MITSUBISHI ELECTRIC

MITSUBISHI CNC

EZMotion-NC E60/E68 Series

PROGRAMMING MANUAL
(MACHINING CENTER SYSTEM)

Introduction

This manual is a guide for using the EZMotion-NC E60/E68.
Programming is described in this manual, so read this manual thoroughly before starting programming.   Thoroughly study the "Precautions for Safety" on the following page to ensure safe use of the NC unit.

Details described in this manual

⚠ CAUTION

⚠ For items described in "Restrictions" or "Usable State", the instruction manual issued by the machine tool builder takes precedence over this manual.

⚠ An effort has been made to note as many special handling methods in this user's manual. Items not described in this manual must be interpreted as "not possible".

⚠ This manual has been written on the assumption that all option functions are added.
Refer to the specifications issued by the machine tool builder before starting use.

⚠ Refer to the Instruction Manual issued by each machine tool builder for details on each machine tool.

⚠ Some screens and functions may differ depending on the NC system or its version, and some functions may not be possible. Please confirm the specifications before use.

General precautions

(1)   Refer to the following documents for details on handling

Operation Manual IB-1500172

Precautions for Safety

Always read the specifications issued by the machine tool builder, this manual, related manuals and attached documents before installation, operation, programming, maintenance or inspection to ensure correct use.
Understand this numerical controller, safety items and cautions before using the unit.
This manual ranks the safety precautions into "DANGER", "WARNING" and "CAUTION".

⚠ **DANGER**	When the user may be subject to imminent fatalities or major injuries if handling is mistaken.
⚠ **WARNING**	When the user may be subject to fatalities or major injuries if handling is mistaken.
⚠ **CAUTION**	When the user may be subject to injuries or when physical damage may occur if handling is mistaken.

Note that even items ranked as " ⚠ **CAUTION**", may lead to major results depending on the situation. In any case, important information that must always be observed is described.

⚠ DANGER

Not applicable in this manual.

⚠ WARNING

Not applicable in this manual.

⚠ CAUTION

1. Items related to product and manual

⚠ For items described as "Restrictions" or "Usable State" in this manual, the instruction manual issued by the machine tool builder takes precedence over this manual.

⚠ An effort has been made to describe special handling of this machine, but items that are not described must be interpreted as "not possible".

⚠ This manual is written on the assumption that all option functions are added. Refer to the specifications issued by the machine tool builder before starting use.

⚠ Refer to the Instruction Manual issued by each machine tool builder for details on each machine tool.

⚠ Some screens and functions may differ depending on the NC system or its version, and some functions may not be possible. Please confirm the specifications before use.

2. Items related to operation

⚠ Before starting actual machining, always carry out dry operation to confirm the machining program, tool offset amount and workpiece offset amount, etc.
If the workpiece coordinate system offset amount is changed during single block stop, the new setting will be valid from the next block.

⚠ **CAUTION**

⚠ Turn the mirror image ON and OFF at the mirror image center.

⚠ If the tool offset amount is changed during automatic operation (including during single block stop), it will be validated from the next block or blocks onwards.

⚠ Do not make the synchronous spindle rotation command OFF with one workpiece chucked by the basic spindle and synchronous spindle during the spindle synchronous control mode.
Failure to observe this may cause the synchronous spindle stop, and hazardous situation.

3. Items related to programming

⚠ The commands with "no value after G" will be handled as "G00".

⚠ " ; ", "EOB", "%", and "EOR" are symbols used for explanation. The actual codes for ISO are "CR, LF" ("LF") and "%".
The programs created on the Edit screen are stored in the NC memory in a "CR, LF" format, however, the programs created with external devices such as the FLD or RS-232C may be stored in an "LF" format.

⚠ The actual codes for EIA are "EOB (End of Block)" and "EOR (End of Record)".
When creating the machining program, select the appropriate machining conditions, and make sure that the performance, capacity and limits of the machine and NC are not

⊘ exceeded. The examples do not consider the machining conditions.
Do not change fixed cycle programs without the prior approval of the machine tool builder.

Contents

1. Control Axes

1.1 Coordinate word and control axis

Function and purpose

In the standard specifications, there are 3 control axes. When using E68, up to 4 axes can be controlled.

The designation of the processing direction responds to those axes and uses a coordinate word made up of alphabet characters that have been decided beforehand.

X-Y table

Program coordinates

Workpiece

X-Y table

Direction of table movement   Bed   Direction of table movement

X-Y and revolving table [E68]

Workpiece

Direction of table movement

Direction of table revolution

Program coordinates

1
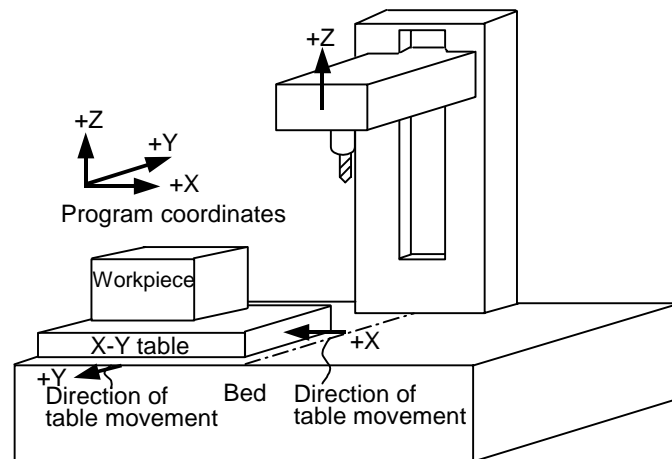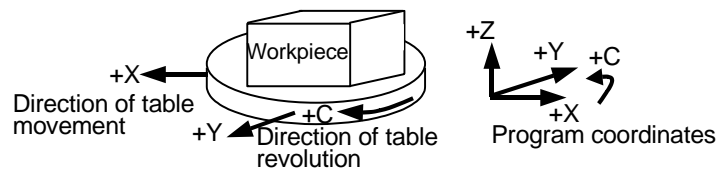
1.2 Coordinate systems and coordinate zero point symbols

Function and purpose
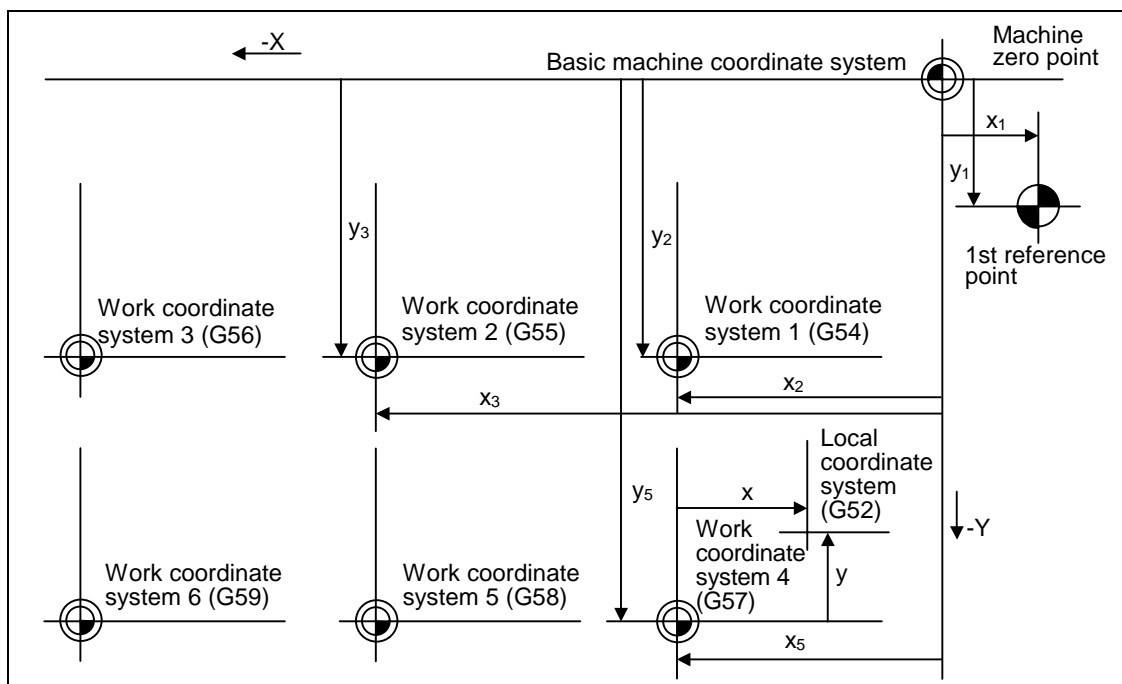
2. Input Command Units

2.1 Input command units

Function and purpose

These are the units used for the movement amounts in the program as commanded by the MDI input and command tape. They are expressed in millimeters, inches or degrees (°).

2.2 Input setting units

Function and purpose

These are the units of setting data which are used, as with the compensation amounts, in common for all axes.
The input setting units and input command units can be selected from the following types for each axis with the parameters. (For further details on settings, refer to the Operation Manual.)

(1) When using E60

	Input unit parameters	Linear axis		Rotary axis (°)
		Millimeter	Inch	
Input command unit	#1015 cunit　= 10	0.001	0.0001	0.001
Input setting unit	#1003 iunit　= B	0.001	0.0001	0.001

(2) When using E68

	Input unit parameters	Linear axis		Rotary axis (°)
		Millimeter	Inch	
Input command unit	#1015 cunit　= 10	0.001	0.0001	0.001
	#1015 cunit　= 1	0.0001	0.00001	0.0001
Input command unit	#1003 iunit　= B	0.001	0.0001	0.001
	#1003 iunit　= C	0.0001	0.00001	0.0001

(Note 1) Inch/metric conversion is performed in either of 2 ways: conversion from the parameter screen ("#1041 I_inch: valid only when the power is switched on) and conversion using the G command (G20 or G21).
However, when a G command is used for the conversion, the conversion applies only to the input command units and not to the input setting units.
Consequently, the tool offset amounts and other compensation amounts as well as the variable data should be preset to correspond to inches or millimeters.

(Note 2) The millimeter and inch systems cannot be used together.

(Note 3) During circular interpolation on an axis where the input command units are different, the center command (I, J, K) and the radius command (R) can be designated by the input setting units. (Use a decimal point to avoid confusion.)

3. Data Formats

3.1 Tape codes

Function and purpose

The tape command codes used for this controller are combinations of alphabet letters (A, B, C, ... Z), numbers (0, 1, 2 ... 9) and signs (+, -, / ...). These alphabet letters, numbers and signs are referred to as characters. Each character is represented by a combination of 8 holes which may, or may not, be present.
These combinations make up what is called codes.
This controller uses, the ISO code (R-840).

(Note 1) If a code not given in the tape code table in Fig. 1 is assigned during operation, program error (P32) will result.

(Note 2) For the sake of convenience, a semicolon " ; " has been used in the CNC display to indicate the end of a block (EOB/IF) which separates one block from another. Do not use the semicolon key, however, in actual programming but use the keys in the following table instead.

⚠ CAUTION

⚠ " ; ", "EOB", "%", and "EOR" are symbols used for explanation. The actual codes for ISO are "CR, LF" ("LF") and "%".

The programs created on the Edit screen are stored in the NC memory in a "CR, LF" format, however, the programs created with external devices such as the FLD or RS-232C may be stored in an "LF" format.

The actual codes for EIA are "EOB (End of Block)" and "EOR (End of Record)".

Detailed description

EOB/EOR keys and displays

Code used / Key used	ISO	Screen display
End of block	LF or NL	;
End of record	%	%

(1) Significant data section (label skip function)

All data up to the first EOB (;), after the power has been turned on or after operation has been reset, are ignored during automatic operation based on tape, memory loading operation or during a search operation. In other words, the significant data section of a tape extends from the character or number code after the initial EOB (;) code after resetting to the point where the reset command is issued.

3. Data Formats

(2) Control out, control in

When the ISO code is used, all data between control out "(" and control in ")" are ignored, although these data appear on the setting and display unit. Consequently, the command tape name, number and other such data not directly related to control can be inserted in this section.

This information (except (B) in the tape codes) will also be loaded, however, during tape loading. The system is set to the "control in" mode when the power is witched on.

Example of ISO code

$^{L}_{F}{}^{C}_{R}$G00X-85000Y-64000(CUTTER$^{S}_{P}$RETURN)$^{L}_{F}$
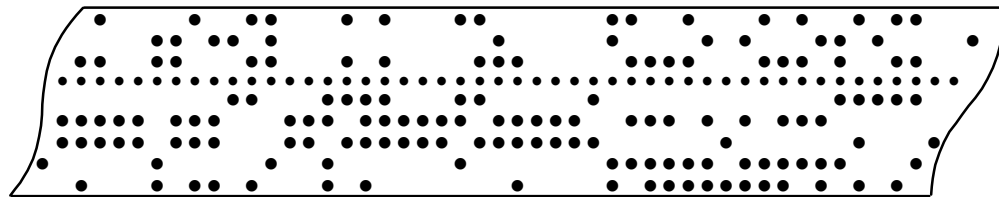
Operator information print-out example

Information in this section is ignored and nothing is executed.

---

(3) EOR (%) code

Generally, the end-or-record code is punched at both ends of the tape. It has the following functions:

(a) Rewind stop when rewinding tape (with tape handler)
(b) Rewind start during tape search (with tape handler)
(c) Completion of loading during tape loading into memory

(4) Tape preparation for tape operation (with tape handler)
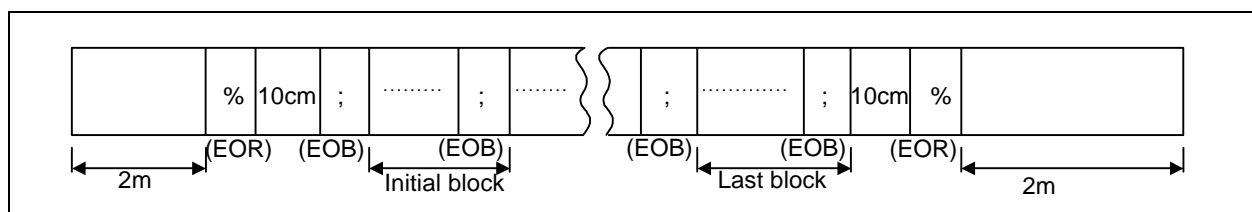
If a tape handler is not used, there is no need for the 2-meter dummy at both ends of the tape and for the head EOR (%) code.

ISO code (R-840)

Feed holes

8 7 6 5 4 | 3 2 1 ← Channel No.

1
2
3
4
5
6
7
8
9
0
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z
+
-
.
,
/
%
LF(Line Feed) or NL
((Control Out)
) (Control In)
:
#
*
=
[
]
SP(Space)
CR(Carriage Return)
BS(Back Space)
HT(Horizontal Tab)
&
!
$
' (Apostrophe)
;
<
>
?
@
"
DEL(Delete)
NULL
DEL(Delete)

A

B

- Under the ISO code, IF or NL is EOB and % is EOR.
- Under the ISO code, CR is meaningless, and EOB will not occur.

Code A are stored on tape but an error results (except when they are used in the comment section) during operation.

The B codes are non-working codes and are always ignored. Parity V check is not executed.

Table of tape codes

6

3.2 Program formats

Function and purpose

The prescribed arrangement used when assigning control information to the controller is known as the program format, and the format used with this controller is called the "word address format".

Detailed description

(1) Word and address

A word is a collection of characters arranged in a specific sequence. This entity is used as the unit for processing data and for causing the machine to execute specific operations. Each word used for this controller consists of an alphabet letter and a number of several digits (sometimes with a "-" sign placed at the head of the number.).

Word configuration
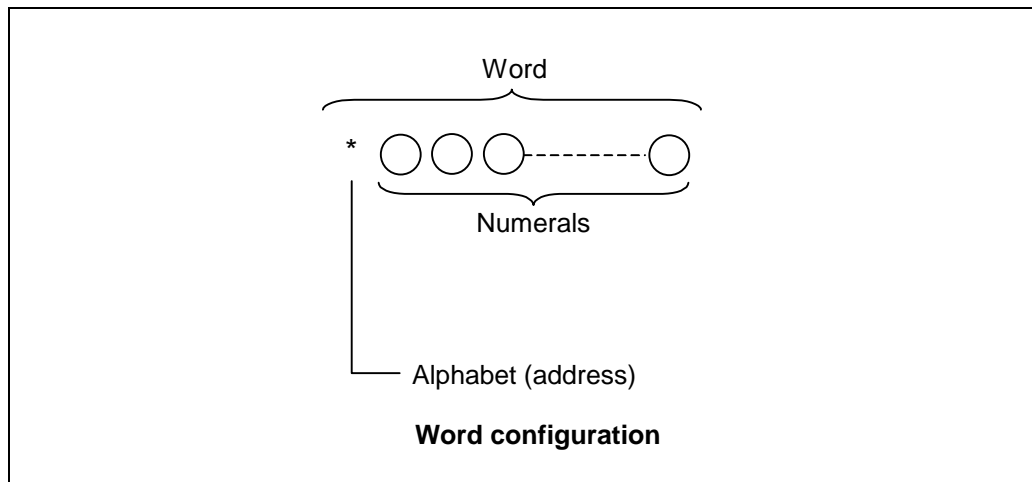
The alphabet letter at the head of the word is the address. It defines the meaning of the numerical information which follows it.

For details of the types of words and the number of significant digits of words used for this controller, refer to containing the "format details".

(2) Blocks

A block is a collection of words. It includes the information which is required for the machine to execute specific operations. One block unit constitutes a complete command. The end of each block is marked with an EOB (end-of-block) code.

(Example 1:)

G0X - 1000 ;
G1X - 2000F500 ; } 2 blocks

(Example 2:)

(G0X - 1000 ;)
G1X - 2000F500 ; } Since the semicolon in the parentheses will not result in an EOB, it is 1 block.

(3) Programs

A program is a collection of several blocks.

3. Data Formats

		Metric command	Inch command	Rotary axis (Metric command)	Rotary axis (Inch command)
Program number		08	←	←	←
Sequence number		N5	←	←	←
Preparatory function		G3/G21	←	←	←
Movement axis	0.001(°) mm/ 0.0001 inch	X+53 Y+53 Z+53 α+53	X+44 Y+44 Z+44 α+44	X+53 Y+53 Z+53 α+53	X+53 Y+53 Z+53 α+53
	0.0001(°) mm/ 0.00001 inch	X+44 Y+44 Z+44 α+44	X+35 Y+35 Z+35 α+35	X+44 Y+44 Z+44 α+44	X+44 Y+44 Z+44 α+44
Arc and cutter radius	0.001(°) mm/ 0.0001 inch	I+53 J+53 K+53 R+53	I+44 J+44 K+44 R+44	I+53 J+53 K+53 R+53	I+44 J+44 K+44 R+44 (Note 5)
	0.0001(°) mm/ 0.00001 inch	I+44 J+44 K+44 R+44	I+35 J+35 K+35 R+35	I+44 J+44 K+44 R+44	I+35 J+35 K+35 R+35 (Note 5)
Feed function	0.001(°) mm/ 0.0001 inch	F63(Feed per minute) F43(Feed per revolution)	F44(Feed per minute) F34(Feed per revolution)	F63(Feed per minute) F43(Feed per revolution)	F44(Feed per minute) F34(Feed per revolution) (Note 6)
	0.0001 (°) mm/ 0.00001 inch	F54(Feed per minute) F34(Feed per revolution)	F35(Feed per minute) F25(Feed per revolution)	F54(Feed per minute) F34(Feed per revolution)	F35(Feed per minute) F25(Feed per revolution) (Note 6)
Dwell		X+53/P+53 or X+53/P+8	←	←	←
Tool offset		H3 D3	←	←	←
Miscellaneous function (M)		M8	←	←	←
Spindle function (S)		S8	←	←	←
Tool function (T)		T8	←	←	←
2nd miscellaneous function		A8/B8/C8	←	←	←
Subprogram		P8 H5 L4	←	←	←
Fixed cycle	0.001(°) mm/ 0.0001 inch	R+53 Q53 P8 L4	←	←	←
	0.0001(°) mm/ 0.00001 inch	R+44 Q44 P8 L4	←	←	←

[Format details]

(Note 1) α indicates the additional axis address, such as A, B or C.

(Note 2) The No. of digits check for a word is carried out with the maximum number of digits of that address.

(Note 3) Numerals can be used without the leading zeros.

(Note 4) The meanings of the details are as follows :
Example 1 : 08   : 8-digit program number
Example 2 : G21   : Dimension G is 2 digits to the left of the decimal point, and 1 digit to the right.
Example 3 : X+53 : Dimension X uses + or - sign and represents 5 digits to the left of the decimal point and 3 digits to the right.
For example, the case for when the X axis is positioned (G00) to the 45.123 mm position in the absolute value (G90) mode is as follows :

G00 X45.123 ;
— 3 digits below the decimal point
— 5 digits above the decimal point, so it's +00045, but the leading zeros and the mark (+) have been omitted.
— G0 is possible.

(Note 5) If an arc is commanded using a rotary axis and linear axis while inch commands are being used, the degrees will be converted into 0.1 inches for interpolation.

(Note 6) While inch commands are being used, the rotary axis speed will be in increments of 10 degrees.
Example: With the F1. (per-minute-feed) command, this will become the 10 degrees/minute command.

(Note 7) The decimal places below the decimal point are ignored when a command, such as an S command, with an invalid decimal point has been assigned with a decimal point.

(Note 8)　This format is the same for the value input from the memory, MDI or setting and display unit.

(Note 9)　Command the program No. in an independent block. Command the program No. in the head block of the program.

(Note 10)　Minimum input setting unit "C" (0.0001 (°) mm/0.00001 inch) is a specification of the E68 system. When using E60, the unit cannot be used.

3.3 Program address check function

Function and purpose

The program can be checked in word units when operating machining programs.

Detailed description

(1) Address check

This function enables simple checking of program addresses in word units. If the alphabetic characters are continuous, the program error (P32) will occur. Availability of this function is selected by the parameter "#1227 aux11/bit4".
Note that an error will not occur for the following:
(1) Reserved words
(2) Comment statements

(2) Word range check

When the operation expression is used for the word data, check whether one word data is enclosed in brackets "[ ]" or not.
If it is not enclosed, the program error (P33) will occur.
Availability of this function is selected by the parameter "#1274 ext10/bit7".

Example of program

(1) Example of program for address check

(Example 1) When there are no numbers following an alphabetic character.
G28X; → An error will occur. Change to "G28 X0;", etc.

(Example 2) When a character string is illegal.
TEST; → An error will occur. Change to "(TEST);", etc.

(2) Example of program for word range check

An error (P33) will occur.		Change as shown below.
X-100.-#24	→	X-[100.+#24]
Z+#500*TAN[#1]		Z+[#500*TAN[#1]]
Y-SIN[60.]		Y-SIN[60.]

The numbers of brackets that can be used for one word data are maximum five.
If six brackets or more are commanded, the program error (P280) will occur.

3.4 Tape memory format

Function and purpose

(1) Storage tape and significant sections

The others are about from the current tape position to the EOB. Accordingly, under normal conditions, operate the tape memory after resetting.

The significant codes listed in "Table of tape codes" in "3.1 Tape Codes" in the above significant section are actually stored into the memory. All other codes are ignored and are not stored.

The data between control out "(" and control in ")" are stored into the memory.

3.5 Optional block skip ; /

Function and purpose

This function selectively ignores specific blocks in a machining program which starts with the "/" (slash) code.

Detailed description

(1) Provided that the optional block skip switch is ON, blocks starting with the "/" code are ignored. They are executed if the switch is OFF.

Parity check is valid regardless of whether the optional block skip switch is ON or OFF.

When, for instance, all blocks are to be executed for one workpiece but specific block are not to be executed for another workpiece, the same command tape can be used to machine different parts by inserting the "/" code at the head of those specific blocks.

Precautions for using optional block skip

(1) Put the "/" code for optional block skip at the beginning of a block. If it is placed inside the block, it is assumed as a user macro, a division instruction.

Example : N20 G1 X25./Y25. ;..... NG (User macro, a division instruction; a program error results.)
/N20 G1 X25. Y25. ; .... OK

(2) Parity checks (H and V) are conducted regardless of the optional block skip switch position.

(3) The optional block skip is processed immediately before the pre-read buffer.
Consequently, it is not possible to skip up to the block which has been read into the pre-read buffer.

(4) This function is valid even during a sequence number search.

(5) All blocks with the "/" code are also input and output during tape storing and tape output, regardless of the position of the optional block skip switch.

3.6 Program/sequence/block numbers ; O, N
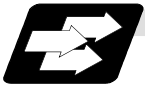
Function and purpose

These numbers are used for monitoring the execution of the machining programs and for calling both machining programs and specific stages in machining programs.

(1) Program numbers are classified by workpiece correspondence or by subprogram units, and they are designated by the address "0" followed by a number with up to 8 digits.

(2) Sequence numbers are attached where appropriate to command blocks which configure machining programs, and they are designated by the address "N" followed by a number with up to 5 digits.

(3) Block numbers are automatically provided internally. They are preset to zero every time a program number or sequence number is read, and they are counted up one at a time unless program numbers or sequence numbers are commanded in blocks which are subsequently read.

Consequently, all the blocks of the machining programs given in the table below can be determined without further consideration by combinations of program numbers, sequence numbers and block numbers.

Machining program	Monitor display		
	Program No.	Sequence No.	Block No.
O12345678 (DEMO, PROG) ;	12345678	0	0
G92 X0 Y0 ;	12345678	0	1
G90 G51 X-150. P0.75 ;	12345678	0	2
N100 G00 X-50. Y-25. ;	12345678	100	0
N110 G01 X250. F300 ;	12345678	110	0
Y-225. ;	12345678	110	1
X-50. ;	12345678	110	2
Y-25.;	12345678	110	3
N120 G51 Y-125. P0.5 ;	12345678	120	0
N130 G00 X-100. Y-75. ;	12345678	130	0
N140 G01 X-200. ;	12345678	140	0
Y-175. ;	12345678	140	1
X-100. ;	12345678	140	2
Y-75. ;	12345678	140	3
N150 G00 G50 X0 Y0 ;	12345678	150	0
N160 M02 ;	12345678	160	0
%			

3.7 Parity H/V
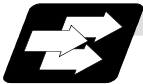
Function and purpose

Parity check provides a mean of checking whether the tape has been correctly perforated or not. This involves checking for perforated code errors or, in other words, for perforation errors. There are two types of parity check: Parity H and Parity V.

(1)　Parity H

Parity H checks the number of holes configuring a character and it is done during tape operation, tape input and sequence number search.
A parity H error is caused in the following cases.

(a)　ISO code

When a code with an odd number of holes in a significant data section has been detected.

Parity H error example

This character causes a parity H error.
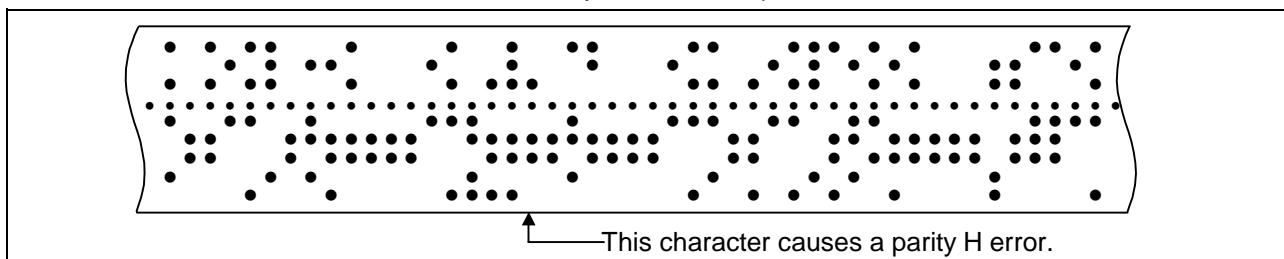
When a parity H error occurs, the tape stops following the alarm code.

(2)　Parity V

A parity V check is done during tape operation, tape input and sequence number search when the I/O PARA #9n15 (n is the unit No.1 to 5) parity V check function is set to "1". It is not done during memory operation.
A parity V error occurs in the following case: when the number of codes from the first significant code to the EOB (;) in the significant data section in the vertical direction of the tape is an odd number, that is, when the number of characters in one block is odd.
When a parity V error is detected, the tape stops at the code following the EOB (;).

(Note 1) Among the tape codes, there are codes which are counted as characters for parity and codes which are not counted as such. For details, refer to the "Table of tape codes" in "3.1 Tape Codes".

(Note 2) Any space codes which may appear within the section from the initial EOB code to the address code or "/" code are counted for parity V check.

3.8 G code lists
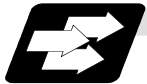
Function and purpose

G code	Group	Function	E60	E68
Δ 00	01	Positioning	○	○
Δ 01	01	Linear interpolation	○	○
02	01	Circular interpolation CW (clockwise)	○	○
03	01	Circular interpolation CCW (counterclockwise)	○	○
02.3				
03.3				
04	00	Dwell	○	○
05	00	High-speed machining mode III	--	○
05.1				
06				
07				
07.1 107	21	Cylindrical interpolation	--	○
08	00	High-accuracy control	--	○
09	00	Exact stop check	○	○
10	00	Program data input (parameter /compensation data/parameter coordinate rotation data)	○	○
11	00	Program parameter input cancel	○	○
12	00	Circular cut CW (clockwise)	○	○
13	00	Circular cut CCW (counterclockwise)	○	○
14				
* 15	18	Polar coordinate command OFF	--	○
16	18	Polar coordinate command ON	--	○
Δ 17	02	Plane selection X-Y	○	○
Δ 18	02	Plane selection Z-X	○	○
Δ 19	02	Plane selection Y-Z	○	○
Δ 20	06	Inch command	○	○
Δ 21	06	Metric command	○	○
22				
23				
24				
25				
26				
27	00	Reference point check	○	○
28	00	Reference point return	○	○
29	00	Start point return	○	○
30	00	2nd to 4th reference point return	○	○
30.1	00	Tool change position return 1	○	○
30.2	00	Tool change position return 2	○	○
30.3	00	Tool change position return 3	○	○
30.4	00	Tool change position return 4	○	○
30.5	00	Tool change position return 5	○	○
30.6	00	Tool change position return 6	○	○
31	00	Skip / Multi-skip function2	○	○

G code	Group	Function	E60	E68
31.1	00	Multi-skip function 1-1	○	○
31.2	00	Multi-skip function 1-2	○	○
31.3	00	Multi-skip function 1-3	○	○
32				
33	01	Thread cutting	○	○
34	00	Special fixed cycle (bolt hole circle)	○	○
35	00	Special fixed cycle (line at angle)	○	○
36	00	Special fixed cycle (arc)	○	○
37	00	Automatic tool length measurement	○	○
37.1	00	Special fixed cycle (grid)	○	○
38	00	Tool radius compensation vector designation	○	○
39	00	Tool radius compensation corner arc	○	○
* 40	07	Tool radius compensation cancel	○	○
41	07	Tool radius compensation left	○	○
42	07	Tool radius compensation right	○	○
40.1				
41.1				
42.1				
43	08	Tool length offset (+)	○	○
44	08	Tool length offset (-)	○	○
43.1	20	1st spindle control ON	--	○
44.1	20	2nd spindle control ON	--	○
47.1	20	Two-spindles simultaneous control ON	--	○
45	00	Tool position offset (expansion)	--	○
46	00	Tool position offset (reduction)	--	○
47	00	Tool position offset (doubled)	--	○
48	00	Tool position offset (halved)	--	○
* 49	08	Tool length offset cancel	○	○
* 50	11	Scaling cancel	○	○
51	11	Scaling ON	○	○
* 50.1	19	G command mirror image cancel	○	○
51.1	19	G command mirror image ON	○	○
52	00	Local coordinate system setting	○	○
53	00	Machine coordinate system selection	○	○
* 54	12	Work coordinate system 1 selection	○	○
55	12	Work coordinate system 2 selection	○	○
56	12	Work coordinate system 3 selection	○	○
57	12	Work coordinate system 4 selection	○	○
58	12	Work coordinate system 5 selection	○	○
59	12	Work coordinate system 6 selection	○	○
54.1	12	Work coordinate system selection 48 sets expanded	○	○
60	00	Uni-directional positioning	○	○
61	13	Exact stop check mode	○	○
61.1	13	High-accuracy control	--	○
62	13	Automatic corner override	○	○
63	13	Tapping mode	○	○
63.1				
63.2				
* 64	13	Cutting mode	○	○
65	00	User macro call	○	○
66	14	User macro modal call A	○	○
66.1	14	User macro modal call B	○	○

G code	Group	Function	E60	E68
* 67	14	User macro modal call cancel	○	○
68	16	Coordinate conversion mode ON	○	○
* 69	16	Coordinate conversion mode OFF	○	○
70	09	User fixed cycle	○	○
71	09	User fixed cycle	○	○
72	09	User fixed cycle	○	○
73	09	Fixed cycle (step)	○	○
74	09	Fixed cycle (reverse tap)	○	○
		Synchronous reverse tapping (pecking/deep-hole)	--	○
75	09	User fixed cycle	○	○
76	09	Fixed cycle (fine boring)	○	○
77	09	User fixed cycle	○	○
78	09	User fixed cycle	○	○
79	09	User fixed cycle	○	○
* 80	09	Fixed cycle cancel	○	○
81	09	Fixed cycle (drill/spot drill)	○	○
82	09	Fixed cycle (drill/counter boring)	○	○
83	09	Fixed cycle (deep drilling)	○	○
84	09	Fixed cycle (tapping)	○	○
		Synchronous tapping (pecking/deep-hole)	--	○
85	09	Fixed cycle (boring)	○	○
86	09	Fixed cycle (boring)	○	○
87	09	Fixed cycle (back boring)	○	○
88	09	Fixed cycle (boring)	○	○
89	09	Fixed cycle (boring)	○	○
Δ 90	03	Absolute value command	○	○
Δ 91	03	Incremental command value	○	○
92	00	Machine coordinate system setting	○	○
93				
Δ 94	05	Asynchronous feed (per-minute feed)	○	○
Δ 95	05	Synchronous feed (per-revolution feed)	○	○
Δ 96	17	Constant surface speed control ON	○	○
Δ 97	17	Constant surface speed control OFF	○	○
* 98	10	Fixed cycle Initial level return	○	○
99	10	Fixed cycle R point level return	○	○
100 to 255	00	User macro (G code call) Max. 10	○	○

(Note 1) Codes marked with * are codes that must be or are selected in the initial state.
The codes marked with Δ are codes that should be or are selected in the initial state by the parameters.

(Note 2) If two or more G codes from the same code are commanded, the latter G code will be valid.

(Note 3) This G code list is a list of conventional G codes. Depending on the machine, movements that differ from the conventional G commands may be included when called by the G code macro. Refer to the Instruction Manual issued by the machine tool builder.

(Note 4) Whether the modal is initialized or not depends on each reset input.
  (1) "Reset 1"
    The modal is initialized when the reset initial parameter "#1151 rstinit" turns ON.
  (2) "Reset 2" and "Reset & rewind"
    The modal is initialized when the signal is input.
  (3) Resetting when emergency stop is canceled
    Follows "Reset 1".
  (4) When modal is automatically reset at the start of individual functions such as reference point return.
    Follows "Reset & rewind".

⚠ CAUTION

⚠ The commands without Nos. following "G" become a "G00" operation during running.

3.9 Precautions before starting machining

Precautions before starting machining

⚠ CAUTION

⚠ When creating the machining program, select the appropriate machining conditions so that the machine, NC performance, capacity and limits are not exceeded. The examples do not allow for the machining conditions.

⚠ Carry out dry operation before actually machining, and confirm the machining program, tool offset and workpiece offset amount.

4. Buffer Register

4.1 Input buffer

Function and purpose

When the pre-read buffer is empty during a tape operation or Rs-232C operation, the contents of the input buffer are immediately transferred to the pre-read buffers and, provided that the data stored in the input buffer do not exceed 250×4 characters, the following data (Max. 250 characters) are read and loaded into the input buffer.

This buffer is designed to eliminate the operational delay originating in the readout time of the tape reader and to smooth out the block joints.

The pre-reading effects are lost, however, when the block execution time is shorter than the tape readout time of the following block.
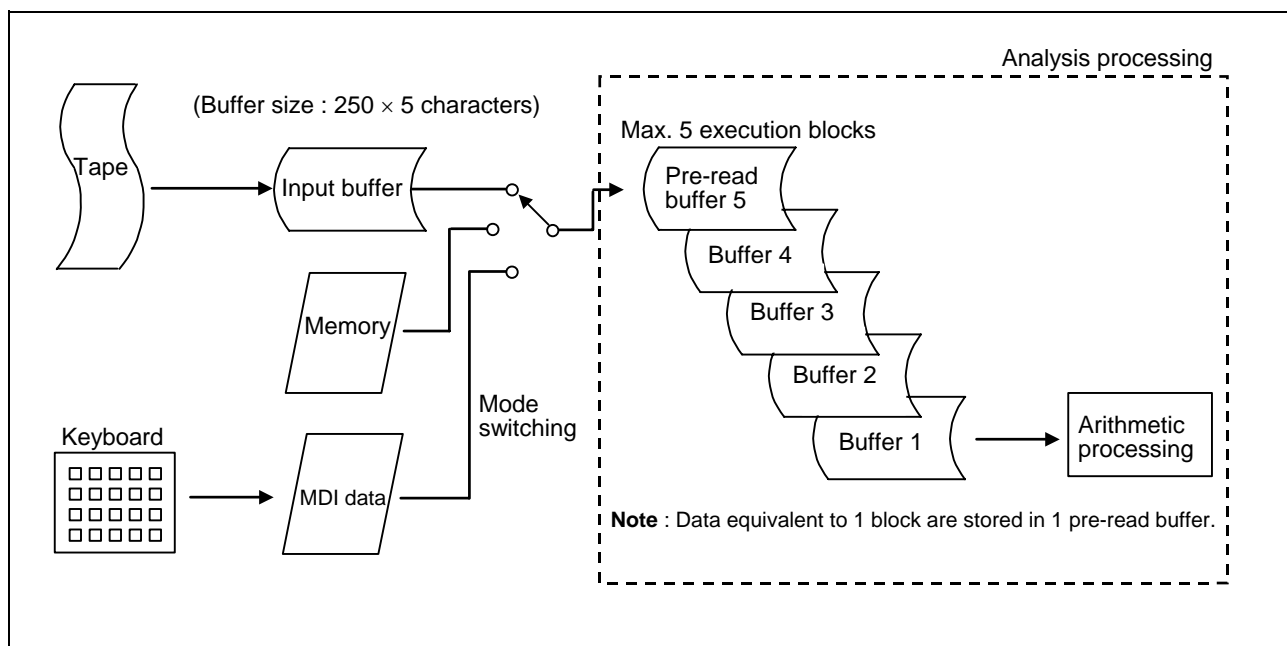
The input buffer has a memory capacity of 250×5 characters (including the EOB code).

(1)   The contents of the input buffer register are updated in 250-character units.
(2)   Only the significant codes in the significant data section are read into the input buffer.
(3)   When codes (including "(" and ")") are sandwiched in the control in or control out mode and the optional block skip function is ON, the data extending from the "/" (slash) code up to the EOB code are read into the input buffer.
(4)   The input buffer contents are cleared with resetting.

(Note 1) The input buffer size (250 characters) differs according to the model.

18

4.2 Pre-read buffers

Function and purpose

During automatic processing, the contents of 1 block are normally pre-read so that program analysis processing is conducted smoothly. However, during tool radius compensation, a maximum of 5 blocks are pre-read for the intersection point calculation including interference check.
The specifications of the data in 1 block are as follows:

(1) The data of 1 block are stored in this buffer.
(2) Only the significant codes in the significant data section are read into the pre-read buffer.
(3) When codes are sandwiched in the control in and control out, and the optional block skip function is ON, the data extending from the "/" (slash) code up to the EOB code are not read into the pre-read buffer.
(4) The pre-read buffer contents are cleared with resetting.
(5) When the single block function is ON during continuous operation, the pre-read buffer stores the following block data and then stops operation.

Other precautions

(1) Depending on whether the program is executed continuously or by single blocks, the timing of the valid/invalid for the external control signals for the block skip and others will differ.
(2) If the external control signal such as optional block skip is turned ON/OFF with the M command, the external control operation will not be effective on the program pre-read with the buffer register.
(3) According to the M command that operates the external controls, it prohibits pre-reading, and the recalculation is as follows:
   The M command that commands the external controls is distinguished at the PLC, and the "recalculation request" for PLC $\rightarrow$ NC interface table is turned ON.
   (When the "recalculation request" is ON, the program that has been pre-read is reprocessed.)

5. Position Commands

5.1 Position command methods ; G90, G91

Function and purpose

By using the G90 and G91 commands, it is possible to execute the next coordinate commands using absolute values or incremental values.
The R-designated circle radius and the center of the circle determined by I, J, K are always incremental value commands.

Command format

G9D Xx1 Yy1 Zz1	
G90	:Absolute value command
G91	:Incremental command

Detailed description

(1) Regardless of the current position, in the absolute value mode, it is possible to move to the position of the workpiece coordinate system that was designated in the program.

N 1 G90 G00 X0 Y0 ;

In the incremental value mode, the current position is the start point (0), and the movement is made only the value determined by the program, and is expressed as an incremental value.

N 2 G90 G01 X200. Y50. F100;

N 2 G91 G01 X200. Y50. F100;

Using the command from the 0 point in the workpiece coordinate system, it becomes the same coordinate command value in either the absolute value mode or the incremental value mode.
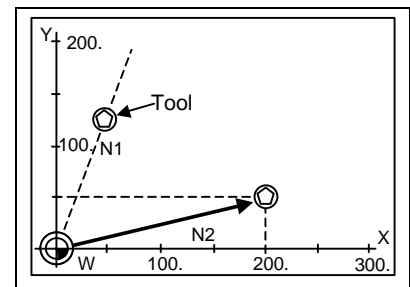
(2) For the next block, the last G90/G91 command that was given becomes the modal.

(G90)   N 3 X100. Y100.;

The axis moves to the workpiece coordinate system X = 100mm and Y = 100mm position.

(G91)   N 3 X–100. Y50.;

The X axis moves to -100.mm and the Y axis to +50.0mm as an incremental value, and as a result X moves to 100.mm and Y to 100.mm.

20

(3) Since multiple commands can be issued in the same block, it is possible to command specific addresses as either absolute values or incremental values.

N 4 G90 X300. G91 Y100.;

The X axis is treated in the absolute value mode, and with G90 is moved to the workpiece coordinate system 300.mm position. The Y axis is moved +100.mm with G91. As a result, Y moves to the 200.mm position. In terms of the next block, G91 remains as the modal and becomes the incremental value mode.
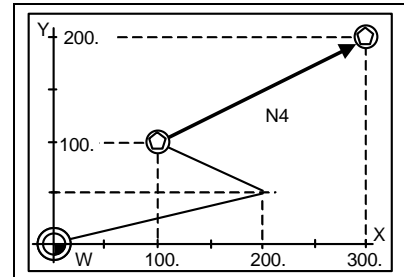
(4) When the power is turned ON, it is possible to select whether you want absolute value commands or incremental value commands with the #1073 I_Absm parameter.

(5) Even when commanding with the manual data input (MDI), it will be treated as a modal from that block.

5.2 Inch/metric command change; G20, G21

Function and purpose

These G commands are used to change between the inch and millimeter (metric) systems.

Command format

G20/G21;	
G20	: Inch command
G21	: Metric command

Detailed description

G20 and G21 selection is meaningful only for linear axes and it is meaningless for rotary axes.
The input unit for G20 and G21 will not change just by changing the command unit.
In other words, if the machining program command unit changes to an inch unit at G20 when the initial inch is OFF, the setting unit of the tool offset amount will remain metric. Thus, take note to the setting value.

(Example 1)   Relationship between input command units and G20/G21 commands
(with decimal point input type 1)

(1) When using E60

Axis	Input command unit type (cunit)	Command example	Metric output (#1016 iout=0)		Inch output (#1016 iout=1)	
			G21	G20	G21	G20
X	10	X100;	0.100mm	0.254mm	0.0039inch	0.0100inch
Y	10	Y100;	0.100mm	0.254mm	0.0039inch	0.0100inch
Z	10	Z100;	0.100mm	0.254mm	0.0039inch	0.0100inch
B	10	B100;	0.100°	0.100°	0.100°	0.100°

(2) When using E68

Axis	Input command unit type (cunit)	Command example	Metric output (#1016 iout=0)		Inch output (#1016 iout=1)	
			G21	G20	G21	G20
X	10	X100;	0.100mm	0.254mm	0.0039inch	0.0100inch
Y	10	Y100;	0.100mm	0.254mm	0.0039inch	0.0100inch
Z	1	Z100;	0.010mm	0.0254mm	0.00039inch	0.00100inch
B	10	B100;	0.100°	0.100°	0.100°	0.100°

(Note 1) When changing between G20 and G21 with program commands, it is necessary in advance, to convert the parameters, variables, and the offsets for the tool diameter, tool position, tool length, to the units in the input settings of the input setting unit system (for each axis) that have inch or metric commands, and make the settings using the parameter tape.

(Note 2) "cunit = 1" can be commanded when only using E68.

(Example 2)   Input setting unit #1015 cunit=10, #1041 I_inch=0
Position command unit 0.001mm
Compensation amount setting unit
..... When the compensation amount is 0.05mm for
0.001mm
In the above example, when changing from G21 to G20, the compensation amount
must be set to 0.002 (0.05 ÷ 25.4 ≒ 0.002).

(Note 3)   Since the data before the change will be executed at the command unit after the change,
command the F speed command for the change so that it is the correct speed command
for the command unit system applied after the change.

Output unit, command unit and setting unit

Select whether to issue the counter or parameter setting and display unit as a command unit determined by the G20/G21 command modal, or as the internal unit determined by parameter "#1041 I_inch". If the command unit setting display is selected when initial metric (internal unit metric) is set with the base specification parameter "#1226 aux10/bit6", the counter and parameters will be displayed as metric units during the G21 command mode. The internal unit metric data will be converted into an inch unit and displayed during the G20 command mode. The command unit for when the power is turned ON and reset can be selected regardless of the internal unit by setting the base specification parameter "#1152 I_G20".

NC axis

| Item | Initial inch OFF (metric internal unit) #1041 I_inch=0 | | | | Initial inch ON (inch internal unit) #1041 I_inch=1 | | | |
| | #1226 aux10/bit6=0 Follows internal unit | | #1226 aux10/bit6=1 Follows command unit | | #1226 aux10/bit6=0 Follows internal unit | | #1226 aux10/bit6=1 Follows internal unit | |
	G21	G20	G21	G20	G21	G20	G21	G20
Movement/ speed command	Metric	Inch	Metric	Inch	Metric	Inch	The "#1226 aux10/bit6" parameter is invalid when initial inch is ON. Even if 1 is set, the setting and display will follow the internal unit in the same manner as when 0 is set.	
Counter display	Metric	Metric	Metric	Inch	Inch	Inch		
Speed display	Metric	Metric	Metric	Inch	Inch	Inch		
User parameter setting/display	Metric	Metric	Metric	Inch	Inch	Inch		
Workpiece/ tool offset setting/display	Metric	Metric	Metric	Inch	Inch	Inch		
Handle feed command	Metric	Metric	Metric	Inch	Inch	Inch		
Setup parameter setting/display	Follows "#1040 M_inch"							

PLC axis

Item	#1042 pcinch=0 (metric)	#1042 pcinch=1 (inch)
Movement/ speed command	Metric	Inch
Counter display	Metric	Inch
User parameter setting/display	Metric	Inch
Setup parameter setting/display	Follows "#1040 M_inch"	

When using both the NC axis and PLC axis as rotation axes, three digits below the decimal point will be displayed for the coordinate data, etc., even for the inch command.

