The specified movement distance is less than the specified cornering or chamfering amount in a block in which chamfering or cornering is specified. 	
	MISSING VALUE AT CNR, CHF	When a block with cornering or chamfering was inserted, the original movement command range was exceeded.	M

Number	Message displayed on CRT	Contents	Remarks
PS 430	CODE ISN'T G01 AFTER CNR, CHF	The command following a block in which cornering or chamfering was specified is not G01. Example) N1 G01X––K–– ; N2 G00Z––I–– ;	
PS 431	MISSING MOVE AFTER CNR, CHF	The direction or movement distance specified in the block following a block in which cornering or chamfering is specified is incorrect.  Direction Movement distance	T
PS 431	MISSING MOVE AFTER CNR, CHF	The block following a block in which cornering or chamfering is specified is not a linear or circular interpolation block.	M
PS 437	ILLEGAL LIFE GROUP NUMBER	A tool group number exceeded the maximum value.	
PS 438	TOOL GROUP NOT SET	A tool group which has not been set was specified in a machining program.	
PS 439	TOO MANY TOOLS	The number of tools specified in a single tool group exceeded the maximum allowed number.	
PS 440	T COMMAND NOT FOUND (G10L3)	No T command was specified in a program which sets a tool group.	
PS 441	NOT USING TOOL IN LIFE GROUP	An H99 or D command was specified when no tool that belongs to a group was being used.	
PS 442	ILLEGAL T COMMAND AT M06	The tool group of the tool specified in the tool command (return tool group) after the M06 command in a machining program does not match the current tool group.	
PS 443	NOT FOUND P, L COMMAND	No P or L command was specified at the beginning of a program that sets a tool group.	
PS 445	ILLEGAL L COMMAND	The value specified in an L command in a program that sets a tool group is 0 or a value greater than the maximum tool life.	
PS 446	ILLEGAL T COMMAND	The value specified in a T command in a program that sets a tool group is greater than the maximum allowed value.	T
PS 446	ILLEGAL H D T COMMAND	The value specified by an H, D, or T command in a program that sets a tool group is greater than the maximum allowed value.	M
PS 447	ILLEGAL TYPE OF TOOL CHANGE	The tool change method is set incorrectly.	M
PS 448	UNUSABLE ADDRESS (G10L3)	A command that cannot be specified in a program that sets a tool group was specified.	
PS 449	NO TOOL LIFE DATA	A tool group was specified in a machining program even though no tool groups have been set.	
PS 450	IN PMC AXIS MODE	In the PMC axis control mode, the CNC issued a move command for the PMC axis. This alarm can be suppressed by setting NPAA of parameter No. 2405 to 1.	
PS 533	S CODE ZERO	The S code for rigid tapping is not specified.	
PS 573	EGB PARAMETER SETTING ERROR	EGB parameter setting error. (1)SYNAXS, bit 0 of parameter 1955, was erroneously set. (2)The rotation axis was not specified (bits 0 and 1 of parameter No. 1006). The synchronous start command with the number of turns was specified for the slave axis. (3)The number of pulses distributed per turn was not specified (parameter Nos. 5996 and 5997).	

Number	Message displayed on CRT	Contents	Remarks
PS 574	EGB FORMAT ERRO	Format error in the block in which the EGB command was specified (1)No command was specified for the master or slave axis. (2)Data which is out of range was specified for the master or slave axis. (3)A travel distance was specified for the master axis although the axis was not an NC–controlled axis (parameter No. 5998 was set to 0).	
PS 575	ILL–COMMAND IN EGB MODE	One of the illegal commands shown below was specified during EGB synchronization. (1)One of commands specified for the slave axis, such as G27, G28, G29, G30, and G53 (2)One of inch/millimeter conversion commands, such as G20 and G21	
PS 577	G80 FORMAT ERROR	Format error in that block in which the G80 code was specified (1)R falls outside the valid range.	
PS 578	G80 PARAMETER SETTING ERROR	Erroneous G80 parameter setting (1)The acceleration/deceleration parameter is invalid.	
PS 580	ENCODE ALARM (PSWD & KEY)	When an attempt was made to read a program, the specified password did not match the password on the tape and password on tape was not equal to 0. When an attempt was made to punch an encrypted tape, the password did not range from one to 99999999 inclusive.	
PS 581	ENCODE ALARM (PARAMETER)	When an attempt was made to punch an encrypted tape, the punch code parameter was set to EIA. An incorrect command was specified for program encryption or protection.	
PS 585	G81 PARAMETER SETTING ERROR	Erroneous G81–parameter setting 1 (1)SYNAXS, bit 0 of parameter 1955, was erroneously set. (2)The slave axis was not specified as a rotation axis (bits 0 and 1 of parameter No. 1006). (3) Parameter No. 5995 was not set. (4)The acceleration/deceleration parameter is invalid. (5)The automatic phase synchronization parameter is invalid.	
PS 586	G81 FORMAT ERROR	Format error in the block in which the G81 code was specified (1)T (number of teeth) was not specified. (2)Data which was out of range was specified for T or L. (3)An overflow occurred in the calculation of synchronous coeffi cients. (4)R falls outside the valid range. (5)The synchronous feedrate for the workpiece axis is greater than or equal to the rapid traverse rate (parameter No. 1420).	
PS 587	ILL–COMMAND IN G81 MODE	One of the illegal commands shown below was specified during G81 synchronization. (1)One of commands specified for the C–axis, such as G27, G28, G29, G30, and G53 (2)One of inch/millimeter conversion commands, such as G20 and G21	
PS 610	ILLEGAL G07.1 AXIS	Axis which cannot perform cylindrical interpolation was specified. More than one axis was specified in a G07.1 block. An attempt was made to cancel cylindrical interpolation for an axis that was not in cylindrical interpolation mode.	
PS 611	ILLEGAL G–CODE USE (G07.1 MODE)	A G code was specified that cannot be specified in cylindrical interpolation mode.	
PS 625	TOO MANY G68 NESTING	3–dimensional coordinate switching (G68) was specified more than twice.	
PS 626	G68 FORMAT ERROR	There is a format error in a G68 block. This alarm occurs in the below cases: (1)When I, J, or K is missing from the block in which G68 is specified (when the coordinate rotation option is not available) (2)When I, J, and K specified in the block in which G68 is specified are all 0 (3)When R is not specified in the block in which G68 is specified	