Usage examples

The screen display will change in the following manner according to the parameter settings.

Parameter setting Program command/ screen operation	#1041 I_inch 0 #1152 I_G20 0 #1226 aux10/bit6 0	Initial metric G21 (metric) mode at reset Setting and display unit is internal unit	#1041 I_inch 0 #1152 I_G20 0 #1226 aux10/bit6 1	Initial metric G21 (metric) mode at reset Setting and display unit is command unit

	[POSITION]
	X 0.000
	Y 0.000
	Z 0.000
	
	  mm.
	
	[TOOL OFFSET]
	#
	1 0.100 11 0.000
	2 0.200 12 0.000
	3 0.300 13 0.000
	  mm.
			[POSITION]
			X 0.000
			Y 0.000
			Z 0.000
			
			  mm.
			
			[TOOL OFFSET]
			#
			1 0.100 11 0.000
			2 0.200 12 0.000
			3 0.300 13 0.000
			  mm.	

Inch command
G20 G0 X1. Y2. Z3. ;
	[POSITION]
	X 25.400
	Y 50.800
	Z 76.200
	
	  in.
	
	[TOOL OFFSET]
	#
	1 0.100 11 0.000
	2 0.200 12 0.000
	3 0.300 13 0.000
	  in.
		Internal data
		25.4(mm)
		50.8(mm)
		76.2(mm)
			Display unit is also inch following the command unit
			
			[POSITION]
			X 1.0000
			Y 2.0000
			Z 3.0000
			
			  in.
			
			[TOOL OFFSET]
			#
			1 0.0039 11 0.0000
			2 0.0078 12 0.0000
			3 0.0118 13 0.0000
			  in.
				Internal data
				25.4(mm)
				50.8(mm)
				76.2(mm)
				
				Internal data
				#10.1(mm)

#(1)(1)
Input
	[TOOL OFFSET]
	#
	1 **1.000** 11 0.000
	⋮
	  in.
		Internal data
		1.0(mm)
			Set with inch unit
			
			[TOOL OFFSET]
			#
			1 **1.0000** 11 0.0000
			⋮
			  in.
				Internal data
				25.4(mm)

Program command/ screen operation ⟍ Parameter setting	#1041 I_inch 0 #1152 I_G20 0 #1226 aux10/bit6 0	Initial metric G21 (metric) mode at reset Setting and display unit is internal unit	#1041 I_inch 0 #1152 I_G20 0 #1226 aux10/bit6 1	Initial metric G21 (metric) mode at reset Setting and display unit is command unit
Reset			Display unit is also metric following the command unit	
	[POSITION] X 25.400 Y 50.800 Z 76.200 mm.		[POSITION] X 25.400 Y 50.800 Z 76.200 mm.	
	[TOOL OFFSET] # 1 1.000 11 0.000 2 0.200 12 0.000 3 0.300 13 0.000 mm.		[TOOL OFFSET] # 1 25.400 11 0.000 2 0.200 12 0.000 3 0.300 13 0.000 mm.	

Precautions

(1) If the tool and workpiece offset data are added with an inch command unit and metric internal unit, an error may occur.

(2) The internal unit is a unit that follows #1041 I_inch.

   Internal unit when #1041 I_inch=0 : mm
   Internal unit when #1041 I_inch=1 : inch

(3) The parameters are input and output with an internal unit.

(4) If the tool data input/output unit selection (#1220 aux04/bit4:1) is valid, the tool data will be input and output with the "#1152 I_G20" setting unit. However, in all other cases, the data will be input/output with the internal unit.

5.3 Decimal point input

Function and purpose

This function enables the decimal point command to be input. It assigns the decimal point in millimeter or inch units for the machining program input information that defines the tool paths, distances and speeds. A parameter #1078 Decpt2 selects whether type I (minimum input command unit) or type II (zero point) is to apply for the least significant digit of data without a decimal point.

Command format

○○○○○.○○○	: Metric command
○○○○.○○○○	: Inch command

Detailed description

(1) The decimal point command is valid for the distances, angles, times, speeds and scaling rate, in machining programs. (Note, only after G51).

(2) In decimal point input type 1 and type 2, the values of the data commands without the decimal points are shown in the table below.

(a) When using E60

Command	Command unit system	Type 1	Type 2
X1;	cunit= 10	1 (μm, 10^{-4}inch,$10^{-3\circ}$)	1 (mm, inch, °)

(b) When using E68

Command	Command unit system	Type 1	Type 2
X1;	cunit= 10	1	1
	cunit= 1	0.1 (μm, 10^{-4}inch,$10^{-3\circ}$)	1 (mm, inch, °)

(3) The valid addresses for the decimal points are X, Y, Z, U, V, W, A, B, C, I, J, K, E, F, P, Q, and R. However, P is valid only during scaling. For details, refer to the list.

(4) See below for the number of significant digits in decimal point commands. (Input command unit cunit = 10)

	Movement command (linear)		Movement command (rotary)		Feed rate		Dwell	
	Integer	**Decimal part**	**Integer**	**Decimal part**	**Integer**	**Decimal part**	**Integer**	**Decimal part**
MM (milli-meter)	0. to 99999.	.000 to .999	0. to 99999.	.000 to.999	0. to 60000.	.00 to .99	0. to 99999.	.000 to .999
INCH (inch)	0. to 9999.	.0000 to .9999	99999. (359.)	.0 to .999	0. to 2362.	.000 to .999	.0 to .99	.000 to .999

(5) The decimal point command is valid even for commands defining the variable data used in subprograms.

(6) While the smallest decimal point command is validated, the smallest unit for a command without a decimal point designation is the smallest command input unit set in the specifications (1μm, 10μm, etc.) or mm can be selected. This selection can be made with parameter #1078 Decpt2.

(7) Decimal point commands for decimal point invalid addresses are processed as integer data only and everything below the decimal point is ignored. Addresses which are invalid for the decimal point are D, H, L, M, N, O, S and T. All variable commands, however, are treated as data with decimal points.

5. Position Commands

Example of program

(1) Example of program for decimal point valid address

Specification division / Program example	Decimal point command 1		Decimal point command 2 1 = 1mm
	When 1 = 1μm	When 1 = 10μm	
G0X123.45 (decimal points are all mm points)	X123.450mm	X123.450mm	X123.450mm
G0X12345	X12.345mm (last digit is 1μm unit)	X123.450mm	X12345.000mm
#111 = 123, #112 = 5.55 X#111 Y#112	X123.000mm, Y5,5550mm	X123.000mm, Y5.550mm	X123.000mm, Y5.550mm
#113 = #111+#112 (addition)	#113 = 128.550	#113 = 128.550	#113 = 128.550
#114 = #111-#112 (subtraction)	#114 = 117.450	#114 = 117.450	#114 = 117.450
#115 = #111∗#112 (multiplication)	#115 = 682.650	#115 = 682.650	#115 = 682.650
#116 = #111/#112 #117 = #112/#111 (division)	#116 = 22.162 #117 = 0.045	#116 = 22.162 #117 = 0.045	#116 = 22.162 #117 = 0.045

Decimal point input I/II and decimal point command valid/invalid

If a command does not use a decimal point at an address where a decimal point command is valid in the table on the following page, it is handled differently between decimal point input I and II modes as explained below.
A command using a decimal point is handled the same way in either the decimal point input I or II mode.

(1) Decimal point input I

The least significant digit place of command data corresponds to the command unit.
(Example) Command "X1" in the 1μm system is equivalent to command "X0.001".

(2) Decimal point input II

The least significant digit place of command data corresponds to the decimal point.
(Example) Command "X1" in the 1μm system is equivalent to command "X1.".

(Note) When a four rules operator is contained, the data will be handled as that with a decimal point.

(Example) When the min. input command unit is 1μm :
　　　G0 x 123 + 0 ; ... X axis 123mm command. It will not be 123μm.

5. Position Commands

Addresses used and valid/invalid decimal point commands

Address	Decimal point command	Application	Remarks
A	Valid	Coordinate position data	
	Invalid	Revolving table, miscellaneous function code	
	Valid	Angle data	
	Invalid	Data settings, axis numbers (G10)	
B	Valid	Coordinate position data	
	Invalid	Revolving table, miscellaneous function code	
C	Valid	Coordinate position data	
	Invalid	Revolving table, miscellaneous function code	
	Valid	Corner chamfering amount	,c
D	Invalid	Offset numbers (tool position, tool radius)	
	Valid	Automatic tool length measurement, deceleration range d	
	Invalid	Data settings byte type data	
E	Valid	Inch thread, number of ridges Precision thread lead	
F	Valid	Feed rate	
	Valid	Thread lead	
	Valid	Automatic tool length measurement Measuring speed	
	Valid	Number of Z axis pitch in synchronous tap	
G	Valid	Preparatory function code	
H	Invalid	Tool length offset number	
	Invalid	Sequence numbers in subprograms	
	Invalid	Program parameter input, bit type data	
	Invalid	Linear-arc intersection selection (Geometric)	
I	Valid	Arc center coordinates	
	Valid	Tool radius compensation vector components	
	Valid	Hole pitch in the special fixed cycle	
	Valid	Circle radius of cut circle (increase amount)	
	Invalid	G0/G1 imposition width	,I
J	Valid	Arc center coordinates	
	Valid	Tool radius compensation vector components	
	Valid	Special fixed cycle's hole pitch or angle	
	Invalid	G0/G1 imposition width	,J
K	Valid	Arc center coordinates	
	Valid	Tool radius compensation vector components	
	Invalid	Number of holes of the special fixed cycle	
L	Invalid	Number of fixed cycle and subprogram repetitions	
	Invalid	Program tool compensation input type selection	L2, L12, L10, L13, L11
	Invalid	Programmable current limitation selection	L14
	Invalid	Program parameter input selection	L50
	Invalid	Program parameter input, 2-word type data	4 bytes
	Invalid	The tool life data	
M	Invalid	Miscellaneous function codes	

(Note 1) All decimal points are valid for the user macro arguments.

Address	Decimal point command	Application	Remarks
N	Invalid	Sequence numbers	
	Invalid	Program parameter input, data numbers	
O	Invalid	Program numbers	
P	Invalid/Valid	Dwell time	Parameter
	Invalid/Valid	Dwell at tap cycle hole base	Parameter
	Invalid	Subprogram program call No.	
	Invalid	Number of holes of the special fixed cycle	
	Invalid	Amount of helical pitch	
	Invalid	Offset number (G10)	
	Invalid	Constant surface speed control axis number	
	Invalid	Program parameter input, broad classification number	
	Invalid	Multi-step skip function 2 signal command	
	Invalid	Subprogram return destination sequence No.	
	Invalid	2nd, 3rd, 4th reference point return number	
	Valid	Scaling magnification	
	Invalid	Extended work coordinate system No.	
	Invalid	High-speed mode type	
	Invalid	The tool life data group No.	
Q	Valid	Cut amount of deep hole drill cycle	
	Valid	Shift amount of back boring	
	Valid	Shift amount of fine boring	
	Invalid	Minimum spindle clamp speed	
	Valid	Starting shift angle for screw cutting	
	Invalid	Tool life data management method	
R	Valid	R-point in the fixed cycle	
	Valid	R-specified arc radius	
	Valid	Corner rounding arc radius	,R
	Valid	Offset amount (G10)	
	Invalid	Synchronous tap/asynchronous tap changeover	
	Valid	Automatic tool length measurement, deceleration range r	
	Valid	Rotation angle	
S	Invalid	Spindle function codes	
	Invalid	Maximum spindle clamp speed	
	Invalid	Constant surface speed control, surface speed	
	Invalid	Program parameter input, word type data	2 bytes
T	Invalid	Tool function codes	
U	Valid	Coordinate position data	
V	Valid	Coordinate position data	
W	Valid	Coordinate position data	
X	Valid	Coordinate position data	
	Valid	Dwell time	
Y	Valid	Coordinate position data	
Z	Valid	Coordinate position data	

(Note 1) All decimal points are valid for the user macro arguments.

6. Interpolation Functions

6.1 Positioning (Rapid traverse); G00
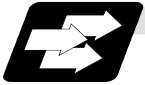
Function and purpose

This command is accompanied by coordinate words. It positions the tool along a linear or non-linear path from the present point as the start point to the end point which is specified by the coordinate words.

Command format

G00 Xx Yy Zz $\alpha\alpha$;

x, y, z, α : Represent coordinates, and could be either absolute values or incremental values, depending on the setting of G90/G91.

Detailed description

(1) Once this command has been issued, the G00 mode is retained until it is changed by another G function or until the G01, G02, G03 or G33 command in the 01 group is issued. If the next command is G00, all that is required is simply that the coordinate words be specified.

(2) In the G00 mode, acceleration and deceleration are always carried out at the start point and end point of the block. After confirming that the current block command is 0, the next block is advanced to after confirming the state of the tracking error of the acceleration/deceleration circuit. The in-position width is set with commanded block's address (,I) or the parameters.

(3) Any G command (G72 to G89) in the 09 group is cancelled (G80) by the G00 command.

(4) The tool path can be selected from linear or non-linear.
The positioning time is the same for the linear and non-linear paths.

    (a) Linear path..........: This is the same as linear interpolation (G01), and the speed is limited by the rapid traverse rate of each axis.
    (b) Non-linear path...: The tool is positioned at the rapid traverse rate independently for each axis.

⚠CAUTION

⚠ The commands without Nos. following "G" become a "G00" operation during running.

Example of program

| G91 G00 X-270000 Y300000 Z150000 ; | (For input setting unit: 0.001mm) |

Precautions

(Note 1) When parameter "#1086 G0Intp" is set to "0", the path along which the tool is positioned is the shortest path connecting the start and end points. The positioning speed is automatically calculated so that the shortest distribution time is obtained in order that the commanded speeds for each axis do not exceed the rapid traverse rate.

When for instance, the Y-axis and Z-axis rapid traverse rates are both 9600mm/min, the tool will follow the path in the figure below if the following is programmed:
G91 G00 X-300000 Y200000 ; (With an input setting unit of 0.001mm)
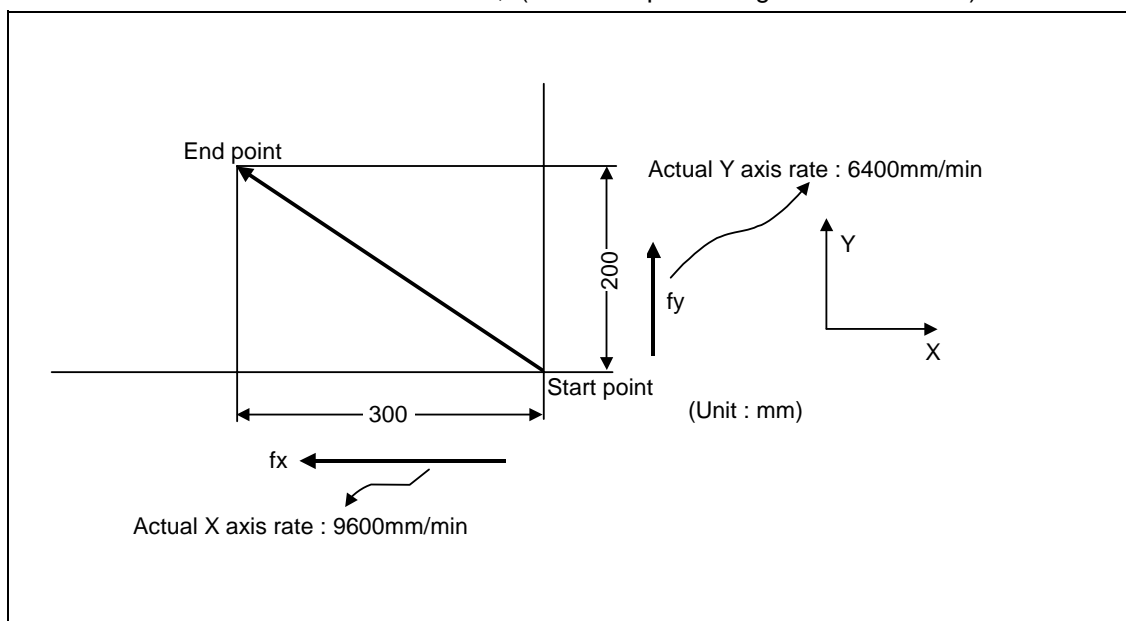
(**Note 2**) When parameter "#1086 G0Intp" is set to 1, the tool will move along the path from the start point to the end point at the rapid traverse rate of each axis.
When, for instance, the Y-axis and Z-axis rapid traverse rates are both 9600mm/min, the tool will follow the path in the figure below if the following is programmed:
G91 G00 X-300000 Y200000 ; (With an input setting unit of 0.001mm
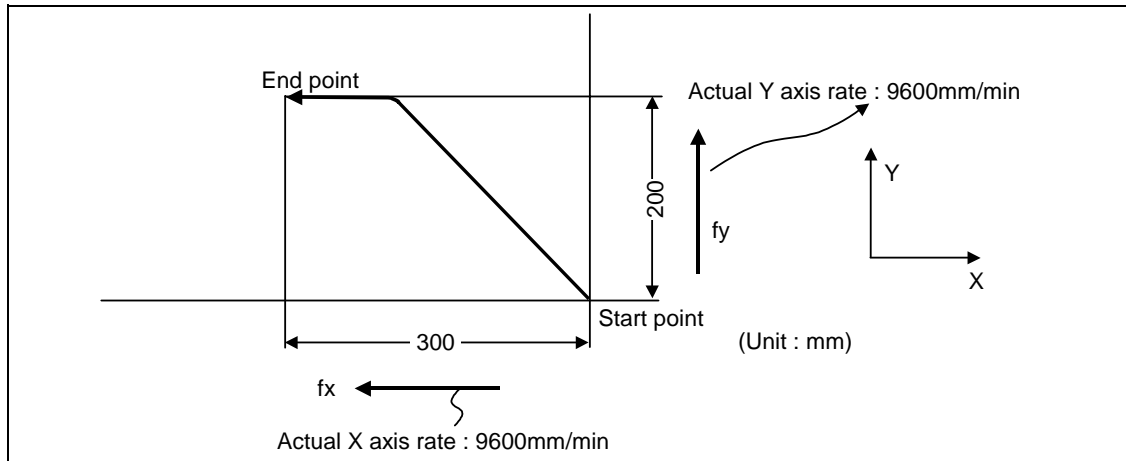
(Note 3) The rapid traverse rate for each axis with the G00 command differs according to the individual machine and so reference should be made to the machine specifications.
(Note 4) Rapid traverse (G00) deceleration check
There are two methods for the deceleration check at rapid traverse; commanded deceleration method and in-position check method. Select a method with the parameter "#1193 inpos".

■ When "inpos" = "1"
Upon completion of the rapid traverse (G00), execute the next block after confirming that the remaining distances for each axis are below the fixed amounts. (Refer to following drawing.)
The confirmation of the remaining distance should be done with the imposition width, LR
• L R is the setting value for the servo parameter "#2224 sv024" or "#2077 G0inps".
The setting for the parameter "#2224 sv024" can be made in increments of 0.0005mm or 0.00005 inch.

The purpose of checking the rapid feedrate is to minimize the time it takes for positioning. The bigger the setting value for the in-position, the longer the reduced time is, but the remaining distance of the previous block at the starting time of the next block also becomes larger, and this could become an obstacle in the actual processing work. The check for the remaining distance is done at set intervals. Accordingly, it may not be possible to get the actual amount of time reduction for positioning with the setting value in-position.

■ When "inpos" = "0"
Upon completion of the rapid traverse (G00), the next block will be executed after the deceleration check time (Td) has elapsed. The deceleration check time (Td) is as follows, depending on the acceleration/deceleration type.

(1) Linear acceleration/linear deceleration .... $Td = Ts + \alpha$

Ts : Acceleration/deceleration time constant
Td : Deceleration check time
$Td = Ts + (0 \sim 14ms)$

(2) Exponential acceleration/linear deceleration .... $Td = 2 \times Ts + \alpha$
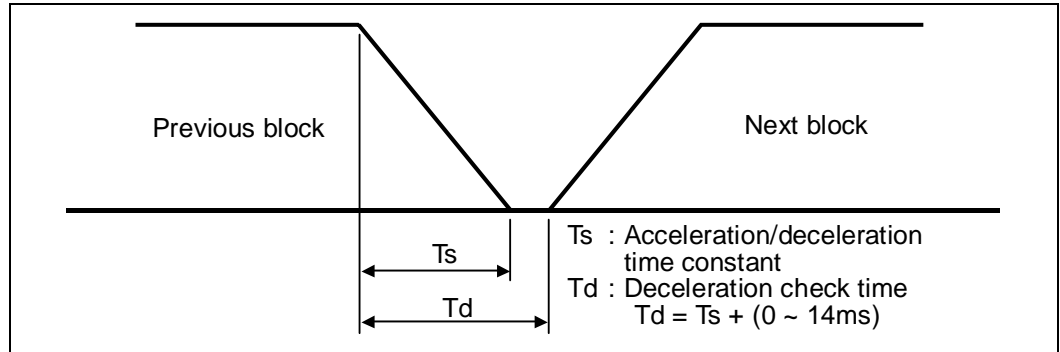
Ts : Acceleration/deceleration time constant
Td : Deceleration check time
$Td = 2 \times Ts + (0 \sim 14ms)$
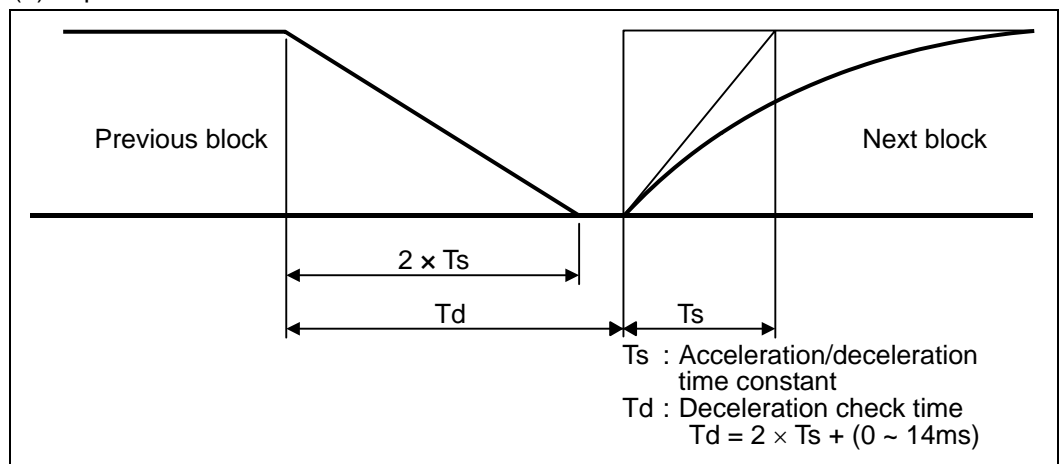
(3) Exponential acceleration/exponential deceleration...... $Td = 2 \times Ts + \alpha$

Ts : Acceleration/deceleration time constant
Td : Deceleration check time
$Td = 2 \times Ts + (0 \sim 14ms)$
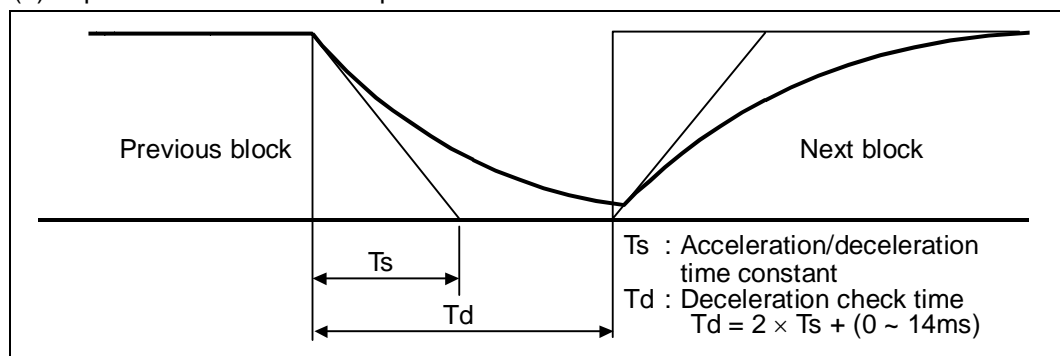
Where Ts is the acceleration time constant, $\alpha = 0$ to 14ms
The time required for the deceleration check during rapid traverse is the longest among the rapid traverse deceleration check times of each axis determined by the rapid traverse acceleration/deceleration time constants and by the rapid traverse acceleration/deceleration mode of the axes commanded simultaneously.

34

Programmable in-position width command for positioning   **[E68]**

This command commands the in-position width for the positioning command from the machining program.

```
G00  X__ Y__ Z__ , I__ ;
```

In-position width
Positioning coordinate value of each axis

,I_	: In-position width. A program error occurs if a decimal point command is issued. This is valid only in the commanded block. A block that does not contain this address will follow the parameter "#1193 inpos" settings.

Operation during in-position check   **[E68]**

Execution of the next block starts after confirming that the position error amount of the positioning (rapid traverse: G00) command block and the block that carries out deceleration check with the linear interpolation (G01) command is less than the in-position width issued in this command.

The in-position width in this command is valid only in the command block, so the deceleration check method set in base specification parameter "#1193 inpos" is used for blocks that do not have the in-position width command.

When there are several movement axes, the system confirms that the position error amount of each movement axis is less than the in-position width issued in this command before executing the next block.

The differences of when the in-position check is validated with the parameter (base specification parameter "#1193 inpos" set to 1; refer to next page for in-position width) and when validated with this command are shown in the following drawing.

Differences between in-position check with this command and in-position check with parameter

In-position check with ",I" address command	In-position check with parameter
After starting deceleration of the command system, the position error amount and commanded in-position width are compared.	After starting deceleration of the command system, the servo system's position error amount and the parameter setting value (in-position width) are compared.
Ts : Acceleration/deceleration time constant Td : Deceleration check time Td = Ts + (0 to 14ms)	

In-position width setting

When the servo parameter "#2224 SV024" setting value is smaller than the setting value of the G0 in-position width "#2077 G0inps" and the G1 in-position width "#2078 G1inps", the in-position check is carried out with the G0 in-position width and the G1 in-position width.

In-position check using the "G0inps" value

Command to motor

Outline of motor movement

G0 in-position

sv024

A stop is judged here.

In-position check using the "G1inps" value

Command to motor

Outline of motor movement

G1 in-position

sv024

A stop is judged here.
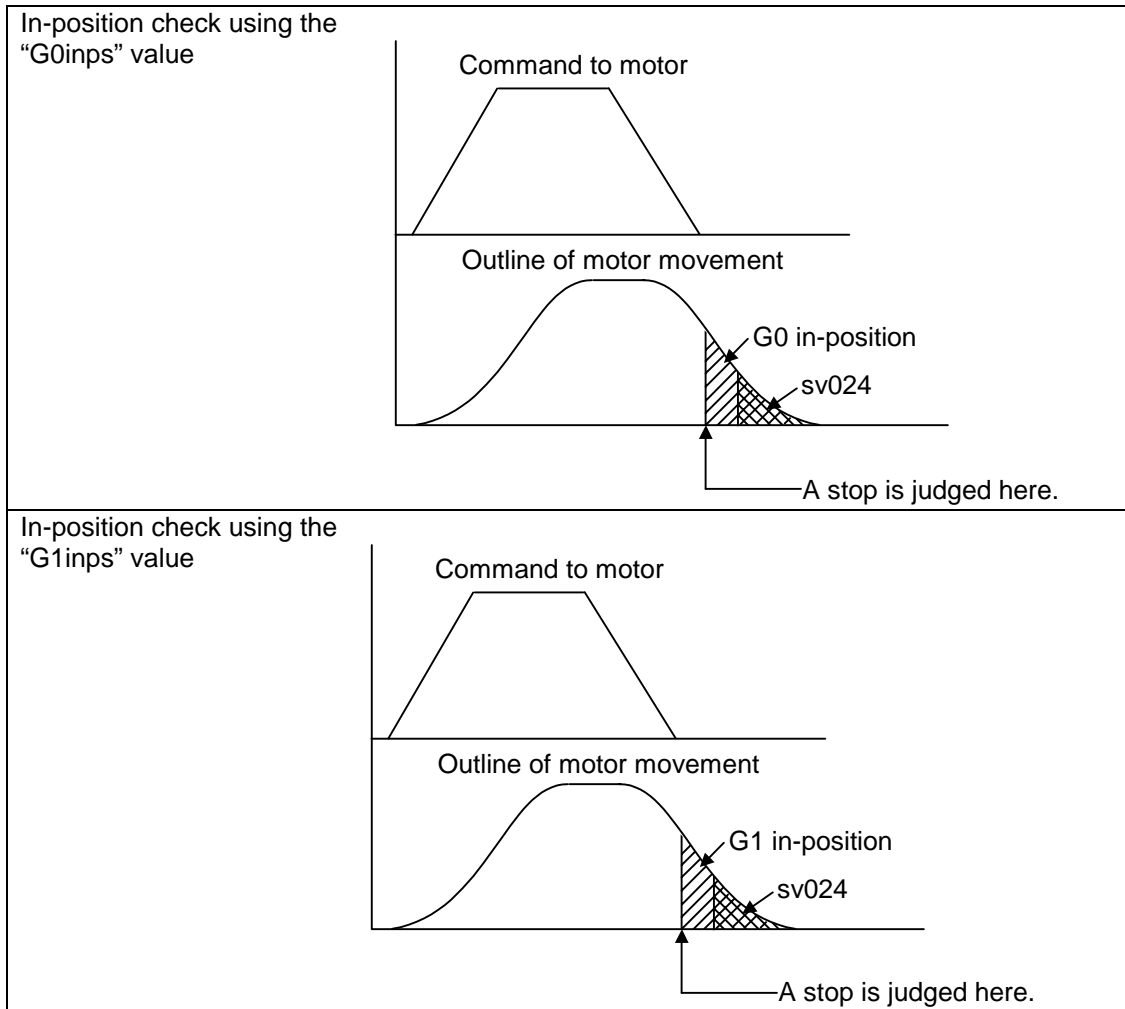
When the SV024 value is larger, the in-position check is completed when the motor position becomes within the specified with SV024.
The in-position check method depends on the method set in the deceleration check parameter.

(Note 1) For E68, when the in-position width (programmable in-position check width) is set in the machining program, either the in-position width set with the parameter (SV024, G0inps, G1inps) or that set in the program, whichever larger, is applied when performing an in-position check.

(Note 2) When the SV024 setting value is larger than the G0 in-position width/G1 in-position width, the in-position check is carried out with the SV024 value.

(Note 3) When the error detect is ON, the in-position check is forcibly carried out.

6.2 Linear interpolation; G01
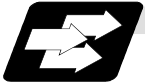
Function and purpose

This command is accompanied by coordinate words and a feedrate command.
It makes the tool move (interpolate) linearly from its present position to the end point specified by the coordinate words at the speed specified by address F. In this case, the feedrate specified by address F always acts as a linear speed in the tool nose center advance direction.

Command format

G01 Xx Yy Zz Ff ;	
x, y, z	:Coordinate values and may be an absolute position or incremental position depending on the G90/G91 state.
f	: Feedrate (mm/min or °/min)

Detailed description

Once this command is issued, the mode is maintained until another G function (G00, G02, G03, G33) in the 01 group which changes the G01 mode is issued. Therefore, if the next command is also G01 and if the feedrate is the same, all that is required to be done is to specify the coordinate words. If no F command is given in the first G01 command block, program error (P62) results.
The feedrate for a rotary axis is commanded by °/min (decimal point position unit). (F300 = 300°/min)
The G functions (G70 - G89) in the 09 group are cancelled (G80) by the G01 command.

Example of program

(Example 1) Cutting in the sequence of $P_1 \to P_2 \to P_3 \to P_4 \to P_1$ at 300 mm/min feedrate
$P_0 \to P_1$ is for tool positioning
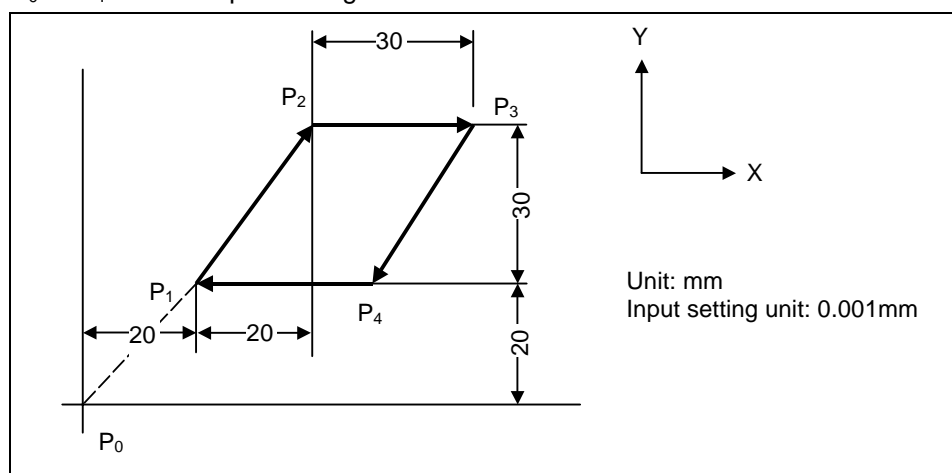
G90	G00	X20000	Y20000 ;		$P_0 \to P_1$
	G01	X20000	Y30000	F300	$P_1 \to P_2$
		X30000	;		$P_2 \to P_3$
		X-20000	Y-30000 ;		$P_3 \to P_4$
		X-30000 ;			$P_4 \to P_1$

Programmable in-position width command for linear interpolation   [E68]

This command commands the in-position width for the linear interpolation command from the machining program.

The commanded in-position width is valid in the linear interpolation command only when carrying out deceleration check.

- When the error detect switch is ON.
- When G09 (exact stop check) is commanded in the same block.
- When G61 (exact stop check mode) is selected.

```
G01  X__ Y__ Z__  F__ , I__ ;
```

In-position width
Feedrate
Linear interpolation coordinate value of each axis

,I_	: In-position width. A program error occurs if a decimal point command is issued. This is valid only in the commanded block. A block that does not contain this address will follow the parameter "#1193 inpos" settings.

(Note 1)   Refer to section "6.1 Positioning (rapid traverse); G00" for details on the in-position check operation.

6.3 Plane selection; G17, G18, G19
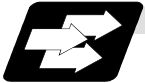
Function and purpose

The plane to which the movement of the tool during the circle interpolation (including helical cutting) and tool diameter compensation command belongs is selected.
By registering the basic three axes and the corresponding parallel axis as parameters, a plane can be selected by two axes that are not the parallel axis. If the rotary axis is registered as a parallel axis, a plane that contains the rotary axis can be selected.

The plane selection is as follows:
- Plane that executes circular interpolation (including helical cutting)
- Plane that executes tool diameter compensation
- Plane that executes fixed cycle positioning.

Command format

G17 ;	**(ZX plane selection)**
G18 ;	**(YZ plane selection)**
G19 ;	**(XY plane selection)**

X, Y and Z indicate each coordinate axis or the parallel axis.

Parameter entry

	#1026 to 1028 base_I,J,K	**#1029 to 1039 aux_I,J,K**
I	X	U
J	Y	
K	Z	V

Example of plane selection parameter entry

As shown in the above example, the basic axis and its parallel axis can be registered.
The basic axis can be an axis other than X, Y and Z.
Axes that are not registered are irrelevant to the plane selection.

Plane selection system

In Fig. 1,
I is the horizontal axis for the G17 plane or the vertical axis for the G18 plane
J is the vertical axis for the G17 plane or the horizontal axis for the G19 plane
K is the horizontal axis for the G18 plane or the vertical axis for the G19 plane

In other words,
G17 ..... IJ plane
G18 ..... KI plane
G19 ..... JK plane

(1) The axis address commanded in the same block as the plane selection (G17, G18, G19) determines which basic axis or parallel axis is used for the plane selection.
For the parameter registration example in Fig. 1.

G17X__Y__ ;	XY plane
G18X__V__ ;	VX plane
G18U__V__ ;	VU plane
G19Y__Z__ ;	YZ plane
G19Y__V__ ;	YV plane

(2) The plane will not changeover at a block where a plane selection G code (G17, G18, G19) is not commanded.

G17X__Y__ ;	XY plane
Y__Z__ ;	XY plane (plane does not change)

(3) If the axis address is omitted in the block where the plane selection G code (G17, G18, G19) is commanded, it will be viewed as though the basic three axes address has been omitted.
For the parameter registration example in Fig. 1.

G17 ;	XY plane
G17U__ ;	UY plane
G18U__ ;	ZU plane
G18V__ ;	VX plane
G19Y__ ;	YZ plane
G19V__ ;	YV plane

(4) The axis command that does not exist in the plane determined by the plane selection G code (G17, G18, G19) is irrelevant to the plane selection.
For the parameter registration example in Fig. 1.
G17U__Z__ ;

(5) If the above is commanded, the UY plane will be selected, and Z will move regardless of the plane.
If the basic axis and parallel axis are commanded in duplicate in the same block as the plane selection G code (G17, G18, G19), the plane will be determined in the priority order of basic axis and parallel axis.
For the parameter registration example in Fig. 1.
G17U__Y__W__-;

If the above is commanded, the UY plane will be selected, and W will move regardless of the plane.
(Note 1) The plane set with parameter "#1025 I_plane" will be selected when the power is turned ON or reset.

6.4 Circular interpolation; G02, G03
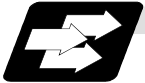
Function and purpose

These commands serve to move the tool along an arc.

Command format

G02 (G03) Xx Yy Ii Jj Kk Ff;	
G02	: Clockwise (CW)
G03	: Counterclockwise (CCW)
Xx, Yy	: End point
Ii, Jj	: Arc center
Ff	: Feedrate

For the arc command, the arc end point coordinates are assigned with addresses X, Y (or Z, or parallel axis X, Y, Z), and the arc center coordinate value is assigned with addresses I, J (or K).

Either an absolute value or incremental value can be used for the arc end point coordinate value command, but the arc center coordinate value must always be commanded with an incremental value from the start point.

The arc center coordinate value is commanded with an input setting unit. Caution is required for the arc command of an axis for which the input command value differs. Command with a decimal point to avoid confusion.

41

Detailed description

(1) G02 (or G03) is retained until another G command (G00, G01 or G33) in the 01 group that changes its mode is issued.
The arc rotation direction is distinguished by G02 and G03.
G02 Clockwise (CW)
G03 Counterclockwise (CCW)

G17(X-Y)plane   G18(Z-X)plane   G19(Y-Z)plane
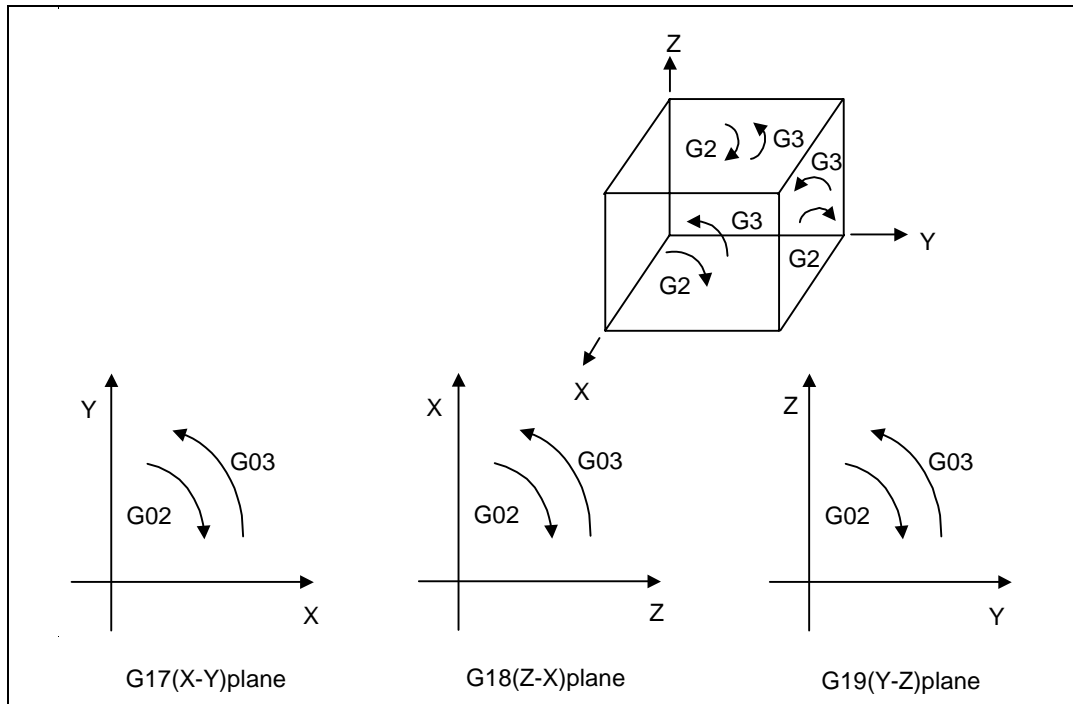
(2) An arc which extends for more than one quadrant can be executed with a single block command.
(3) The following information is needed for circular interpolation.
    (a) Plane selection.................. : Is there an arc parallel to one of the XY, ZX or YZ planes?
    (b) Rotation direction .... : Clockwise (G02) or counterclockwise (G03)?
    (c) Arc end point coordinates .. : Given by addresses X, Y, Z
    (d) Arc center coordinates .... : Given by addresses I, J, K (incremental commands)
    (e) Feed rate .... : Given by address F

Example of program
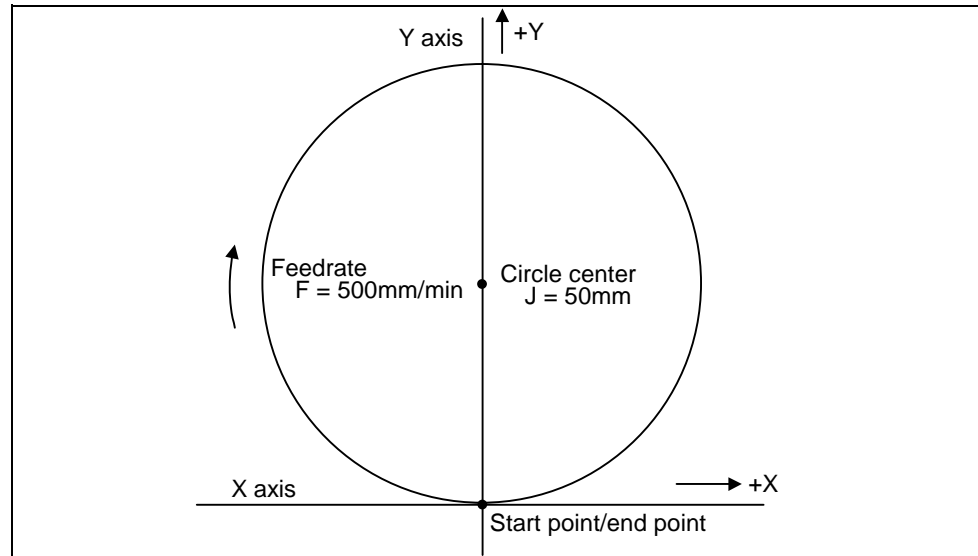
(Example 1)

G02 J50000 F500 ;	Circle command

(Example 2)
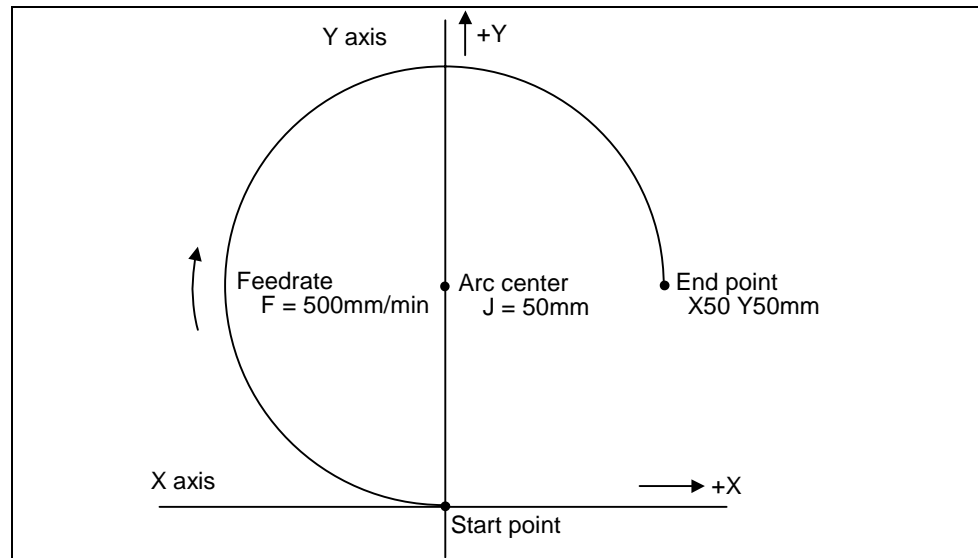
G91 G02 X50000 Y50000 J50000 F500 ;	3/4 command

Plane selection

The planes in which the arc exists are the following three planes (refer to the detailed drawings), and are selected with the following method.

XY plane
G17;   Command with a (plane selection G code)
ZX plane
G18;   Command with a (plane selection G code)
YZ plane
G19;   Command with a (plane selection G code)

Precautions for circular interpolation

(1) The terms "clockwise" (G02) and "counterclockwise" (G03) used for arc operations are defined as a case where in a right-hand coordinate system, the negative direction is viewed from the position direction of the coordinate axis which is at right angles to the plane in question.

(2) When all the end point coordinates are omitted or when the end point is the same position as the start point, a 360° arc (full circle) is commanded when the center is commanded using I, J and K.

(3) The following occurs when the start and end point radius do not match in an arc command :

(a) Program error (P70) results at the arc start point when error ΔR is greater than parameter "#1084 RadErr".

(G91) G02X9.899I 5. ;

#1084 RadErr parameter value 0.100
Start point radius = 5.000
End point radius = 4.899
Error ΔR = 0.101

Alarm stop

Start point   Center   End point
Start point radius   End point radius ΔR
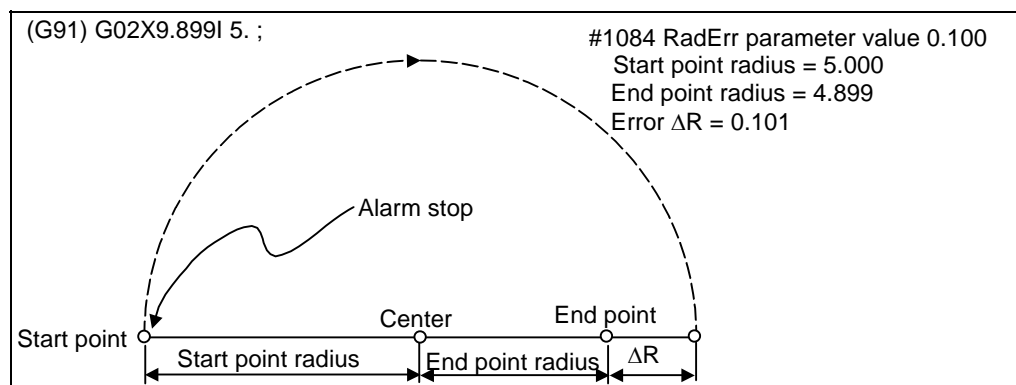
(b) Spiral interpolation in the direction of the commanded end point results when error ΔR is less than the parameter value.

(G91) G02X9.9I 5. ;

Spiral interpolation

#1084 RadErr parameter value 0.100
Start point radius = 5.000
End point radius = 4.900
Error ΔR = 0.100

Start point   Center   End point
Start point radius   End point radius ΔR
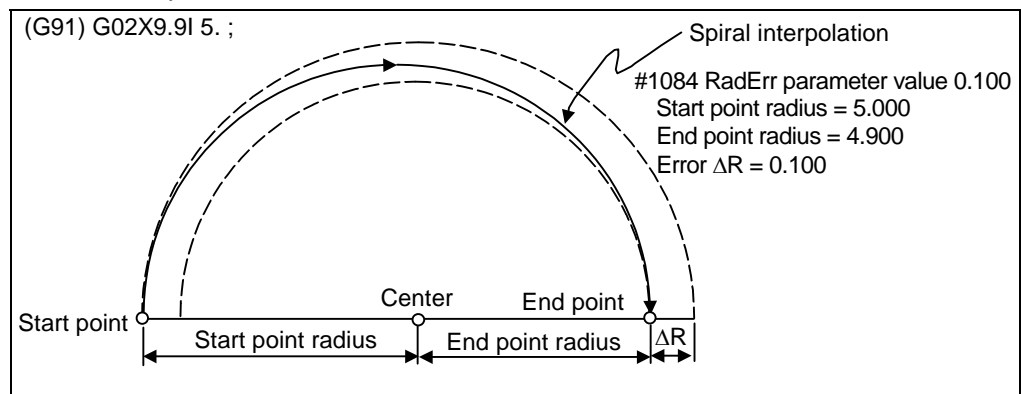
The parameter setting range is from 0.001mm to 1.000mm.

6.5 R-specified circular interpolation; G02, G03

Function and purpose

Along with the conventional circular interpolation commands based on the arc center coordinate (I, J, K) designation, these commands can also be issued by directly designating the arc radius R.

Command format

G02 (G03) Xx Yy Rr Ff ;
x   : X-axis end point coordinate
y   : Y-axis end point coordinate
r   : Arc radius
f   : Feedrate

The arc radius is commanded with an input setting unit. Caution is required for the arc command of an axis for which the input command value differs. Command with a decimal point to avoid confusion.

Detailed description

The arc center is on the bisector line which is perpendicular to the line connecting the start and end points of the arc. The point, where the arc with the specified radius whose start point is the center intersects the perpendicular bisector line, serves as the center coordinates of the arc command.
If the R sign of the commanded program is plus, the arc is smaller than a semisphere; if it is minus, the arc is larger than a semisphere.
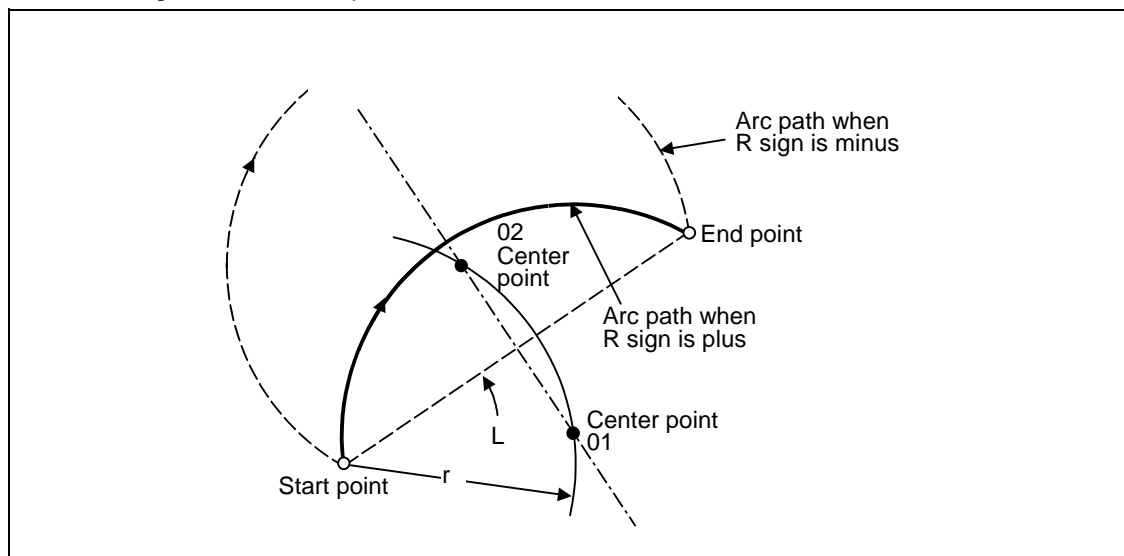
The following condition must be met with an R-specified arc interpolation command:

$L/(2xr) \leq 1$ An error will occur when $L/2 - r >$ (parameter : #1084 RadErr)

Where L is the line from the start point to end point.
When the R specification and I, J, K specification are contained in the same block, the R specification has priority in processing.
When the R specification and I, J, K specification are contained in the same block, the R specification has priority in processing.
The plane selection is the same as for the I, J, K-specified arc command.

Example of program

(Example 1)

G02 Xx_1 Yy_1 Rr_1 Ff_1 ;	XY plane R-specified arc

(Example 2)

G03 Zz_1 Xx_1 Rr_1 Ff_1 ;	ZX plane R-specified arc

(Example 3)

G02 Xx_1 Yy_1 Ii_1 Jj_1 Rr_1 Ff_1 ;	XY plane R-specified arc (When the R specification and I, J, (K) specification are contained in the same block, the R specification has priority in processing.)

(Example 4)

G17 G02 Ii_1 Jj_1 Rr_1 Ff_1 ;	XY plane This is an R-specified arc, but as this is a circle command, it is already completed.

6.6 Helical interpolation ; G17 to G19, G02, G03

Function and purpose

While circular interpolating with G02/G03 within the plane selected with the plane selection G code (G17, G18, G19), the 3rd axis can be linearly interpolated.

Normally, the helical interpolation speed is designated with the tangent speed F' including the 3rd axis interpolation element as shown in the lower drawing of Fig. 1. However, when designating the arc plane element speed, the tangent speed F on the arc plane is commanded as shown in the upper drawing of Fig. 1.

The NC automatically calculates the helical interpolation tangent speed F' so that the tangent speed on the arc plane is F.
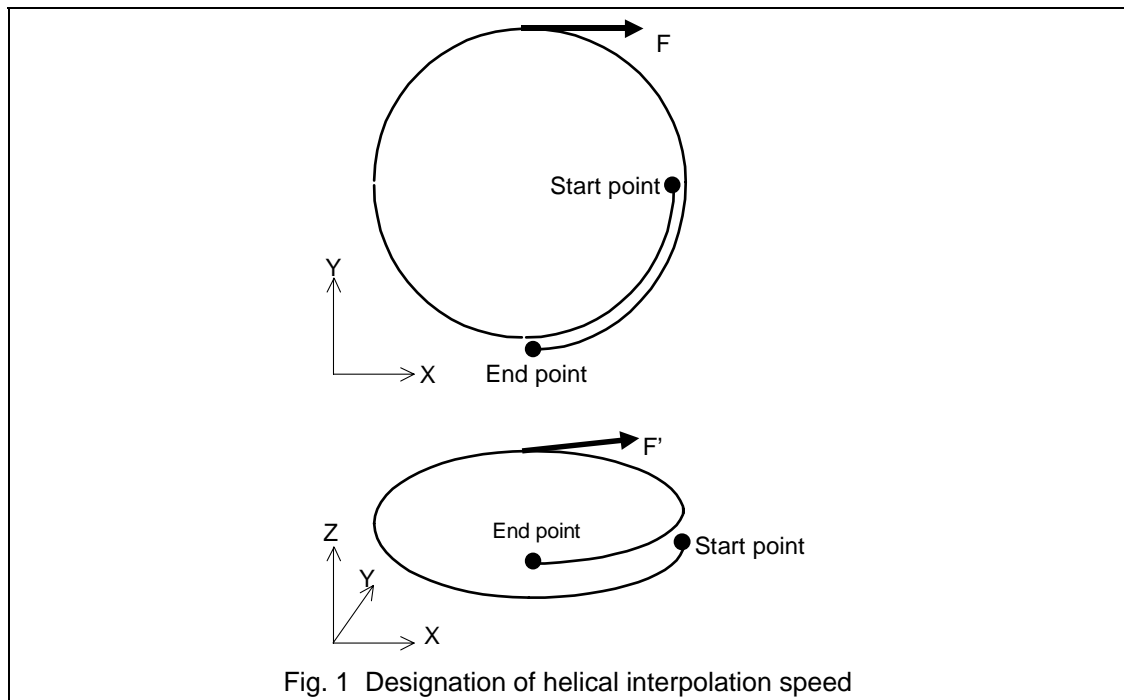
Fig. 1 Designation of helical interpolation speed

Command format

G17 G02 (G03) Xx$_1$ Yy$_1$ Zz$_1$ Ii$_1$ Jj$_1$ Pp$_1$ Ff$_1$;
G17 G02 (G03) Xx$_2$ Yy$_2$ Zz$_2$ Rr$_2$ Ff$_2$;

Xx$_1$ Yy$_1$ Xx$_2$ Yy$_2$	: Arc end point coordinate
Zz$_1$ Zz$_2$	: Linear axis end point coordinate
Ii$_1$ Jj$_1$	: Arc center coordinate
Pp$_1$	: Pitch No.
Ff$_1$ Ff$_2$	: Feedrate
Rr$_2$	: Arc radius

The arc center coordinate value and arc radius value are commanded with an input setting input. Caution is required for the helical interpolation command of an axis for which the input command value differs.

Command with a decimal point to avoid confusion.

Normal speed designation
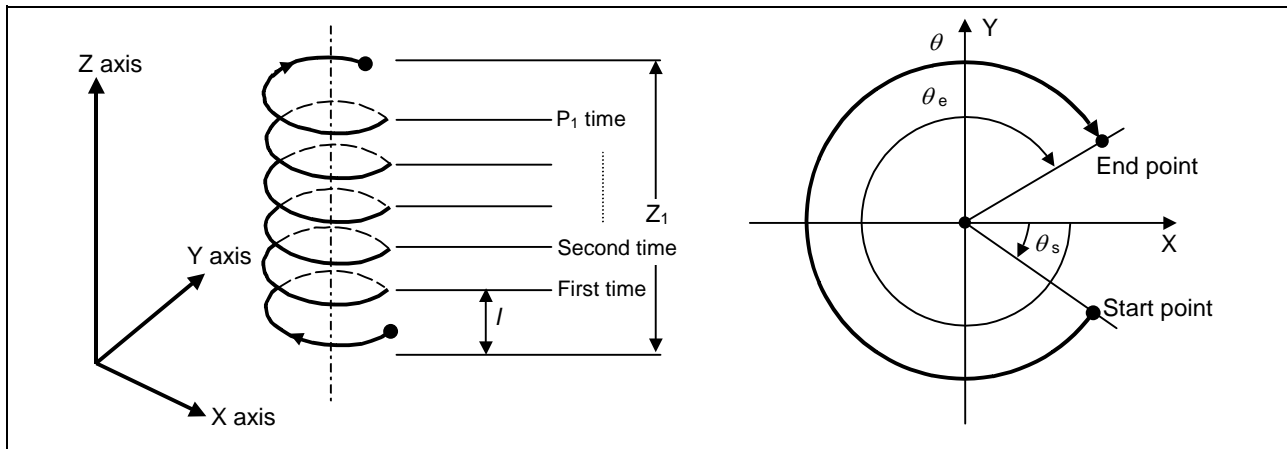
(1) For this command, command a linear axis (multiple axes can be commanded) that does not contain a circular axis in the circular interpolation command.

(2) For feedrate F, command the X, Y Z axis composite element directions speed.

(3) Pitch l is obtained with the following expression.

$$l = \frac{Z1}{(2\pi \cdot P1 + \theta) / 2\pi}$$

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

Where xs, ys are the start point coordinates from the arc center
xe, ye are the end point coordinates from the arc center

(4) If pitch No. is 0, address P can be omitted.

(Note) The pitch No. P command range is 0 to 99.
The pitch No. designation (P command) cannot be made with the R-specified arc.

(5) Plane selection
The helical interpolation arc plane selection is determined with the plane selection mode and axis address as for the circular interpolation. For the helical interpolation command, the plane where circular interpolation is executed is commanded with the plane selection G code (G17, G18, G19), and the 2 circular interpolation axes and linear interpolation axis (axis that intersects with circular plane) 3 axis addresses are commanded.

   XY plane circular, Z axis linear
      Command the X, Y and Z axis addresses in the G02 (G03) and G17 (plane selection G code) mode.
   ZX plane circular, Y axis linear
      Command the X, Y and Z axis addresses in the G02 (G03) and G18 (plane selection G code) mode.
   YZ plane circular, X axis linear
      Command the X, Y and Z axis addresses in the G02 (G03) and G19 (plane selection G code) mode.

The plane for an additional axis can be selected as with circular interpolation.

UY plane circular, Z axis linear
> Command the U, Y and Z axis addresses in the G02 (G03) and G19 (plane selection G code) mode.

In addition to the basic command methods above, the command methods following the program example can be used. Refer to the section "6.4 plane selection for the arc planes selected" with these command methods.

Example of program
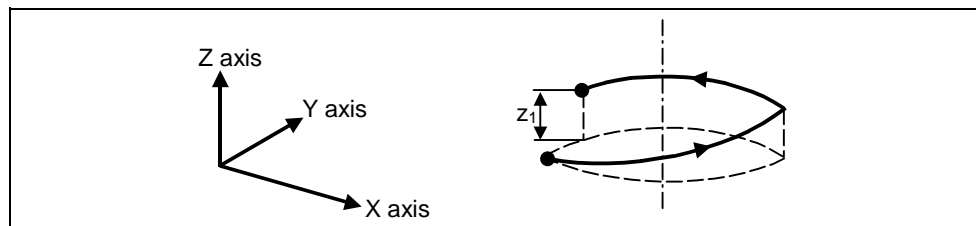
(Example 1)

G17 ;	XY plane
G03 Xx_1 Yy_1 $Zz1$ $Ii1$ $Jj1$ P0 $Ff1$;	XY plane arc, Z axis linear

(Note) If pitch No. is 0, address P can be omitted.
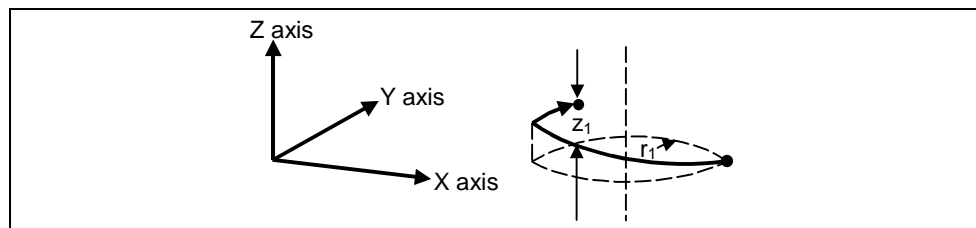
(Example 2)

G17 ;	XY plane
G02 $Xx1$ $Yy1$ $Zz1$ $Rr1$ $Ff1$;	XY plane arc, Z axis linear

(Example 3)
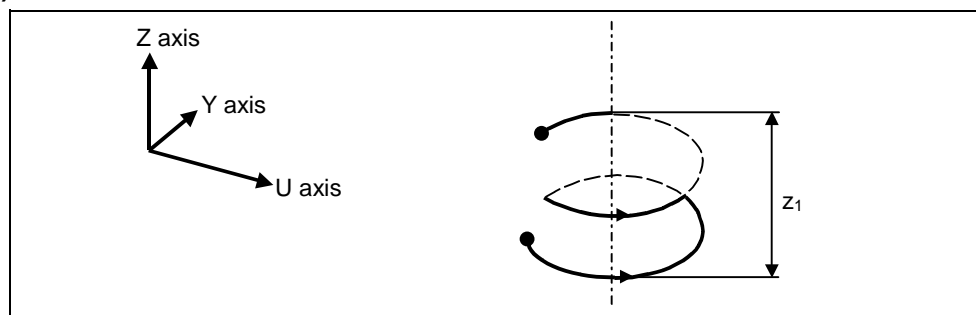
G17 G03 Uu_1 Yy_1 Zz_1 Ii_1 Jj_1 P2 Ff_1 ;	UY plane arc, Z axis linear

(Example 4)

G18 G03 Xx_1 Uu_1 Zz_1 I i1 Kk_1 Ff_1 ;	ZX plane arc, U axis linear

(Note) If the same system is used, the standard axis will perform circular interpolation and the additional axis will perform linear interpolation.

(Example 5)

G18 G02 Xx_1 Uu_1 Yy_1 Zz_1 Ii_1 Jj_1 Kk_1 Ff_1 ;	ZX plane arc, U axis, Y axis linear (The J command is ignored)

(Note) Two or more axes can be designated for the linear interpolation axis. When using E60, only one axis can be designated.

6.7 Thread cutting

6.7.1 Constant lead thread cutting ; G33
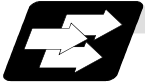
Function and purpose

The G33 command exercises feed control over the tool which is synchronized with the spindle rotation and so this makes it possible to conduct constant-lead straight thread-cutting and tapered thread-cutting.

Command format

G32 Zz Ff$_1$ Qq ;	(Normal lead thread cutting commands)
Zz	: Thread cutting direction axis address (X, Y, Z) and thread length
Ff	: Lead of long axis (axis which moves most) direction.
Qq	: Thread cutting start shift angle, (0.000 to 360.000°)

G33 Zz Ee$_1$ Qq ;	(Precision lead thread cutting commands)
Zz	: Thread cutting direction axis address (X, Y, Z) and thread length
Ee	: Lead of long axis (axis which moves most) direction
Qq	: Thread cutting start shift angle, (0.000 to 360.000°)

Detailed description

(1) The E command is also used for the number of ridges in inch thread cutting, and whether the ridge number or precision lead is to be designated can be selected by parameter setting. (Precision lead is designated by setting the parameter "#1229 set 01/bit 1" set to 1.)

(2) The lead in the long axis direction is commanded for the taper thread lead.
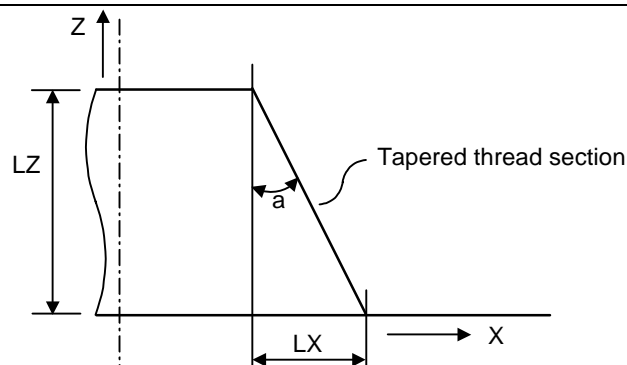
When a<45° lead is LZ
When a>45° lead is LX
When a=45° lead can be in either LX or LZ

(a) When using E60

Thread cutting Metric input   Thread cutting Inch input

Input unit system	0.001mm			0.0001inch		
Command address	F (mm/rev)	E (mm/rev)	E (threads/inch)	F (inch/rev)	E (inch/rev)	E (threads/inch)
Minimum command unit	1 (= 1.000), (1.=1.000)	1 (= 1.00001), (1.=1.00001)	1 (= 1.00), (1.=1.00)	1(=1.0000), (1.=1.0000)	1(=1.000000), (1.=1.000000)	1 (= 1.0000), (1.=1.0000)
Command range	0.001 to 999.999	0.00001 to 999.99999	0.03 to 999.99	0.0001 to 99.9999	0.000001 to 39370078	0.0255 to 9999.9999

(b) When using E68

Thread cutting Metric input

Input unit system	B (0.001mm)			C (0.0001mm)		
Command address	F (mm/rev)	E (mm/rev)	E (threads/inch)	F (mm/rev)	E (mm/rev)	E (threads/inch)
Minimum command unit	1(=1.000) (1.=1.000)	1(=1.00000) (1.=1.00000)	1(=1.00) (1.=1.00)	1(=1.0000) (1.=1.0000)	1(=1.00000) (1.=1.00000)	1(=1.000) (1.=1.000)
Command range	0.001 to 999.999	0.00001 to 999.99999	0.03 to 999.99	0.0001 to 99.9999	0.000001 to 99.999999	0.255 to 9999.999

Thread cutting Inch input

Input unit system	B (0.0001inch)			C (0.00001inch)		
Command address	F (inch/rev)	E (inch/rev)	E(threads/inch)	F (inch/rev)	E (inch/rev)	E (threads / inch)
Minimum command unit	1(=1.0000) (1.=1.0000)	1(=1.000000) (1.=1.0000000)	1(=1.0000) (1.=1.0000)	1(=1.00000) (1.=1.00000)	1(=1.000000) (1.=1.000000)	1(=1.0000) (1.=1.0000)
Command range	0.0001 to 99.9999	0.000001 to 39.370078	0.0255 to 9999.9999	0.00001 to 3.93700	0.000001 to 3.937007	0.25401 to 999.99999

(Note 1) It is not possible to assign a lead where the feed rate as converted into per-minute feed exceeds the maximum cutting feed rate.

(3) The constant surface speed control function should not be used for taper thread cutting commands or scrolled thread cutting commands.

(4) The spindle rotation speed should be kept constant throughout from the rough cutting until the finishing.

(5) If the feed hold function is employed during thread cutting to stop the feed, the thread ridges will lose their shape. For this reason, feed hold does not function during thread cutting.
Note that this is valid from the time the thread cutting command is executed to the time the axis moves.
If the feed hold switch is pressed during thread cutting, block stop will result at the end point of the block following the block in which thread cutting is completed (no longer G33 mode).

(6) The converted cutting feedrate is compared with the cutting feed clamp rate when thread cutting starts, and if it is found to exceed the clamp rate, an operation error will result.

(7) In order to protect the lead during thread cutting, a cutting feedrate which has been converted may sometimes exceed the cutting feed clamp rate.

(8) An illegal lead is normally produced at the start of the thread and at the end of the cutting because of servo system delay and other such factors.
Therefore, it is necessary to command a thread length which is determined by adding the illegal lead lengths $\delta 1$ and $\delta 2$ to the required thread length.

(9) The spindle rotation speed is subject to the following restriction:

$$1 \leq R \leq \frac{\text{Maximum feedrate}}{\text{Thread lead}}$$

Where $R \leq$ Tolerable speed of encoder (r/min)
R = Spindle rotation speed (r/min)
Thread lead = mm or inches
Maximum feedrate = mm/min or inch/mm
(this is subject to the restrictions imposed by the machine specifications).

(10) A program error (P97) may occur when the result of the expression (9) is R<1 because the thread lead is very large to the highest cutting feedrate.

(11) Dry run is valid for thread cutting but the feedrate based on dry run is not synchronized with the spindle rotation.

The dry run signal is checked at the start of thread cutting and any switching during thread cutting is ignored.

(12) Synchronous feed applies for the thread cutting commands even with an asynchronous feed command (G94).

(13) Spindle override or cutting feed override is invalid and 100% fixed during thread cutting.

(14) When a thread cutting command is programmed during tool radius compensation, the compensation is temporarily canceled and the thread cutting is executed.

(15) When the mode is switched to another automatic mode while G33 is executed, the following block which does not contain a thread cutting command is first executed and then the automatic operation stops.

(16) When the mode is switched to the manual mode while G33 is executed, the following block which does not contain a thread cutting command is first executed and then the automatic operation stops. In the case of a single block, the following block which does not contain a thread cutting command (When G33 mode is cancelled) is first executed and then the automatic operation stops. Note that automatic operation is stopped until the G33 command axis starts moving.

(17) The thread cutting command waits for the single rotation sync signal of the rotary encoder and starts movement.
Note that when using two systems, if one system issues a thread cutting command during ongoing thread cutting by another system, the movement will start without waiting for the rotary encoder single rotation sync signal. Therefore, carry out synchronization between systems before issuing a thread cutting command with multiple systems.

(18) The automatic handle interrupt/interruption is valid during thread cutting.

(19) The thread cutting start shift angle is not a modal. If there is no Q command with G33, this will be handled as "Q0".

(20) If a value exceeding 360.000 is commanded with G33 Q, a program error (P35) will occur..

(21) G33 cuts one row with one cycle. To cut two rows, change the Q value, and issue the same command.

Example of program
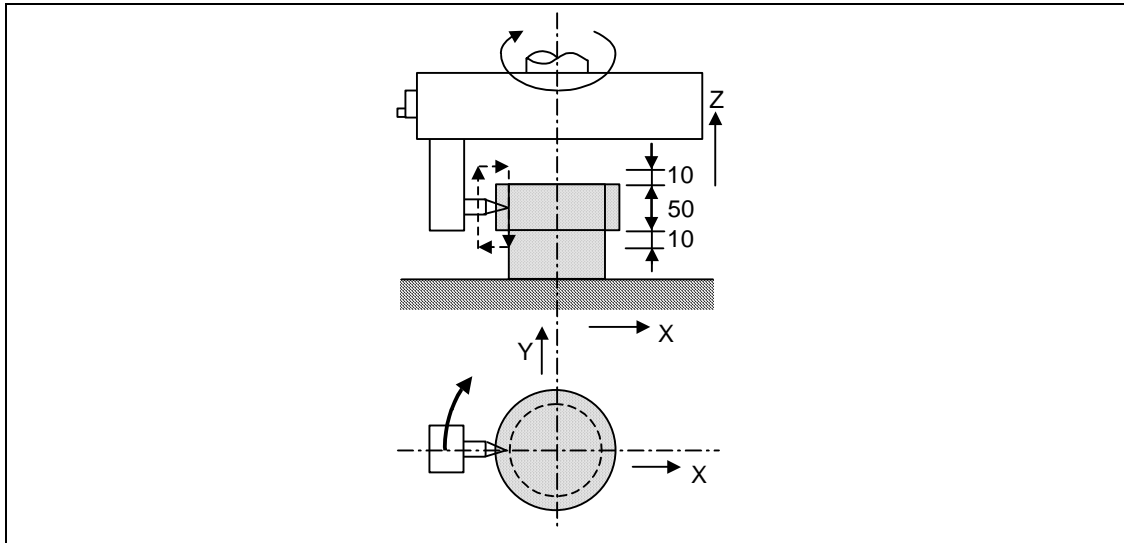
N110 G90 G0 X-200. Y-200. S50 M3 ; N111 Z110. ;	The spindle center is positioned to the workpiece center, and the spindle rotates in the forward direction.
N112 G33 Z40. F6.0 ;	The first thread cutting is executed. Thread lead = 6.0mm
N113 M19 ;	Spindle orientation is executed with the M19 command.
N114 G0X-210. ;	The tool is evaded in the X axis direction.
N115 Z110. M0 ;	The tool rises to the top of the workpiece, and the program stops with M00. Adjust the tool if required.
N116 X-200. ; M3 ;	Preparation for second thread cutting is done.
N117 G04 X5.0 ;	Command dwell to stabilize the spindle rotation if necessary.
N118 G33 Z40. ;	The second thread cutting is executed.

6.7.2 Inch thread cutting; G33
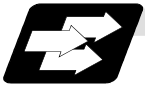
Function and purpose

If the number of ridges per inch in the long axis direction is assigned in the G33 command, the feed of the tool synchronized with the spindle rotation will be controlled, which means that constant-lead straight thread-cutting and tapered thread-cutting can be performed.

Command format

G33 Zz Ee Qq ;	
Zz	: Thread cutting direction axis address (X, Y, Z) and thread length
Ee	: Number of ridges per inch in direction of long axis (axis which moves most) (decimal point command can also be assigned)
Qq	: Thread cutting start shift angle, 0.000 to 360.000°.

Detailed description

(1) The number of ridges in the long axis direction is assigned as the number of ridges per inch.

(2) The E code is also used to assign the precision lead length, and whether the ridge number of precision lead length is to be designated can be selected by parameter setting. (The number of ridges is designated by setting the parameter "#1229 set01/bit1" to 0.)

(3) The E command value should be set within the lead value range when the lead is converted.
(4) The other matters are the same as uniform lead thread cutting.

Example of program

Thread lead ..... 3 threads/inch (= 8.46666 ...)
When programmed with δ_1= 10mm,
δ_2 = 10mm using metric input
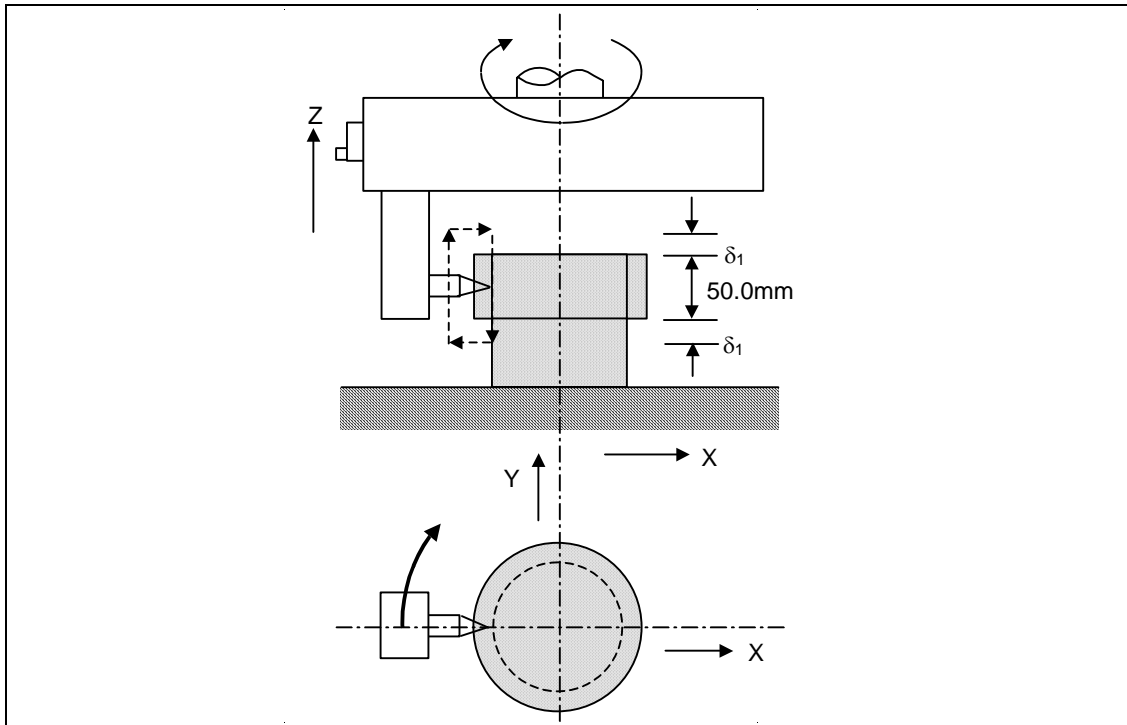
N210 G90 G0X-200. Y-200. S50M3;	
N211 Z110.;	
N212 G91 G33 Z-70.E3.0;	(First thread cutting)
N213 M19;	
N214 G90 G0X-210.;	
N215 Z110.M0;	
N216 X-200.; M3;	
N217 G04 X2.0;	
N218 G91 G33 Z-70.;	(Second thread cutting)

6.8 Uni-directional positioning; G60

Function and purpose

The G60 command can position the tool at a high degree of precision without backlash error by locating the final tool position from a single determined direction.

Command format

G60 Xx Yy Zz αα ;
α   : Optional axis

Detailed description

(1) The creep distance for the final positioning as well as the final positioning direction is set by parameter.
(2) After the tool has moved at the rapid traverse rate to the position separated from the final position by an amount equivalent to the creep distance, it move to the final position in accordance with the rapid traverse setting where its positioning is completed.
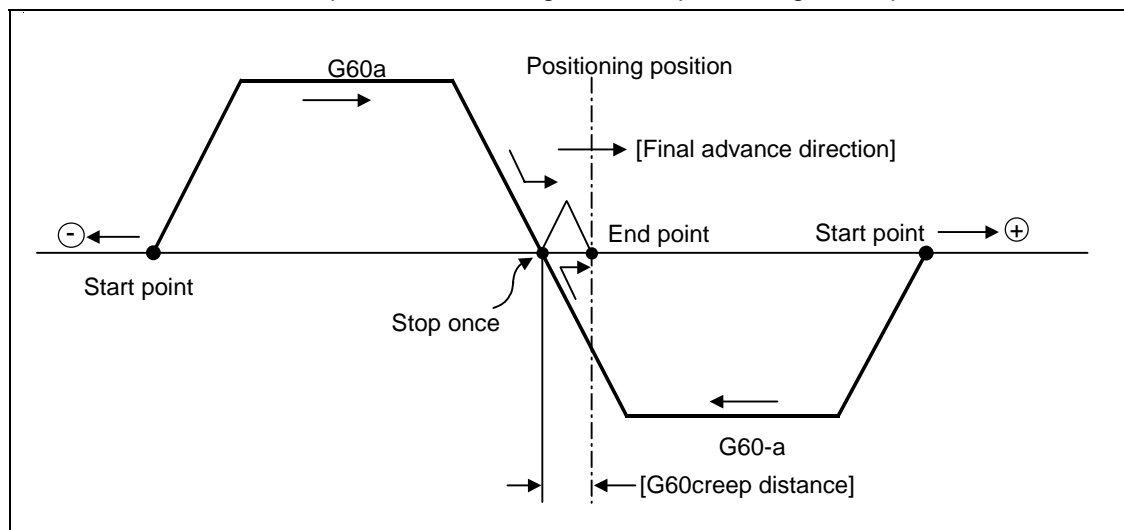
(3) The above positioning operation is performed even when Z-axis commands have been assigned for Z-axis cancel and machine lock. (Display only)
(4) When the mirror image function is ON, the tool will move in the opposite direction as far as the intermediate position due to the mirror image function but the operation within the creep distance during its final advance will not be affected by mirror image.
(5) The tool moves to the end point at the dry run speed during dry run when the G0 dry run function is valid.
(6) Feed during creep distance movement with final positioning can be stopped by resetting, emergency stop, interlock, feed hold and rapid traverse override zero.
    The tool moves over the creep distance at the rapid traverse setting. Rapid traverse override is valid.
(7) Uni-directional positioning is not performed for the drilling axis during drilling fixed cycles.
(8) Uni-directional positioning is not performed for shift amount movements during the fine boring or back boring fixed cycle.
(9) Normal positioning is performed for axes whose creep distance has not been set by parameter.
(10) Uni-directional positioning is always a non-interpolation type of positioning.
(11) When the same position (movement amount of zero) has been commanded, the tool moves back and forth over the creep distance and is positioned at its original position from the final advance direction.
(12) Program error (P61) results when the G60 command is assigned with an NC system which has not been provided with this particular specification.

6.9 Circular interpolation; G07.1   [E68]
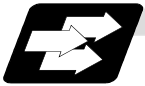
Function and purpose

This function develops a shape with a cylindrical side (shape in cylindrical coordinate system) into a plane. When the developed shape is programmed as the plane coordinates, that is converted into the linear axis and rotation axis movement in the cylindrical coordinates and the contour is controlled during machining.
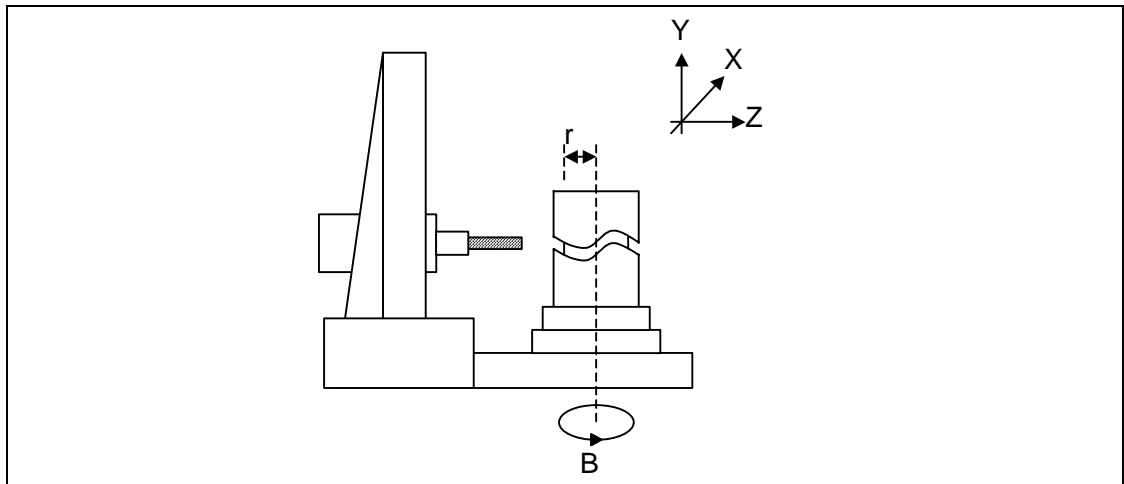
As programming can be carried out with a shape with which the side on the cylinder is developed, this is effective for machining cylindrical cams, etc. When programmed with the rotation axis and its orthogonal axis, slits, etc., can be machined on the cylinder side.
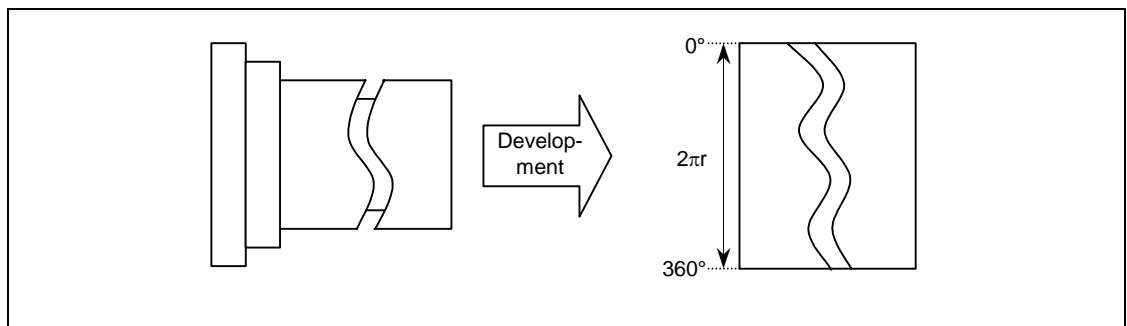
Command format

G07.1 Cc ;	(Cylindrical interpolation mode start/cancel)
Cc	Cylinder radius value
	• Radius value $\neq$ 0: Cylindrical interpolation mode start
	• Radius value = 0: Cylindrical interpolation mode cancel

(Note) The above format applies when the name of the rotation axis is "C". If the name is not "C", command the name of the rotation axis being used instead of "C".

(1) The coordinates commanded in the interval from the start to cancellation of the cylindrical interpolation mode will be the cylindrical coordinate system.

    G07.1 C Cylinder radius value;   Cylindrical interpolation mode start
     :   (Cylindrical interpolation will start)
     :   (The coordinate commands in this interval will be
     :   the cylindrical coordinate system)
    G07.1 C0 ;   Cylindrical interpolation mode cancel
                                               (Cylindrical interpolation will be canceled)

(2) G107 can be used instead of G07.1.

(3) Command G07.1 in an independent block. A program error (P33) will occur if this command is issued in the same block as another G code.

(4) Program the rotation axis with an angle degree.

(5) Linear interpolation or circular interpolation can be commanded during the cylindrical interpolation mode. Note that the plane selection command must be issued just before the G07.1 block.

(6) The coordinates can be commanded with either an absolute command or incremental command.

(7) Tool radius compensation can be applied on the program command. Cylindrical interpolation will be executed on the path after tool radius compensation.

(8) Command the segment feed in the cylinder development with F. The F unit is mm/min or inch/min.

Detailed description

(1) Cylindrical interpolation accuracy

In the cylindrical interpolation mode, the movement amount of the rotation axis commanded with an angle is converted on the circle periphery, and after operating the linear and circular interpolation between the other axes, the amount is converted into an angle again.

Thus, the actual movement amount may differ from the commanded value such as when the cylinder radius is small.

Note that the error generated at this time is not cumulated.

(2) Related parameters

#	Item		Details	Setting range
1516	mill_ax	Milling axis name	Set the name of the rotation axis for cylindrical interpolation. Only one of the rotation axes can be set.	A to Z
8111	Milling Radius		Select the diameter and radius of the linear axis for cylindrical interpolation. 　0: Radius command for all axes 　1: Each axis setting (follows #1019 dia diameter designation axis)	0 / 1
1267 (PR)	ext03 (bit0)	G code type	The type of high-accuracy G code is changed. 　0: G61.1 type 　1: G8 type	0 / 1
1270 (PR)	ext06 (bit7)	Handle C axis coordinate during cylindrical interpolation	Specify whether the rotary axis coordinate before the cylindrical interpolation start command is issued is kept during the cylindrical interpolation or not. 　0: Do not keep 　1: keep	0 / 1

(3) Plane selection

The axis used for cylindrical interpolation must be set with the plane selection command.
(Note)
The correspondence of the rotation axis to an axis' parallel axis is set with the parameters (#1029, #1030, #1031).
The circular interpolation and tool radius compensation, etc., can be designated on that plane.
The plane selection command is set immediately before or after the G07.1 command. If not set and a movement command is issued, a program error (P485) will occur.

(Example) G19 Z0. C0. ;

G07.1 C100. ;

:

G07.1 C0 ;

	G17	G18	G19
Basic coordinate system X, Y, Z	Y↑ →X	X↑ →Z	Z↑ →Y
Cylindrical coordinate system C, Y, Z (Rotation axis is X axis' parallel axis) #1029	Y↑ →C G17	C↑ →Z G18	✕
Cylindrical coordinate system X, C, Z (Rotation axis is Y axis' parallel axis) #1030	C↑ →X G17	✕	Z↑ →C G19
Cylindrical coordinate system X, Y, C (Rotation axis is Z axis' parallel axis) #1031	✕	X↑ →C G18	C↑ →Y G19
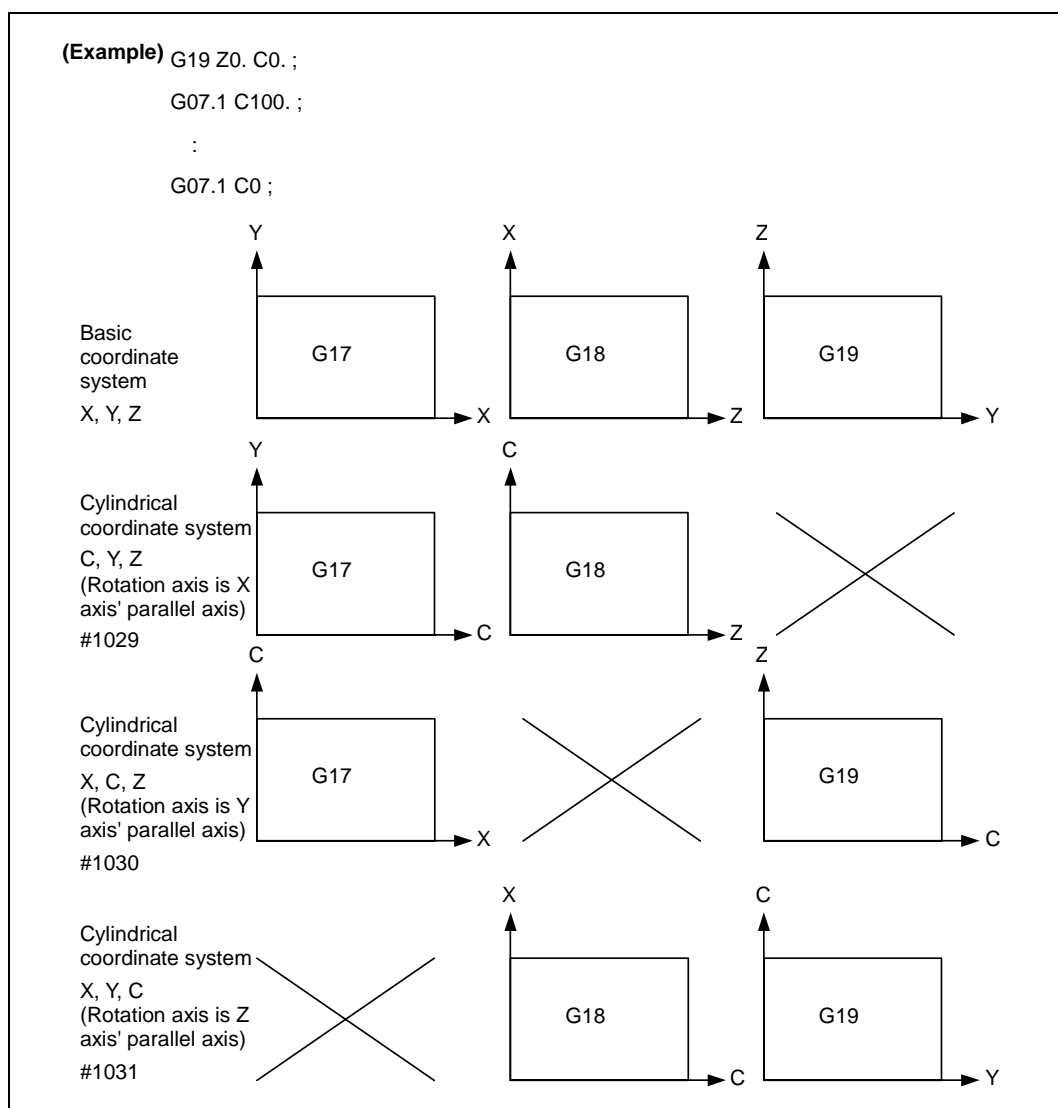
(Note) Depending on the model or version, the Z-C plane (Y-Z cylinder plane) will be automatically selected with G07.1 and G19.
The circular interpolation and tool radius compensation, etc., can be designated on that plane.