Number	Message displayed on CRT	Contents	Remarks
PS 630	HPCC:ILLEGAL START U/CANCEL	G05P10000 was specified in a mode which cannot enter the HPCC mode (See Section 2.1.1) The HPCC mode was canceled without canceling the cutter compensation mode (G41/ G42).(G40 is required before G50P0.)	
PS 631	HPCC:NOT READY STATE	The RISC processor board is not ready for operation.	
PS 632	HPCC:ILLEGAL PARAME-TER	The setting of a parameter is invalid.	
PS 633	HPCC:CRC OFS REMAIN AT CANCEL	The HPCC mode was canceled with a cutter compensation value retained.	
PS 634	HPCC:ILLEGAL COMMAND CODE	An invalid code was specified in the HPCC mode. (Refer to 2.1.2). In a block that includes positioning, or auxiliary function, cutter compensation (G41, G42), cancel (G40), vector retention (G38), corner arc (G39), or macro call was specified.	
PS 710	ILLEGAL COMMAND IN SPACE–CIR	An error was found in the specification for three dimensional circular interpolation.	
PS 711	ILLEGAL COMBINATION OF M CODES	M codes of the same group are specified in the same block. Or, an M code which must be specified alone is specified with an other M code in a block.	
PS 805	ILLEGAL COMMAND	A command is specified incorrectly in a block.	
PS 891	ILLEGAL COMMAND G05	G05 was specified in a state in which G05 cannot be specified.	M
PS 895	ILLEGAL PARAMETER IN G02.3/G03.3	The setting (parameter No. 7636) that specifies the axis for which to perform exponential interpolation is incorrect.	M
PS 896	ILLEGAL FORMAT IN G02.3/G03.3	The format for specifying exponential interpolation is incorrect (addresses I, J, or R are not specified, are set to 0, or the span value is negative).	M
PS 897	ILLEGAL COMMAND IN G02.3/G03.3	An illegal value was specified in exponential interpolation (i.e. parameter 1n is negative, etc.)	
PS 898	ILLEGAL PARAMETER IN G54.2	An illegal parameter (6068 to 6076) was specified for fixture off-set.	
PS 900	G72.1 NESTING ERROR	G72.1 was specified again during G72.1 rotation copying.	
PS 901	G72.2 NESTING ERROR	G72.2 was specified again during G72.2 parallel copying.	
PS 920	ILL–COMMAND IN G05.1 Q1	An invalid command was specified in G05.1 Q1 (look–ahead acceleration/deceleration before interpolation) mode. (1)F1–digit command	
PS 935	ILLEGAL FORMAT IN G02.2/G03.2	The end point of an involute curve, or I, J, K, or R was not specified.	
PS 936	ILLEGAL COMMAND IN G02.2/G03.2	An illegal value was specified in involute interpolation. (1)The start point or end point is specified inside the basic circle. (2)I, J, K, or R was specified as 0.	
PS 937	OVER TOLERANCE OF END POINT	The end point is not positioned on the involute curve that passes through the start point (the value specified in parameter No. 2510 is out of range).	
PS 990	SPL : ERROR	The spline interpolation axis command is invalid. In NURBS interpolation by high–precision contour control using a RISC processor: (1)An invalid rank was specified. (2)No knot, or an invalid knot, was specified. (3)Too many axes were specified. (4)Another program error was detected.	
PS 991	SPL : ERROR	A G06.1 command was specified in a G code mode in which the command is not supported.	