	G17	G18	G19
Basic coordinate system X,Y,Z	Y↑ →X	X↑ →Z	Z↑ →Y
Cylindrical coordinate system			Z→ G19 ↓C
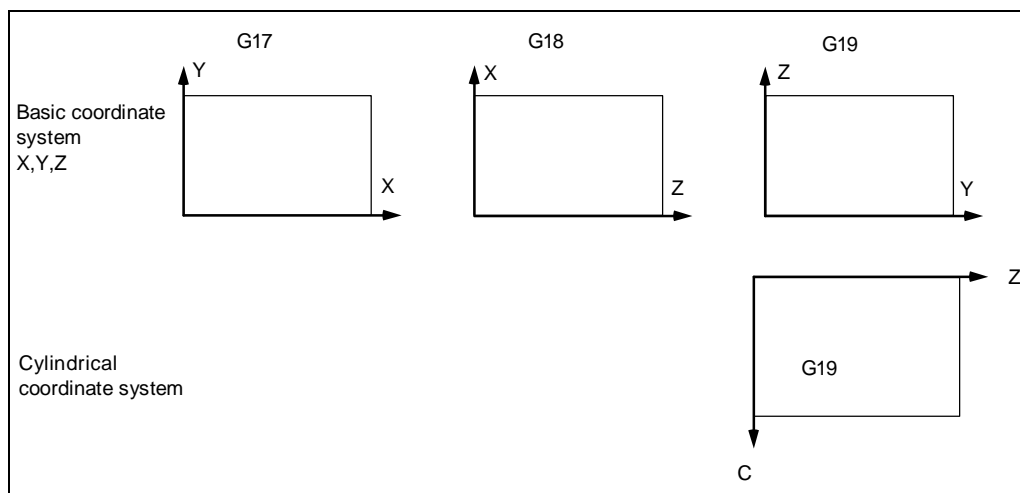
Relation with other functions

(1) Circular interpolation

- (a) Circular interpolation between the rotation axis and linear axis is possible during the cylindrical interpolation mode.

- (b) An R specification command can be issued with circular interpolation. (I, J and K cannot be designated.)

(2) Tool radius compensation

The tool radius can be compensated during the cylindrical interpolation mode.

- (a) Command the plane selection in the same manner as circular interpolation.
  When using tool radius compensation, start up and cancel the compensation within the cylindrical interpolation mode.

- (b) A program error (P485) will occur if G07.1 is commanded during tool radius compensation.

- (c) If the G07.1 command is issued with no movement command given after the tool radius compensation is canceled, the position of the axis in the G07.1 command block is interpreted as the position applied after the tool radius compensation is canceled and the following operations are performed.

(3) Cutting asynchronous feed

- (a) The asynchronous mode is forcibly set when the cylindrical interpolation mode is started.

- (b) When the cylindrical interpolation mode is canceled, the synchronization mode will return to the state before the cylindrical interpolation mode was started.

- (c) A program error (P485) will occur if G07.1 is commanded in the constant surface speed control mode (G96).

(4) Miscellaneous functions

- (a) The miscellaneous function (M) and 2nd miscellaneous function can be issued even in the cylindrical interpolation mode.

- (b) The S command in the cylindrical interpolation mode issues the rotary tool's rotation speed instead of the spindle rotation speed.

- (c) Issue the T command before cylindrical interpolation is started. A program error (P485) will occur if the T command is issued in the cylindrical interpolation mode.

  ```
      :
      :
  T12 ;                  ... T command before cylindrical interpolation → Valid
  T0 X100. Z0 ;
  G19 Z C ;
  G07.1 C100. ;
      :
  T11 ;                  ... T command in cylindrical interpolation mode →Program error
      :
  G07.1 C0 ;
  ```

- (d) Complete the tool compensation movement (movement of tool length and wearcompensation amount) before executing the cylindrical interpolation. If the tool compensation movement is not completed when the cylindrical interpolation start command has been issued, the followings will be resulted:
  - Machine coordinate is not changed even if G07.1 is executed.
  - The workpiece coordinate is changed to that of the post tool length compensation when G07.1 is executed. (Even if canceling the cylindrical interpolation, this workpiece coordinate will not be canceled.)

(5) F command during cylindrical interpolation

As for the F command in the cylindrical interpolation mode, whether the previous F command is used or not depends on that the mode just before G07.1 is the feed per minute command (G94) or feed per rotation command (G95).

(a) When G94 is commanded just before G07.1
If there is no F command in the cylindrical interpolation, the previous F command feedrate will be used.
The feedrate after the cylindrical interpolation mode is canceled will remain the F command feedrate issued when the cylindrical interpolation mode was started or the final F command feedrate set during cylindrical interpolation.

(b) When G95 is commanded just before G07.1
The previous F command feedrate cannot be used during cylindrical interpolation, thus a new F command must be issued.
The feedrate after the cylindrical interpolation mode is canceled will return to that applied before the cylindrical interpolation mode was started.

When there is no F command in G07.1

Previous mode	No F command	After G07.1 is canceled
G94	Previous F is used	←
G95	Program error (P62)	F just before G07.1 is used

When F is commanded in G07.1

Previous mode	No F command	After G07.1 is canceled
G94	Commanded F is used	←
G95	Commanded F is used *1	F just before G07.1 is used

*1) Moves with the feed per minute command during G07.1.

Restrictions and precautions

(1) The following G code commands can be used during the cylindrical interpolation mode.

G code	Details
G00	Positioning
G01	Linear interpolation
G02	Circular interpolation (CW)
G03	Circular interpolation (CWW)
G04	Dwell
G09	Exact stop check
G40-42	Tool nose R compensation
G61	Exact stop check mode
G64	Cutting mode
G65	Macro call (simple call)
G66	Macro modal call A (modal call)
G66.1	Macro modal call B (block call per macro)
G67	Macro modal call cancel (modal call cancel)
G80-89	Hole drilling fixed cycle
G90/91	Absolute/incremental value command
G94	Asynchronous feed
G98	Hole drilling cycle initial return
G99	Hole drilling cycle R point return

A program error (P481) may occur if a G code other than those listed above is commanded during cylindrical interpolation.

(2) The cylindrical interpolation mode is canceled when the power is turned ON or reset.

(3) A program error (P484) will occur if any axis commanded for cylindrical interpolation has not completed reference point return.

(4) Tool radius compensation must be canceled before the cylindrical interpolation mode can be canceled.

(5) When the cylindrical interpolation mode is canceled, the mode will change to the cutting mode, and the plane will return to that selected before cylindrical interpolation.

(6) The program of the block during the cylindrical interpolation cannot be restarted (program restart).

(7) A program error (P486) will occur if the cylindrical interpolation command is issued during the mirror image.

(8) When the cylindrical interpolation mode is started or canceled, the deceleration check is performed.

(9) A program error (P486) will occur if the cylindrical interpolation is commanded during the cylindrical interpolation mode.

Example of program

<Program>

N01 G28XZC;
N02 T02;
N03 G97S100M23;
N04 G00X50.Z0.;
N05 G94G01X40.F100.;
N06 G19C0Z0; ◀────── Command of plane selection for cylindrical interpolation and command of two interpolation axes
N07 G07.1C20.; ◀────── Cylindrical interpolation start
N08 G41;
N09 G01Z-10.C80.F150;
N10 Z-25.C90.;
N11 Z-80.C225;
N12 G03Z-75.C270.R55.;
N13 G01Z-25;
N14 G02Z-20.C280.R80.;
N15 G01C360.
N16 G40;
N17 G07.1C0; ◀────── Cylindrical interpolation cancel
N18 G01X50.;
N19 G0X100.Z100.;
N20 M25;
N21 M30;

<Parameter>
#1029 aux_I
#1030 aux_J   C
#1031 aux_K

65
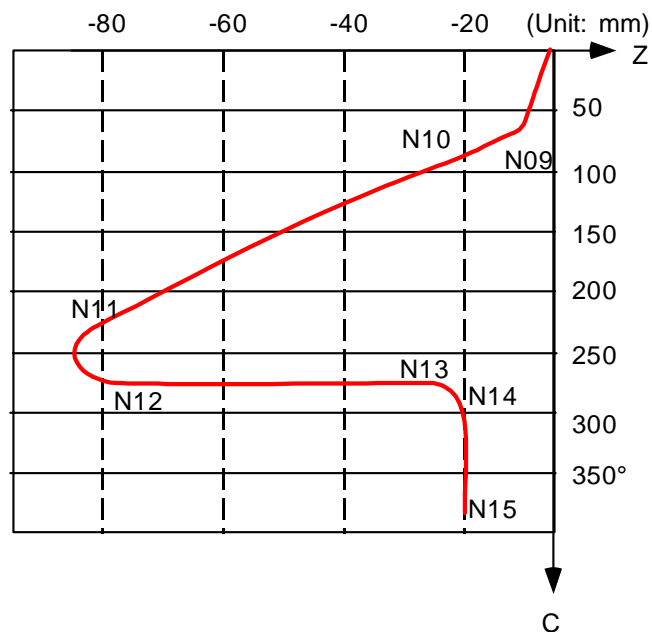
7. Feed Functions

7.1 Rapid traverse rate

Function and purpose

The rapid traverse rate can be set independently for each axis. The available speed ranges are from 1 mm/min to 240,000 mm/min (E60) or from 1 mm/min to 1,000,000 mm/min (E68) for input setting units of 1μm or 10μm. The upper limit is subject to the restrictions imposed by the machine specifications.
Refer to the specifications manual of the machine for the rapid traverse rate settings.
The feedrate is valid for the G00, G27, G28, G29, G30 and G60 commands.
Two paths are available for positioning: the interpolation type where the area from the start point to the end point is linearly interpolated or the non-interpolation type where movement proceeds at the maximum speed of each axis. The type is selected with parameter "#1086 G0Intp". The positioning time is the same for each type.

7.2 Cutting feed rate

Function and purpose

The cutting feedrate is assigned with address F and 8 digits (F8-digit direct designation).
The F8 digits are assigned with a decimal point for a 5-digit integer and a 3-digit fraction. The cutting feedrate is valid for the G01, G02, G03 and G33 commands.

(Examples)

Feedrate	
G1 X100. Y100. F200 ;	200.0mm/min F200. or F200.000 gives the same rate.
G1 X100. Y100. F123.4 ;	123.4mm/min
G1 X100. Y100. F56.789 ;	56.789mm/min

Speed range that can be commanded

(1) When using E60

Command mode	F command range	Feed rate command range	Remarks
mm/min	0.001 to 240000.000	0.01 to 240000.00 mm/min	
inch/min	0.0001 to 9448.8188	0.001 to 9448.818 inch/min	
°/min	0.001 to 240000.000	0.01 to 240000.00 °/min	

(2) When using E68

Command mode	F command range	Feed rate command range	Remarks
mm/min	0.001 to 1000000.000	0.01 to 1000000.00 mm/min	
inch/min	0.0001 to 39370.0787	0.001 to 100000.0000 inch/min	
°/min	0.001 to 1000000.000	0.01 to 1000000.00 °/min	

(Note 1) A program error (P62) results when there is no F command in the first cutting command (G01, G02, G03) after the power has been switched on.

7.3 F1-digit Feed

Function and purpose

By setting the F1-digit feed parameter, the feedrate which has been set to correspond to the 1-digit number following the F address serves as the command value.
When F0 is assigned, the rapid traverse rate is established and the speed is the same as for G00. (G modal does not change.)
When F1 to F5 is assigned, the feedrate set to correspond to the command serves as the command value.
The command greater than F6 is considered to be the normal cutting feedrate.
The F1-digit command is valid in a G01, G02, G03, G02.1 or G03.1 modal.
The F1-digit command can also be used for fixed cycle.

Detailed description

(1)   Use of both the F1-digit command and normal cutting feedrate command is possible when the F1-digit is valid.
  (Example 1)
        F0   Rapid traverse rate
        F1 to F5   F1-digit
        F6 or more   Normal cutting feedrate command
(2)   F1 to F5 are invalid in the G00 mode and the rapid traverse rate is established instead.
(3)   If F0 is used in the G02 or G03 mode, a program error (P121) will result.
(4)   When F1. to F5. (with decimal point) are assigned, the 1mm/min to 5mm/min direct commands are established instead of the F1-digit command.
(5)   When the commands are used with the millimeter or degree units, the feedrate set to correspond to F1 to F5 serves as the assigned speed mm (°)/min.
(6)   When the commands are used with inch units, one-tenth of the feedrate set correspond to F1 to F5 serves at the assigned speed inch/min.
(7)   During a F1-digit command, the F1-digit number and F1-digit command signal are output as the PLC signals.
(8) F1-digit and G commands

  (a)01 group G command in same block as F1-digit commands

	Executed feedrate	Modal display rate	G modal
G0F0 F0G0	Rapid traverse rate	0	G0
G0F1 F1G0	Rapid traverse rate	1	G0
G1F0 F0G1	Rapid traverse rate	0	G1
G1F1 F1G1	F1 contents	1	G1

  (b)F1-digit and unmodal commands may be assigned in the same block. In this case, the unmodal command is executed and at the same time the F1-digit modal command is updated.

67

7.4 Synchronous feed; G94, G95

Function and purpose

Using the G95 command, it is possible to assign the feed amount per rotation with an F code.
When this command is used, the rotary encoder must be attached to the spindle.
When the G94 command is issued the per-minute feed rate will return to the designated per-minute feed (asynchronous feed) mode.

Command format

G94;
G95;
G94   : Per-minute feed (mm/min) (asynchronous feed) (F1 = 1mm/min)
G95   : Per-revolution feed (mm/rev) (synchronous feed) (F1 = 0.01mm/rev)

The G95 command is a modal command and so it is valid until the G94 command (per-minute feed) is next assigned.

(1) The F code command range is as follows.
The movement amount per spindle revolution with synchronous feed (per-revolution feed) is assigned by the F code and the command range is as shown in the table below.
(a) When using E60

Input unit system	**0.001mm** Metric input		**0.0001inch** Inch input	
Command mode	Feed per minute	Feed per rotation	Feed per minute	Feed per rotation
Command address	F (mm/min)	E (mm/rev)	F (inch/min)	E (inch/rev)
Minimum command unit	1 (= 1.00), (1. = 1.00)	1 (= 0.01), (1. = 1.00)	1 (= 1.000), (1. = 1.000)	1 (= 0.001), (1. = 1.000)
Command range	0.01 to 1000000.00	0.001 to 999.999	0.001 to 100000.0000	0.0001 to 999.9999

(b) When using E68
Metric input

Input unit system	**B (0.001mm)**		**C (0.0001mm)**	
Command mode	Feed per minute	Feed per rotation	Feed per minute	Feed per rotation
Command address	F (mm/min)	E (mm/rev)	F (mm/min)	E (mm/rev)
Minimum command unit	1 (= 1.00), (1. = 1.00)	1 (= 0.01), (1. = 1.00)	1 (= 1.000), (1. = 1.000)	1 (= 0.01), (1. = 1.00)
Command range	0.01 to 1000000.00	0.001 to 999.999	0.001 to 100000.000	0.0001 to 99.9999

Inch input

Input unit system	**B (0.0001inch)**		**C (0.00001inch)**	
Command mode	Feed per minute	Feed per rotation	Feed per minute	Feed per rotation
Command address	F (inch/min)	E (inch/rev)	F (inch/min)	E (inch/rev)
Minimum command unit	1 (= 1.000), (1. = 1.000)	1 (= 0.001), (1. = 1.000)	1 (= 1.0000), (1. = 1.0000)	1 (= 0.001), (1. = 1.000)
Command range	0.001 to 100000.0000	0.0001 to 999.9999	0.0001 to 10000.00000	0.00001 to 99.99999

(2) The effective speed (actual movement speed of machine) under per-revolution feed conditions is given in the following formula (Formula 1).

$FC = F \times N \times OVR$ ..... (Formula 1)

Where FC = Effective rate (mm/min, inch/min)
       F   = Commanded feedrate (mm/rev, inch/rev)
       N   = Spindle speed (r/min)
       OVR = Cutting feed override

When a multiple number of axes have been commanded at the same time, the effective rate FC in formula 1 applies in the vector direction of the command.

(**Note 1)** The effective rate (mm/min or inch/min), which is produced by converting the commanded speed, the spindle speed and the cutting feed override into the per-minute speed, appears as the FC on the monitor 1. Screen of the setting and display unit.

(Note 2) When the above effective rate exceeds the cutting feed clamp rate, it is clamped at that clamp rate.

(Note 3) If the spindle speed is zero when synchronous feed is executed, operation alarm "105" results.

(Note 4) During machine lock high-speed processing, the rate will be 240,000 mm/min (or 9448 inch/min, 240,000 °/min) when using E60 or 1,000,000 mm/min (or 39,370 inch/min, 1,000,000 °/min) when using E68 regardless of the commanded speed and spindle speed. When high-speed processing is not undertaken, the rate will be the same as for non-machine lock conditions.

(Note 5) Under dry run conditions, asynchronous speed applies and movement results at the externally set rate (mm/min, inch/min, °/min).

(Note 6) The fixed cycle G84 (tapping cycle) and G74 (reverse tapping cycle) are executed to the feed mode that is already designated.

(Note 7) Whether asynchronous feed (G94) or synchronous feed (G95) is to be established when the power is switched on or when M02 or M30 is executed is set with parameter "#1074 I_Sync".

7.5 Feedrate designation and effects on control axes

Function and purpose

It has already been mentioned that a machine has a number of control axes. These control axes can be divided into linear axes which control linear movement and rotary axes which control rotary movement. The feedrate is designed to assign the displacement speed of these axes, and the effect exerted on the tool movement speed which poses problems during cutting differs according to when control is exercised over the linear axes or when it is exercised over the rotary axes.

The displacement amount for each axis is assigned separately for each axis by a value corresponding to the respective axis. The feedrate is not assigned for each axis but assigned as a single value. Therefore, when two or more axes are to be controlled simultaneously, it is necessary to understand how this will work for each of the axes involved.

The assignment of the feedrate is described with the following related items.

When controlling linear axes

Even when only one machine axis is to be controlled or there are two or more axes to be controlled simultaneously, the feed rate which is assigned by the F code functions as a linear speed in the tool advance direction.

(Example)   When the feedrate is designated as "f" and linear axes (X and Y) are to be controlled.

$$\text{Feedrate for X axis} = f \times \frac{x}{\sqrt{x^2 + y^2}}$$

$$\text{Feedrate for Y axis} = f \times \frac{y}{\sqrt{x^2 + y^2}}$$

P₂ (Tool end point)

Speed in this direction is "f"

P (Tool start point)
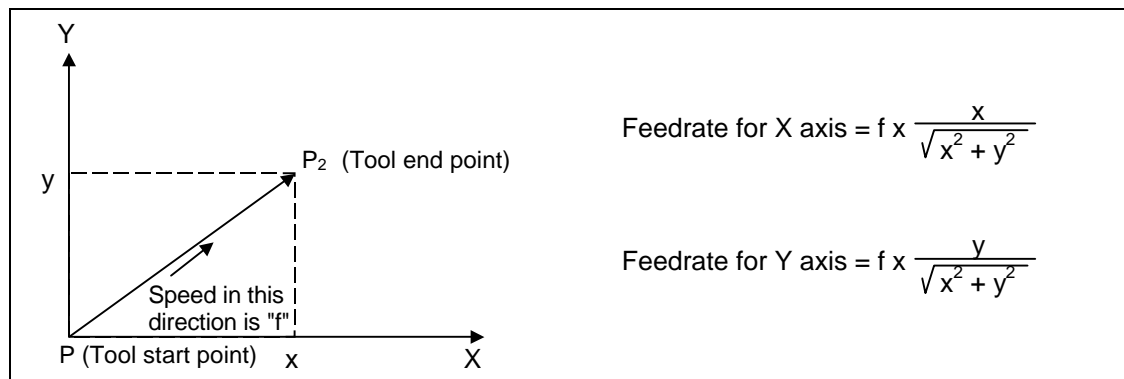
When only linear axes are to be controlled, it is sufficient to designate the cutting feed in the program. The feedrate for each axis is such that the designated rate is broken down into the components corresponding to the movement amounts.

(Note)   When the circular interpolation function is used and the tool is moved along the circumference of an arc by the linear control axis, the rate in the tool advance direction, or in other words the tangential direction, will be the feedrate designated in the program.

Linear speed is "f"
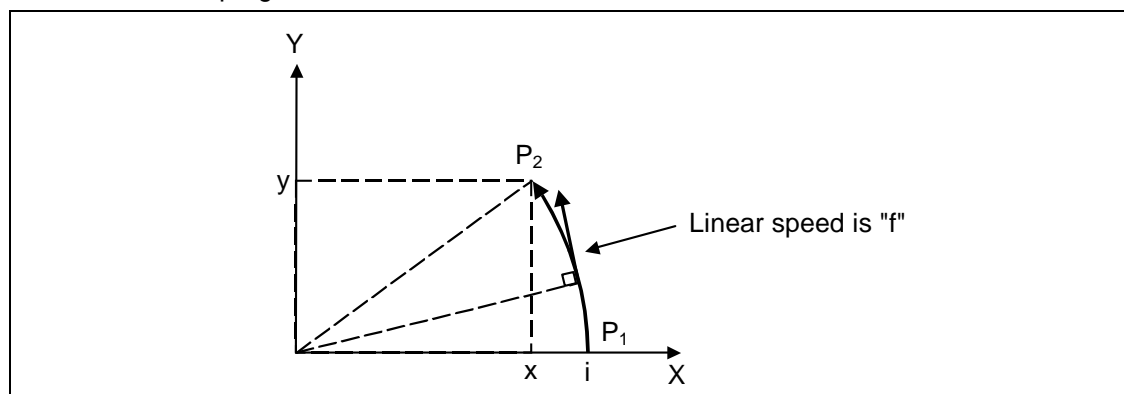
7. Feed Functions

7.5 Feedrate designation and effects on control axes

(Example) When the feedrate is designated as "f" and the linear axes (X and Y) are to be controlled using the circular interpolation function.
In this case, the feed rate of the X and Z axes will change along with the tool movement. However, the combined speed will always be maintained at the constant value "f".

When controlling rotary axes

When rotary axes are to be controlled, the designated feedrate functions as the rotary speed of the rotary axes or, in other words, as an angular speed.
Consequently, the cutting feed in the tool advance direction, or in other words the linear speed, varies according to the distance between the center of rotation and the tool. This distance must be borne in mind when designating the feedrate in the program.

(Example) When the feedrate is designated as "f" and rotary axis (CA) is to be controlled ("f" units = °/min)

P_2(tool end point)

Angular speed is "f"

Linear speed is : $\dfrac{\pi \cdot r \cdot f}{180}$

c

Rotation center

P_1 (tool start point)

r
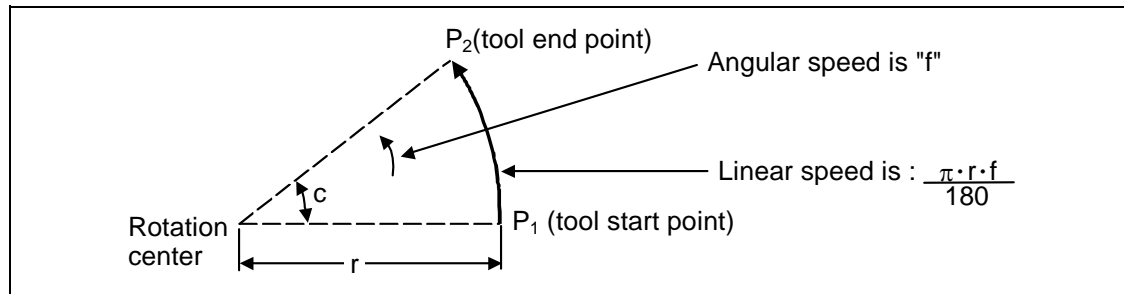
In this case, in order to make the cutting feed (linear feed) in the tool advance direction "fc" :

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

Therefore, the feedrate to be designated in the program must be :

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

When linear and rotary axes are to be controlled at the same time

The controller proceeds in exactly the same way whether linear or rotary axes are to be controlled. When a rotary axis is to be controlled, the numerical value assigned by the coordinate word (A, B, C) is the angle and the numerical values assigned by the feedrate (F) are all handled as linear speeds. In other words, 1° of the rotary axis is treated as being equivalent to 1mm of the linear axis. Consequently, when both linear and rotary axes are to be controlled simultaneously, the components for each axis of the numerical values assigned by F will be the same as previously described "When controlling linear axes". However, although in this case both the size and direction of the speed components based on linear axis control do not vary, the direction of the speed components based on rotary axis control will change along with the tool movement (their size will not change). This means, as a result, that the combined tool advance direction feedrate will vary along with the tool movement.

7. Feed Functions

(Example) When the feed rate is designated as "f" and Linear (X) and rotary © axes are to be controlled simultaneously.
In the X-axis incremental command value is "x" and the C-axis incremental command values is "c":

Size and direction are fixed for fx.
Size is fixed for fc but direction varies.
Both size and direction vary for ft.
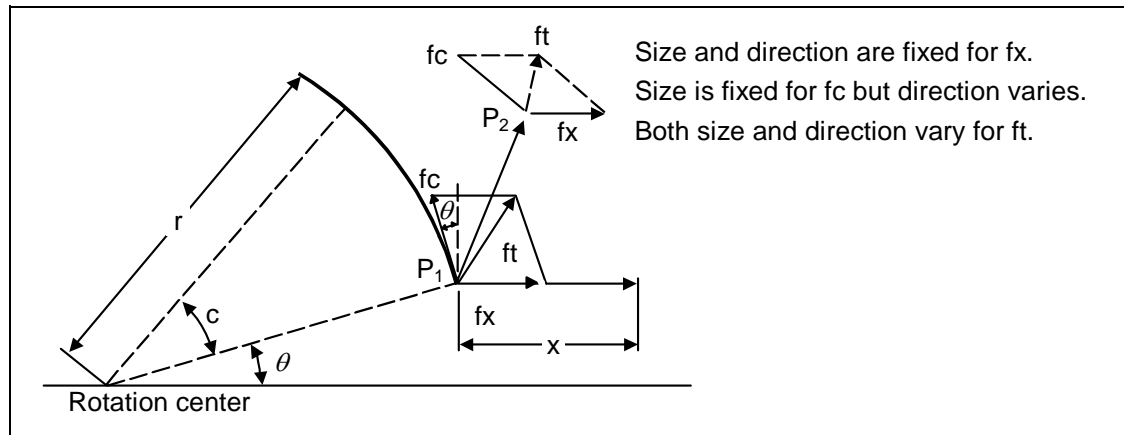
X-axis feedrate (linear speed) "fx" and C-axis feedrate (angular speed) "ω" are expressed as:

$$fx = f \times \frac{x}{\sqrt{x^2 + c^2}} \quad \text{...} \quad (1)$$

$$\omega = f \times \frac{c}{\sqrt{x^2 + c^2}} \quad \text{...} \quad (2)$$

Linear speed "fc" based on C-axis control is expressed as:

$$fc = \omega \times \frac{\pi \times r}{180} \quad \text{...} \quad (3)$$

If the speed in the tool advance direction at start point P1 is "ft" and the component speeds in the X-axis and Y-axis directions are "ftx" and "fty", respectively, then these can be expressed as:

$$ftx = -r\sin\left(\frac{\pi}{180}\,\theta\right) \times \frac{\pi}{180}\,\omega + fx \quad \text{...} \quad (4)$$

$$fty = -r\cos\left(\frac{\pi}{180}\,\theta\right) \times \frac{\pi}{180}\,\omega \quad \text{...} \quad (5)$$

Where r is the distance between center of rotation and tool (in mm units), and θ is the angle between the P1 point and the X axis at the center of rotation (in units °).
The combined speed "ft" according to (1), (2), (3), (4) and (5) is:

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

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

Consequently, feedrate "f" designated by the program must be as follows:

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

"ft" in formula (6) is the speed at the P1 point and the value of θ changes as the C axis rotates, which means that the value of "ft" will also change.
Consequently, in order to keep the cutting feed "ft" as constant as possible the angle of rotation which is designated in one block must be reduced to as low as possible and the extent of the change in the θ value must be minimized.

7.6 Exact stop check; G09
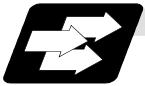
Function and purpose

In order to prevent roundness during corner cutting and machine shock when the tool feedrate changes suddenly, there are times when it is desirable to start the commands in the following block once the in-position state after the machine has decelerated and stopped or the elapsing of the deceleration check time has been checked. The exact stop check function is designed to accomplish this purpose.

Whether to control with the deceleration check time or with the in-position state can be selected with the parameter. (Refer to the section "7.8 Deceleration Check".)

The in-position width is set into the servo parameter "#2224 sv024" or "#2077 G0inps", "#2078 G1inps".

Command format

G09 ;

The exact stop check command G09 has an effect only with the cutting command (G01 - G03) in its particular block.

Example of program

N001 G09 G01 X100.000 F150 ;	The following block is started once the deceleration check time or in-position state has been checked after the machine has decelerated and stopped.
N002 Y100.000 ;	

Solid line indicates speed pattern with G09 command.
Broken line indicates speed pattern without G09 command.

Fig. 1   Exact stop check result
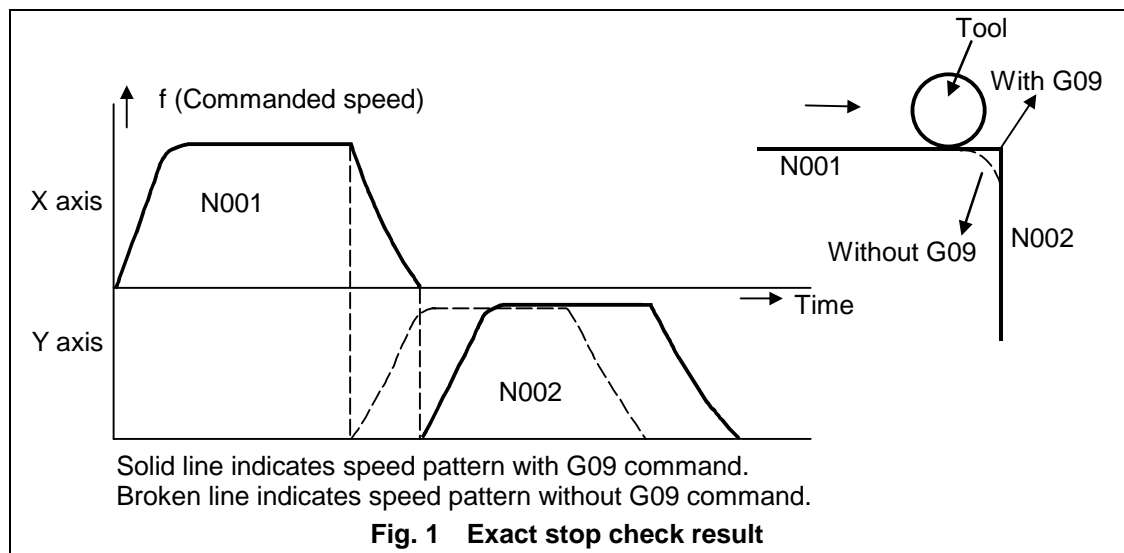
Detailed description

(1) With continuous cutting feed

Previous block

Next block

Ts

Fig. 2 Continuous cutting feed command

(2) With cutting feed in-position check

Previous block

Next block

Lc (in-position width)

Ts   Ts
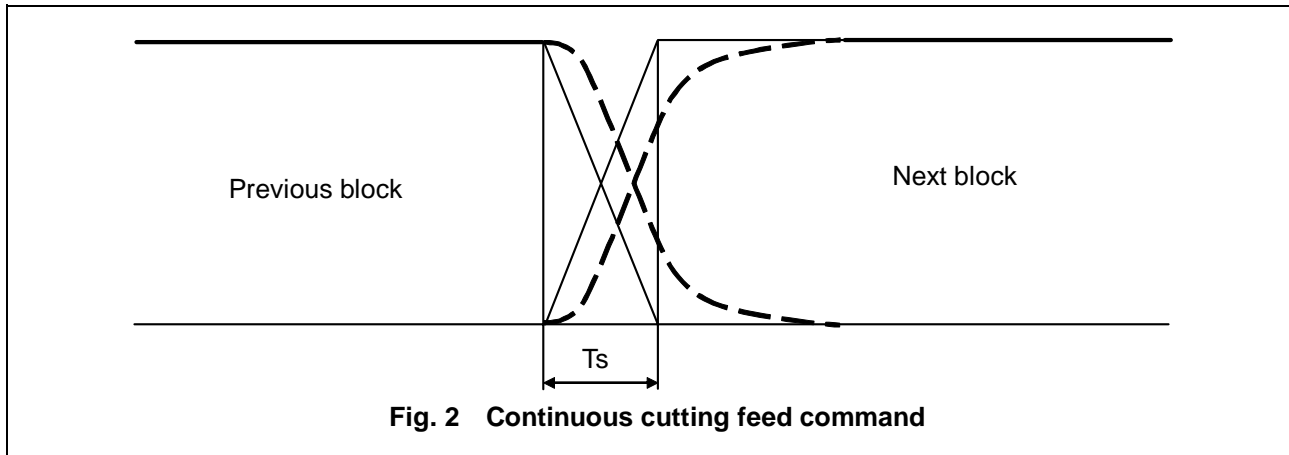
Fig. 3 Block joint with cutting feed in-position check
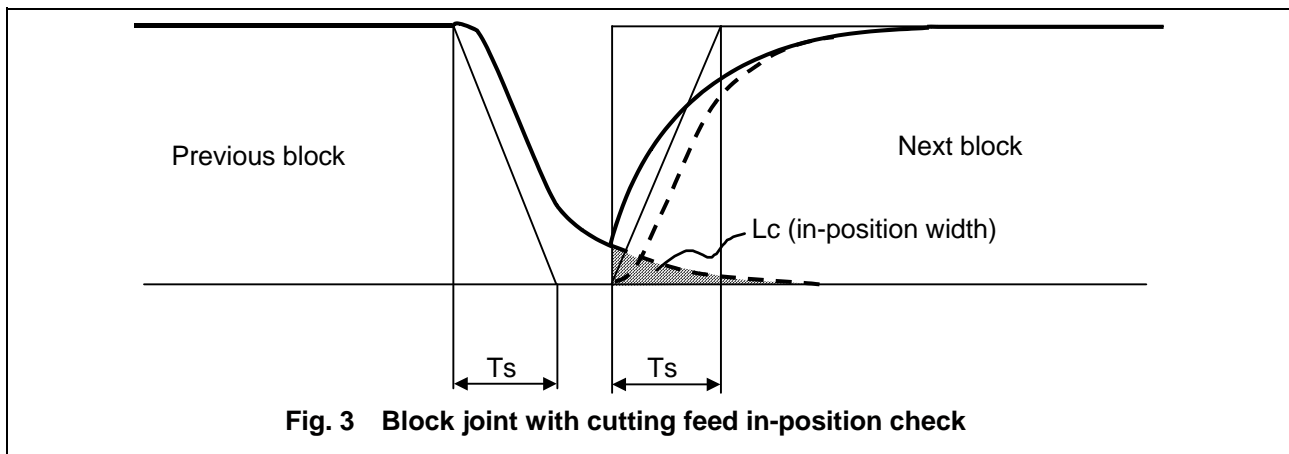
In Figs. 2 and 3 :

Tc = Cutting feed acceleration/deceleration time constant

Lc = In-position width

As shown in Fig. 3, the in-position width "Lc" can be set in the "#2224 sv024" servo parameter as the remaining distance (shaded area in Fig. 3) of the previous block when the next block is started. The in-position width is designed to reduce the roundness at the workpiece corners to below the constant value.

Lc

Next block

Previous block
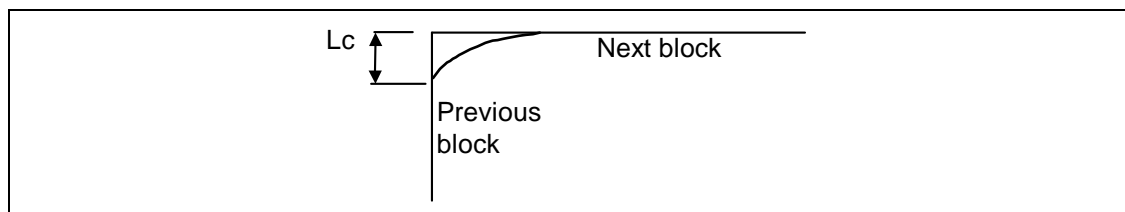
To eliminate corner roundness, set the value as small as possible to servo parameter "#2224 SV024" and perform an in-position check or assign the dwell command (G04) between blocks.

(3)   With deceleration check

(a)   With linear acceleration/deceleration
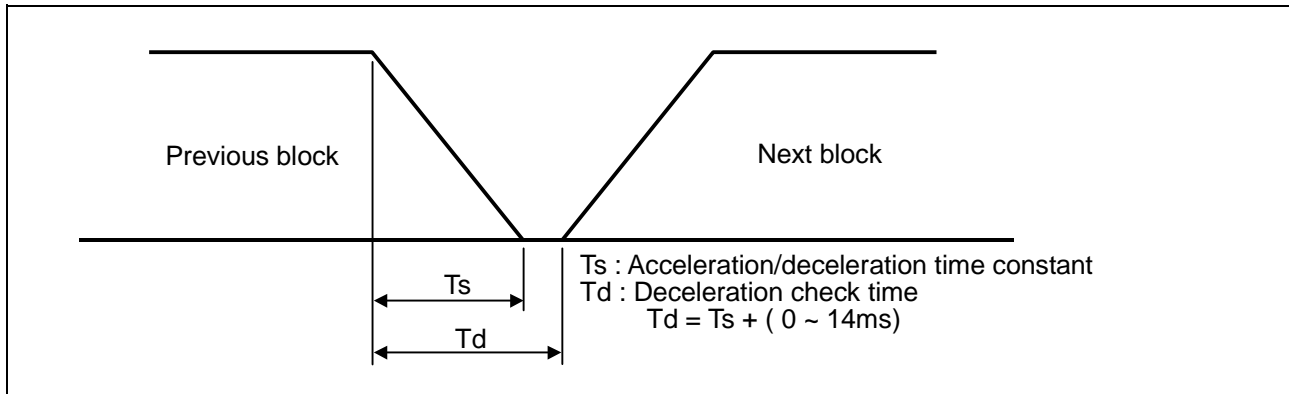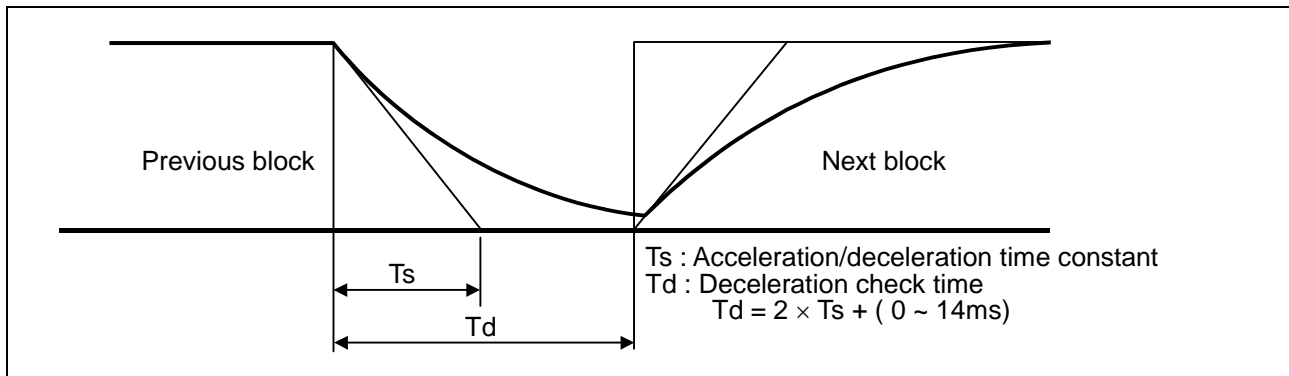
Previous block

Next block

Ts : Acceleration/deceleration time constant
Td : Deceleration check time
    Td = Ts + (0 ~ 14ms)

(b)   With exponential acceleration/deceleration

Previous block

Next block

Ts : Acceleration/deceleration time constant
Td : Deceleration check time
    $Td = 2 \times Ts + (0 ~ 14ms)$

(c)   With exponential acceleration/linear deceleration

Previous block

Next block

2 x Ts

Td

Ts

Ts : Acceleration/deceleration time constant
Td : Deceleration check time
    $Td = 2 \times Ts + (0 ~ 14ms)$
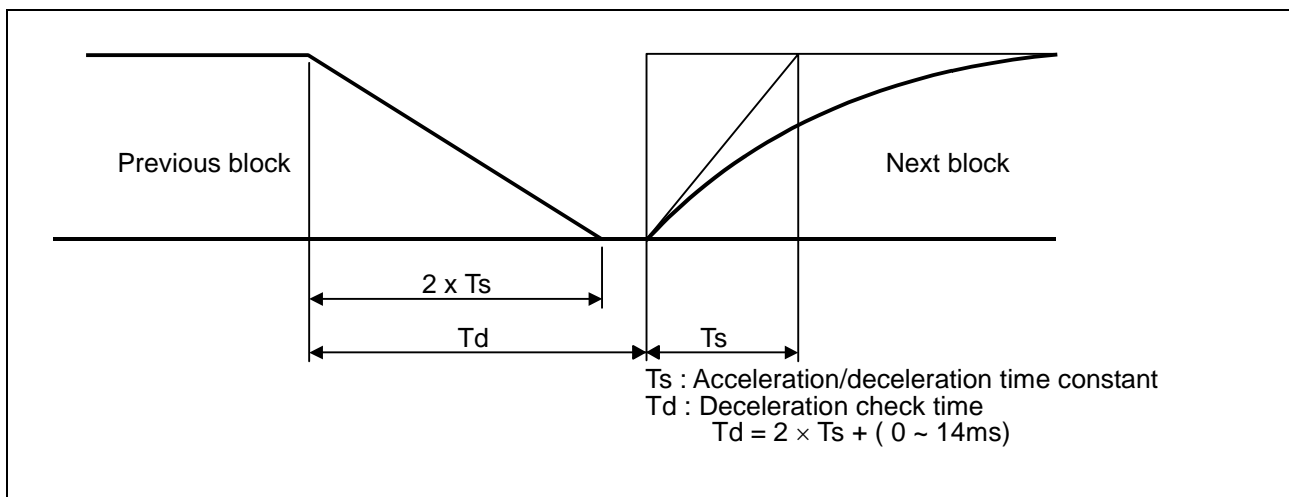
The time required for the deceleration check during cutting feed is the longest among the cutting feed deceleration check times of each axis determined by the cutting feed acceleration/deceleration time constants and by the cutting feed acceleration/deceleration mode of the axes commanded simultaneously.

7.7 Exact stop check mode ; G61
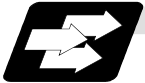
Function and purpose

Whereas the G09 exact stop check command checks the in-position status only for the block in which the command has been assigned, the G61 command functions as a modal. This means that deceleration will apply at the end points of each block to all the cutting commands (G01 to G03) subsequent to G61 and that the in-position status will be checked. G61 is released by high-accuracy control mode (G61.1), automatic corner override (G62), tapping mode (G63), or cutting mode (G64).

Command format

G61 ;

In-position check is executed in the G61 block, and thereafter, the in-position check is executed at the end of the cutting command block is executed until the check mode is canceled.

7.8 Deceleration check

Function and purpose

The deceleration check is a function that determines the method of the check at the completion of the axis movement block's movement.
The deceleration check includes the in-position check and commanded speed check method.
The G0 and G1 deceleration check method combination can be selected.
(Refer to section "Deceleration check combination".)
With this function, the deceleration check in the reverse direction of G1 $\rightarrow$ G0 or G1 $\rightarrow$ G1 can be changed by changing the parameter setting.

(1) Types of deceleration check

Commanded speed check	
With the commanded speed check, the completion of deceleration is judged when the command to the motor is completed	Command to motor / movement of motor / Deceleration start point / Judges the stop here
In-position check	
With the in-position check, the completion of deceleration is judged when the motor moves to the in-position width designated with the parameter.	G0/G1 in-position width / Deceleration start point / Judges the stop here

76

(2)   Designating deceleration check

The deceleration check by designating a parameter includes "deceleration check specification type 1" and "deceleration check specification type 2". The setting is selected with the parameter "#1306 InpsTyp".

(a) Deceleration check specification type 1 ("#1306 InpsTyp" = 0)

The G0 and G1 deceleration check method can be selected with the base specification parameter deceleration check method 1 (#1193 inpos) and "deceleration check method 2" (#1223 aux07/bit1).

Parameter	Rapid traverse command
Inpos (#1193)	G0→XX (G0+G9→XX)
0	Command deceleration check
1	In-position check

Parameter	Other than rapid traverse command (G1 : other than G0 command)	
AUX07/BIT-1 (#1223/BIT-1)	G1+G9→XX	G0→XX
0	Command deceleration check	No deceleration check
1	In-position check	No deceleration check

(Note) XX expresses all commands.

(b) Deceleration check specification type 2 ("#1306 InpsTyp" = 1)

Rapid traverse and cutting in-position are designated with the "inpos" parameter.

Parameter	Command block		
#1193 Inpos	**G0**	**G1+G9**	**G1**
0	Command deceleration check	Command deceleration check	No deceleration check
1	In-position check	In-position check	No deceleration check

(Note) "G0" means the rapid traverse, and "G1" means the cutting feed.

7.8.1 G1 → G0 deceleration check

Detailed description

(1)   In G1 → G0 continuous blocks, the parameter "#1502 G0Ipfg" can be changed to change the deceleration check in the reverse direction.

Command deceleration

Example of program

When there is a deceleration check in the movement of several axes:

(1)	G91 G1 X100. Y100. F4000 ; G0 X-100. Y120. ;	A deceleration check is carried out, because the X axis moves in the reverse direction in the program above.

(2)	G91 G1 X100. Y-100. F4000 ; G0 X80. Y100. ;	A deceleration check is carried out, because the Y axis moves in the reverse direction in the program above.

(3)	G90 G1 X100. Y100. F4000 ; G0 X80. Y120. ;	A deceleration check is carried out, because the X axis moves in the reverse direction in the program above. (When the program start position is X0 Y0)

(4)	G91 G1 X100. Y100. F4000 ; G0 X100. Y100. ;	A deceleration check is not carried out, because both the X axis and the Y axis move in the same direction in the program above.

(5)	G91 G1 X100. Y80. F4000 ; G0 X80. ;	A deceleration check is not carried out, because the X axis moves in the same direction, and there is no Y axis movement command in the program above.

7.8.2 G1 → G1 deceleration check

Detailed description

(1) In G1 → G1 continuous blocks, the parameter "#1503 G1Ipfg" can be changed to change the deceleration check of the reverse direction.

	Same direction	**Reverse direction**
G1Ipfg: 0	G1 G1	G1 G1
G1Ipfg: 1	G1 G1	G1 G1 Command deceleration

Example of program

When there is a deceleration check in the movement of several axes:

(1)	G91 G1 X100. Y100. F4000 ; G1 X-100. Y120. ;	A deceleration check is carried out, because the X axis moves in the reverse direction in the program above.

(2)	G91 G1 X100. Y-100. F4000 ; G1 X80. Y100. ;	A deceleration check is carried out, because the Y axis moves in the reverse direction in the program above.

(3)	G90 G1 X100. Y100. F4000 ; G1 X80. Y120. ;	A deceleration check is carried out, because the X axis moves in the reverse direction in the program above. (When the program start position is X0 Y0)

(4)	G91 G1 X100. Y100. F4000 ; G1 X100. Y100. ;	A deceleration check is not carried out, because both the X axis and the Y axis move in the same direction in the program above.

(5)	G91 G1 X100. Y80. F4000 ; G1 X80. ;	A deceleration check is not carried out, because the X axis moves in the same direction, and there is no Y axis movement command in the program above.

7.9 Automatic corner override ; G62

Function and purpose

With tool radius compensation, this function reduces the load during inside cutting of automatic corner R, or during inside corner cutting, by automatically applying override to the feed rate.
Automatic corner override is valid until the tool radius compensation cancel (G40), exact stop check mode (G61), high-accuracy control mode (G61.1), tapping mode (G63) and cutting mode (G64) command is issued.

Command format

G62 ;

Machining inside corners

When cutting an inside corner as in Fig. 1, the machining allowance amount increases and a greater load is applied to the tool. To remedy this, override is applied automatically within the corner set range, the feedrate is reduced, the increase in the load is reduced and cutting is performed effectively.
However, this function is valid only when finished shapes are programmed.
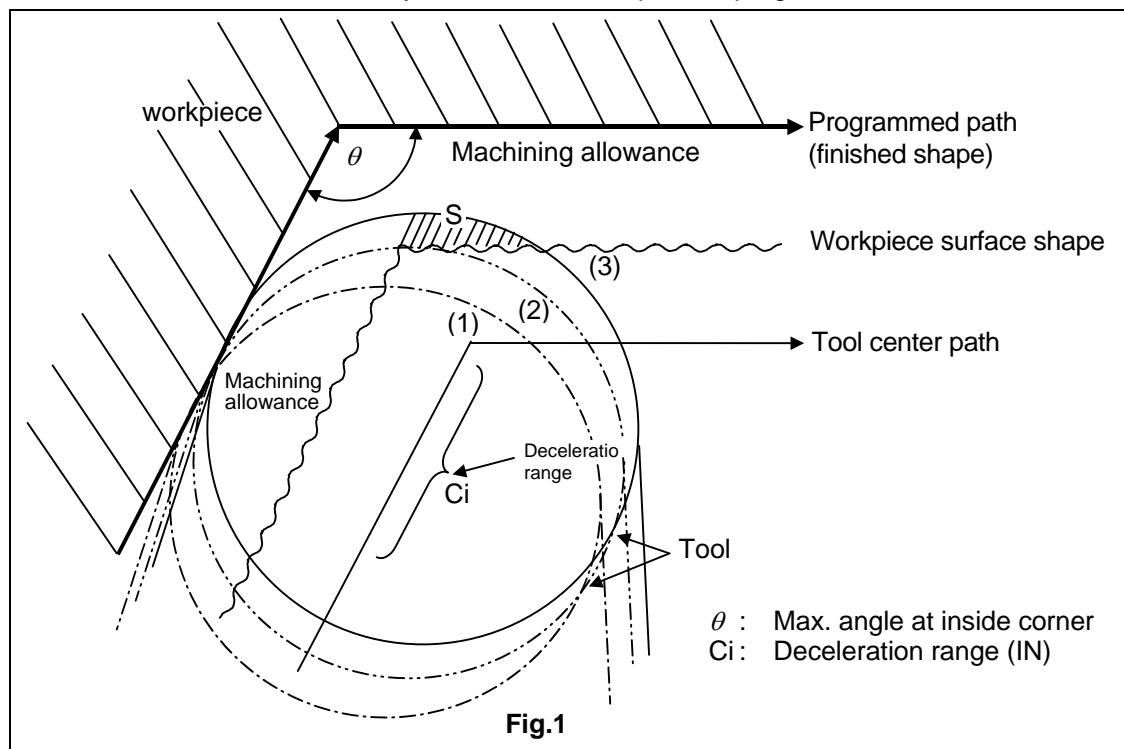
Fig.1

(1) Operation

(a) When automatic corner override is not to be applied :
When the tool moves in the order of (1) → (2) → (3) in Fig. 1, the machining allowance at (3) increases by an amount equivalent to the area of shaded section S and so the tool load increases.

(b) When automatic corner override is to be applied :
When the inside corner angle θ in Fig. 1 is less than the angle set in the parameter, the override set into the parameter is automatically applied in the deceleration range Ci.

(2) Parameter setting

The following parameters are set into the machining parameters :

#	Parameter	Setting range
#8007	OVERRIDE	0 to 100%
#8008	MAX ANGLE	0 to 180°
#8009	DSC. ZONE	0 to 99999.999mm or 0 to 3937.000 inches

Refer to the Operation Manual for details on the setting method.

Automatic corner
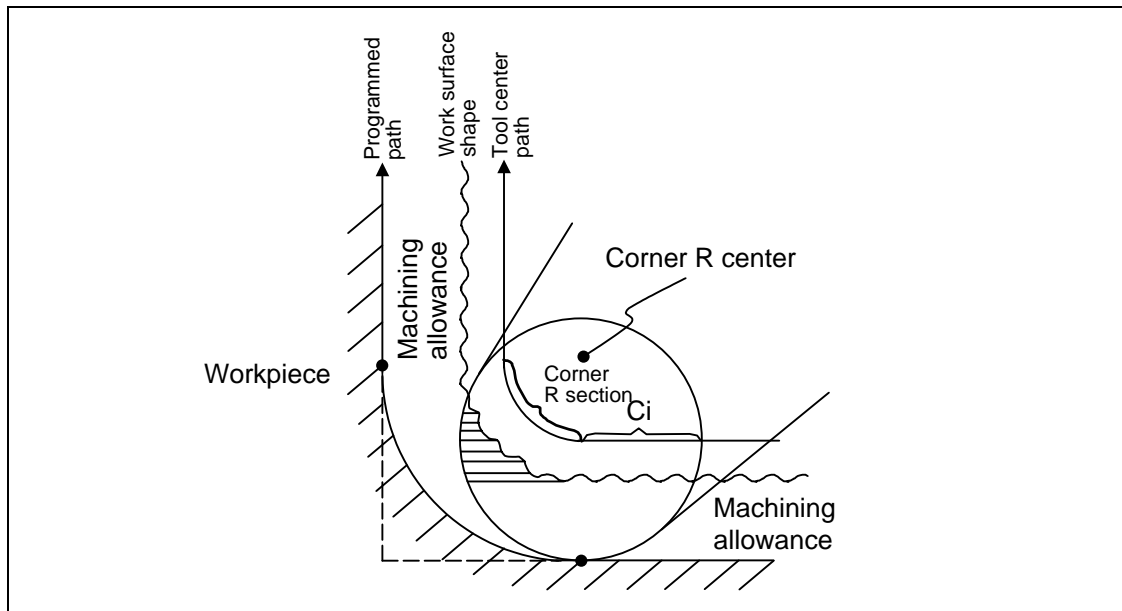
(1) The override set in the parameter is automatically applied at the deceleration range Ci and corner R section for inside offset with automatic corner R. (There is no angle check.)

7. Feed Functions

Application example

(1)   Line - line corner
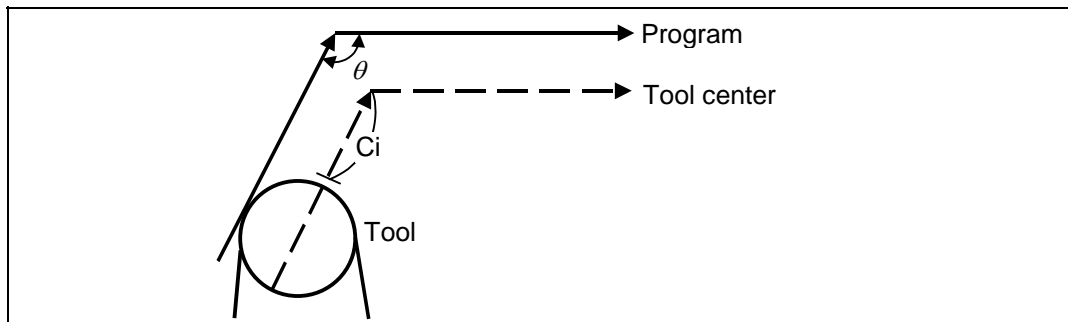
The override set in the parameter is applied at Ci.

(2)   Line - arc (outside) corner
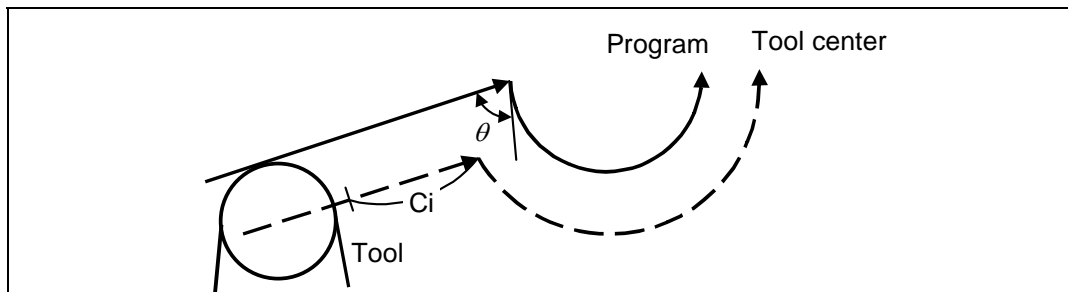
The override set in the parameter is applied at Ci.

(3)   Arc (inside offset) - line corner
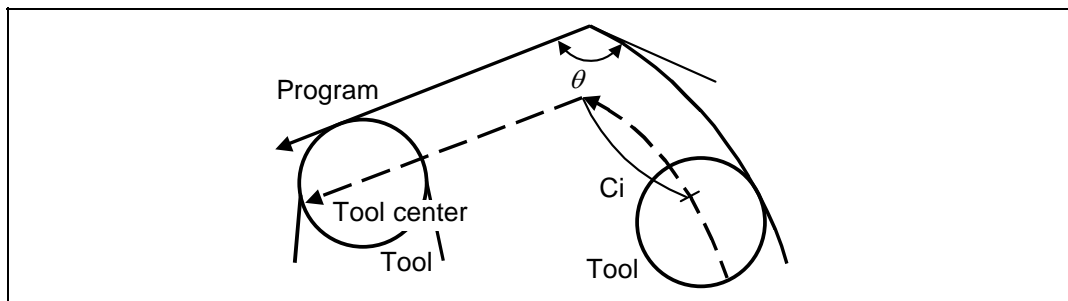
The override set in the parameter is applied at Ci.
(**Note**)   The deceleration range Ci where the override is applied is the length of the arc with an arc command.

(4)   Arc (inside offset) - arc (outside offset) corner
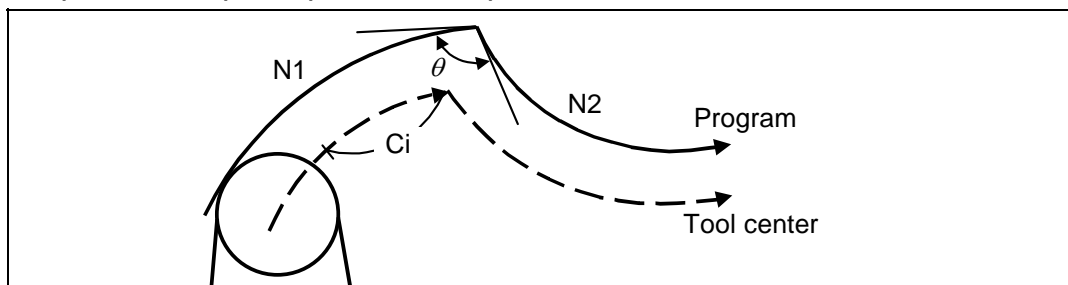
The override set in the parameter is applied at Ci.

Relation with other functions

Function	Override at corner
Cutting feed override	Automatic corner override is applied after cutting feed override has been applied.
Override cancel	Automatic corner override is not canceled by override cancel.
Speed clamp	Valid after automatic corner override
Dry run	Automatic corner override is invalid.
Synchronous feed	Automatic corner override is applied to the synchronous feedrate.
Thread cutting	Automatic corner override is invalid.
G31 skip	Program error results with G31 command during tool radius compensation.
Machine lock	Valid
Machine lock high speed	Automatic corner override is invalid.
G00	Invalid
G01	Valid
G02, G03	Valid

Precautions

(1) Automatic corner override is valid only in the G01, G02, and G03 modes; it is not effective in the G00 mode. When switching from the G00 mode to the G01 (or G02 or G03) mode at a corner (or vice versa), automatic corner override will not be applied at that corner in the G00 block.

(2) Even if the automatic corner override mode is entered, the automatic corner override will not be applied until the tool diameter compensation mode is entered.

(3) Automatic corner override will not be applied on a corner where the tool diameter compensation is started or canceled.
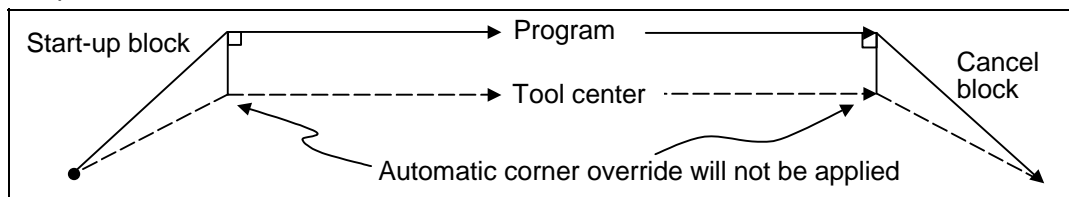
(4) Automatic corner override will not be applied on a corner where the tool diameter compensation I, K vector command is issued.
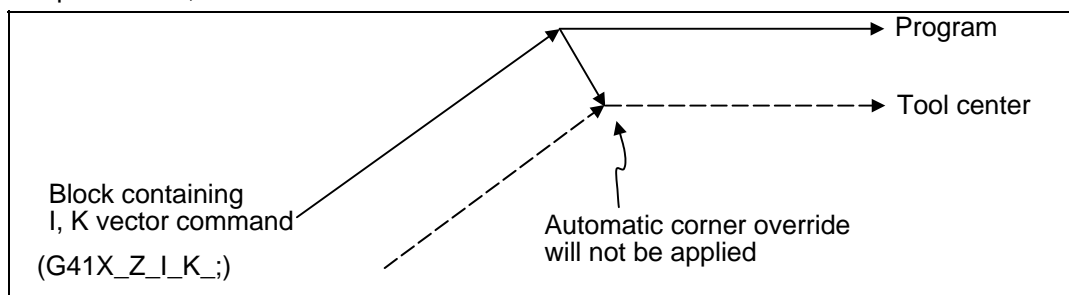
(5) Automatic corner override will not be applied when intersection calculation cannot be executed.
Intersection calculation cannot be executed in the following case.
(a) When the movement command block does not continue for four or more times.

(6) The deceleration range with an arc command is the length of the arc.

(7) The inside corner angle, as set by parameter, is the angle on the programmed path.

(8) Automatic corner override will not be applied when the maximum angle in the parameter is set to 0 or 180.

(9) Automatic corner override will not be applied when the override in the parameter is set to 0 or 100.

84

7.10 Tapping mode ; G63

Function and purpose

The G63 command allows the control mode best suited for tapping to be entered, as indicated below :
  1) Cutting override is fixed at 100%.
  2) Deceleration commands at joints between blocks are invalid.
  3) Feed hold is invalid.
  4) Single block is invalid.
  5) In-tapping mode signal is output.
G63 is released by the exact stop check mode (G61), high-accuracy control (G61.1), automatic corner override (G62), cutting mode (G64) command.

Command format

G63 ;

7.11 Cutting mode ; G64

Function and purpose

The G64 command allows the cutting mode in which smooth cutting surfaces are obtained to be established. Unlike the exact stop check mode (G61), the next block is executed continuously with the machine not decelerating and stopping between cutting feed blocks in this mode.
G64 is released by the exact stop mode (G61), high-accuracy control (G61.1), automatic corner override (G62), tapping mode (G63) command.
This cutting mode is established in the initialized status.

Command format

G64 ;

85

8. Dwell

The G04 command can delay the start of the next block.

8.1 Per-second dwell ; G04

Function and purpose

The machine movement is temporarily stopped by the program command to make the waiting time state. Therefore, the start of the next block can be delayed. The waiting time state can be canceled by inputting the skip signal.

Command format

| G04 X__; or G04 P__; |
| X, P ········Dwell time |

The input command unit for the dwell time depends on the parameter.

Detailed description

(1) When designating the dwell time with X, the decimal point command is valid.

(2) When designating the dwell time with P, the availability of the decimal point command can be selected with the parameter (#8112). When the decimal point command is invalid in the parameter setting, the command below the decimal point issued with P is ignored.

(3) When the decimal point command is valid or invalid, the dwell time command range is as follows.

Command range when the decimal point command is valid	Command range when the decimal point command is invalid
0.001 ~ 99999.999 (s)	1 ~ 99999999 (ms)

(4) The dwell time setting unit applied when there is no decimal point can be made 1s by setting 1 in the parameter "#1078 Decpt2". This is effect only for X and P for which the decimal command is valid.

(5) When a cutting command is in the previous block, the dwell command starts calculating the dwell time after the machine has decelerated and stopped. When it is commanded in the same block as an M, S, T or B command, the calculation starts simultaneously.

(6) The dwell function is valid during interlock.

(7) Dwell is valid even for machine lock.

(8) The dwell can be canceled by setting the parameter "#1173 dwlskp" beforehand. If the set skip signal is input during the dwell time, the remaining time is discarded, and the following block will be executed.

86
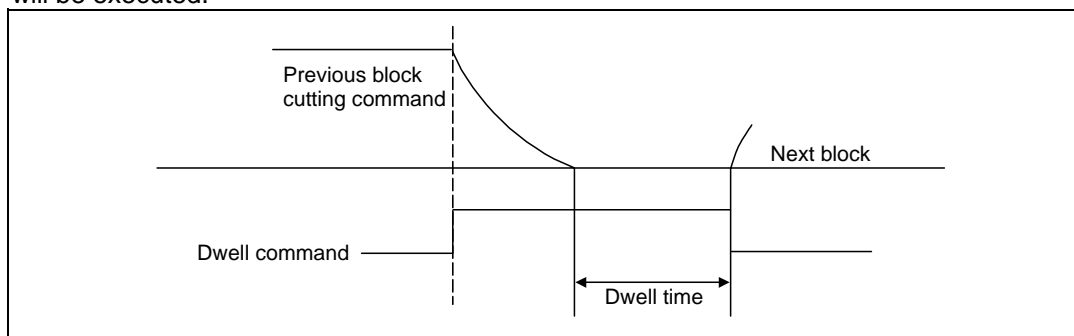
8. Dwell

Example of program

Command	Dwell time [sec]			
	#1078 Decpt2 = 0		**#1078 Decpt2 = 1**	
	DECIMAL PNT-N	**DECIMAL PNT-P**	**DECIMAL PNT-N**	**DECIMAL PNT-P**
G04 X500 ;	0.5		500	
G04 X5000 ;	5		5000	
G04 X5. ;	5		5	
G04 X#100 ;	1000		1000	
G04 P5000 ;	5		5	5000
G04 P12.345 ;	0.012	12.345	0.012	12.345
G04 P#100 ;	1	1000	1	1000

(Note 1) The above examples are the results under the following conditions.
- Input setting unit 0.001mm or 0.0001inch
- #100 = 1000 ;

(Note 2) "DECIMAL PNT-P" is a control parameter (#8112).

(Note 3) If the input setting unit is "C"(0.0001mm), the X before G04 will be multiplied by 10. For example for "X5. G04 ;", the dwell time will be 50 sec.

Precautions and restrictions

(1) When using this function, command X after G04 in order to make sure that the dwell is based on X.

9. Miscellaneous Functions

9.1 Miscellaneous functions (M8-digits BCD)
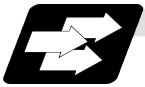
Function and purpose

The miscellaneous (M) functions are also known as auxiliary functions, and they include such numerically controlled machine functions as spindle forward and reverse rotation, operation stop and coolant ON/OFF.

These functions are designated by an 8-digit number (0 to 99999999) following the address M with this controller, and up to 4 groups can be commanded in a single block.

(Example) G00 Xx Mm1 Mm2 Mm3 Mm4 ;

When five or more commands are issued, only the last four will be valid.

The output signal is an 8-digit BCD code and start signal.

The six commands of M00, M01, M02, M30, M96, M97, M98 and M99 are used as auxiliary commands for specific objectives and so they cannot be used as general auxiliary commands. This therefore leaves 92 miscellaneous functions which are usable as such commands. Reference should be made to the instructions issued by the machine manufacturer for the actual correspondence between the functions and numerical values.

When the M00, M01, M02, M96, M97 and M30 functions are used, the next block is not read into the pre-read buffer due to pre-read inhibiting.

If the M function is designated in the same block as a movement command, the commands may be executed in either of the following two orders. Which of these sequences actually applies depends on the machine specifications.

(1) The M function is executed after the movement command.

(2) The M function is executed at the same time as the movement command.

Processing and completion sequences are required in each case for all M commands except M96, M97, M98 and M99.

The 8M functions used for specific purposes will now be described.

Program stop : M00

When the tape reader has read this function, it stops reading the next block. As far as the NC system's functions are concerned, only the tape reading is stopped. Whether such machine functions as the spindle rotation and coolant supply are stopped or not differs according to the machine in question.

Re-start is enabled by pressing the automatic start button on the machine operation board.

Whether resetting can be initiated by M00 depends on the machine specifications.

9. Miscellaneous Functions

9.1 Miscellaneous functions (M8-digits BCD)

Optional stop : M01

If the tape reader reads the M01 command when the optional stop switch on the machine operation board is ON, it will stop and the same effect as with the M00 function will apply.
If the optional stop switch is OFF, the M01 command is ignored.
(Example)

:
 N10 G00 X1000 ;
 N11 M01 ;
 N12 G01 X2000 Z3000 F600 ;
 :
	Optional stop switch status and operation
	    Stops at N11 when switch is ON
	    Next command (N12) is executed without
	    stopping at N11 when switch is OFF

Program end : M02 or M30

This command is normally used in the final block for completing the machining, and so it is primarily used for tape rewinding. Whether the tape is actually rewound or not depends on the machine specifications.
Depending on the machine specifications, the system is reset by the M02 or M30 command upon completion of tape rewinding and any other commands issued in the same block.
(Although the contents of the command position display counter are not cleared by this reset action, the modal commands and compensation amounts are canceled.)
The next operation stops when the rewinding operation is completed (the in-automatic operation lamp goes off). To restart the unit, the automatic start button must be pressed or similar steps must be taken.
When the program is restarted after M02 and M30 are completed, if the first movement command is designated only with a coordinate word, the interpolation mode will function when the program ends. It is recommended that a G function always be designated for the movement command designated first.

(Note 1) Independent signals are also output respectively for the M00, M01, M02 and M30 commands and these outputs are each reset by pressing the reset key.

(Note 2) M02 or M30 can be assigned by manual data input (MDI). At this time, commands can be issued simultaneously with other commands just as with the tape.

Macro interrupt: M96, M97

M96 and M97 are M codes for user macro interrupt control.
The M code for user macro interrupt control is processed internally, and is not output externally.
To use M96 and M97 as an auxiliary code, change the setting to another M code with the parameter (#1109 subs_M and #1110 M96_M, #1111 M97_M).

Subprogram call/completion : M98, M99

These commands are used as the return instructions from branch destination subprograms and branches to subprograms.
M98 and M99 are processed internally and so M code signals and strobe signals are not output.

Internal processing with M00/M01/M02/M30 commands

Internal processing suspends pre-reading when the M00, M01, M02 or M30 command has been read. Other tape rewinding operations and the initialization of modals by resetting differ according the machine specifications.

9.2 Secondary miscellaneous functions (B8-digits, A8 or C8-digits)

Function and purpose

These serve to assign the indexing table positioning and other such functions. In this controller, they are assigned by an 8-digit number from 0 to 99999999 following address A, B or C. The machine tool builder determines which codes correspond to which positions.

The output signal is BCD signal (8-digit) and start signal.

If the A, B or C function is designated in the same block as a movement command, the commands may be executed in either of the following two orders. The machine specifications determine which sequence applies.

(1)   The A, B or C function is executed after the movement command.

(2)   The A, B or C function is executed simultaneously with the movement command.

Processing and completion sequences are required for all secondary miscellaneous functions.
The table below given the various address combinations. It is not possible to use an address which is the same for the axis name of an additional axis and secondary miscellaneous function.

Secondary miscellaneous function \ Additional axis name	A	B	C
A	×	○	○
B	○	×	○
C	○	○	×

(Note)   When A has been assigned as the secondary miscellaneous function address, the following commands cannot be used.

(1) Linear angle commands (,A can be used.)

(2) Geometric commands

90

9.3 Index table indexing   [E68]

Function and purpose

Index table indexing can be carried out by setting the index axis. The indexing command only issues the indexing angle to the axis set for indexing. It is not necessary to command special M codes for table clamping and unclamping, thus simplifying the program.

Detailed description

The index table index function carries out operations as follows.

(Example) G00 B90 ;

　　　　　　　　　　→　The axis that was designated as the index axis with parameter "#2076 index_x".

(1) Set the "index_x" parameter (#2076) for the axis in which index table indexing will be carried out to "1".

(2) The movement command (either absolute or incremental) for the selected axis is executed with the program command.

(3) An unclamp process are carried out before the axis movement.

(4) The commanded axis movement starts after the unclamp process completes.

(5) The clamp process is carried out after the movement is completed.

(6) The next block is processed after the unclamp process completes.

<Operation time chart>

Program command	G0 B90. ;
Unclamp command	
Unclamp completed	
B axis movement	

T10 FIN WAIT 0800　　　　　　　　T10 FIN WAIT 0800

91
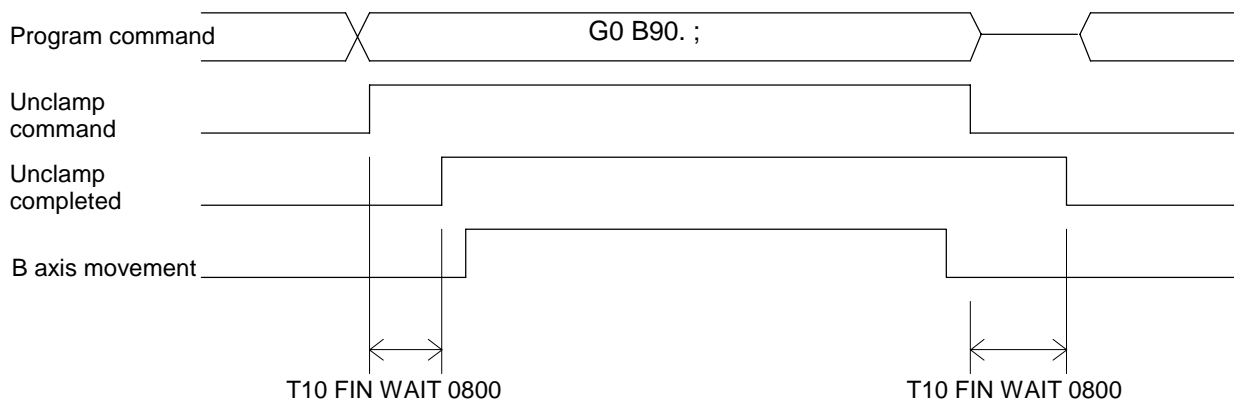
Precautions

(1)   Several axes can be set as index table indexing axes.

(2)   The movement speed of index table indexing axes follows the feedrate of the modal (G0/G1) at that time.

(3)   The unclamp process for the indexing axes is also issued when the index table indexing axes are commanded in the same block as other axes. Thus, the movement of other axes commanded in the same block is not carried out until the unclamp process completes.
Note that the movement of other axes commanded in the same block is carried out for non-interpolation commands.

(4)   Index table indexing axes are used as normal rotation axes, but this function performs an unclamp process even for linear axes.

(5)   If some error that makes unclamp command OFF occurs during indexing axis movement in automatic operation, the unclamp state will be remained, and the indexing axis will execute a deceleration stop.
Other axes commanded in the same block will also execute a deceleration stop, except for non-interpolation commands.

(6)   If the axis movement is interrupted by an interlock, etc., during indexing axis movement, the unclamp state will be remained.

(7)   The clamp and unclamp process are not executed when the movement commands of the index table indexing axis are continuous.
Note that the clamp and unclamp process are executed even when the movement commands are continued during single block operation.

(8)   Make sure that the command position is at a position where clamping is possible.

10. Spindle Functions

10.1 Spindle functions S functions

Function and purpose

The spindle functions are also known simply as S functions and they assign the spindle speed. If the S function is designated in the same block as a movement command, the commands may be executed in either of the following two orders. The machine specifications determine which sequence applies.

(1)   The S function is executed after the movement command.

(2)   The S function is executed simultaneously with the movement command.

10.1.1 Spindle functions (S6-digits Analog)

Function and purpose

When the S6-digits function is added, a 6-digit value (0 to 999999) can be designated after the S code.

By assigning a 6-digit number following the S code, these functions enable the appropriate gear signals, voltages corresponding tot he commanded spindle speed (r/min) and start signals to be output.

If the gear step is changed manually other than when the S command is being executed, the voltage will be obtained from the set speed at that gear step and the previously commanded speed, and then will be output.

The analog signal specifications are given below.

(1)   Output voltage.............. 0 to 10V

(2)   Resolution .... 1/4096 (2^{-12})

(3)   Load conditions............ 10kΩ

(4)   Output impedance........ 220Ω

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

(1)   Parameters corresponding to individual gears .... Limit speed, maximum speed, shift speed and maximum speed during tapping.

(2)   Parameters corresponding to all gears .... Minimum speed, orientation rotation speed

10.1.2 Spindle functions (S8-digits)

Function and purpose

These functions are assigned with an 8-digit (0 to 99999999) number following the address S, and one group can be assigned in one block.

The output signal is a 32-bit binary data with sign and start signal. Processing and completion sequences are required for all S commands.

10.2 Constant surface speed control; G96, G97
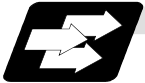
Function and purpose

These commands automatically control the spindle speed in line with the changes in the radius coordinate values as cutting proceeds in the diametrical direction, and they serve to keep the cutting point speed constant during the cutting.

Command format

G96 Ss Pp; Constant surface speed ON	
Ss	: Peripheral speed
Pp	: Assignment of constant surface speed control axis

G97 ; Constant surface speed cancel

Detailed description

(1) The constant surface speed control axis is set by parameter (#1181 G96_ax).
    0 : Fixed at 1st axis (P command invalid)
    1 : 1st axis
    2 : 2nd axis
    3 : 3rd axis

(2) When the above-mentioned parameter is not zero, the constant surface speed control axis can be assigned by address P.
    (Example) With G96_ax (1)

Program	Constant surface speed control axis
G96 S100 ;	1st axis
G96 S100 P3 ;	3rd axis

(3) Example of selection program and operation

G90 G96 G01 X50. Z100. S200 ;

G97 G01 X50. Z100. F300 S500 ;

M02 ;

The spindle speed is controlled so that the peripheral speed is 200m/min.

The spindle speed is controlled to 500r/min.

The modal returns to the initial setting.

10.3 Spindle/C Axis Control   [E68]

Function and purpose

This function enables one spindle to also be used as a C axis (rotation axis) by an external signal.

Detailed description

(1) Spindle/C axis changeover
Changeover between the spindle and C axis is done by the C axis SERVO ON signal.

	Spindle	C axis	Spindle
Servo ON			

At servo OFF .... Spindle (C axis control not possible)
  The C axis is in a reference point return incomplete state.

At servo ON.................. C axis (spindle control not possible)
  The C axis is in a reference point return incomplete state.
  A reference point return (orientation) is carried out at the rising edge
  of the SERVO ON signal by the parameter setting.
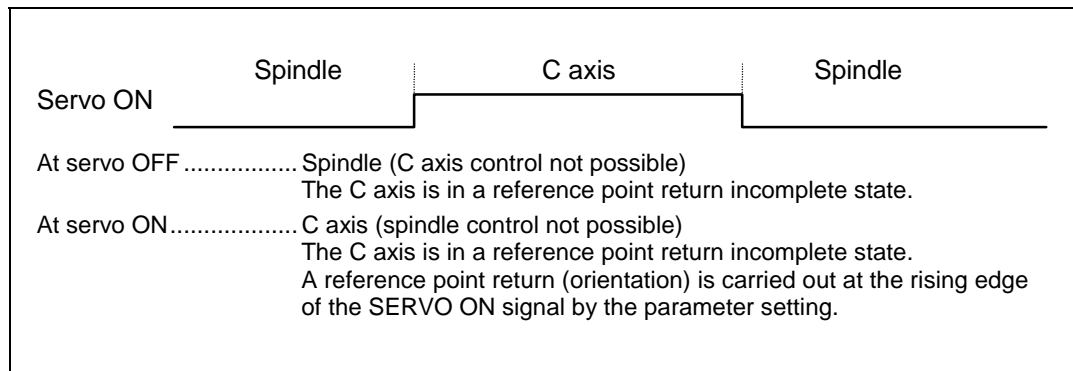
(2) Changeover timing chart example
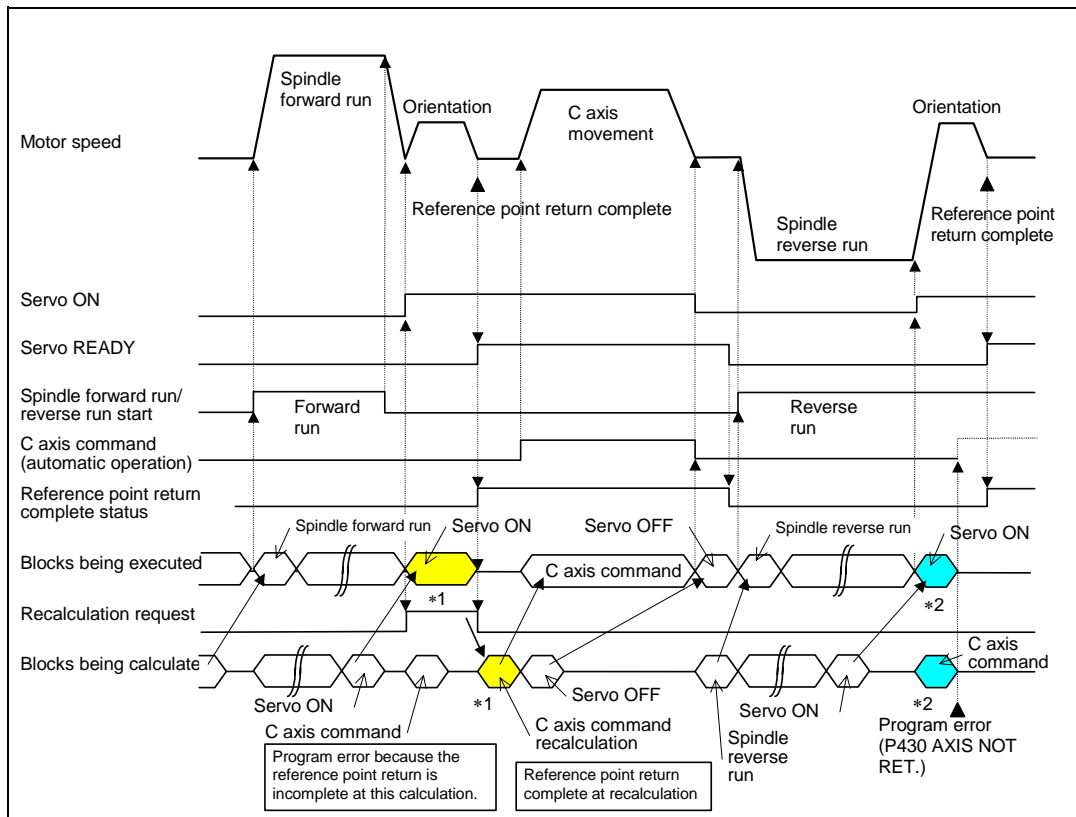
(Note) For axis commands, the reference point return complete is checked at calculation. Thus, when the C axis servo ON command and C axis command are continuous, the program error (P430) will occur as shown above in ∗2.
In response to this kind of situation, the following two processes must be carried out on user PLC, as shown above in ∗1.

- Input the recalculation request signal with a servo ON command.

- Wait for the completion of the servo ON command until the C axis enters a servo READY state.

(3) C axis gain
The C axis gain is changed over (the optimum gain is selected) by the C axis cutting condition. During C axis cutting feed, cutting gain is applied. During other axis' cutting feed (C axis face turning), non-cutting stop gain is applied. Non-cutting gain is applied in all other cases.

For the gain adjustment, refer to the parameter of C axis used.
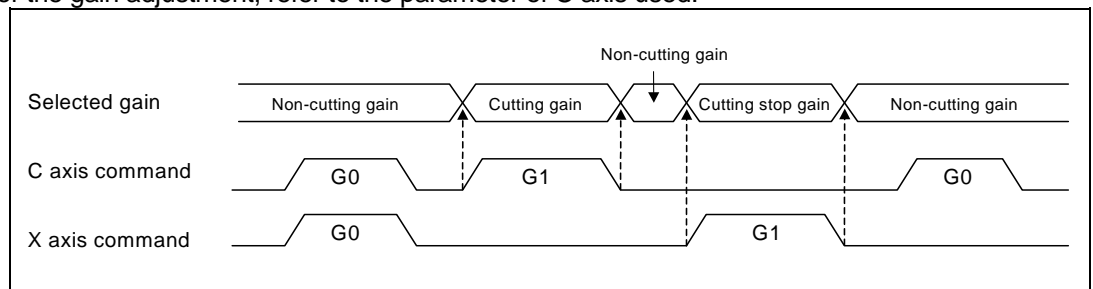
(Note 1) There are 1st to 3rd cutting gains, which are selected with the ladder.

(4) Deceleration check in movement including spindle/C-axis
The deceleration check in a movement command including the spindle/C-axis is as the table described below when the following condition is fulfilled.
When the different values are set for the position loop gain in non-cutting mode (spindle parameter #3203 PGCO) and the position loop gain in cutting mode (spindle parameter #3330 PGC1 to #3333 PGC4).
That is because a vibration and so on occurs in the machine when the gain is changed during the axis movement.

Parameter	Rapid traverse command
Inpos (#1193)	G0→XX (G0+G9→XX)
0	Command deceleration check
1	In-position check

Parameter	Other than rapid traverse command (G1 : other than G0 command)	
AUX07/BIT-1 (#1223/BIT-1)	G1+G9→XX (G1+G9→XX)	G1→ G1
0	In-position check (Applicable only to SV024)	No deceleration check
1		

(Note 1) When G1 command is issued, the in-position check is performed regardless of the deceleration check parameter.

(Note 2) XX expresses all commands.

Precautions and Restrictions

(1)   A reference point return cannot be executed by the orientation when there is no Z phase in the detector (PLG, ENC, other).
Replace the detector with one having a Z phase, or if using the detector as it is, set the position control changeover to "After deceleration stop" in the parameters (Spindle parameters "#3329 SP129 bitE: 1"), and set the axis to "Axis without zero point" (Zero point return parameters "#2031 noref: 1").

(2)   The program error (P430) will occur if a C axis command is issued during servo OFF or during orientation.

(3)   Do not execute a servo OFF during a C axis command.
The remaining C axis commands will be cleared at servo ON.
(If servo OFF is executed during C axis control, the feed will stop and spindle control will occur.)

(4)   If servo ON is executed during spindle rotation, the rotation will stop and C axis control will occur.

(5)   Dog-type reference point return is not possible for the C axis.
Set the reference point return to the orientation method in the parameters (Spindle parameters "#3329 SP129 bitE: 0"), or set the axis to "Axis without zero point" (Zero point return parameters "#2031 noref: 1").

10.4 Spindle Synchronization Control   [E68]

Function and purpose

In a machine having two spindles, this function controls the rotation speed and phase of one spindle (basic spindle) in synchronization with the rotation of the other spindle (synchronous spindle).

The function is used when the rotation speed of the two spindles must be matched, for example, if a workpiece grasped by the 1st spindle is to be grasped by a 2nd spindle, or if the spindle rotation speed has to be changed when one workpiece is grasped by both the 1st and 2nd spindles.

There are two types of spindle synchronization: Spindle synchronization I and Spindle synchronization II.

The spindle synchronization control I	The designation of the synchronous spindle and start/stop of the synchronization are executed by commanding G codes in the machining program. This type cannot be used with M system.
The spindle synchronization function II	The selections of the synchronized spindle and synchronization start, etc., are all designated from the PLC. Refer to the instruction manual issued by the machine tool builder for details.

Common setting for the spindle synchronization control I and II

When the spindle synchronization control is carried out, the followings must be set.
    Chuck close
    Error temporary cancel
    Multi-speed acceleration/deceleration
For details, refer to the "10.4.2 Precautions for Using Spindle Synchronization Control".

10.4.1 Spindle Synchronization Control II
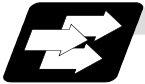
Function and purpose

With the spindle synchronous control II, selection of the synchronized spindle and synchronization start, etc., are all designated from the PLC.

Basic spindle and synchronous spindle selection

Select the basic spindle and synchronous spindle for synchronization control from the PLC.

Device No.	Signal name	Abbrev.	Explanation
R446	Basic spindle selection	-	Select a serially connected spindle to be controlled as the basic spindle. (0: 1st spindle), 1: 1st spindle, 2: 2nd spindle **(Note 1)** Spindle synchronization control will not take place if a spindle not connected in serial is selected. **(Note 2)** If "0" is designated, the 1st spindle will be controlled as the basic spindle.
R447	Synchronous spindle selection	-	Select a serially connected spindle to be controlled as the synchronous spindle. (0: 2nd spindle), 1: 1st spindle, 2: 2nd spindle **(Note 3)** Spindle synchronization control will not take place if a spindle not connected in serial is selected or if the same spindle as the basic spindle is selected. **(Note 4)** If "0" is designated, the 2nd spindle will be controlled as the synchronous spindle.

Starting spindle synchronization

The spindle synchronization control mode is entered by inputting the spindle synchronization control signal (SPSYC). The synchronous spindle will be controlled in synchronization with the rotation speed commanded for the basic spindle during the spindle synchronization control mode. When the difference of the basic spindle and synchronous spindle rotation speeds reaches the spindle synchronization rotation speed reach level setting value (#3050 sprlv), the spindle rotation speed synchronization complete signal (FSPRV) will be output.

The synchronous spindle's rotation direction is designated with the spindle synchronization rotation direction designation as the same as the basic spindle or the reverse direction.

Device No.	Signal name	Abbrev.	Explanation
Y398	Spindle synchronization control signal	SPSYC	The spindle synchronization control mode is entered when this signal turns ON.
X308	In spindle synchronization control	SPSYN1	This notifies that the mode is the spindle synchronization control.
X309	Spindle rotation speed synchronization complete signal	FSPRV	This turns ON when the difference of the basic spindle and synchronous spindle rotation speeds reaches the spindle rotation speed reach level setting value during the spindle synchronization control mode. This signal turns OFF when the spindle synchronization control mode is canceled, or when an error exceeding the spindle rotation speed reach level setting value occurs during the spindle synchronization control mode.
Y39A	Spindle synchronization rotation direction designation	-	Designate the basic spindle and synchronous spindle rotation directions for spindle synchronization control. 0: The synchronous spindle rotates in the same direction as the basic spindle. 1: The synchronous spindle rotates in the reverse direction of the basic spindle.

Spindle phase alignment

Spindle phase synchronization starts when the spindle phase synchronization control signal (SPPHS) is input during the spindle synchronization control mode. The spindle phase synchronization complete signal is output when the spindle synchronization phase reach level setting value (#3051 spplv) is reached.
The synchronous spindle's phase shift amount can also be designated from the PLC.

Device No.	Signal name	Abbrev.	Explanation
Y399	Spindle phase synchronization control signal	SPPHS	Spindle phase synchronization starts when this signal is turned ON during the spindle synchronization control mode. **(Note 1)** If this signal is turned ON in a mode other than the spindle synchronization control mode, it will be ignored.
X30A	Spindle phase synchronization complete signal	FSPPH	This signal is output when the spindle synchronization phase reach level is reached after starting spindle phase synchronization.
R448	Phase shift amount setting	-	Designate the synchronous spindle's phase shift amount. Unit: 360°/4096
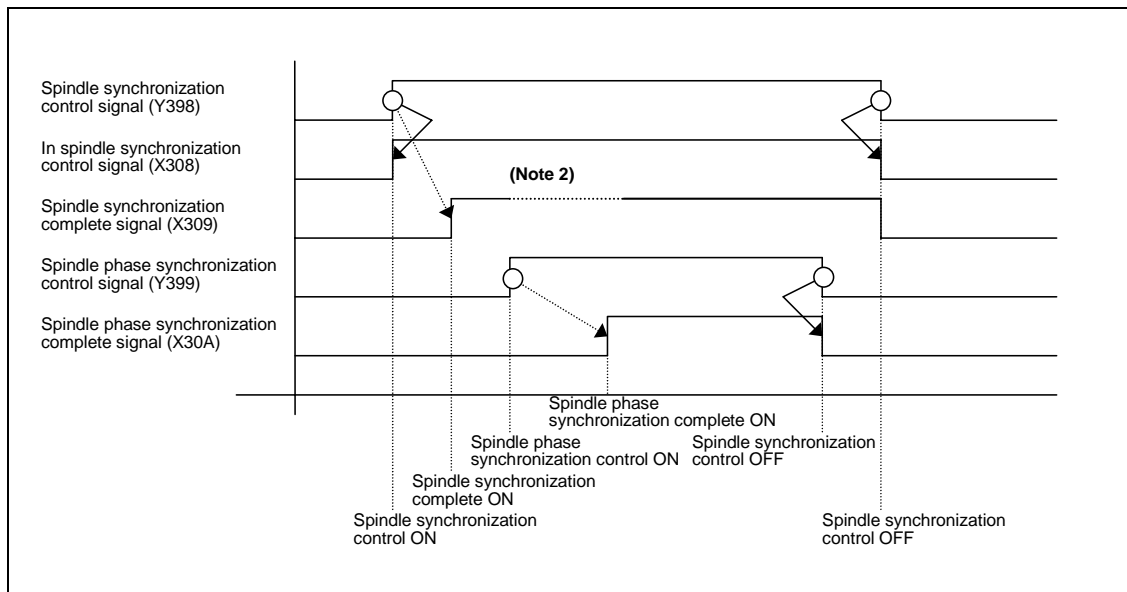
(Note 2) Turns OFF temporarily to change the rotation speed during phase synchronization.

Calculating the spindle synchronization phase shift amount and requesting phase offset

The spindle phase shift amount calculation function obtains and saves the phase difference of the basic spindle and synchronous spindle by turning the PLC signal ON during spindle synchronization. When calculating the spindle phase shift, the synchronous spindle can be rotated with the handle, so the relation of the phases between the spindles can also be adjusted visually.
If the spindle phase synchronization control signal is input while the phase offset request signal (SSPHF) is ON, the phases will be aligned using the position shifted by the saved phase shift amount as a reference.
This makes aligning of the phases easier when grasping the material that the shape of one end differ from another end.