Number	Message displayed on CRT	Contents	Remarks
PS 992	SPL : ERROR	Movement was specified for an axis other than those used for spline interpolation. In NURBS interpolation by high–precision contour control using a RISC processor: (1) A program error was detected in the block to be read in advance. (2) Increase in knots is not monotonous. (3) A mode which cannot be used in NURBS interpolation mode was specified.	
PS 993	SPL : ERROR	A three–dimensional tool offset vector cannot be generated. In NURBS interpolation by high–precision contour control using a RISC processor, the first control point for NURBS is invalid.	
PS 994	SPL : ERROR	In NURBS interpolation by high–precision contour control using a RISC processor, NURBS interpolation was resumed after manual intervention with the manual absolute switch set to ON.	
PS 996	G41.3/G40 FORMAT ERROR	A move command was specified in the block in which the G41.3 or G40 code was specified. A G or M code which suppresses buffering was specified in the block in which the G41.3 code was specified.	
PS 997	ILLEGAL COMMAND IN G41.3	A G code other than G00 or G01 in group 01 was specified in the continuous state of the G41.3 code. An offset command (G code in group 07) was specified in the G41.3 mode. The block following the block in which G41.3 (start up) was specified did not have a move command.	
PS 998	G41.3 ILLEGAL START UP	The G41.3 code (start up) was specified in the continuous state of a G code other than G00 or G01 in group 01. The angle formed by the tool–direction vector and the movement–direction vector was 0 or 180 degrees at start up.	
PS 999	ILLEGAL PARAMETER IN G41.3	A parameter which determines the relationship between the rotation axis and rotation plane was erroneously set (parameter Nos. 1022 and 6080 to 6089)	

Number	Message displayed on CRT	Contents	Remarks
SR 160	INTERNAL DATA OVER FLOW	Coordinates specified in an absolute or machine coordinate system are out of range. In general, this alarm is generated when the rotational axis is rotated a number of times and an absolute coordinate becomes greater than +99999.999 or less than –99999.999 (for increment system IS–B). Perform reference position return when this alarm occurs.	
SR 310	H, S, C ERROR	G10.3, G11.3, or G65.3 is used incorrectly.	
SR 311	H, S, C ERROR	A command not permitted during high–speed machining was specified.	
SR 312	H, S, C ERROR	Address P, Q, or L is out of range.	
SR 313	H, S, C ERROR	High–speed machining data does not exist for the number that was called.	
SR 314	H, S, C ERROR	No cluster exists for the number that was called.	
SR 315	H, S, C ERROR	An attempt was made to register a number for which high–speed machining data already exists.	
SR 316	H, S, C ERROR	Memory is full.	
SR 317	H, S, C ERROR	The maximum number of sets of high–speed machining data has been registered (no additional high–speed machining data can be registered).	
SR 318	H, S, C ERROR	An attempt was made to register high–speed machining data during background editing.	
SR 319	H, S, C ERROR	There is an error in the format of high–speed machining data.	
SR 320	H, S, C ERROR	A macro interrupt occurred during high–speed machining.	
SR 424	OVER MAXIMAM TOOL DATA	An attempt was made to register new tool data even though the maximum number of sets of tool data has already been set (tool No. specified for tool offset).	
SR 590	TH ERROR	TH alarm (A character with a parity error was entered in a significant data area.)	
SR 591	TV ERROR	TV alarm (The number of characters in a block was not even.)	
SR 592	END OF RECORD	EOR (End of Record) was specified partway through a block.	
SR 600	PARAMETER OF RESTART ERROR	There is an error in the specification of the program restart parameter.	
SR 805	ILLEGAL COMMAND	An attempt was made to use an illegal command for the reader/punch interface. This is generally a system error.	
SR 806	DEVICE TYPE MISS MATCH	An attempt was made to use a command which is invalid for the currently selected I/O device. For example, this alarm occurs when an attempt is made to perform a file head search even though the medium is not a FANUC floppy disk.	
SR 807	PARAMETER SETTING ERROR	An I/O interface which was not selected as an option was specified, or the settings or parameters related to the I/O interface are set incorrectly.	
SR 810	PTR NOT READY	The ready signal for the tape reader is off.	
SR 812	OVERRUN ERROR (PTR)	The next character was received by the tape reader before it could read a previously received character.	
SR 820	DR OFF(1)	The data set ready signal for reader/punch interface 1 went off.	
SR 821	CD OFF(1)	The signal quality detection signal for reader/punch interface 1 went off.	
SR 822	OVERRUN ERROR(1)	The next character was received by reader/punch interface 1 before it could read a previously received character.	
SR 823	FRAMING ERROR(1)	No stop bit was detected for a character received by reader/punch interface 1.	

Number	Message displayed on CRT	Contents	Remarks
SR 824	BUFFER OVERFLOW(1)	The NC received more than ten characters of data from reader/ punch interface 1 even though the NC sent a stop code (DC3) during data reception.	
SR 830	DR OFF(2)	The data set ready signal for reader/punch interface 2 went off.	
SR 831	CD OFF(2)	The signal quality detection signal for reader/punch interface 2 went off.	
SR 832	OVERRUN ERROR(2)	The next character was received by reader/punch interface 2 before it could read a previously received character.	
SR 833	FRAMING ERROR(2)	No stop bit was detected for a character received by reader/ punch interface 2.	
SR 834	BUFFER OVERFLOW(2)	The NC received more than ten characters of data from reader/ punch interface 2 even though the NC sent a stop code (DC3) during data reception.	
SR 840	DR OFF(3)	The data set ready signal for reader/punch interface 2 went off.	
SR 841	CD OFF(3)	The signal quality detection signal for reader/punch interface 3 went off.	
SR 842	OVERRUN ERROR(3)	The next character was received by reader/punch interface 3 before it could read a previously received character	
SR 843	FRAMING ERROR(3)	No stop bit was detected for a character received by reader/ punch interface 3.	
SR 844	BUFFER OVERFLOW(3)	The NC received more than ten characters of data from reader/ punch interface 3 even though the NC sent a stop code (DC3) during data reception.	
SR 860	DATA SET READY DOWN (ASR33/43)	The data set ready signal for the 20 mA current loop interface went off.	
SR 861	CARRIER DETECT DOWN (ASR33/43)	The signal quality detection signal for the 20 mA current loop interface went off.	
SR 862	OVERRUN ERROR (ASR33/43)	The next character was received by the 20 mA current loop inter- face before it could read a previously received character.	
SR 863	FRAMING ERROR (ASR33/ASR44)	No stop bit was detected for a character received by the 20 mA current loop interface.	
SR 864	BUFFER OVERFLOW (ASR33/ASR44)	The NC received more than ten characters of data from the 20 mA current loop interface even though the NC sent a stop code (DC3) during data reception.	
SR 870	DATA SET READY DOWN (RS422)	The data mode signal for the RS–422 interface went off.	
SR 872	OVERRUN ERROR (RS422)	The next character was received by the RS–422 interface before it could read a previously received character.	
SR873	FRAMING ERROR (RS422)	No stop bit was detected for a character received by the RS–422 interface.	
SR 874	BUFFER OVERFLOW (RS422)	The NC received more than ten characters of data from the RS–422 interface even though the NC sent a stop code (DC3) during data reception.	
SR 880	NOT DRAWING	Communication cannot be made with the graphic CPU.	
SR 890	CHECK SUM ERROR (G05)	A check sum error occurred (high–speed remote buffer).	

Number	Message displayed on CRT	Contents	Remarks
OH 000	MOTOR OVERHEAT	A servo motor overheated.	Axis type
OH 001	LOCKER OVERHEAT	The NC cabinet overheated.	

Axis type : Error codes of this type are specified for each axis for which it occurs.

Number	Message displayed on CRT	Contents	Remarks
SB 010	GRAPHIC RAM PARITY	A parity error occurred in the graphic ROM.	
SB 011	GRAPHIC RAM PARITY (WORK)	A parity error occurred in the graphic work RAM.	
SB 011	GRAPHIC RAM PARITY (RED)	A parity error occurred in the RAM for red graphics.	
SB 011	GRAPHIC RAM PARITY (GREEN)	A parity error occurred in the RAM for green graphics.	
SB 011	GRAPHIC RAM PARITY (BLUE)	A parity error occurred in the RAM for blue graphics.	

Number	Message displayed on CRT	Contents	Remarks
SW 000	PARAMETER ENABLE SWITCH ON	Parameter setting is enabled (PWE, a bit of parameter No. 8000 is 1)	