Device No.	Signal name	Abbrev.	Explanation
Y39B	Phase shift calculation request	SSPHM	If spindle synchronization is carried out while this signal is ON, the phase difference of the basic spindle and synchronous spindle will be obtained and saved.
Y39C	Phase offset request	SSPHF	If spindle phase synchronization is carried out while this signal is ON, the phases will be aligned using the position shifted by the saved phase shift amount as a basic position.
R474	Phase difference output	-	The delay of the synchronous spindle in respect to the basic spindle is output. Unit: 360°/4096 **(Note 1)** If either the basic spindle or synchronous spindle has not passed through the Z phase, etc., and the phase cannot be calculated, -1 will be output. **(Note 2)** This data is output only while calculating the phase shift or during spindle phase synchronization.
R490	Phase offset data	-	The phase difference saved with phase shift calculation is output. Unit: 360°/4096 **(Note 3)** This data is output only during spindle synchronization control.
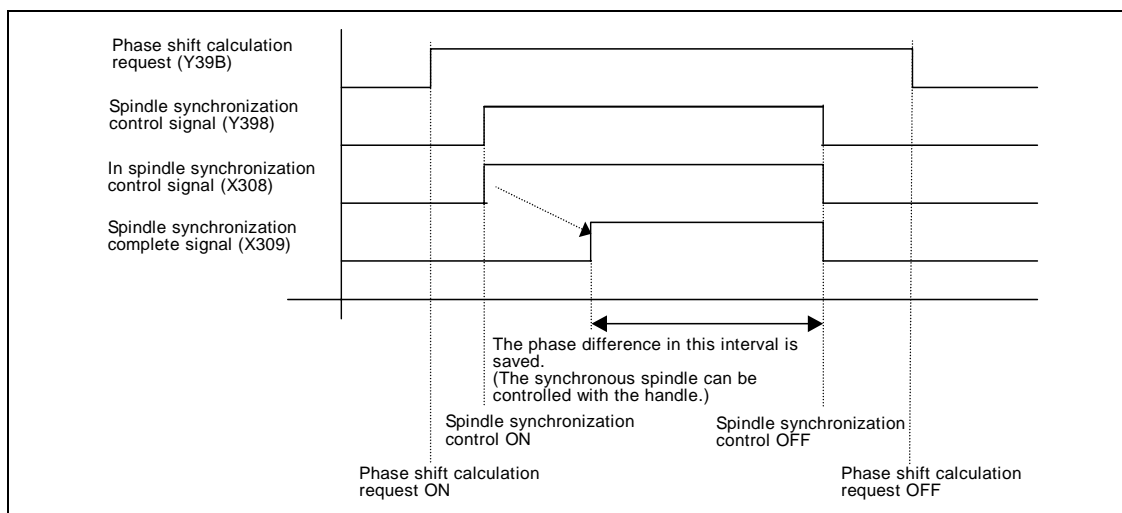
(Note 4) The phases cannot be aligned while calculating the phase shift.
(Note 5) The synchronous spindle cannot be rotated with the handle when the manual operation mode is set to the handle mode.

Precautions and restrictions

(1) When carrying out spindle synchronization, a rotation command must be issued to both the basic spindle and synchronous spindle. The synchronous spindle's rotation direction will follow the basic spindle rotation direction and spindle synchronization rotation direction designation regardless of whether a forward or reverse run command is issued.

(2) The spindle synchronization control mode will be entered even if the spindle synchronization control signal is turned ON while the spindle rotation speed command is ON. However, synchronous control will not actually take place. Synchronous control will start after the rotation command has been issued to the basic spindle, and then the spindle synchronization complete signal will be output.

(3) The spindle rotating with spindle synchronization control will stop when emergency stop is applied.

(4) An operation error will occur if the spindle synchronization control signal is turned ON while the basic spindle and synchronous spindle designations are illegal.

(5) The rotation speed clamp during spindle synchronization control will follow the smaller clamp value set for the basic spindle or synchronous spindle.

(6) Orientation of the basic spindle and synchronous spindle is not possible during the spindle synchronization. To carry out orientation, turn the spindle synchronization control signal OFF first.

(7) The rotation speed command is invalid for the synchronous spindle during the spindle synchronization. The commanded rotation speed will be validated after spindle synchronization is canceled.

(8) The constant surface speed control is invalid for the synchronous spindle during the spindle synchronization.

(9) If the phase offset request signal is turned ON before the phase shift is calculated and then spindle phase synchronization is executed, the shift amount will not be calculated.

(10) The spindle Z-phase encoder position parameters are invalid when phase offset is carried out.

(11) If spindle phase synchronization is started while the phase shift calculation request signal is ON, the error "M01 operation error 1106" will occur.

(12) Turn the phase shift calculation request signal ON when the basic spindle and synchronous spindle are both stopped. If the phase shift calculation request signal is ON while either of the spindles is rotating, the error "M01 operation error 1106" will occur.

(13) The phase shift amount saved in the NC is held until the next phase shift is calculated. (This value is saved even when the power is turned OFF.)

(14) Always execute the "Chuck close". If the "Chuck close" is not execute, an excessive load may be applied on the machine or an alarm may occur.

⚠ **CAUTION**

⚠ Do not make the synchronous spindle rotation command OFF with one workpiece chucked by the basic spindle and synchronous spindle during the spindle synchronous control mode.
Failure to observe this may cause the synchronous spindle stop, and hazardous situation.

10.4.2 Precautions for Using Spindle Synchronization Control
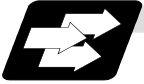
Function and purpose

Some PLC signals must be set when spindle synchronization control I or II is used. If these signals are not set, an excessive load or an alarm may occur. Refer to the instruction manual issued by the machine tool builder for details. In this section, each function and the signal are explained.

Chuck close signal

The synchronous spindle side carries out droop compensation while the chuck is opened, and aligns itself with the basic spindle. However, when the chuck is closed, the droop compensation is added, and the synchronization error with the base increases. Droop compensation is prevented with the chuck close signal and the position where the chuck is grasped is maintained with position compensation.

Device No.	Signal name	Abbrev.	Explanation
Y2E9	Chuck close signal	-	This turns ON when the chuck is closed. When this signal turns ON, the compensation between the basic spindle and synchronous spindle will change from droop compensation to position compensation.
X30E	Chuck close confirmation signal	-	This turns ON when the chuck close signal is received during the spindle synchronization control mode.
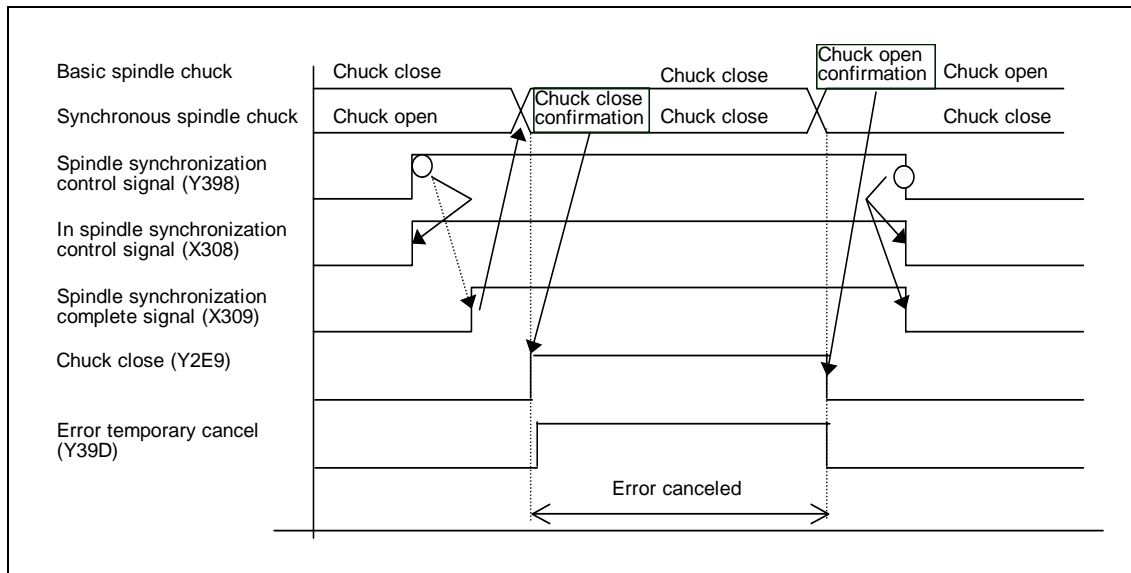
(Note 1)   Use the error temporary cancel only when there is still an error between the spindle and synchronization with the chuck close signal.

Error temporary cancel function

When spindle synchronization is carried out while grasping the workpiece with the basic spindle and rotating, if the chuck is closed to grasp the workpiece with the synchronous spindle, the speed will fluctuate due to external factors and an error will occur. If spindle synchronization is continued without compensating this error, the workpiece will twist.
This torsion can be prevented by temporarily canceling this error.

Device No.	Signal name	Abbrev.	Explanation
Y39D	Error temporary cancel signal	SPDRP0	The error is canceled when this signal is ON.

Phase error monitor

The phase error can be monitored during spindle phase synchronization.

Device No.	Signal name	Abbrev.	Explanation
R477	Phase error monitor	-	The phase error during spindle phase synchronization control is output as a pulse unit.
R478	Phase error monitor (lower limit value)	-	The lower limit value of the phase error during spindle phase synchronization control is output as a pulse unit.
R479	Phase error monitor (upper limit value)	-	The upper limit value of the phase error during spindle phase synchronization control is output as a pulse unit.

Multi-speed acceleration/deceleration

Up to eight steps of acceleration/deceleration time constants for spindle synchronization can be selected according to the spindle rotation speed.

(1) Time required from stopped state to sptc1 setting rotation speed

spt ∗ (sptc1/maximum rotation speed)

(2) Time required from sptc1 to sptc2 setting rotation speed

spt ∗ ((sptc2−sptc1)/maximum rotation speed) ∗ spdiv1

(3) Time required from sptc2 to sptc3 setting rotation speed

spt ∗ ((sptc3−sptc2)/maximum rotation speed) ∗ spdiv2
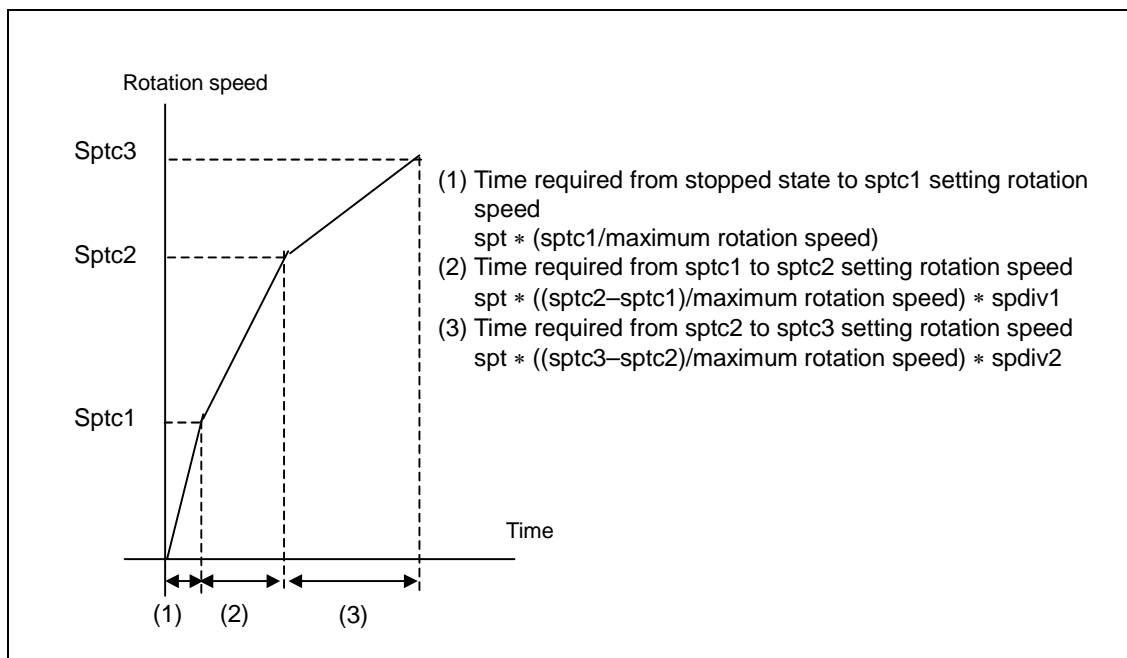
10.5 Multiple Spindle Control   [E68]

Function and purpose

Multiple spindle control is a function used to control the sub-spindle in a machine tool that has a main spindle (1st spindle) and a sub-spindle (2nd spindle).
The spindle control method differs between multiple spindle control I and multiple spindle control II.
Whether to use multiple spindle control I or multiple spindle control II depends on the parameter (#1300 ext36/bit0) setting.

Multiple spindle control I: (ext36/bit0 = 0)	Control following the spindle selection command (G43.1, etc.) and spindle control command ([S***** ;] or [S○=***** ;]), etc.
Multiple spindle control II: (ext36/bit0 = 1)	Control following the external signal (spindle command selection signal, spindle selection signal) and spindle control command ([S***** ;] only), etc. The spindle selection command [S○=***** ;] cannot be used.

10.5.1 Multiple Spindle Control I (Multiple Spindle Command)

Function and purpose

In addition to the S∗∗∗∗∗ command, the 1st and 2nd spindles can be identified and commanded with the SO=∗∗∗∗∗ command.

Command format

SO=∗∗∗∗∗ ;	
O	Designate the spindle number with one numeric character (1: 1st spindle/2: 2nd spindle). A variable can be designated.
∗∗∗∗∗	Rotation speed or constant surface speed command. A variable can be designated.

(Note 1) A program error (P35) will occur if the O value is not 1 or 4.

(Note 2) A program error (P33) will occur if G47.1 is issued in a modal.

Detailed description

(1) Each spindle command is delimited by the details of O.

   (Example)

   S1 = 3500 ; 1st spindle 3500(r/min) command
   S2 = 1500 : 2nd spindle 1500(r/min) command

(2) Multiple spindles can be commanded in one block.

(3) If two or more commands are issued to the same spindle in a block, the command issued last will be valid.

   (Example) S1 = 3500 S1 = 3600 S1 = 3700 ;   S1 = 3700 will be valid.

(4) The S∗∗∗∗∗ command and SO=∗∗∗∗∗ command can be used together.
   The spindle targeted for the S∗∗∗∗∗ command is delimited by the spindle selection command.
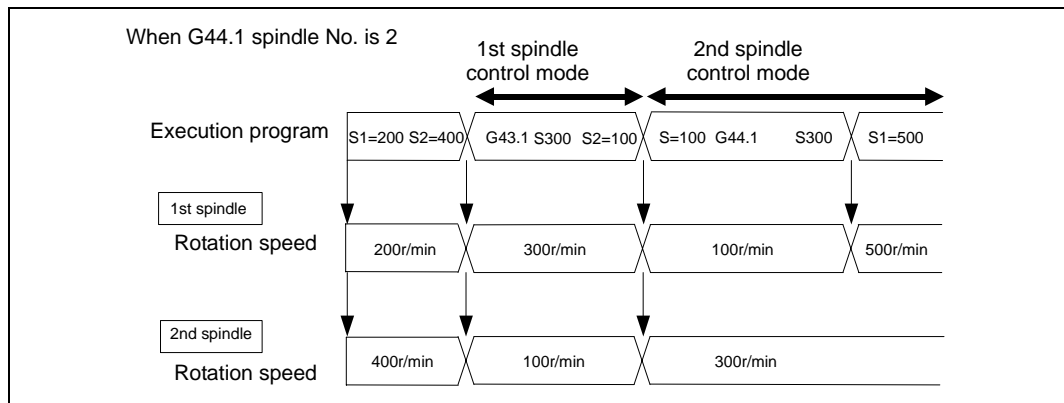
10.5.2 Multiple Spindle Control I (Spindle Selection Command)
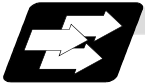
Function and purpose

The following S commands (S∗∗∗∗) can be changed for the 1st spindle or 2nd spindle by issuing the spindle selection command (G43.1, etc. [G group 20]).

Command format

G43.1;	**1st spindle control mode ON**
G44.1;	**Selected spindle control mode ON (Selected spindle No. is set with SnG44.1)**
G47.1;	**All spindles simultaneous control mode ON**

Detailed description

(1) The selected spindle No. is set with the parameter (#1534 SnG44.1).

(2) The spindle selection commands are modal G codes.

(3) A program error (P33) will occur if the spindle selection command is issued while the multiple spindle control II mode is valid.

(4) The spindle control mode entered when the power is turned ON or reset can be set with the parameters. The state when the power is turned ON or reset is as follows.

G group 20 modal state	Set with setup parameter "#1199 Sselect". 0:G43.1 1:G44.1 2:G47.1
G44.1 spindle No.	Set with setup parameter "#1534 SnG44.1". 0:2nd spindle 1:1st spindle 2:2nd spindle

(5) If the S command is issued in the same block as the spindle selection command, it will be valid for the spindle switched to with the spindle selection command.

Example)		
G43.1 S100 ;	→	Command 100 rotations for 1st spindle
S200 G44.1 S300 ; (SnG44.1=2)	→	After commanding 200 rotations for 1st spindle, command 300 rotations for 2nd spindle

(6) If a spindle that does not exist is set, the 2nd spindle will be used. Note that if there is only one spindle, the 1st spindle will be used.

Relation with other functions

The following functions change after the spindle selection command.

(1) S command (S∗∗∗∗)
The S command in G97 (rotation speed command)/G96 (constant surface speed command) will be the command for the spindle designated with the spindle selection command.

(2) Per rotation command (synchronous feed)
Even if F is commanded in the G95 mode, the per rotation feedrate for the spindle designated with the spindle selection command will be applied.

(3) Constant surface speed control spindle changeover
The constant surface speed control will also follow the spindle selection command mode.

If spindle other than the current mode is commanded with SO=∗∗∗∗∗, the spindle rotation speed command designated with O will have the priority.

(Example)

S2=∗∗∗∗ in G43 system mode ....2nd spindle rotation speed
command
S1=∗∗∗∗ in G44 system 2nd spindle selection mode ....1st spindle rotation speed
command

Note that the constant surface speed command in the constant surface speed control (G96) will be modal information independent from the rotation speed command.

When G44.1 spindle No. is 2

Execution program: S1=200 S2=300 | G43.1 | G90 G0 X100. | G96 S300 | G01 X50. F200 | G97 | G44.1 | G96 S300 | G01 X20. F200 | G97

1st spindle Rotation speed: 200r/min | 300m/min | 300×1000/(50×3.14)r/min

2nd spindle Rotation speed: 300r/min | 300m/min | 300×1000/(20×3.14)r/min
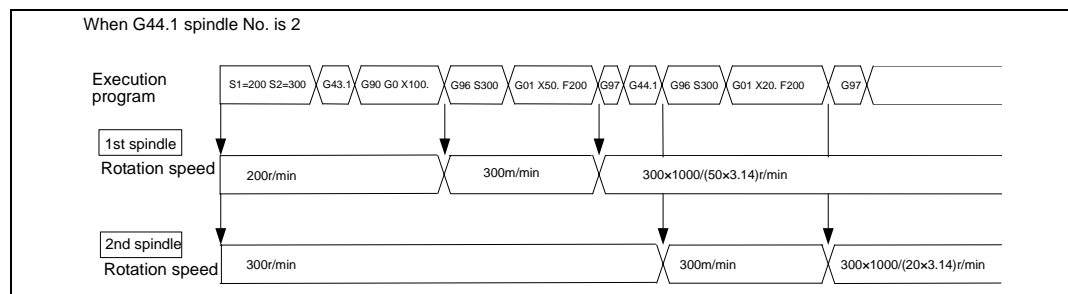
10.5.3 Multiple Spindle Control II
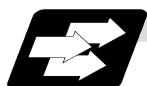
Function and purpose

Multiple spindle control II is a function that designates which spindle to select with the signals from PLC. The command is issued to the spindle with one S command.

Detailed description

(1) Spindle command selection, spindle selection

The S command to the spindle is output as the rotation speed command to the selected spindle when the spindle selection signal (SWS) from the PLC turns ON. The selected spindle rotates at the output rotation speed. The spindle whose selection is canceled when the spindle selection signal (SWS) turns OFF maintains the speed at which it was rotating at before being canceled. This allows each axis to be simultaneously rotated at differing rotation speeds.
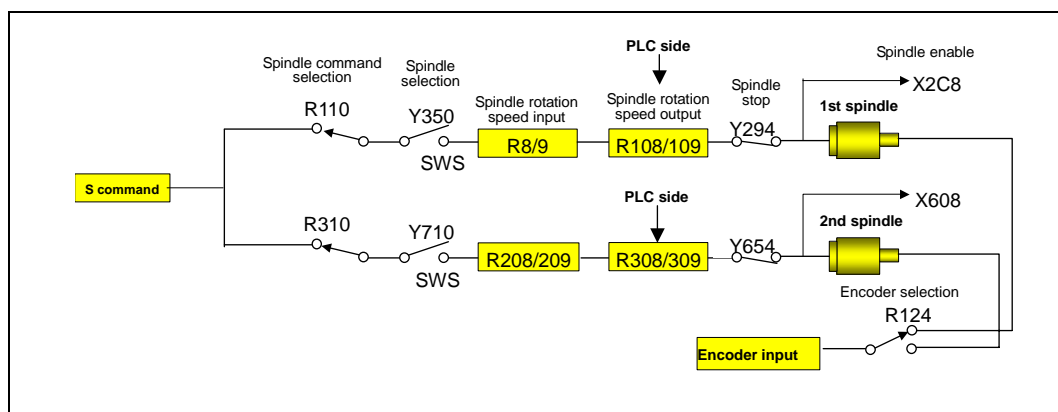
(Note) Refer to the PLC Interface Manual for details on each signal.

Relation with other functions

(1) Constant surface speed control
Constant surface speed control can be applied on all spindles.
The spindle rotation speed is automatically controlled during constant surface speed control, so when machining with constant surface speed, the spindle selection signal (SWS) for that spindle must be left ON.
The spindle not selected with the spindle selection signal (SWS) maintains the speed at which it was rotating at before being canceled.

(2) Thread cutting/synchronous feed
The threads are cut with the spindle selected with the spindle selection signal (SWS). The encoder feedback selected with the encoder selection signal is used.

(3) Synchronous tap
The synchronous tap spindle is selected with the spindle selection signal (SWS).
Select the synchronous tap spindle before issuing the synchronous tap command. Do not change the synchronous tap spindle selection signal during the synchronous tapping mode.
If a C axis mode command is issued to the synchronous tap spindle, the "M01 operation error 1026" will occur. When the C axis command is canceled, the error will be canceled and machining will resume.

(4) Asynchronous tap
The asynchronous tap spindle is selected with the spindle selection signal (SWS).
Select the asynchronous tap spindle before issuing the tap command. Input a calculation request to change the asynchronous tap spindle selection. Do not change the asynchronous tap spindle selection signal during the asynchronous tapping mode.

(5) Tap return
The tap return spindle is selected with the spindle selection signal (SWS).
Select the spindle for which the tap cycle execution in stopped before turning the tap return signal ON.
If tap return is executed when a different spindle is selected, the "M01 operation error 1032" will occur. Do not change the spindle selection signal during tap return.

Restrictions

(1) The S manual value command is invalid when multiple spindle control II is valid.

(2) Setup parameter "#1199 Sselect" is invalid when multiple spindle control II is valid.

(3) The spindle control mode changeover G code cannot be used when multiple spindle control II is valid. A program error (P34) will occur.

(4) The "S1=∗∗∗" and "S2=∗∗∗" commands are invalid when multiple spindle control II is valid. A program error (P33) will occur.

(5) The spindle gear shift command output signal (GR1/GR2) is not output when multiple spindle control II is valid.

11. Tool Functions

11.1 Tool functions (T8-digit BCD)

Function and purpose

The tool functions are also known simply as T functions and they assign the tool numbers and tool offset number. They are designated with a 8-digit number following the address T, and one set can be commanded in one block. The output signal is an 8-digit BCD signal and start signal.

If the T function is designated in the same block as a movement command, the commands may be executed in either of the following two orders. The machine specifications determine which sequence applies.

(1)   The T function is executed after the movement command.

(2)   The T function is executed simultaneously with the movement command.

Processing and completion sequences are required for all T commands.

12. Tool Offset Functions

12.1 Tool offset

Function and purpose

The basic tool offset function includes the tool length offset and tool diameter compensation. Each offset amount is designated with the tool offset No. Each offset amount is input from the setting and display unit or the program.

Tool offset memory

There are two types of tool offset memories for setting and selecting the tool offset amount. (The type used is determined by the machine tool builder specifications.)
The offset amount or offset amount settings are preset with the setting and display unit.

Type 1 is selected when parameter "#1037 cmdtyp" is set to "1", and type 2 is selected when set to "2".

Type of tool offset memory	Classification of length offset, diameter compensation	Classification of shape offset, wear compensation
Type 1	Not applied	Not applied
Type 2	Applied	Applied

Type 1

Type 1One offset amount corresponds to one offset No. as shown on the right. Thus, these can be used commonly regardless of the tool length offset amount, tool diameter offset amount, shape offset amount and wear offset amount.

$(D1) = a_1$, $(H1) = a_1$
$(D2) = a_2$, $(H2) = a_2$
 : :
$(Dn) = a_n$, $(Hn) = a_n$

Offset No.	Offset amount
1	a_1
2	a_2
3	a_3
•	•
•	•
n	a_n

Type 2

The shape offset amount related to the tool length, wear offset amount, shape offset related to the tool diameter and the wear offset amount can be set independently for one offset No. as shown below.

The tool length offset amount is set with H, and the tool diameter offset amount with D.

$(H1) = b1 + c1$, $(D1) = d1 + e1$
$(H2) = b2 + c2$, $(D2) = d2 + e2$
 : :
$(Hn) = bn + cn$, $(Dn) = dn + en$

Offset No.	Tool length (H)		Tool diameter(D)/ (Position offset)	
	Shape offset amount	Wear offset amount	Shape offset amount	Wear offset amount
1	b1	c1	d1	e1
2	b2	c2	d2	e2
3	b3	c3	d3	e3
•	•	•	•	•
•	•	•	•	•
n	bn	cn	dn	en

⚠ CAUTION

⚠ If the tool offset amount is changed during automatic operation (including during single block stop), it will be validated from the next block or blocks onwards.

Tool offset No. (H/D)

This address designates the tool offset No.

(1) H is used for the tool length offset, and D is used for the tool position offset and tool diameter offset.

(2) The tool offset No. that is designated once does not change until a new H or D is designated.

(3) The offset No. can be commanded once in each block. (If two or more Nos. are commanded, the latter one will be valid.)

(4) The No. of offset sets that can be used will differ according to the machine.
For E60 (200 sets): Designate with the H01 to H200 (D01 to D200) numbers.
For E68 (400 sets): Designate with the H01 to H400 (D01 to D400) numbers.

(5) If a value larger than this is set, the program error "P170" will occur.

(6) The setting value ranges are as follows for each No.
The offset amount for each offset No. is preset with the setting and display unit.

(a) When using E60

Input setting unit	Shape offset amount		Wear offset amount	
	Metric system	**Inch system**	**Metric system**	**Inch system**
#1015 cunit=10	±9999.999mm	±999.9999 inch	±999.999 mm	±99.9999 inch

(b) When using E68

Input setting unit	Shape offset amount		Wear offset amount	
	Metric system	**Inch system**	**Metric system**	**Inch system**
#1015 cunit=10	±9999.999mm	±999.9999 inch	±999.999 mm	±99.9999 inch
#1015 cunit=1	±999.9999 mm	±99.99999 inch	±99.9999 mm	±9.99999 inch

12.2 Tool length offset/cancel; G43, G44/G49
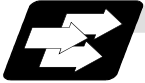
Function and purpose

The end position of the movement command can be offset by the preset amount when this command is used. A continuity can be applied to the program by setting the actual deviation from the tool length value decided during programming as the offset amount using this function.

Command format

When tool length offset is +

```
G43 Zz Hh ;  Tool length offset + start
   :
G49 Zz ; Tool length offset cancel
```

When tool length offset is −

```
G44 Zz Hh ;  Tool length offset − start
   :
G49 Zz ;
```

Detailed description

(1) Tool length offset movement amount

The movement amount is calculated with the following expressions when the G43 or G44 tool length offset command or G49 tool length offset cancel command is issued.

$$
\begin{array}{llll}
& \text{Z axis movement amount} & & \\
\text{G43 Zz H}n_1 \text{ ;} & z + & (\lambda h_1) & \text{Offset in + direction by tool offset amount} \\
\text{G44 Zz Hh}_1 \text{ ;} & z - & (\lambda h_1) & \text{Offset in − direction by tool offset amount} \\
\text{G49 Zz} \quad\;\; \text{ ;} & z -(+) & (\lambda h_1) & \text{Offset amount cancel.} \\
& & \lambda h_1 : & \text{Offset amount for offset No. } h_1
\end{array}
$$

Regardless of the absolute value command or incremental value command, the actual end point will be the point offset by the offset amount designated for the programmed movement command end point coordinate value.

The G49 (tool length offset cancel) mode is entered when the power is turned ON or when M02 has been executed.

(Example 1) For absolute value command
```
            H01 = −100000
         N1  G28  Z0  T01  M06 ;
         N2  G90  G92  Z0 ;
         N3  G43  Z5000  H01 ;
         N4  G01  Z-50000  F500 ;
```

(Example 2) For incremental value command
```
            H01 = −100000
         N1  G28  Z0  T01  M06 ;
         N2  G91  G92  Z0 ;
         N3  G43  Z5000  H01 ;
         N4  G01  Z-55000  F500 ;
```


117
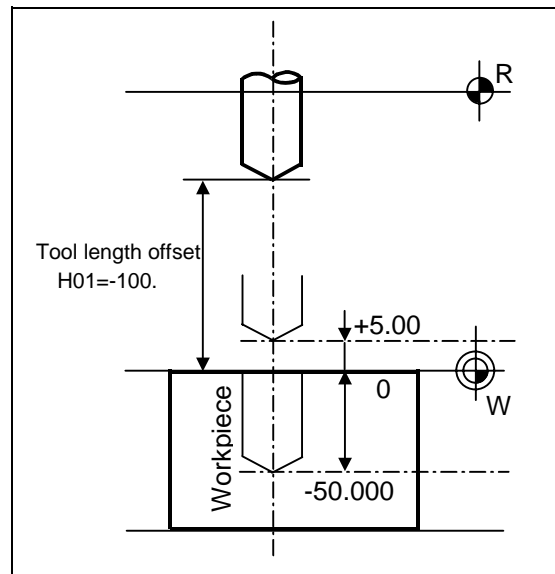
(2) Offset No.

(a) The offset amount differs according to the compensation type.

Type 1
G43 Hh_1 ;
When the above is commanded, the offset amount lh_1 commanded with offset No. h_1 will be applied commonly regardless of the tool length offset amount, tool diameter offset amount, shape offset amount or wear offset amount.
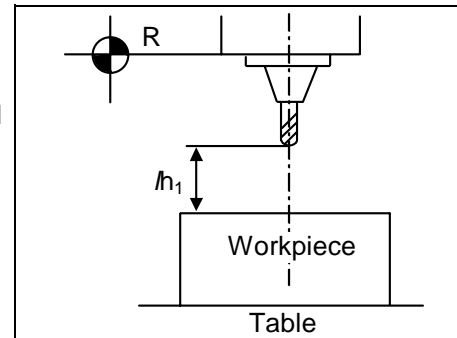
Type 2
G43 Hh_1 ;
When the above is commanded, the offset amount lh_1 commanded with offset No. h_1 will be as follows.
lh_1: Shape offset (Note) + wear offset amount
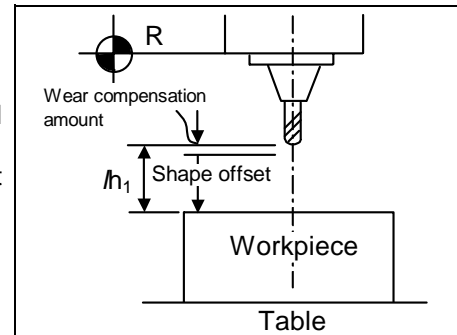
(b) The valid range of the offset No. will differ according to the specifications (No. of offset sets).

(c) If the commanded offset No. exceeds the specification range, the program error "P170" will occur.

(d) Tool length cancel will be applied when H0 is designated.

(e) The offset No. commanded in the same block as G43 or G44 will be valid for the following modals.

(Example 3)

G43 Z$z1$ H$h1$;........... Tool length offset is executed with h1.
 :
G45 X$x1$ Y$y1$ H$h6$;
 :
G49 Z$z2$; .... The tool length offset is canceled.
 :
G43 Z$z2$; .... Tool length offset is executed again with h1.

(f) If G43 is commanded in the G43 modal, an offset of the difference between the offset No. data will be executed.

(Example 4)

G43 Z$z1$ H$h1$; .... Becomes the z1 + (λh1) movement.
 :
G43 Z$z2$ H$h2$; .... Becomes the z2 + (λh2 - lh1) movement.
 :
The same applies for the G54 command in the G59 modal.

(3) Axis valid for tool length offset

(a) When parameter #1080 Dril_Z is set to "1", the tool length offset is always applied on the Z axis.

(b) When parameter #1080 Dril_Z is set to "0", the axis will depend on the axis address commanded in the same block as G43. The order of priority is shown below.
Zp > Yp > Xp

(Example 5)

```
G43  Xx1  Hh1 ; ................+ offset to X axis
     :
G49  Xx2 ;
     :
G44  Yy1  Hh2 ; ................−offset to Y axis
     :
G49  Yy2 ;
     :
G43  Xx3  Yy3  Zz3 ; .........Offset is applied on Z axis
     :
G49  ;
```

The handling of the additional axis will follow the parameters #1029 to 1031 aux_I, J and K settings.
If the tool length offset is commanded for the rotary axis, set the rotary axis name for one of the parallel axes.

(c) If H (offset No.) is not designated in the same block as G43, the Z axis will be valid.

(Example 6)

```
G43  Hh1 ; ........................Offset and cancel to X axis
     :
49  ;
```

(4) Movement during other commands in tool length offset modal

(a) If reference point return is executed with G28 and manual operation, the tool length offset will be canceled when the reference point return is completed.

(Example 7)

```
G43  Zz1  Hh1 ;
     :
G28  Zz2 ; .......................Canceled when reference point is reached.
     :
G43  Zz2  Hh2 ;                              (Same as G49)
     :
G49  G28  Zz2 ; ................After the Z axis is canceled, reference point
                                 return is executed.
```

(b) The movement is commanded to the G53 machine coordinate system, the axis will move to the machine position when the tool offset amount is canceled. When the G54 to G49 workpiece coordinate system is returned to, the position returned to will be the coordinates shifted by the tool offset amount.

12.3 Tool radius compensation

Function and purpose

This function compensates the radius of the tool. The compensation can be done in the random vector direction by the radius amount of the tool selected with the G command (G38 to G42) and the D command.

Command format

G40X___Y___ ; : Tool radius compensation cancel
G41X___Y___ ; : Tool radius compensation (left)
G42X___Y___ ; : Tool radius compensation (right)
G38I___J___ ;   : Change or hold of compensation vector ⎫ Can be commanded only
G39X___Y___ ; : Corner changeover ⎬ during the radius compensation mode.

Detailed description

The No. of compensation sets will differ according to the machine model.
(The No. of sets is the total of the tool length offset, tool position offset and tool radius compensation sets.)
The H command is ignored during the tool radius compensation, and only the D command is valid.
The compensation will be executed within the plane designated with the plane selection G code or axis address 2 axis, and axes other than those included in the designated plane and the axes parallel to the designated plane will not be affected. Refer to the section on plane selection for details on selecting the plane with the G code.

12.3.1 Tool radius compensation operation

Tool radius compensation cancel mode

The tool radius compensation cancel mode is established by any of the following conditions.

(1)   After the power has been switched on
(2)   After the reset button on the setting and display unit has been pressed
(3)   After the M02 or M30 command with reset function has been executed
(4)   After the tool radius compensation cancel command (G40) has been executed

The offset vectors are zero in the compensation cancel mode, and the tool nose point path coincides with the programmed path.
Programs including tool radius compensation must be terminated in the compensation cancel mode.

Tool radius compensation start (start-up)

Tool radius compensation starts when all the following conditions are met in the compensation cancel mode.

(1) The movement command is issued after G41 or G42.
(2) The tool radius compensation offset No. is $0 < D \leq$ max. offset No.
(3) The movement command of positioning (G00) or linear interpolation (G01) is issued.

At the start of compensation, the operation is executed after at least three movement command blocks (if three movement command blocks are not available, after five movement command blocks) have been read regardless of the continuous operation or single block operation.
During compensation, 5 blocks are pre-read and the compensation is arithmetically processed.

Control mode transition diagram
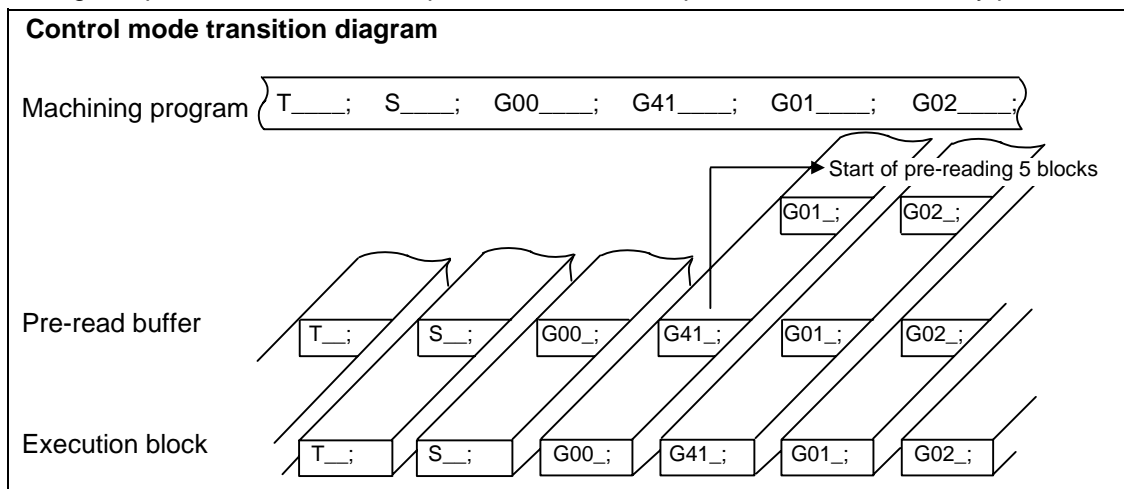
There are two ways of starting the compensation operation: type A and type B.
The type can be selected with bit 2 of parameter #1229 set 01. This type is used in common with the compensation cancel type.

Type A	When the start-up and cancel commands are operated during tool radius compensation, their blocks are not handled by intersection operation processing; they are handled as offset vectors in the direction vertical to that of the commands.
Type B	When the start-up and cancel commands are operated during tool radius compensation, the intersection operation processing of the command block and the next block is executed.

In the following explanatory figure, "S" denotes the single block stop point.

12. Tool Offset Functions

Start of movement for tool radius compensation

(1)　For inner side of corner
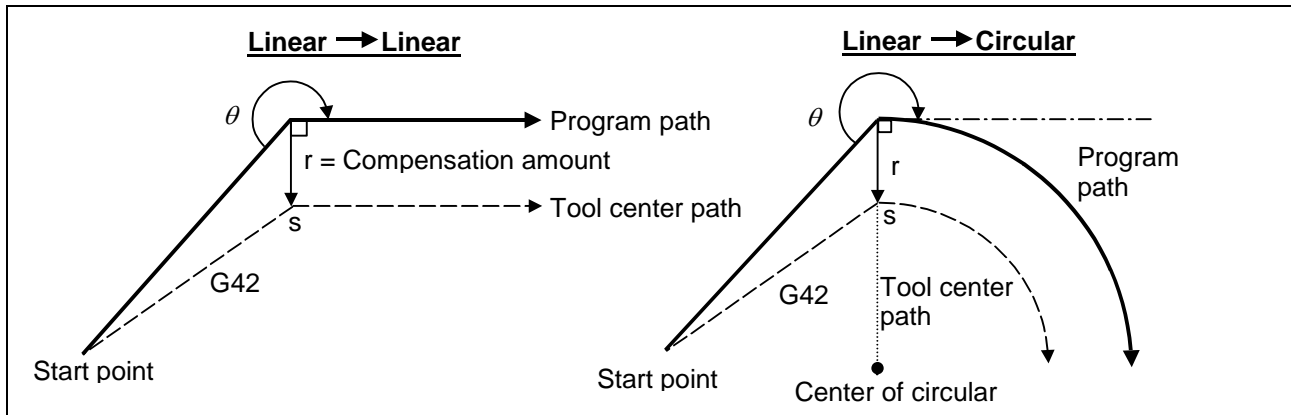
(2)　For outer side of corner (obtuse angle) [90°≦0<180°]
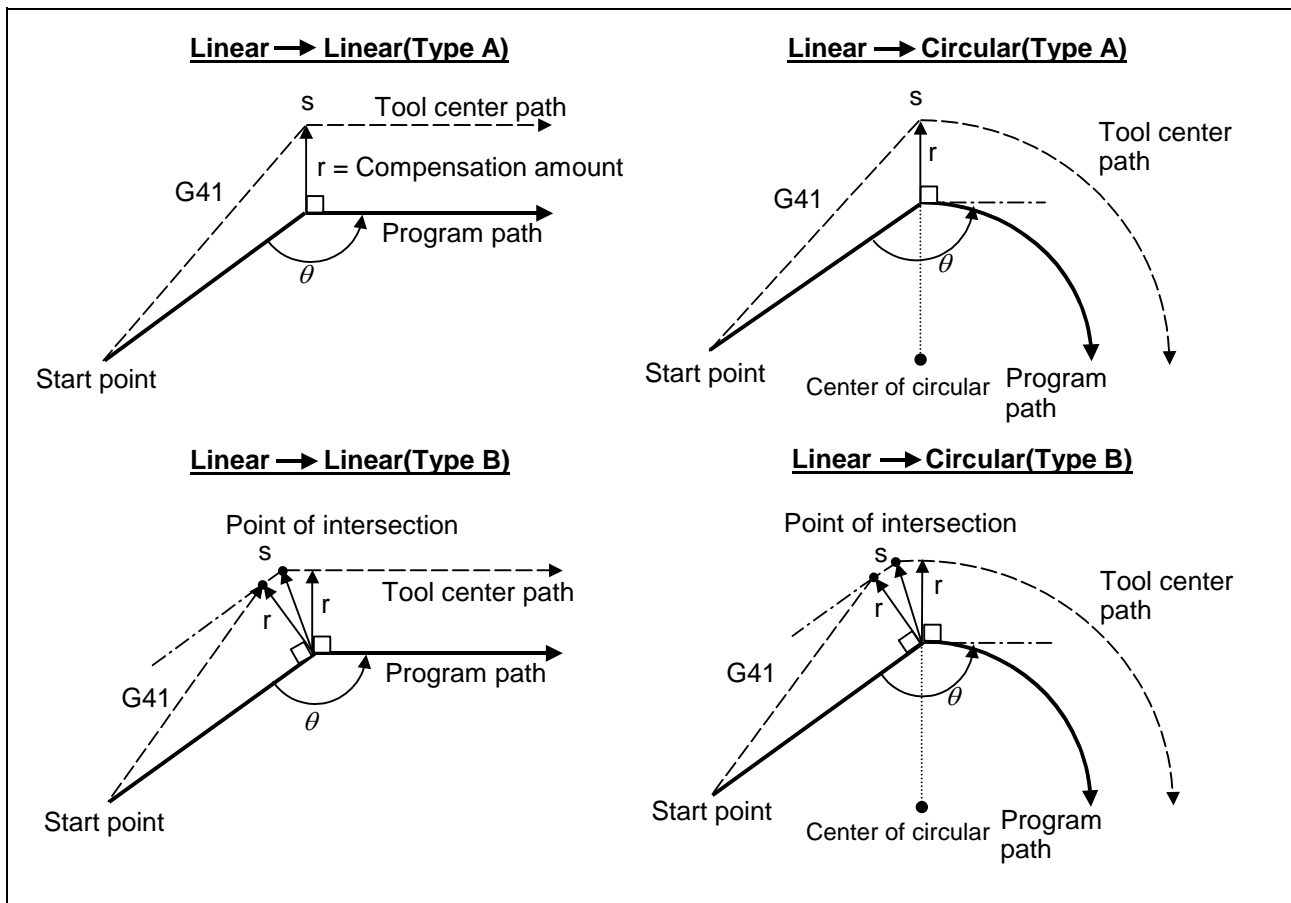
(3) For outer side of corner (obtuse angle) [0<90°]
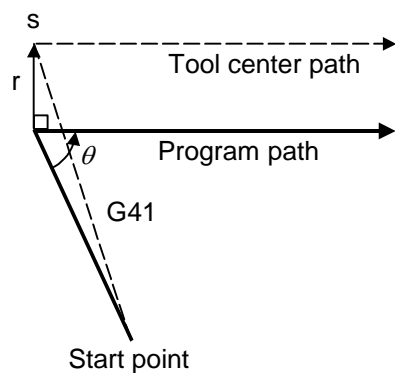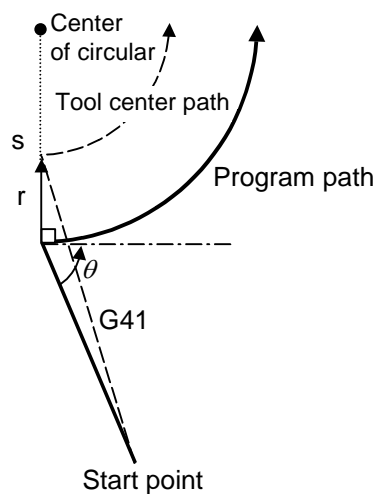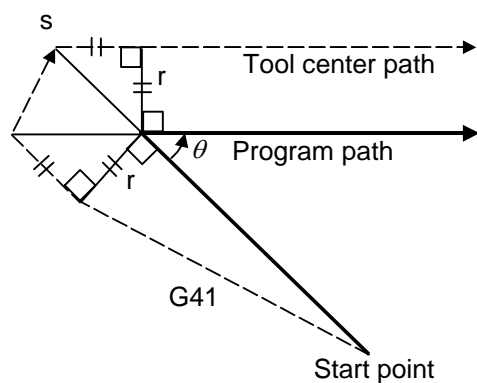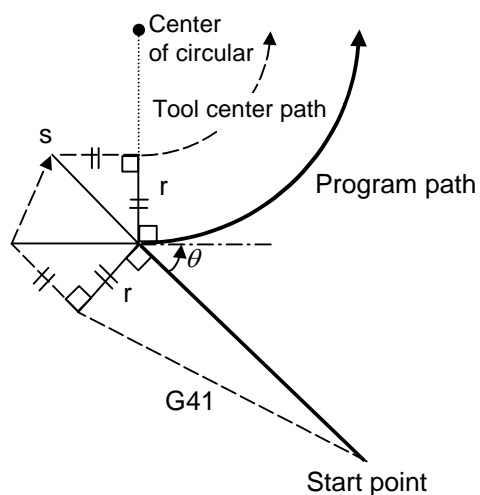
(Note 1) If there is axis movement command in the same block as G41 or G42, compensation is performed perpendicularly to the next block's direction.

Operation in compensation mode

Relative to the program path (G00, G01, G02, G03), the tool center path is found from the straight line/circular arc to make compensation.