Number	Message displayed on CRT	Contents	Remarks
OT 001	+OVERTRAVEL (SOFT1)	This alarm occurs in the following cases: –While moving in a positive direction, the tool exceeded the limit set in stored stroke limit 1. –While moving in the positive direction of a hypothetical axis, the tool exceeded the limit set in the hypothetical axis stroke limit. –While moving in the positive direction of a hypothetical axis, the eccentric tool length (R) was exceeded.	Axis type
OT 002	–OVERTRAVEL (SOFT1)	This alarm occurs in the following cases: –While moving in a negative direction, the tool exceeded the limit set in stored stroke limit 1. –While moving in the negative direction of a hypothetical axis, the tool exceeded the limit set in the hypothetical axis stroke limit. –While moving in the negative direction of a hypothetical axis, the eccentric tool length (R)was exceeded.	Axis type
OT 003	+OVERTRAVEL (SOFT2)	While moving in a positive direction, the tool exceeded the limit set in stored stroke limit 2, or the tool entered the out–of–bounds chuck/tool stock area.	Axis type
OT 004	–OVERTRAVEL (SOFT2)	While moving in a negative direction, the tool exceeded the limit set in stored stroke limit 2, or the tool entered the out–of–bounds chuck/tool stock area.	Axis type
OT 005	+OVERTRAVEL (SOFT3)	While moving in a positive direction, the tool exceeded the limit set in stored stroke limit 3.	Axis type T
OT 006	–OVERTRAVEL (SOFT3)	While moving in a negative direction, the tool exceeded the limit set in stored stroke limit 3.	Axis type T
OT 007	+OVERTRAVEL (HARD)	The stroke limit switch located at the positive side was turned on.	Axis type
OT 008	–OVERTRAVEL (HARD)	The stroke limit switch located at the negative side was turned on.	Axis type
OT 021	+OVERTRAVEL (PRE–CHECK)	The tool exceeded the limit in the positive direction during the stroke check before movement.	Axis type
OT 022	–OVERTRAVEL (PRE–CHECK)	The tool exceeded the limit in the negative direction during the stroke check before movement.	Axis type
OT 030	EXCESS ERROR ALARM 1	Alarm 1 for excess synchronous error was generated.	Axis type
OT 031	SYNCHRONIZE ADJUST MODE	The system is in synchronous adjustment mode (bit 1 of parameter No. 1803 is set to 1).	Axis type
OT 032	NEED ZRN (ABS PCDR)	The counter value of the absolute pulse coder does not match the machine coordinates.	Axis type
OT 034	BATTERY ZERO (ABS PCDR)	The battery voltage of the absolute pulse coder is 0 V. Check whether the battery is connected properly. If the battery is connected properly, replace it.	Axis type
O T036	BATTERY DOWN (SPLC)	The battery voltage of the serial pulse coder is low. (Built–in unit)	
O T037	BATTERY DOWN (EX)	The battery voltage of the additional detector is low. (Separate unit)	
O T038	BATTERY ZERO (SPLC)	The serial pulse coder battery voltage is 0 V. (Built–in unit)	
O T039	BATTERY ZERO (EX)	The serial pulse coder battery voltage is 0 V. (Separate unit)	
OT 100	SPINDLE ALARM	The spindle motor alarm was generated, but the code for specifying the cause of the alarm is 0.	
OT 101	SPINDLE MOTOR OVERHEAT	The spindle motor overheated.	
OT 102	EXCESS VELOCITY ERROR	The difference between the speed set for the spindle motor and the actual speed of the spindle motor was too great (15% to 20%).	
OT 103	FUSE F7 BLEW	Fuse F7 for the speed control unit of the spindle motor has blown.	

Number	Message displayed on CRT	Contents	Remarks
OT 104	FUSE F1, F2 OR F3 BLEW	Fuse F1, F2, or F3 for the speed control unit of the spindle motor has blown.	
OT 105	FUSE AF2 OR AF3 BLEW	Fuse AF2 or AF3 for the speed control unit of the spindle motor has blown.	
OT 106	EXCESS VELOCITY (ANA-LOG)	The speed of the spindle motor exceeded the maximum rating (analog detection).	
OT 107	EXCESS VELOCITY (DIGI-TAL)	The speed of the spindle motor exceeded the maximum rating (digital detection).	
OT 108	VOLTAGE (+24V) TOO HIGH	The voltage of the 24 V power supply for the speed control unit of the spindle is too high.	
OT 109	POWER SEMICONDUCTOR OVERLOAD	The power semiconductor overheated.	
OT 110	VOLTAGE (+15) TOO LOW	The voltage of the 15 V power supply for the speed control unit of the spindle motor is too low.	
OT 111	VOLTAGE EXCESS (DC LINK)	The voltage in the DC link for the speed control unit of the spindle motor is too high.	
OT 112	CURRENT EXCESS (DC LINK)	The current in the DC link for the speed control unit of the spindle motor is excessive.	
OT 113	CPU ERROR	The CPU or a peripheral circuit of the speed control unit of the spindle motor is faulty.	
OT 114	ROM ERROR	The ROM of the speed control unit of the spindle motor is faulty.	
OT 115	OPTION ALARM	Optional alarm for the spindle motor.	
OT 116	DISCONNECTION POS CODER	The position coder is disconnected.	
OT 117	SPINDLE OVERHEAT	Spindle alarm was detected by the function for detecting fluctua-tions in spindle speed.	
OT 120	UNASSIGNED ADDRESS (HIGH)	The upper four bits (EIA4 to EIA7) of an external data I/O inter-face address signal are set to an undefined address (high bits).	
OT 121	UNASSIGNED ADDRESS (LOW)	The lower four bits (EIA0 to EIA3) of an external data I/O inter-face address signal are set to an undefined address (low bits).	
OT 122	TOO MANY MESSAGES	Requests were made to display more than four external operator messages or external alarm messages at the same time.	
OT 123	MESSAGE NUMBER NOT FOUND	An external operator message or external alarm message cannot be cancelled because no message number is specified.	
OT 124	OUTPUT REQUEST ERROR	An output request was issued during external data output, or an output request was issued for an address that has no output data.	
OT 125	TOO LARGE NUMBER	A number outside of the range 0 to 999 was specified as the number for an external operator message or an external alarm message.	
OT 126	SPECIFIED NUMBER NOT FOUND	This alarm occurs in the following cases: –The number specified for a program number or sequence number search during external data input could not be found. –The number specified for a workpiece number search could not be found. –There was an I/O request issued for a pot number or offset (tool data), but either no tool numbers have been input since power–on or there is no data for the entered tool number.	
OT 127	DI. EIDHW OUT OF RANGE	The value input by external data input signals EID32 to EID47 is out of range.	
OT 128	DI. EIDLL OUT OF RANGE	The value input by external data input signals EID0 to EID31 is out of range.	
OT 129	NEGATE POS CODER 1 REV ON	The CPU or a peripheral circuit of the position coder is faulty.	

Number	Message displayed on CRT	Contents	Remarks
OT 130	SEARCH REQUEST NOT ACCEPTED	No requests can be accepted for a program No. or sequence No. search because the system is not in memory mode or the reset state.	
OT 131	EXT–DATA ERROR (OTHER)	· An input request was received for a pot No.or offset value while tool data was being registered with G10. · An error occurred during reading or writing of tool data.	
OT 132	RETURN ERROR (PTRR)	The tool did not return to the stored position along the axis in tool retraction mode.	Axis type
OT 150	A/D CONVERT ALARM	The A/D converter is faulty.	
OT 151	A/D CONVERT ALARM	The A/D converter is faulty.	
OT 184	PARAMETER ERROR IN TORQUE	An invalid parameter was specified for torque control. (1)The torque constant parameter is set to 0.	
OT 200	INTERFERENCE DATA ERROR	The tool figure data specified for the tool post interference check is incorrect (I > X or K > Z in tool figure data).	TT
OT 208	INTERFERENCE X1 MINUS	An interference alarm occurred while the first tool post was moving in the negative direction along the X axis.	TT
OT 209	INTERFERENCE X1 PLUS	An interference alarm occurred while the first tool post was moving in the positive direction along the X axis.	TT
OT 210	INTERFERENCE Z1 MINUS	An interference alarm occurred while the first tool post was moving in the negative direction along the Z axis.	TT
OT 211	INTERFERENCE Z1 PLUS	An interference alarm occurred while the first tool post was moving in the positive direction along the Z axis.	TT
OT 212	INTERFERENCE X2 MINUS	An interference alarm occurred while the second tool post was moving in the negative direction along the X axis.	TT
OT 213	INTERFERENCE X2 PLUS	An interference alarm occurred while the second tool post was moving in the positive direction along the X axis.	TT
OT 214	INTERFERENCE Z2 MINUS	An interference alarm occurred while the second tool post was moving in the negative direction along the Z axis.	TT
OT 215	INTERFERENCE Z2 PLUS	An interference alarm occurred while the second tool post was moving in the positive direction along the Z axis.	TT
OT 300	S–SPINDLE LSI ERROR	An error occurred in communication with the spindle.	
OT 301	MOTOR OVERHEAT	Motor overheat	
OT 302	EX DEVIATION SPEED	Excessive deviation in speed	
OT 307	OVER SPEED	Excessive speed	
OT 309	OVERHEAT MAIN CIRCUIT	Excessive load on the main circuit	
OT 310	LOW VOLT INPUT POWER	Low voltage in the input power supply	
OT 311	OVERVOLT POW CIRCUIT	Excessive voltage in the DC link	
OT 312	OVERCURRENT POW CIRCUIT	Excessive current in the DC link	
OT 313	DATA MEMORY FAULT CPU	Error in CPU internal data memory	
OT 318	SUMCHECK ERROR PGM DATA	Sum check error in program ROM	
OT 319	EX OFFSET CURRENT U	Excessive offset in U–phase current detection circuit	
OT 320	EX OFFSET CURRENT V	Excessive offset in V–phase current detection circuit	
OT 324	SERIAL TRANSFER ERROR	Error in serial data transfer	
OT 325	SERIAL TRANSFER STOP	Serial data transfer was stopped.	
OT 326	DISCONNECT C–VELO DETCT	The Cs axis control speed detection signal is disconnected.	
OT 327	DISCONNECT POS–CODER	The position coder signal is disconnected.	

Number	Message displayed on CRT	Contents	Remarks
OT 328	DISCONNECT C–POS DETCT	The Cs axis control position detection signal is disconnected.	
OT 329	OVERLOAD	Temporary overload	
OT 330	OVERCURRENT POW CIRCUIT	Excessive current in input circuit	
OT 331	MOTOR LOCK OR V–SIG LOS	The motor locked up alarm is issued or the speed detection signal is disconnected.	
OT 332	RAM FAULT SERIAL LSI	Faulty RAM exists in LSI for serial data transfer.	
OT 333	SHORTAGE POWER CHARGE	Insufficient power charge in DC link	
OT 334	PARAMETER SETTING ERROR	Parameter data is out of range.	
OT 335	EX SETTING GEAR RATIO	An excessive gear ratio was set.	
OT336	OVERFLOW ERROR COUNTER	The error counter overflowed.	
OT 399	S–SPINDLE ERROR	A miscellaneous spindle amplifier alarm occurred.	
OT 512	EXCESS VELOCITY	The feedrate specified during polar coordinate interpolation mode (G12.1) was greater than the maximum cutting feedrate.	
OT 513	SYNC EXCESS ERROR	The difference between the positional deviations for the synchronized axes during simple synchronization was greater than the limit set in parameter No. 7799.	Axis type

Number	Message displayed on CRT	Contents	Remarks
PC 010	PC ERROR	A parity error occurred in a PMC ROM.	
PC 020	PC ERROR	A parity error occurred in a PMC RAM.	
PC 030	PC ERROR	An I/O unit was not allocated correctly.	
PC 500	WATCH DOG ALARM	A PMC watch dog alarm occurred.	