Even if the same compensation command (G41, G42) is issued in the compensation mode, the command will be ignored.

When 4 or more blocks not accompanying movement are commanded continuously in the compensation mode, overcutting or undercutting will result.

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

(1)   Machining an outer wall
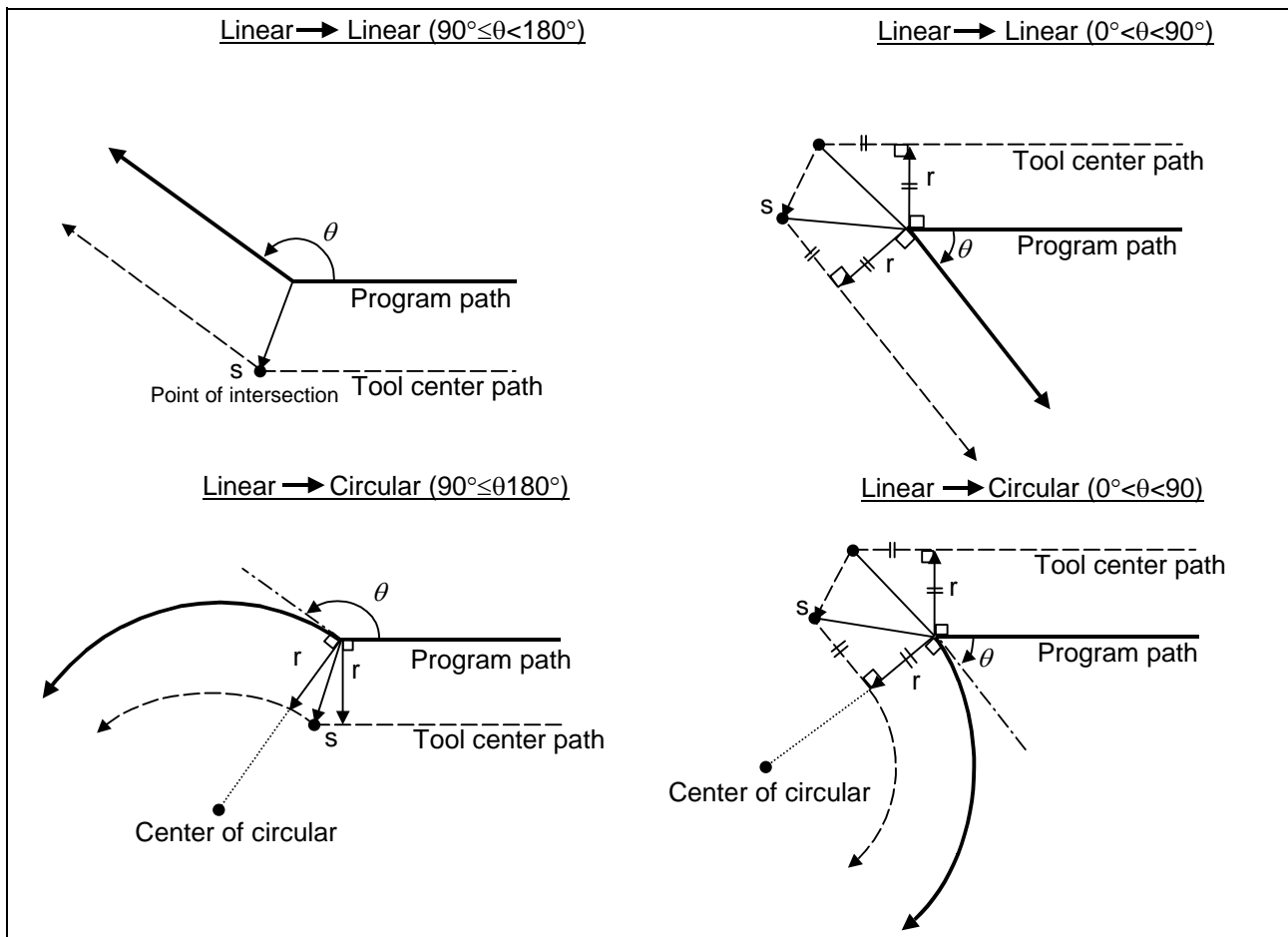
Linear → Linear (90°≤θ<180°)

Linear → Linear (0°<θ<90°)

Linear → Circular (90°≤θ180°)

Linear → Circular (0°<θ<90)

Circular → Linear (90°≤θ<180°)

Center of circular

Program path

θ

Tool center path

r

r

s

Point of intersection

Circular → Linear (0°<θ<90°)

Program path

θ

r

Tool center path

r

Center of circular

s

Circular → Circular (90°≤θ<180°)

Center of circular

Program path

θ

r

r

s

Point of intersection

Center of circular

Tool center path

Circular → Circular (0°<θ<90°)

Program path

θ

r

r

Tool center path

Center of circular

Center of circular

s
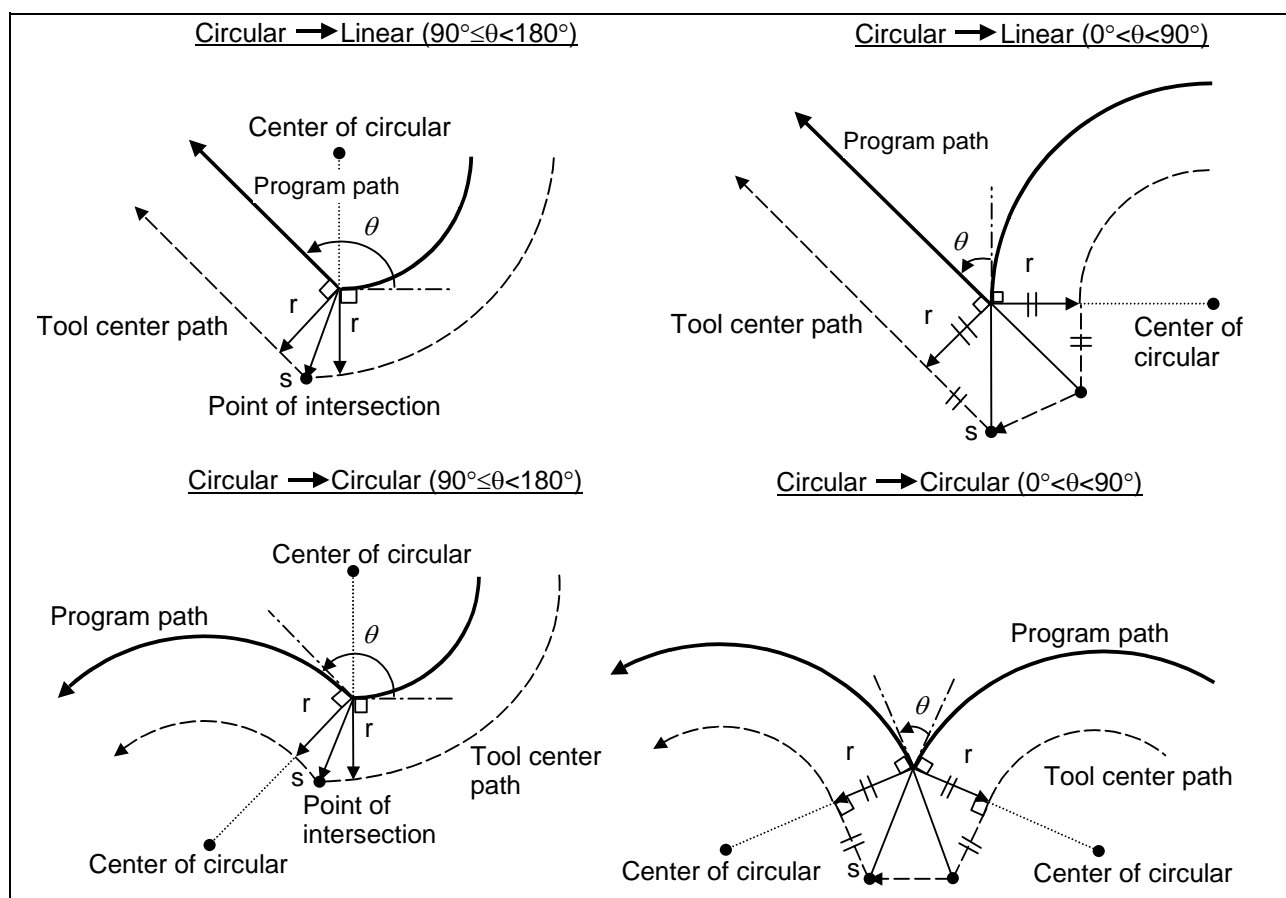
(2) Machining an inner wall

Linear ➔ Linear (Obtuse angle)

θ

r

s

Point of
intersection

Program path

Tool center path

Linear ➔ Linear (Acute angle)

θ

r

s

r

Program path

Tool center path
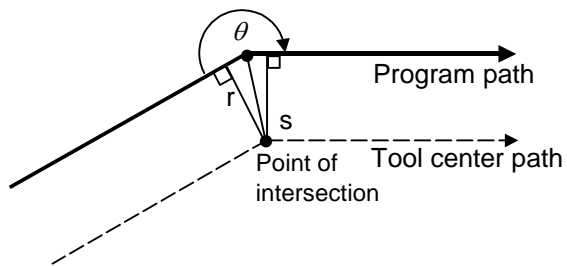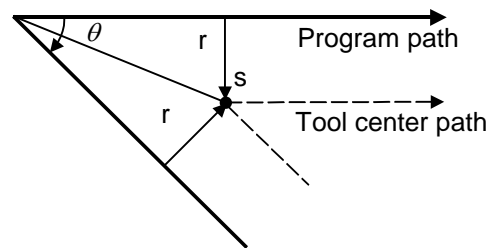
Linear ➔ Circular (Obtuse angle)

θ

Program path

Tool center path

s

Point of
intersection

r

Center of circular
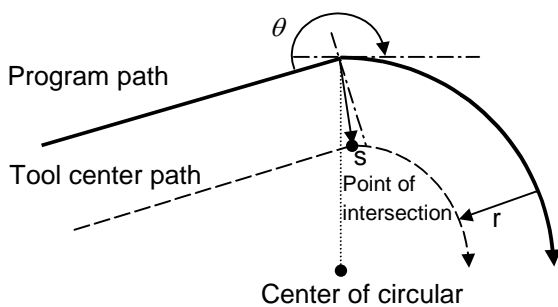
Linear ➔ Circular (Acute angle)
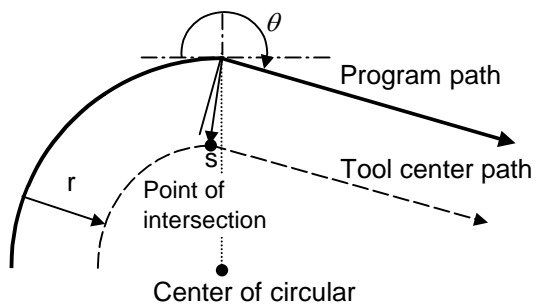
Program path

Tool center path

r

θ

s

Point of
intersection

Center of
circular

r

Circular ➔ Linear (Obtuse angle)

θ

Program path

Tool center path

r

s

Point of
intersection

Center of circular

Circular ➔ Linear (Acute angle)

Center of circular

θ

s

Point of
intersection

Program path

Tool center path

r

127

Circular → Linear (Obtuse angle)

Circular → Linear (Acute angle)
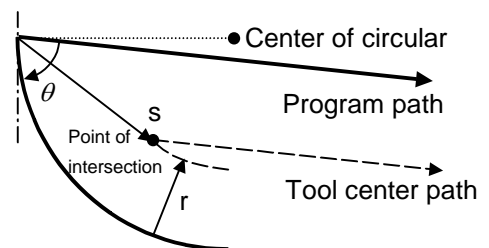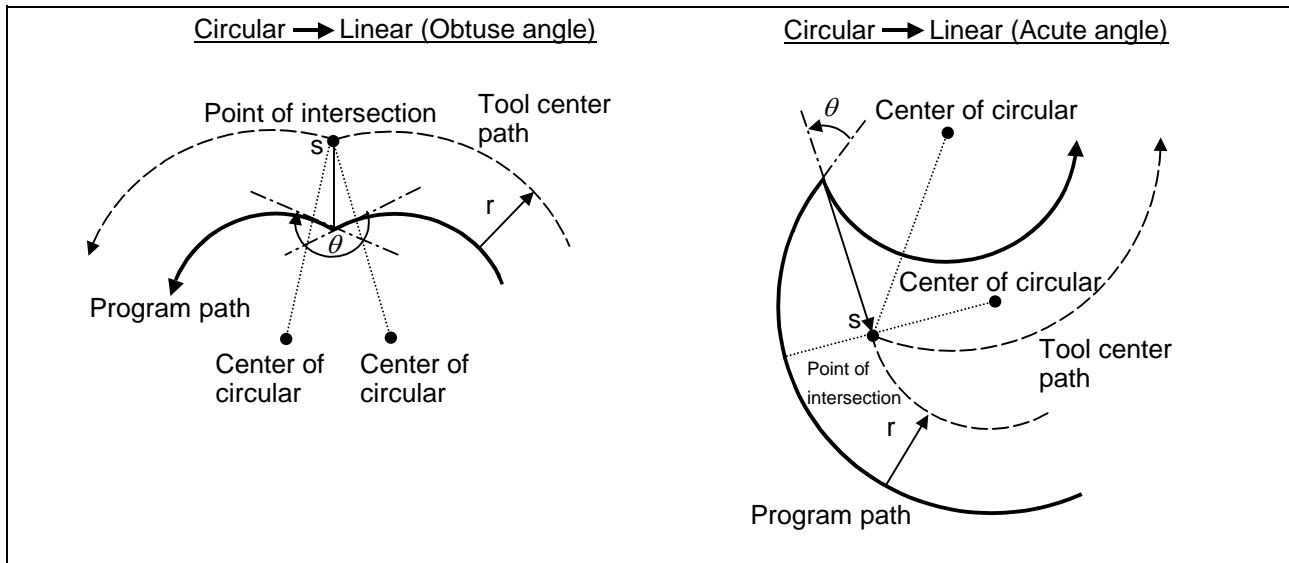
(3) When the arc end point is not on the arc

For spiral arc ....A spiral arc will be interpolated from the start to end point of the arc.

For normal arc command...........If the error after compensation is within parameter #1084 RadErr, the area from the arc start point to the end point is interpolated as a spiral arc.
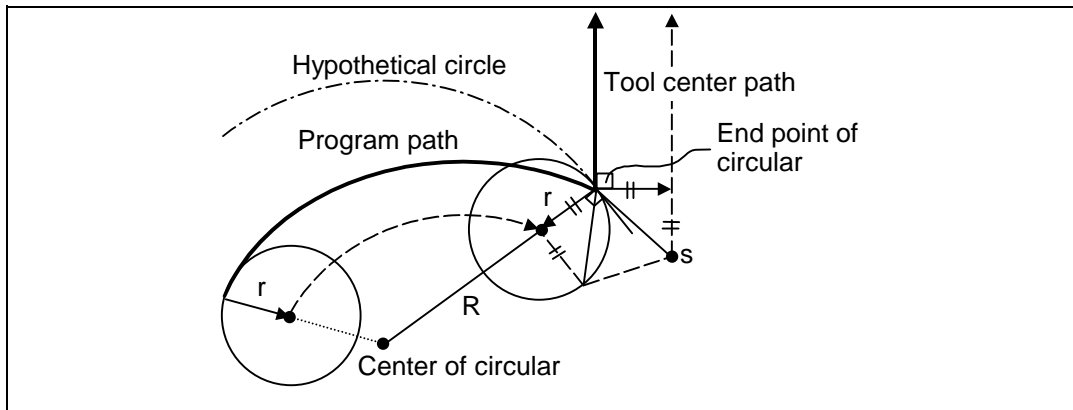
(4) When the inner intersection point does not exist

In an instance such as that shown in the figure below, the intersection point of arcs A and B may cease to exist due to the offset amount. In such cases, program error "P152" appears and the tool stops at the end point of the previous block.
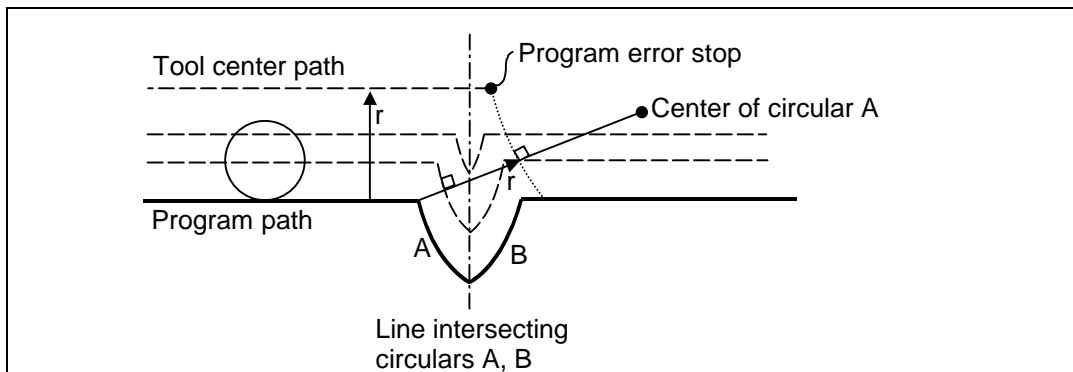
Tool radius compensation cancel

If either of the following conditions is met in the tool radius compensation mode, the compensation will be canceled. However, the movement command must be a command which is not a circular command.
If the compensation is canceled by a circular command, program error "P151" results.
(1)   The G40 command has been executed.
(2)   The D00 tool number has been executed.
The cancel mode is established once the compensation cancel command has been read, 5-block pre-reading is suspended an 1-block pre-reading is made operational.

Tool radius compensation cancel operation

(1) For inner side of corner
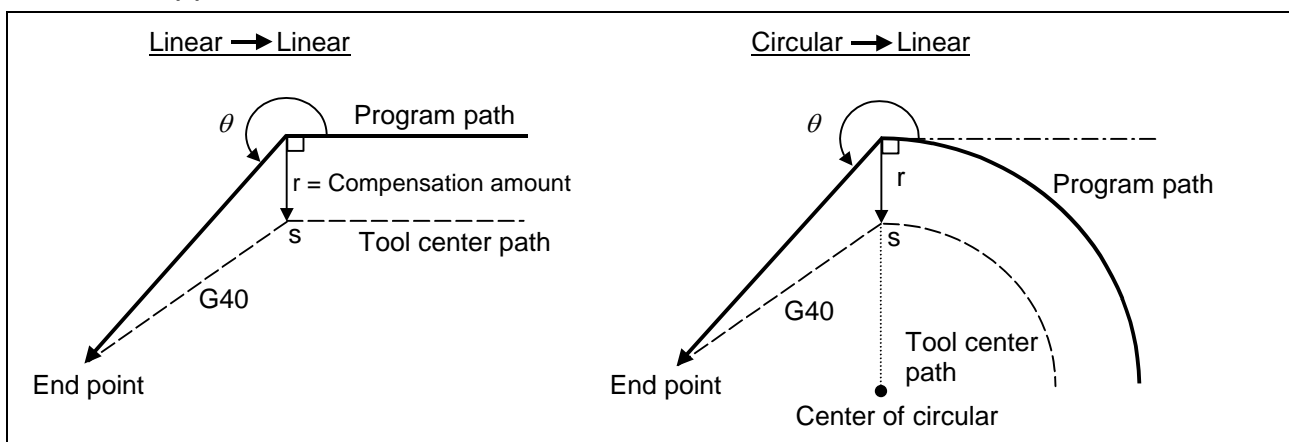
(2) For outer side of corner (obtuse angle)

Linear ⟶ Linear (Type A)
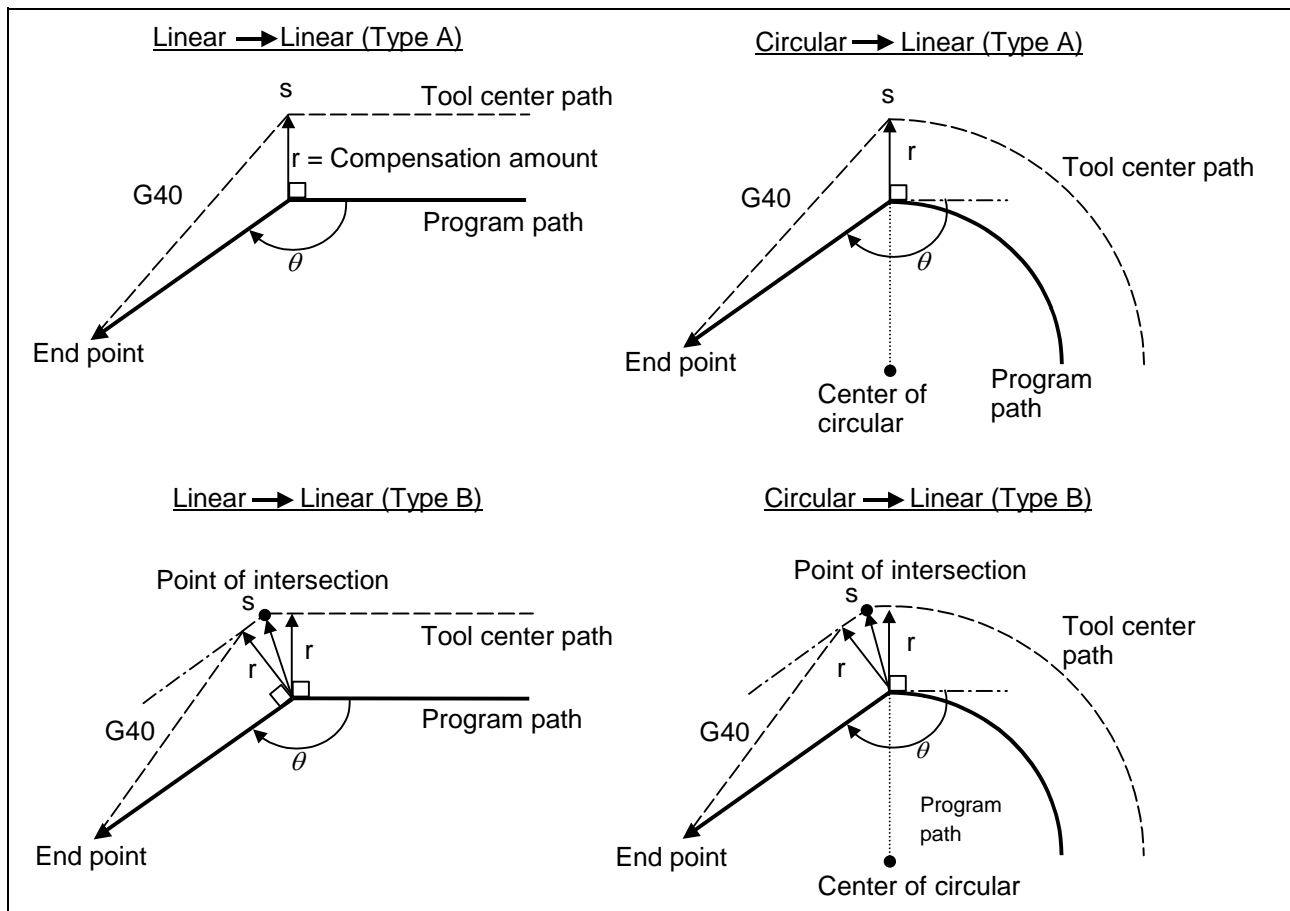
Circular ⟶ Linear (Type A)

Linear ⟶ Linear (Type B)

Circular ⟶ Linear (Type B)

(3) For outer side of corner (acute angle)

Linear → Linear (Type A)
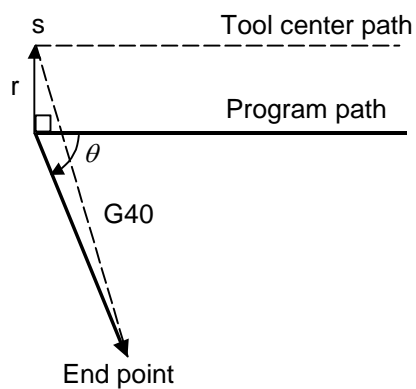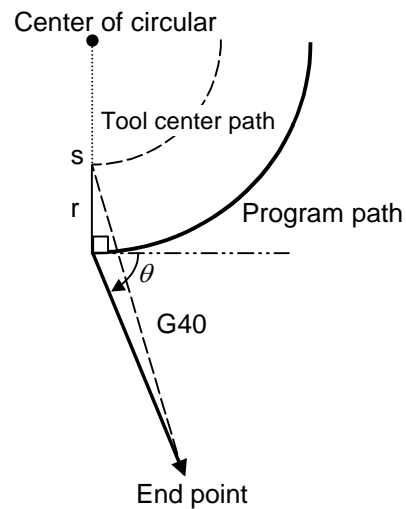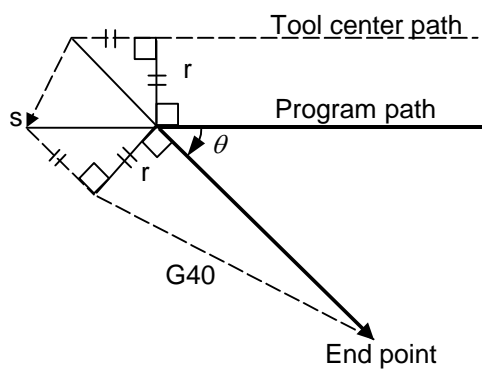
Circular → Linear (Type A)

Linear → Linear (Type B)

Circular → Linear (Type B)
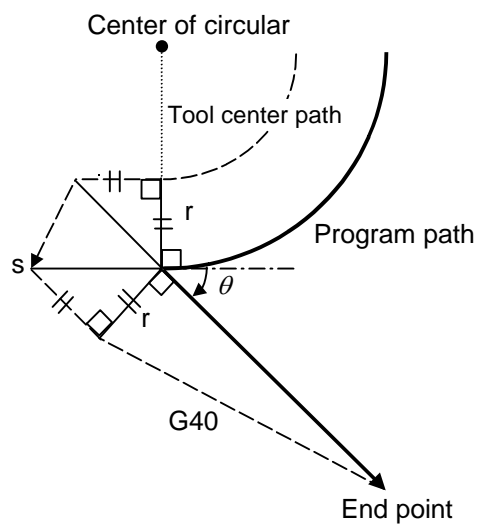
12.3.2 Other operations during tool radius compensation

Insertion of corner arc

An arc that uses the compensation amount as the radius is inserted without calculating the point of intersection at the workpiece corner when G39 (corner arc) is commanded.

For outer side compensation   For inner side compensation

N1   G28X0Y0 ;
N2   G91G01G42X20.Y20.D1F100 ;
N3   G39X40. ;
N4   G39Y40. ;
N5   G39X-40. ;
N6   Y-40. ;
N7   G40X-20.Y-20. ;
N8   M02 ;
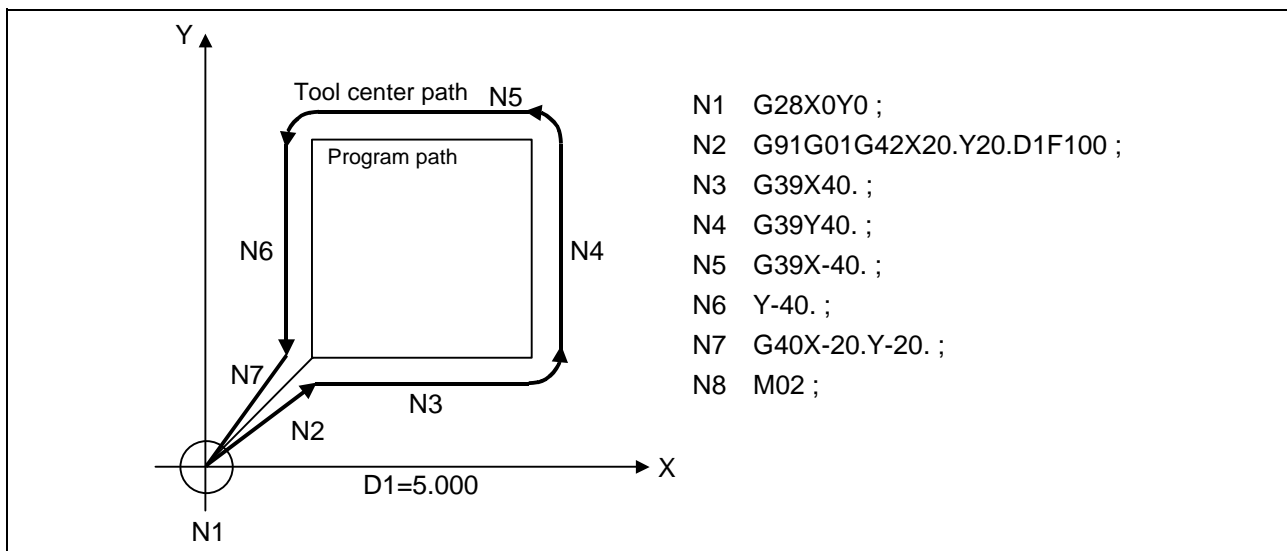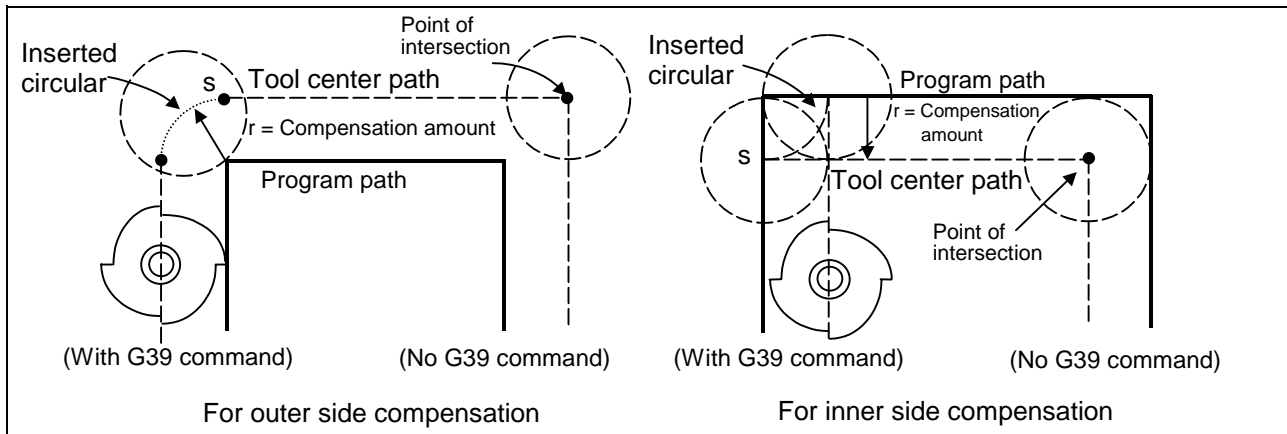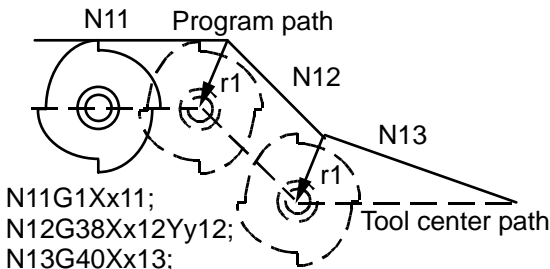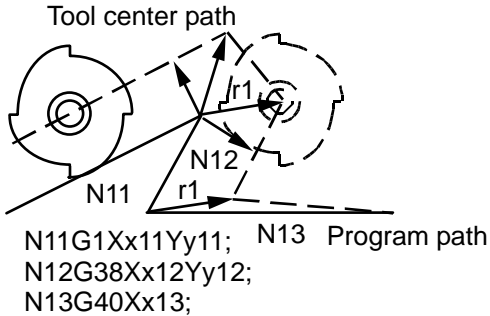
Changing and holding of compensation vector
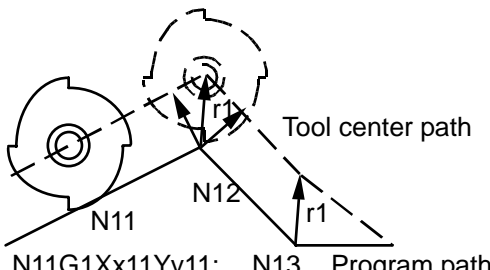
The compensation vector can be changed or held during tool diameter compensation by using the G38 command.

(1) Holding of vector: When G38 is commanded in a block having a movement command, the point of intersection will not be calculated at the program end point, and instead the vector of the previous block will be held.

G38 Xx Yy ;

This can be used for pick feed, etc.
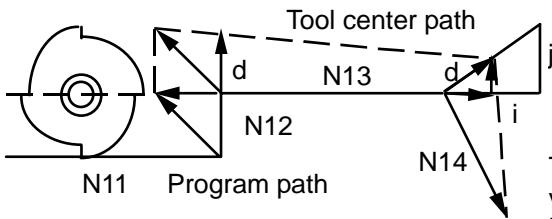
(1) Holding the inside compensation vector	(2) Holding the outside compensation acute angle
N11G1Xx11; N12G38Xx12Yy12; N13G40Xx13; r1: Vector at N11-N12 block intersection calculation	N11G1Xx11Yy11; N12G38Xx12Yy12; N13G40Xx13; r1: Vector at N11-N12 block intersection calculation

(3) Holding the outside compensation obtuse angle

N11G1Xx11Yy11;
N12G38Xx12Yy12;
N13G40Xx13;

r1: Vector at N11-N12 block intersection calculation

(2) Changing of vector: A new compensation vector direction can be commanded with I, J and K, and a new offset amount with D.
(These can be commanded in the same block as the movement command.)

G38 Ii Jj Dd ; (I, J and K will differ according to the selected plane.)

N11G1Xx11;
N12Yy12;
N13G38Xx13IiJjDd;
N14G40Xx14Yy14;

The compensation amount "d" vector is created in the commanded i and j vector direction.

(Note)   If G38 is commanded in the same block as the circular block (G02/G03) I and J commands, I and J will be handled as the G38 vector, and an error will occur.

Changing the compensation direction during tool diameter compensation

The compensation direction is determined by the tool diameter compensation commands (G41, G42) and compensation amount sign.

Compensation amount sign G code	+	–
G41	Left-hand compensation	Right-hand compensation
G42	Right-hand compensation	Left-hand compensation

The compensation direction can be changed by changing the compensation command in the compensation mode without the compensation having to be first canceled. However, no change is possible in the compensation start block and the following block.
Refer to section 12.3.5 "Precautions for tool diameter compensation" for the movement when the symbol is changed.

Linear → Linear

Tool center path

Point of intersection

Program path
G41   G41   G42

If there is no point of intersection when the compensation direction is changed.

Linear ↔ Circular

G41   G42   G41   G41   G42

Program path

Tool center path

Circular ↔ Circular

Tool center path

Circular center

G41   G42

Program path

G41

G41   G41   G42

Circular center

Linear return

G41

Tool center path

G42

r

Program path

Arc exceeding 360° due to compensation

G42

Program path

Tool center path

G41

G42

Uncut section
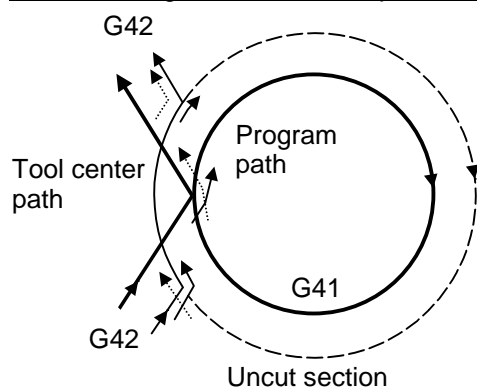
In the case below, it is possible that the arc may exceed 360°

a. With offset direction selection based on G41/G42

b. I, J, K was commanded in G40.

In cases like this the tool center path will pass through a section where the arc is doubled due to the compensation and a section will be left uncut.

Command for eliminating offset vectors temporarily

When the following command is issued in the compensation mode, the offset vectors are temporarily eliminated and a return is then made automatically to the compensation mode.
In this case, the compensation is not canceled, and the tool goes directly from the intersection point vector to the point without vectors or, in other words, to the programmed command point.
When a return is made to the compensation mode, it goes directly to the intersection point.

(1)   Reference point return command

Intermediate point

| N5 | N6 | N7 | N8 |

```
(G41)
N5    G91   G01   X60.   Y30.   ;  ← Temporarily no compensation vectors at intermediate
N6    G28         X50.   Y-40.  ;     point.
N7                X30.   Y-60.  ;     (Reference point when there is no intermediate point)
N8                X70.   Y40.   ;
```
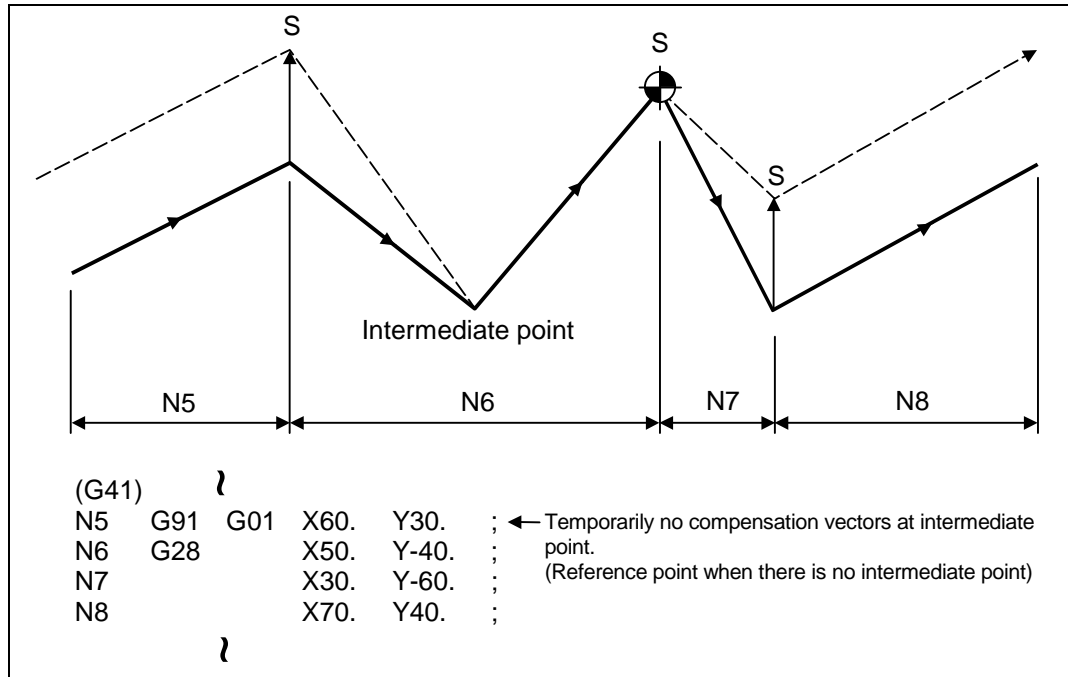
(2)   G33 thread cutting command

Tool nose radius compensation does not apply to the G33 block.
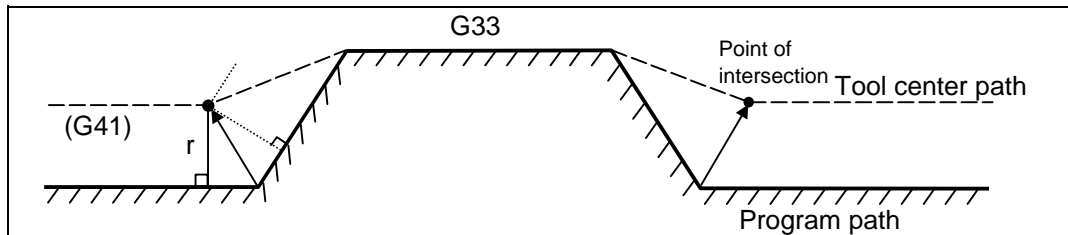
(3)   The compensation vector will be eliminated temporarily with the G53 command (basic machine coordinate system selection).

(Note 1) The offset vectors do not change with the coordinate system setting (G92) command.

Blocks without movement and pre-read inhibit M command

The following blocks are known as blocks without movement.

a. M03 ; .... M command
b. S12 ; .... S command
c. T45 ; .... T command
d. G04 X500 ; .... Dwell
e. G22 X200. Y150. Z100 ; .... Machining inhibit region setting
f. G10 L10 P01 R50 ; .... Offset amount setting
g. G92 X600. Y400. Z500. ; ... Coordinate system setting
h. (G17) Z40. ; .... Movement but not on offset plane
i. G90 ; .... G code only
j. G91 X0 ; .... Zero movement amount ..... Movement amount is zero

No movement

M00, M01, M02 and M30 are handled as pre-read inhibit M codes.

(1)   When command is assigned at start of the compensation

Perpendicular compensation will be applied on the next movement block.

```
N1  X30.  Y60.  ;
N2  G41  D10  ;  ← Block without
                    movement
N3  X20. Y-50.  ;
N4  X50. Y-20.  ;
```
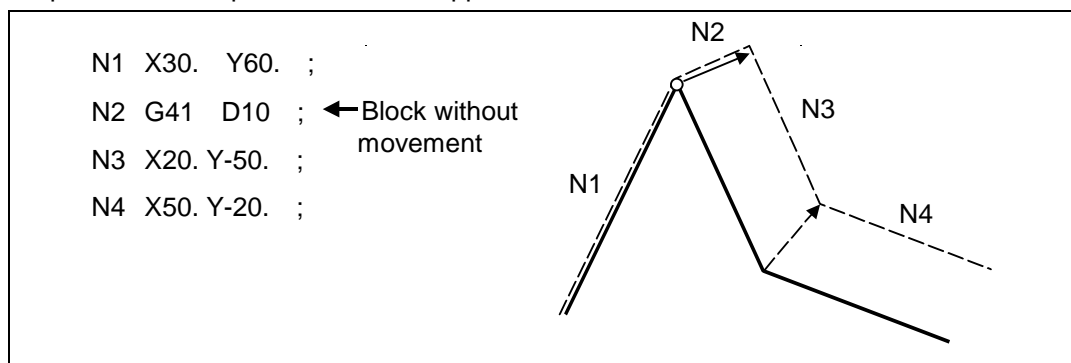
If four or more blocks containing no move command continue or if there is a pre-read inhibit M code, no compensation vector is generated.

```
N1  X30.   Y60.  ;
N2  G41   D10 ;
N3  G4   X1000 ;
N4  F100 ;           Block without
N5  S500 ;           movement
N6  M3 ;
N7  X20. Y-50. ;
N8  X50. Y-20. ;
```

```
N1  G41   X30.  Y60.   D10 ;
N2  G4   X1000 ;
N3  F100 ;          Block without
N4  S500 ;          movement
N5  M3 ;
N6  X20.   Y-50. ;
N7  X50.   Y-20. ;
```
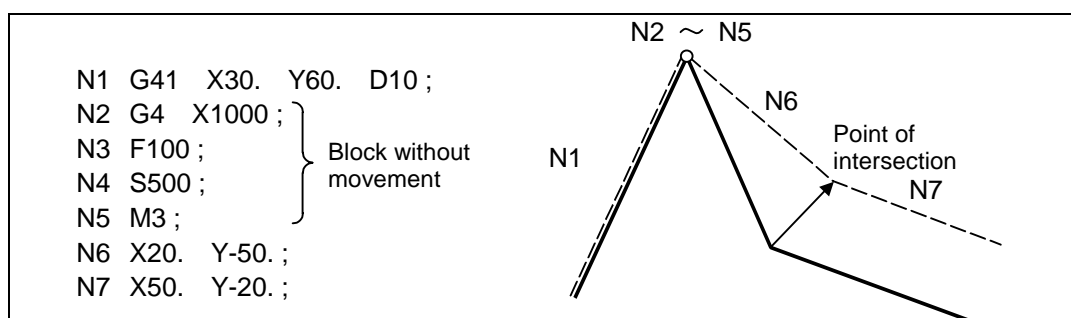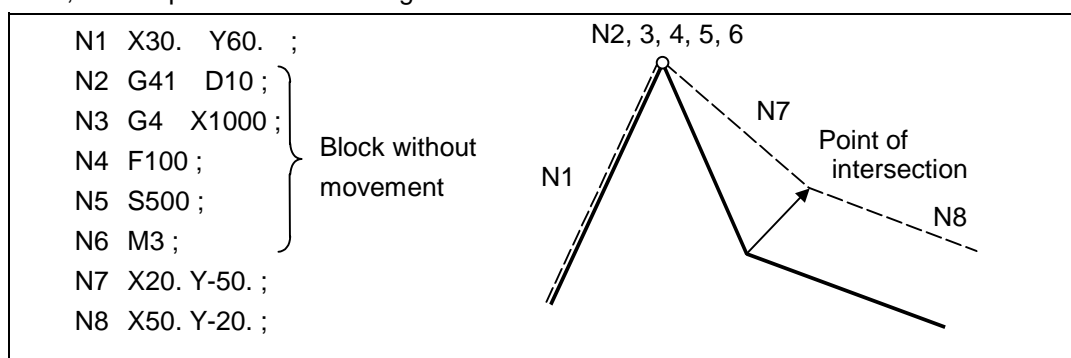
(2)   When command is assigned in the compensation mode

When 4 or more blocks without movement follow in succession in the compensation mode or when there is no pre-read inhibit M code, the intersection point vectors will be created as usual.

```
N6  G91 X100.   Y200. ;

N7  G04 X   P1000 ;·················· Block without
                                      movement
N8  X200. ;
```
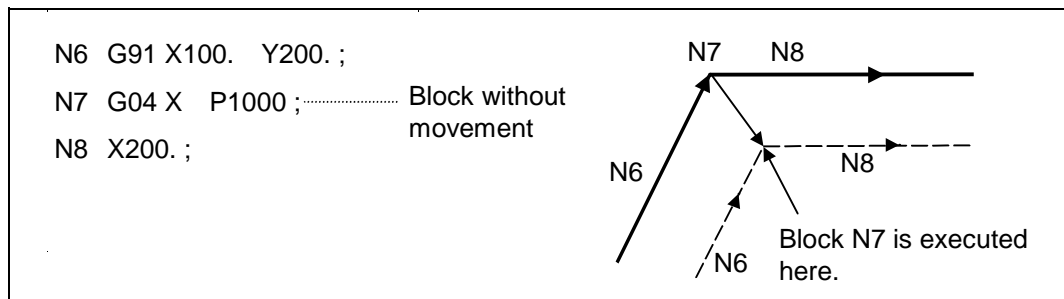
When 4 or more blocks without movement follow in succession or if there is a pre-read inhibit M code, the offset vectors are created perpendicularly at the end point of the previous block.

```
N6    X100.      Y200. ;

N7    G4   X1000 ;

N8    F100 ;

N9    S500 ;            Block without
                        movement
N10   M4 ;

N11   W100. ;

In this case, a cut results.
```

138
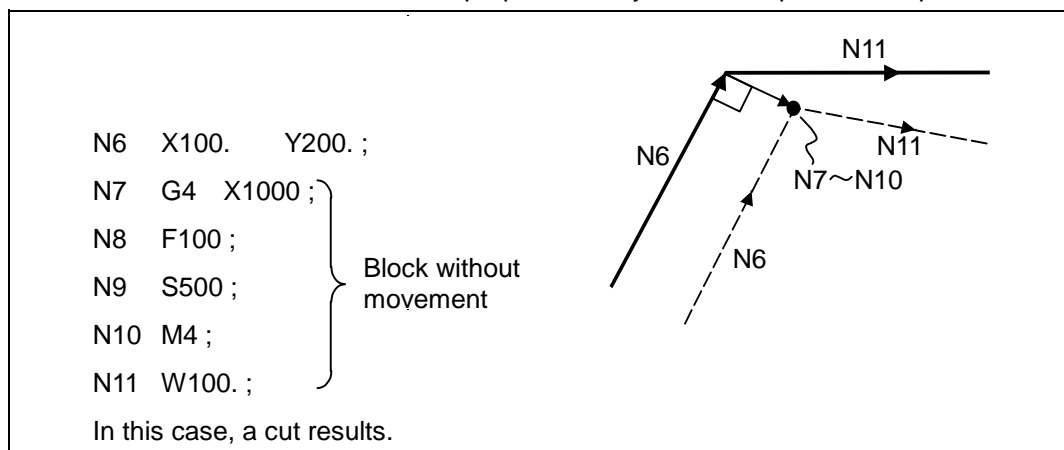
(3) When commanded together with compensation cancel

```
N6   X100. Y200.   ;
N7   G40   M5 ;
N8   X100. Y50.   ;
```
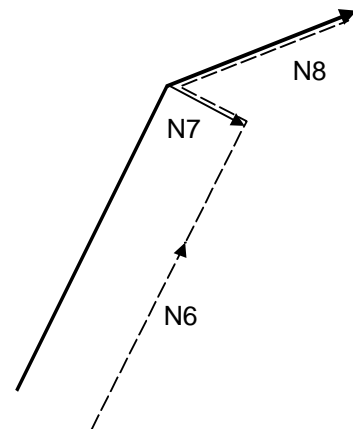
When I, J, K are commanded in G40

(1) If the final movement command block in the four blocks before the G40 block is the G41 or G42 mode, it will be assumed that the movement is commanded in the vector I, J or K direction from the end point of the final movement command. After interpolating between the hypothetical tool center path and point of intersection, it will be canceled. The compensation direction will not change.

```
N1 (G41) G1X_   ;
N2 G40XaYbIiJj;
```
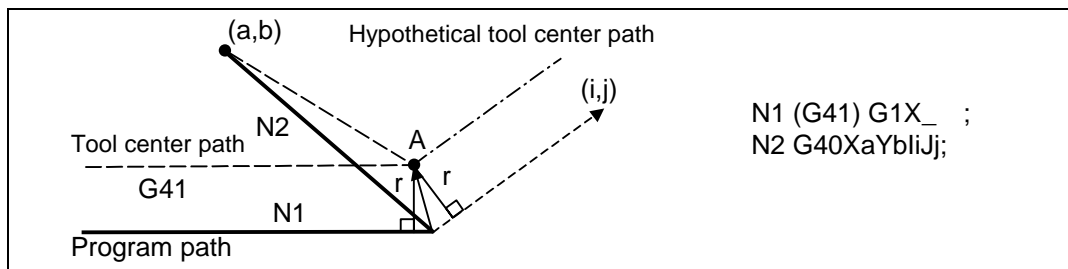
In this case, the point of intersection will always be obtained, regardless of the compensation direction, even when the commanded vector is incorrect as shown below.