Number	Message displayed on CRT	Contents	Remarks
SV 000	TACHOGENERATER DIS-CONNECT	The tachogenerator is disconnected.	Axis type
SV 001	EXCESS CURRENT IN SERVO	The servo motor was overloaded.	Axis type
SV 002	BREAKER (FEED_CTL_UNIT) OFF	The breaker for the velocity control circuit tripped.	Axis type
SV 003	ABNORMAL CURRENT IN SERVO	Excessive current was detected in the velocity control circuit.	Axis type
SV 004	EXCESS V TO MOTOR	Excessive voltage was detected in the velocity control circuit.	Axis type
SV 005	EXCESS DISCHARGE I FROM MOTOR	Excessive discharge current from the motor was detected in the velocity control circuit.	Axis type
SV 006	VELOCTY UNIT POWER TOO LOW	Low voltage was detected in the velocity control circuit.	Axis type
SV 008	EXCESS ERROR (STOP)	Position deviation when stopped is larger than the value set in parameter No. 1829.	Axis type
SV 009	EXCESS ERROR (MOVING)	Position deviation while moving is larger than the value set in parameter No. 1828.	Axis type
SV 010	EXCESS DRIFT COM-PENSATION	Drift compensation is excessive (exceeded 500VELO).	Axis type
SV 011	LSI OVERFLOW	The value specified for position deviation compensation is not within the range of –32767 to +32767, or the command value of the D/A converter is not within the range of –8192 to +8192. In general, this alarm occurs due to incorrect settings.	Axis type
SV 012	MOTION VALUE OVER-FLOW	A speed greater than 512,000 pulses per second (frequency converted into detection units) was specified. In general, this alarm occurs due to incorrect setting of parameter CMR or the feedrate.	Axis type
SV 013	IMPROPER V–READY OFF	The speed control ready signal (VRDY) went off even though the position control ready signal (PRDY) was on. When a multiaxis amplifier is being used, specifying controlled axis detachment for any axis driven by the amplifier causes alarm SV013 to be issued for the other axes driven by that amplifier.	Axis type
SV 014	IMPROPER V–READY ON	The speed control ready signal (VRDY) went on even though the position control ready signal (PRDY) was off.	Axis type
SV 015	PULSE CODER DISCON-NECTED	The pulse coder is disconnected.	Axis type
SV 017	ILL POSITION CONTROL LSI	A fault occurred in the LSI for position control.	Axis type
SV 018	INCORRECT DSCG FRE-QUENCY	A fault was detected in a resolver or inductsyn feedback frequency check.	Axis type
SV 019	INCORRECT PULSE CODER PULSE	A fault was detected in the feedback pulse from the pulse coder.	Axis type
SV 020	INCORRECT PHASE SHIFT VALUE	The resolver or induction phase shift cannot be obtained correctly.	Axis type
SV 021	1–REV PULSE SIGNAL INCORRECT	The single–revolution signal for the pulse coder went on at an incorrect position.	Axis type
SV 022	1–REV PULSE MISSING	The single–revolution signal for the pulse coder did not go on within the appropriate range.	Axis type
SV 023	SV OVERLOAD	A servo motor overloaded.	Axis type
SV 024	EXCESS ERROR ALARM 2	Alarm 2 for excess synchronous error was generated.	
SV 025	V–READY ON (INITIALIZING)	During servo control, the speed control ready signal (VRDY) is on even though it is supposed to be off.	Axis type
SV 026	ILLEGAL AXIS ARRANGE-MENT	Parameter No. 1023 for specifying the arrangement of servo axes is set incorrectly.	

Number	Message displayed on CRT	Contents	Remarks
SV 027	ILL DGTL SERVO PARAMETER	There is an illegal digital servo parameter. – A correct motor model type is not set in parameter No. 1874. – The number of pulses per motor rotation is not set in parameter No. 1876. – The direction of motor rotation set in parameter No. 1879 is neither 111 nor –111.	
SV 030	EMERGENCY STOP	Emergency stop occurred (when ENR, a bit of parameter No. 2001, is 1).	
SV 031	EXCESS SPINDLE DIST (SPDL)	Excess position deviation occurred for the spindle during rigid tapping.	
SV 032	LSI OVERFLOW (SPDL)	Data for the LSI used for spindle control overflowed during rigid tapping.	
SV 050	ILLEGAL AXIS SYNCHRO-NIZATION	Parameter No. 1023 is set incorrectly.	
SV 055		Parameter No. 1023 is set incorrectly.	
SV 056		Bit #6 of parameter No. 1817 is set incorrectly.	
SV 99	SPINDLE–AXIS ERROR	An amplifier alarm occurred during spindle contour control.	
SV 100	SACOMP, VALUE OVER-FLOW	The value set for straightness compensation is not within the range of –32767 to +32767.	M
SV 101	DATA ERROR (ABS PCDR)	A correct machine position cannot be obtained because the absolute pulse coder is faulty or the machine moved too far during power–on.	Axis type
SV 110	PULSE CORDER ALARM (SERIAL A)	An error was detected in the serial pulse coder (serial A) or feedback cable. Replace the cable or pulse coder. (For details on the alarm, refer to the "FANUC AC Servo Amplifier Maintenance Manual.")	Axis type
SV 111	PULSE CORDER ALARM (SERIAL C)	An error was detected in the serial pulse coder (serial C) or feedback cable. Replace the cable or pulse coder. (For details on the alarm, refer to the "FANUC AC Servo Amplifier Maintenance Manual.")	Axis type
SV 114	ABNORMAL REV. DATA (SERIAL A)	The serial pulse coder (serial A) is abnormal. Replace it.	Axis type
SV 115	ABNORMAL COMMUNICATION (SPLC)	A serial pulse coder communication failure occurred. Replace the pulse coder, feedback cable, or CNC axis board.	Axis type
SV 116	MCC WELDING ARARM	The contact of the magnetic contractor (MCC) in the servo amplifier has melted. For details, refer to the "FANUC AC Servo Amplifier Maintenance Manual."	Axis type
SV 117	ABNORMAL CURRENT OFFSET	Digital servo current conversion is abnormal. Replace the CNC axis board.	Axis type
SV 118	DETECT ABNORMAL TORQUE	An abnormal servo motor load was detected. Reset to clear the alarm.	Axis
SV 125	EXCESS VELOCITY IN TORQUE	In torque control, the specified maximum allowable velocity was exceeded.	
SV 126	EXCESS ERROR IN TORQUE	In torque control, the maximum allowable accumulated movement, specified with a parameter, was exceeded.	
SV 323	DISCONNECT ALARM BY SOFT	Disconnection of the serial pulse coder was detected. (Software)	
SV 324	DISCONNECT ALARM BY HARD	Disconnection of the serial pulse coder was detected. (Hardware)	
SV 325	PULSE CODER ALARM1 (SPLC)	An abnormality was detected in the serial pulse coder or feedback cable. (Built–in unit)	
SV 326	PULSE CODER ALARM3 (SPLC)	An abnormality was detected in the serial pulse coder or feedback cable. (Built–in unit)	

Number	Message displayed on CRT	Contents	Remarks
SV 327	ABNORMAL REV. DATA (SPLC)	The serial pulse coder is abnormal. Replace the pulse coder. (Built–in unit)	
SV 330	ABNORMAL COMMUNICATION (SPLC)	A serial pulse coder communication error occurred. (Built–in unit)	
SV 331	PULSE CODER ALARM2 (SPLC)	An abnormality was detected in the serial pulse coder or feedback cable. (Built–in unit)	
SV 333	PULSE CODER ALARM4 (EX–LIN)	An abnormality was detected in the serial pulse coder or feedback cable. (Separate linear)	
SV 334	PULSE CODER ALARM5 (EX–ROT)	An abnormality was detected in the serial pulse coder or feedback cable. (Separate rotary)	
SV 335	ABNORMAL REV. DATA (EX–ROT)	The serial pulse coder is abnormal.	
SV 336	BATTERY ZERO (EX)	The serial pulse coder battery voltage is 0 V. (Separate unit)	
SV 338	DETECTOR OVER LOAD (EX)	The serial pulse coder is overloaded. (Separate unit)	
SV 339	ABNORMAL COMMUNICATION (EX)	A communication error occurred in the serial pulse coder. (Separate unit)	
SV 340	PULSE CODER DISCONNECT (EX)	Disconnection of the serial pulse coder was detected. (Separate unit)	

Number	Message displayed on CRT	Contents	Remarks
IO 030	CHECK SUM ERROR	The check sum for a page of NC memory is incorrect.	
IO 031	INVALID CODE	An invalid code was read from NC memory.	
IO 032	OUT OF RANGE MEMORY ACCESS	An attempt was made to read data from or write data to an address outside of bubble memory.	

Number	Message displayed on CRT	Contents	Remarks
PW 000	POWER MUST BE OFF	A parameter was set for which the power must be turned off then on again.	
PW 100	ILLEGAL PARAMETER	The parameter for setting straightness compensation or the parameter for setting inclination compensation is specified incorrectly.	

INDEX

S

T

U

V

W

Revision Record

FANUC Series 15/150–MODEL B For Machining Center PROGRAMMING OPERATOR'S MANUAL (B–62564E)

Revision	Date	Contents	Revision	Date	Contents
02	Jan., '97	• Addition of new functions • Correction of errors			
01	Aug., '94				