When the I and j symbols in the above program example are incorrect
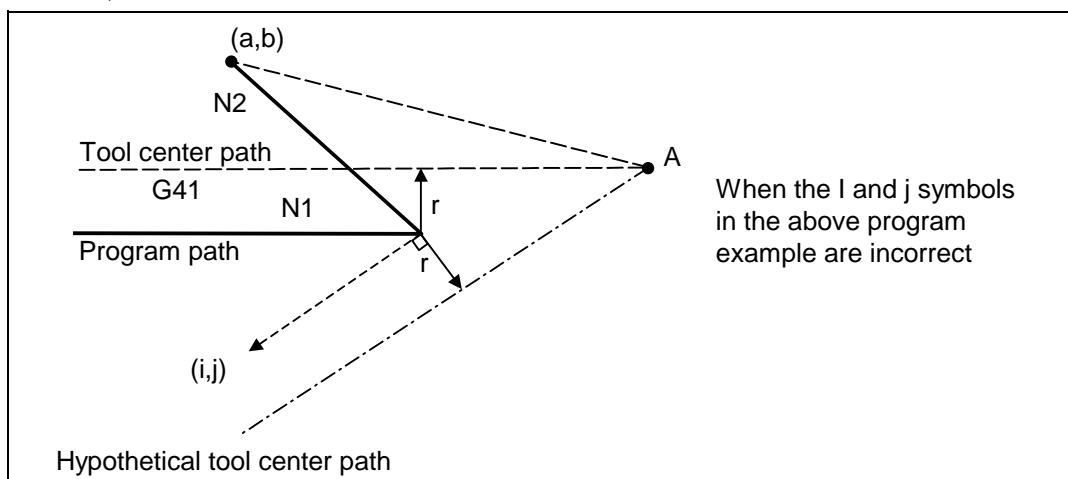
If the compensation vector obtained with point of intersection calculation is extremely large, a perpendicular vector will be created in the block before G40.
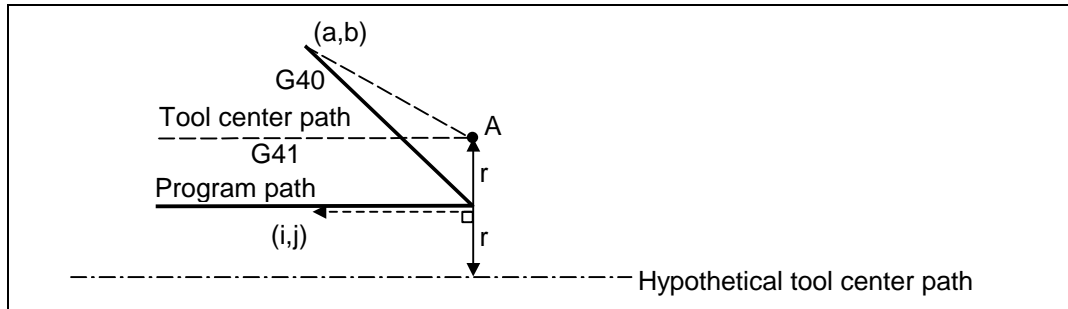
(2) If the arc is 360° or more due to the details of I, J and K at G40 after the arc command, an uncut section will occur.

N1 (G42,G91) G01X200. ;
N2 G02 J150. ;
N3 G40 G1X150. Y-150. I-100. J100. ;
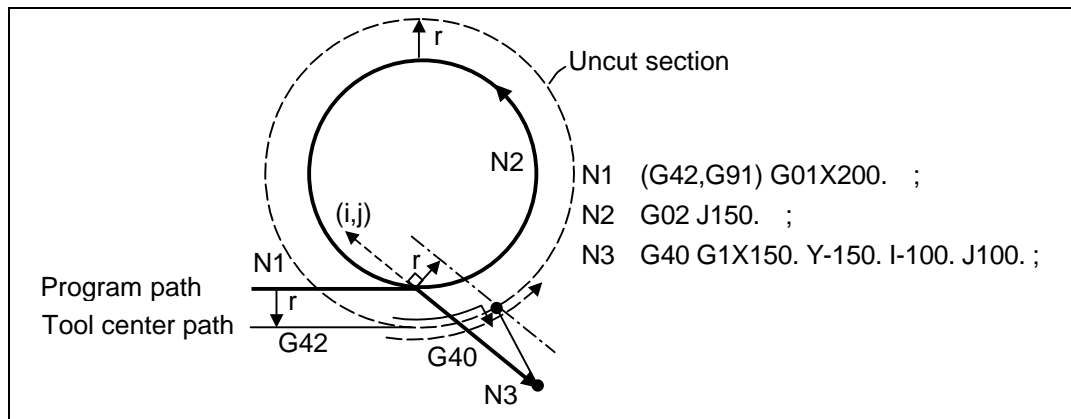
Corner movement

When a multiple number of offset vectors are created at the joints between movement command blocks, the tool will move in a straight line between those vectors.
This action is called corner movement.
When the vectors do not coincide, the tool moves in order to machine the corner although this movement is part and parcel of the joint block. Consequently, operation in the single block mode will execute the previous block + corner movement as a single block and the remaining joining movement + following block will be executed as a single block in the following operation.
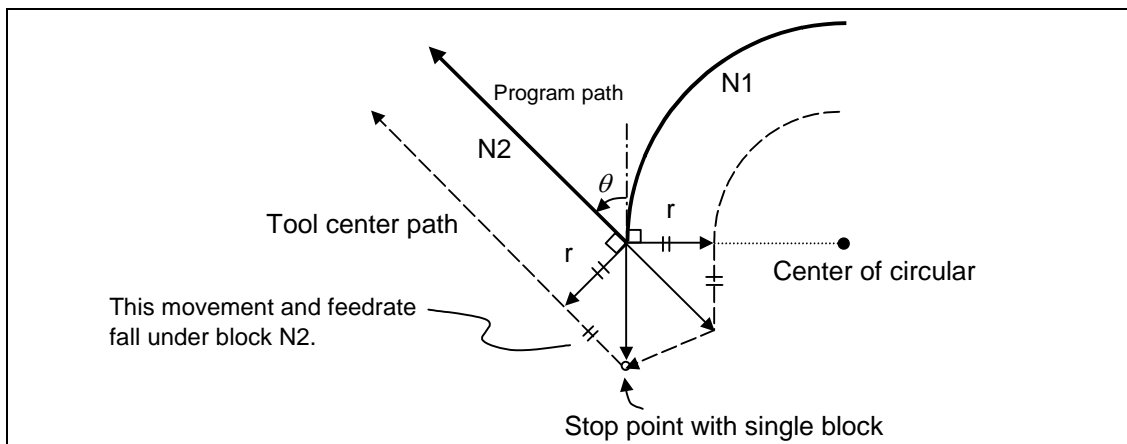
12.3.3 G41/G42 commands and I, J, K designation
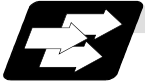
Function and purpose

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

Command format

```
G17 (XY plane)  G41/G42 X__ Y__ I__ J__ ;
G18 (ZX plane)  G41/G42 X__ Z__ I__ K__ ;
G19 (YZ plane)  G41/G42 Y__ Z__ J__ K__ ;
```

Assign an linear command (G00, G01) in a movement mode.

I, J type vectors (G17 XY plane selection)

The new I, J type vector (G17 plane) created by this command is now described. (Similar descriptions apply to vector I, K for the G18 plane and to J, K for the G19 plane.)
As shown in the figures, the vectors with a size equivalent to the offset amount are made to serve as the I, J type compensation vector perpendicularly to the direction designated by I, J without the intersection point of the programmed path being calculated. the I, J vector can be commanded even in the mode (G41/G42 mode in the block before) and even at the compensation start (G40 mode in the block before).

(1) When I, J is commanded at compensation start


```
N100   G91   G41   X100. Y100.   I150. D1 ;
N110   G04   X1000 ;
N120   G01   F1000 ;
N130   S500 ;
N140   M03 ;
N150   X150. ;
```

Program path
Tool center path
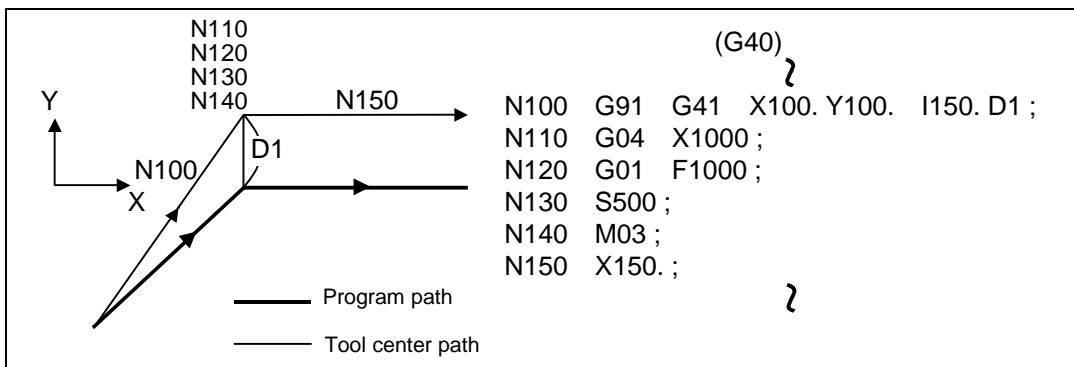
(2) When there are no movement commands at the compensation start.


```
N1   G41   I150. D1 ;
N2   G91   X100. Y100. ;
N3   X150. ;
```
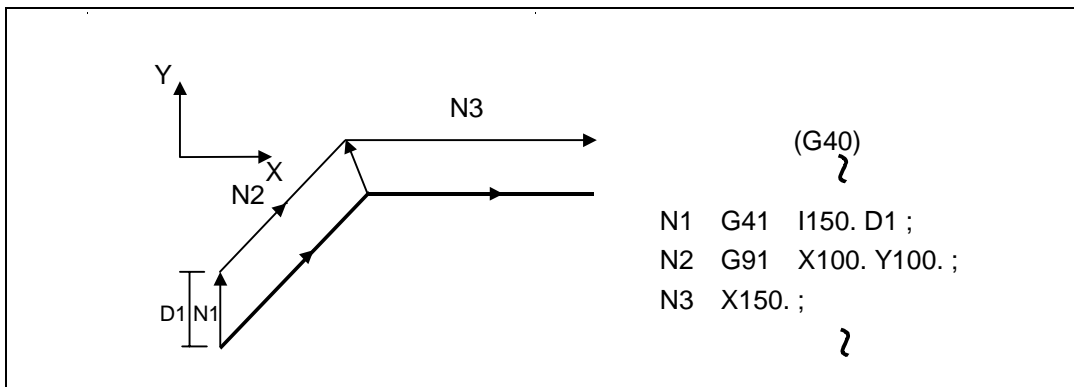
(3)   When I, J has been commanded in the G41/G42 mode (G17 plane)
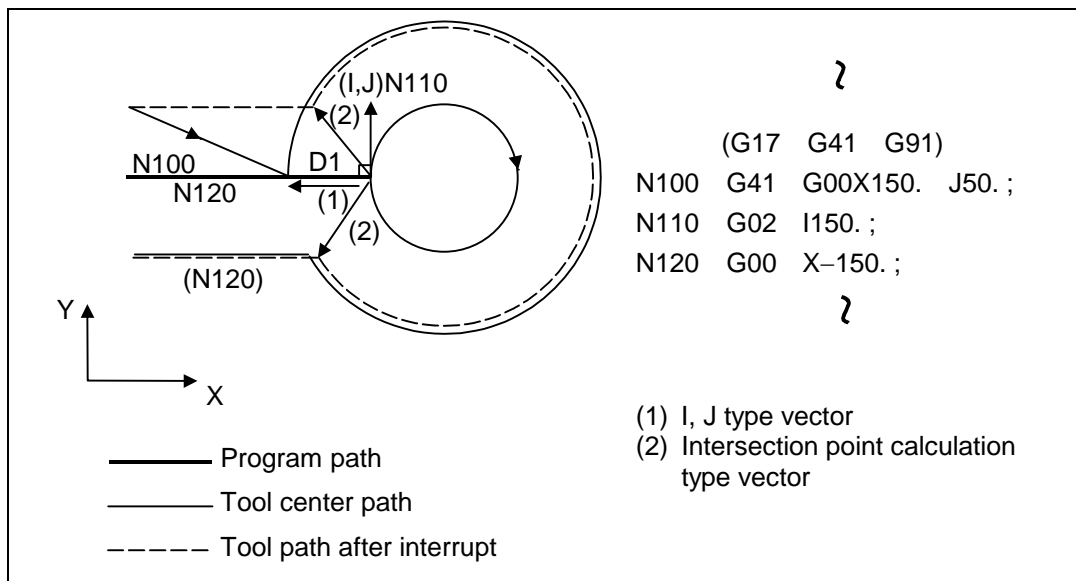
(G17   G41   G91)
N100   G41   G00X150.   J50. ;
N110   G02   I150. ;
N120   G00   X−150. ;

(1) I, J type vector
(2) Intersection point calculation
     type vector

——————— Program path

——————— Tool center path

− − − − − Tool path after interrupt

(Reference)
(a)   G18 plane

(G18   G41   G91)
N100   G41   G00 Z150.   I50. ;
N110   G02   K50. ;
N120   G00   Z−150. ;

(b)   G19 plane

(G19   G41   G91)
N100   G41   G00 Y150.   K50. ;
N110   G02   J50. ;
N120   G00   Y−150. ;
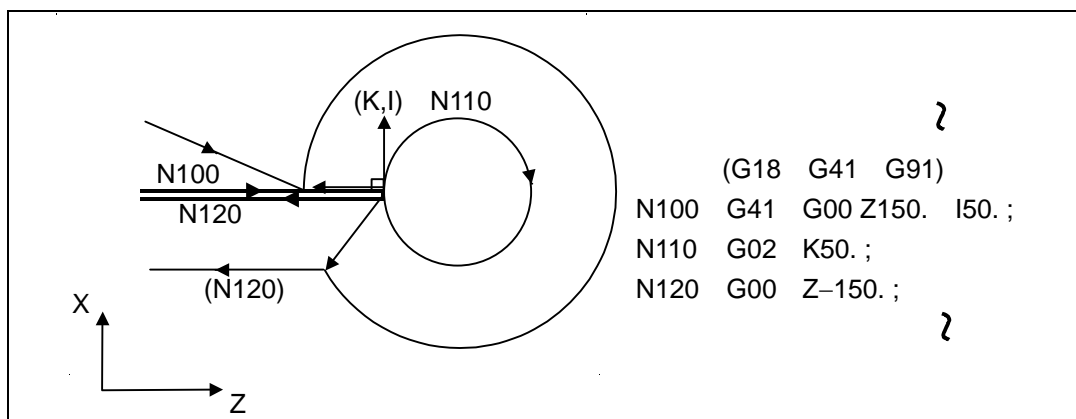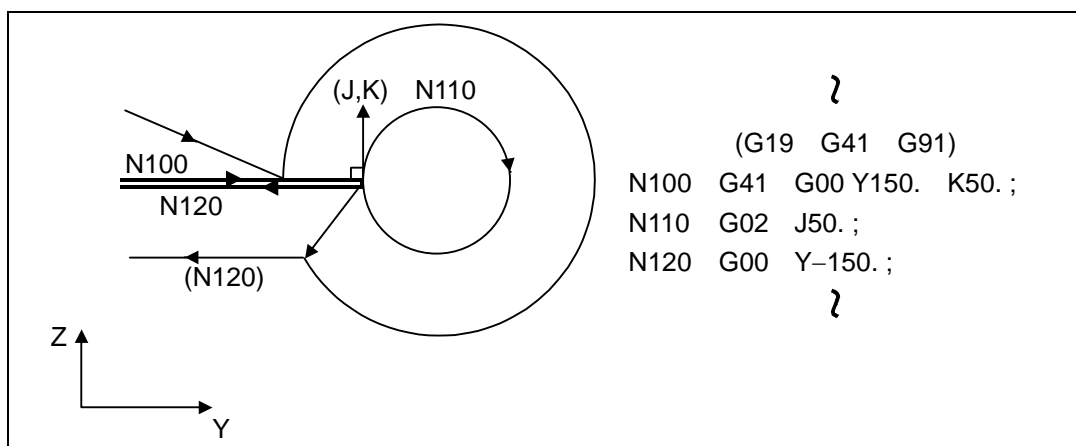
(4) When I, J has been commanded in a block without movement


```
N1  G41  D1  G01  F1000 ;
N2  G91  X100. Y100. ;
N3  G41  I50. ;
N4  X150. ;
N5  G40 ;
```
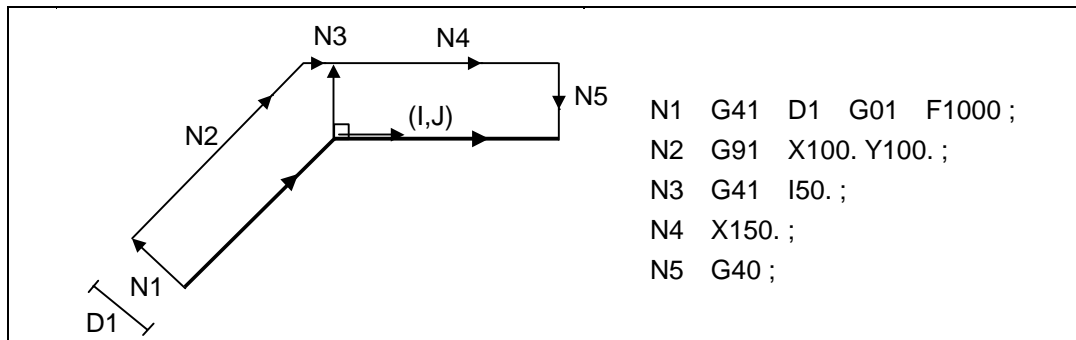
Direction of offset vectors

(1) In G41 mode

Direction produced by rotating the direction commanded by I, J through 90° to the left from the forward direction of the Z axis (axis 3) as seen from the zero point
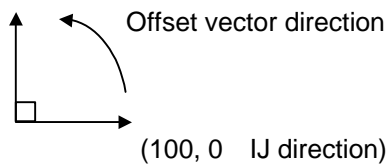
(Example 1) With I100.　　　　　　　**(Example 2)** With I-100.
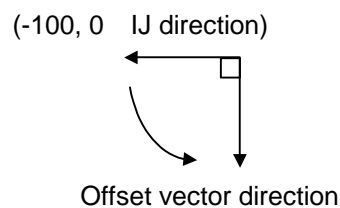
(2) In G42 mode

Direction produced by rotating the direction commanded by I, J through 90° to the right from the forward direction of the Z axis (axis 3) as seen from the zero point
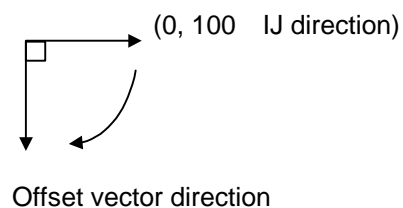
(Example 1) With I100.　　　　　　　**(Example 2)** With I-100.
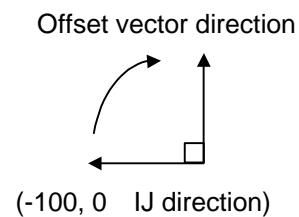
12. Tool Offset Functions

Selection of offset modal

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


```
N1   G28   X0 Y0 ;
N2   G41   D1   F1000 ;
N3   G01   G91   X100. Y100. ;
N4   G42   X100.   I100.   J-100.   D2 ;
N5   X100.   Y-100. ;
N6   G40 ;
N7   M02 ;
%
```
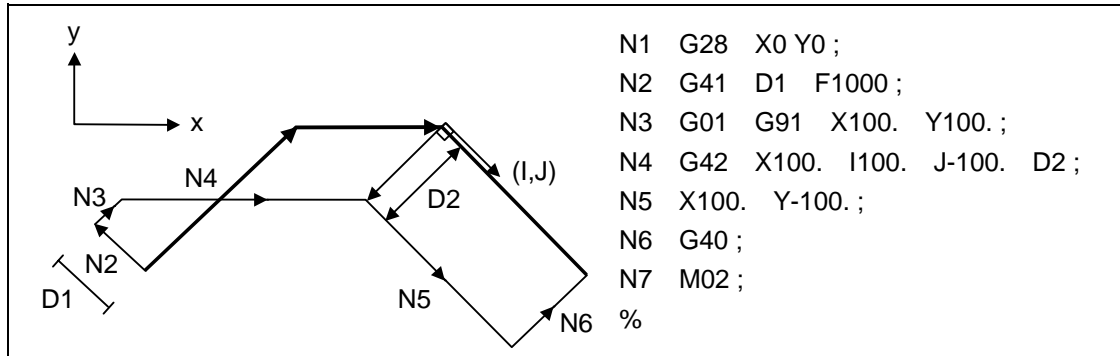
Offset amount for offset vectors

The offset amounts are determined by the offset number (modal) in the block with the I, J designation.

< Example 1>


```
(G41   D1   G91)
        ?
N100   G41   X150.   I50. ;
N110   X100.   Y-100. ;
        ?
```
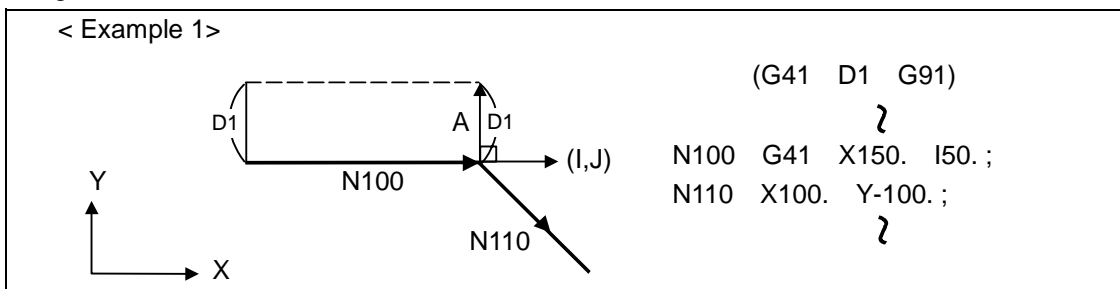
Vector A is the offset amount entered in offset number modal D1 in the N200 block.

< Example 2>


```
(G41   D1   G91)
        ?
N200   G41   X150.   I50.   D2 ;
N210   X100.   Y-100. ;
        ?
```
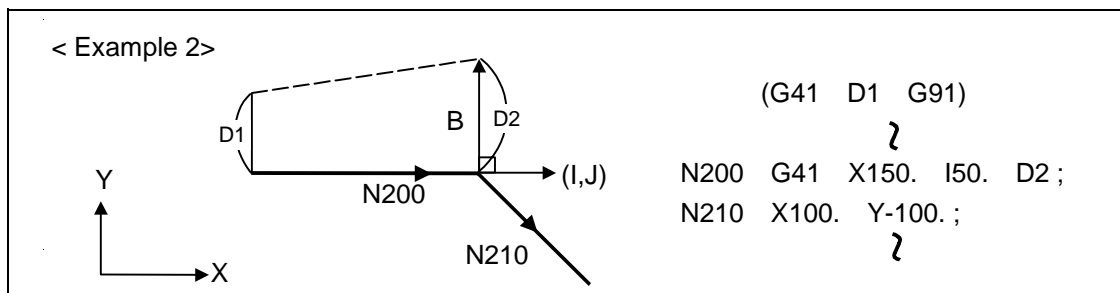
Vector B is the offset amount entered in offset number modal D2 in the N200 block.

144

Precautions

(1) Issue the I, J type vector in a linear mode (G0, G1). If it is issued in an arc mode at the start of compensation, program error "P151" (G2, 3 CMP. ERR) will result.
An IJ designation in an arc mode functions as an arc center designation in the offset mode.

(2) When the I, J type vector has been designated, it is not deleted (avoidance of interference) even if there is interference.
Consequently, overcutting may arise in such a case.


```
N1   G28   X0Y0 ;
N2   G42   D1   F1000 ;
N3   G91   X100. ;
N4   G42   X100.  Y100. I10. ;
N5   X100.  Y-100. ;
N6   G40 ;
N7   M02 ;
```
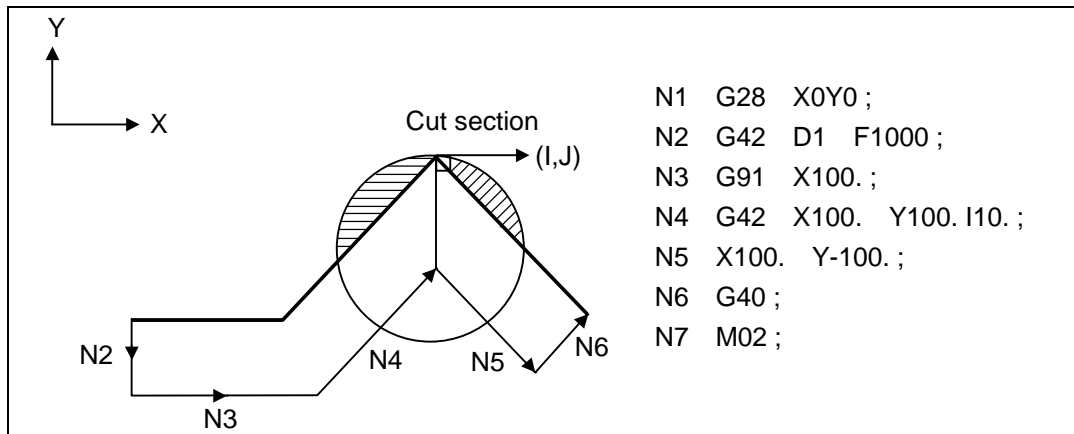
(3) The vectors differ for the G38 I _J_ (K_) command and the G41/G42 I_J_(K_) command.

	G38	**G41/G42**
Example	↕ (G41) ↕ G38 G91 X100. I50. J50. ; ↕ 	↕ (G41) ↕ G41 G91 X100. I50. J50. ; ↕
	Vector in IJ direction having an offset amount size	Vector perpendicular in IJ direction and having an offset amount size

(4) Refer to the following table for the offset methods based on the presence and/or absence of the G41 and G42 commands and I, J, (K) command.

G41/G42	I, J (K)	Offset method
No	No	Intersection point calculation type vector
No	Yes	Intersection point calculation type vector
Yes	No	Intersection point calculation type vector
Yes	Yes	I, J, type vector No insertion block

N1   G91   G01   G41   X200. D1   F1000 ;
N2   X-150.   Y150. ;
N3   G41   X300.   I50. ;
N4   X-150.   Y-150. ;
N5   G40   X-200. ;

During the I, J type vector compensation,
the A insertion block will not exist.
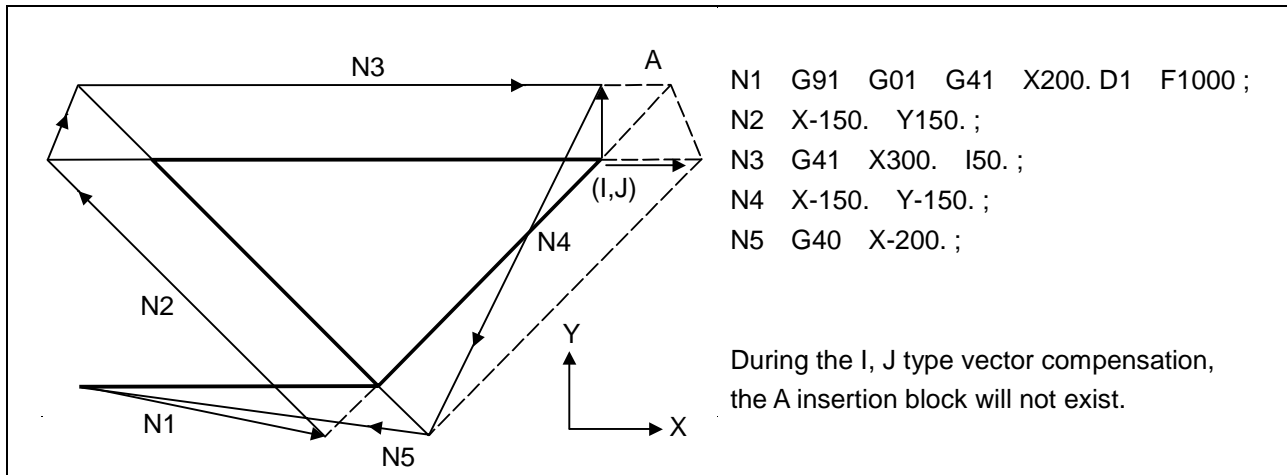
12.3.4 Interrupts during tool radius compensation

MDI interrupt

Tool radius compensation is valid in any automatic operation mode-whether tape, memory or MDI operation.
An interrupt based on MDI will give the result as in the figure below after block stop during tape or memory operation.

(1) Interrupt without movement (tool path does not change)

```
N1 G41D1;

N2 X20. Y50. ;          MDI interrupt

N3 G3 X40. Y-40. R70. ; S1000 M3;
```

S (Stopping position
    for single block)

N2   N3
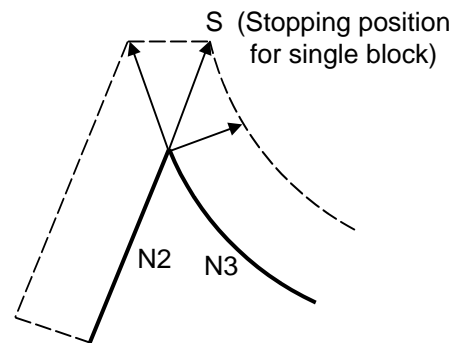
(2) Interrupt with movement

The offset vectors are automatically re-ca
lculated at the movement block after interrupt.

With linear interrupt

```
N1 G41D1;

N2 X20. Y50. ;          MDI interrupt

N3 G3 X40.Y-40. R70. ;  X50. Y-30. ;

                        X30.  Y50.  ;
```

S

S

N2

N3
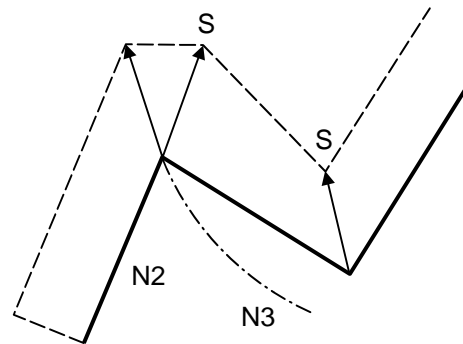
With circular interrupt

```
N1 G41 D1 ;

N2 X20. Y50. ;          MDI interrupt

N3 G3 X40. Y-40. R70.   G2 X40. Y-40. R70. ;

                        G1 X40. ;
```

S

S

N2   N3
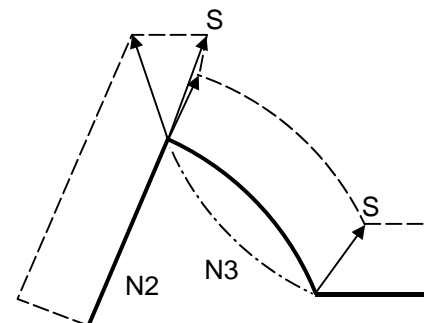
Manual interrupt

(1)   Interrupt with manual absolute OFF.
The tool path is shifted by an amount equivalent to the interrupt amount.

Tool path after interrupt

Interrupt

Tool path after compensation

Program path
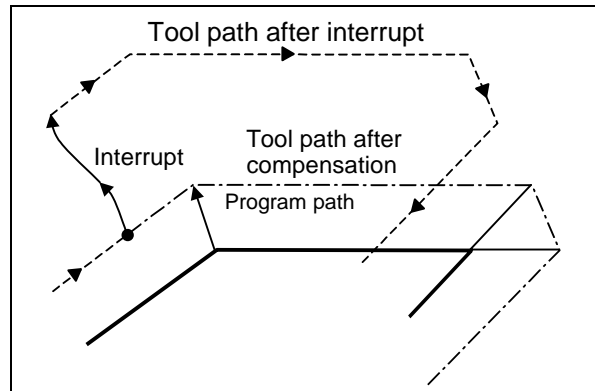
(2)   Interrupt with manual absolute ON.

In the incremental value mode, the same operation results as with manual absolute OFF.
In the absolute value mode, however, the tool returns to its original path at the end point of the block following the interrupted block, as shown in the figure.

Interrupt

Interrupt
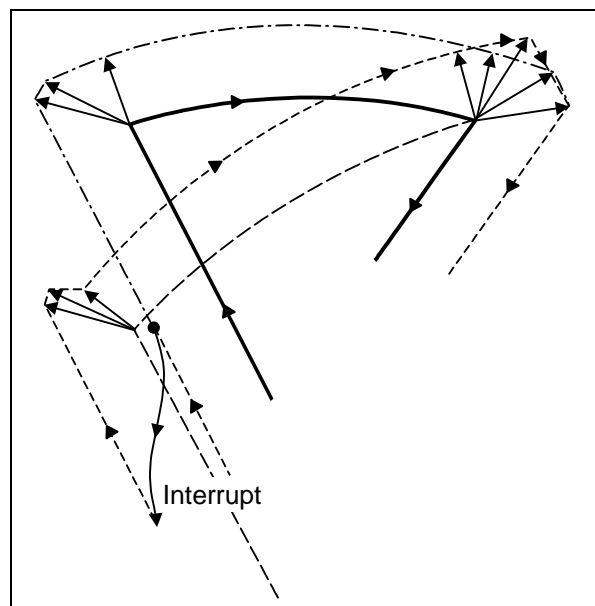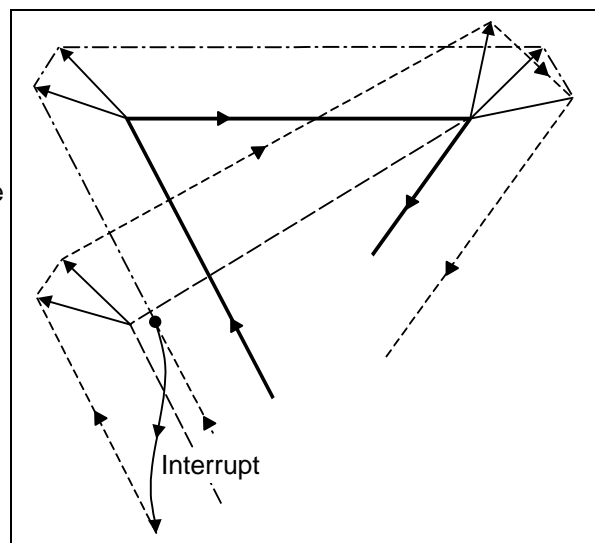
12.3.5 General precautions for tool radius compensation

Precautions

(1) Designating the offset amounts

The offset amounts can be designated with the D code by designating an offset amount No. Once designated, the D code is valid until another D code is commanded. If an H code is designated, the program error "P170" No COMP No will occur. Besides being used to designate the offset amounts for tool radius compensation, the D codes are also used to designate the offset amounts for tool position offset.

(2) Changing the offset amounts

Offset amounts are normally changed when a different tool has been selected in the compensation cancel mode. However, when an amount is changed in the compensation mode, the vectors at the end point of the block are calculated using the offset amount designated in that block.

(3) Offset amount symbols and tool center path

If the offset amount is negative (−), the figure will be the same as if G41 and G42 are interchanged. Thus, the axis that was rotating around the outer side of the workpiece will rotate around the inner side, and vice versa.
An example is shown below. Normally, the offset amount is programmed as positive (+). However, if the tool path center is programmed as shown in (a) and the offset amount is set to be negative (−), the movement will be as shown in (b). On the other hand, if the program is created as shown in (b) and the offset amount is set to be negative (−), the movement will be as shown in (a). Thus, only one program is required to execute machining of both male and female shapes. The tolerance for each shape can be randomly determined by adequately selecting the offset amount.
(Note that a circle will be divided with type A when compensation is started or canceled.)

G41 offset amount (+) or G42 offset amount (−)	G41 offset amount (−) or G42 offset amount (+)
(a)	(b)
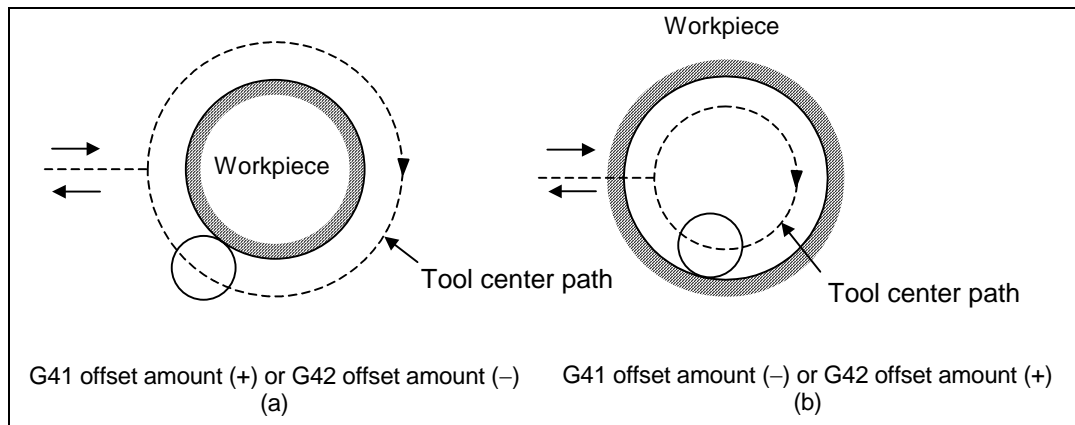
12.3.6 Changing of offset No. during compensation mode

Function and purpose

As a principle, the offset No. must not be changed during the compensation mode. If changed, the movement will be as shown below.
When offset No. (offset amount) is changed:

```
G41  G01 ............................  Dr1  ;
                                                        α = 0, 1, 2, 3
N101          G0α      Xx1    Yy1   ;
N102          G0α      Xx2    Yy2   Dr2  ;  .................................. Offset No. changed
N103                   Xx3    Yy3   ;
```

(1) During linear → linear
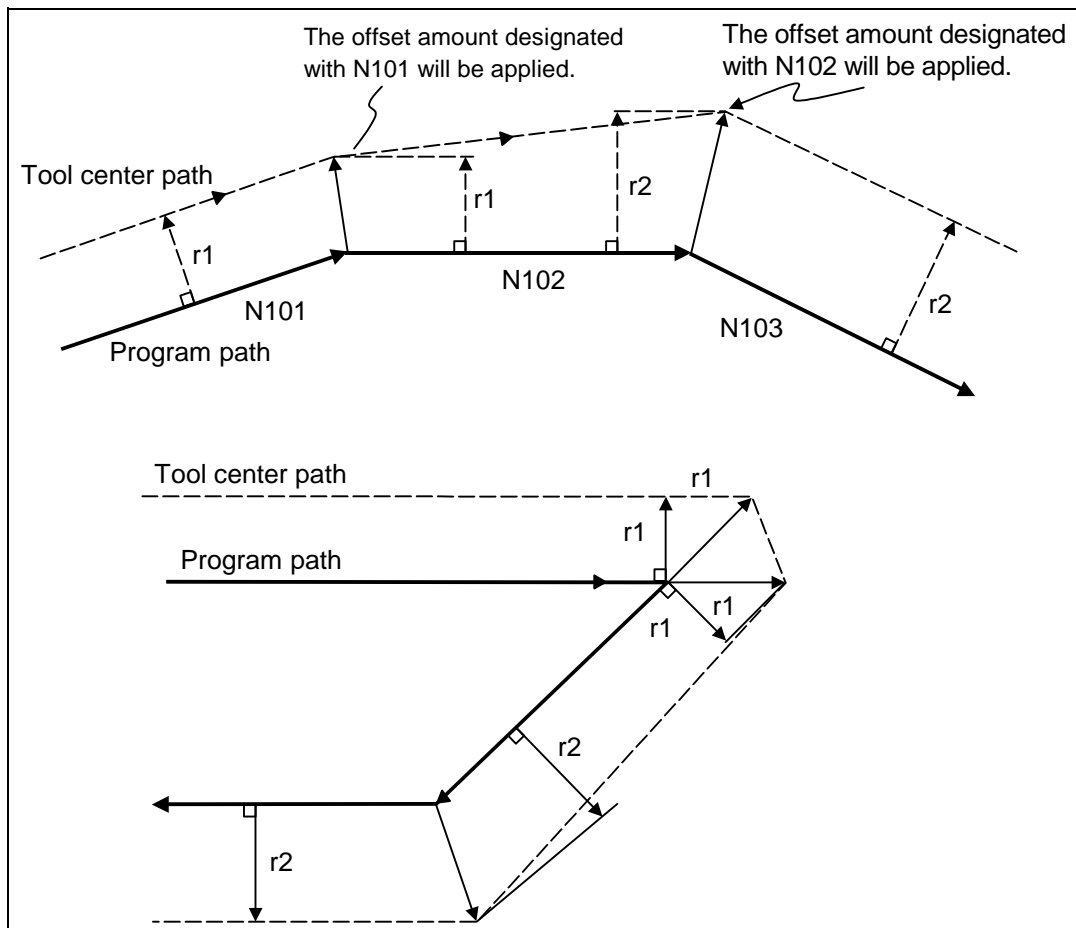
(2) Linear → circular
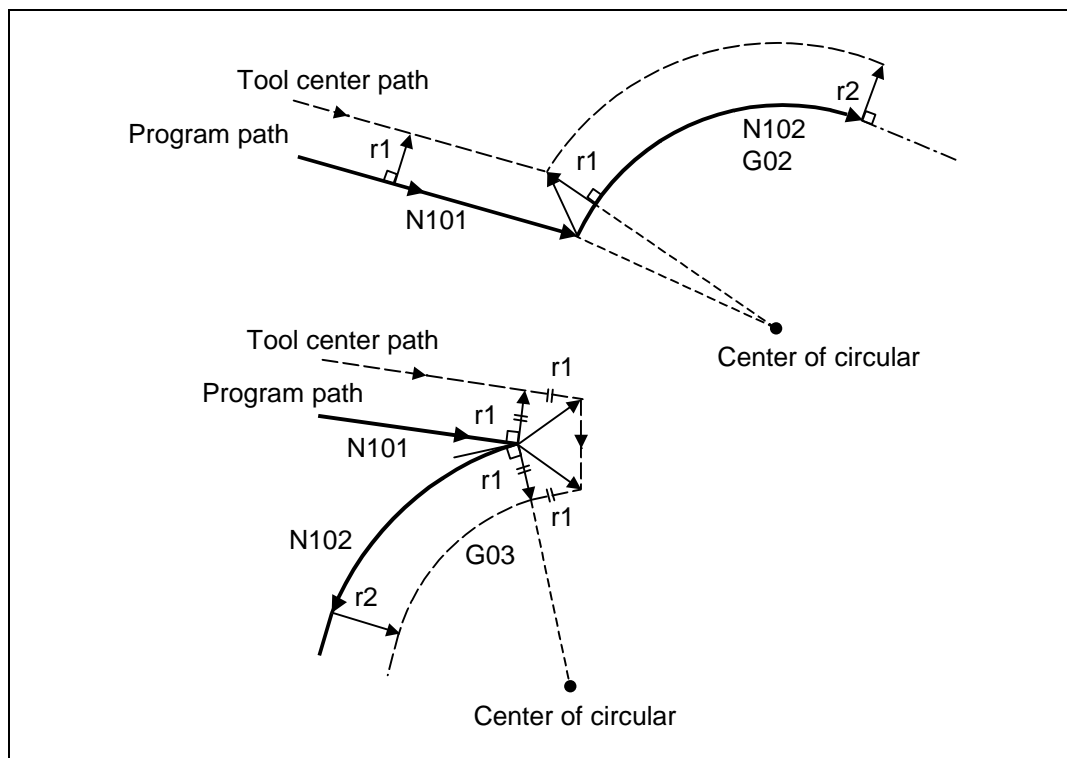
(3) Circular → circular
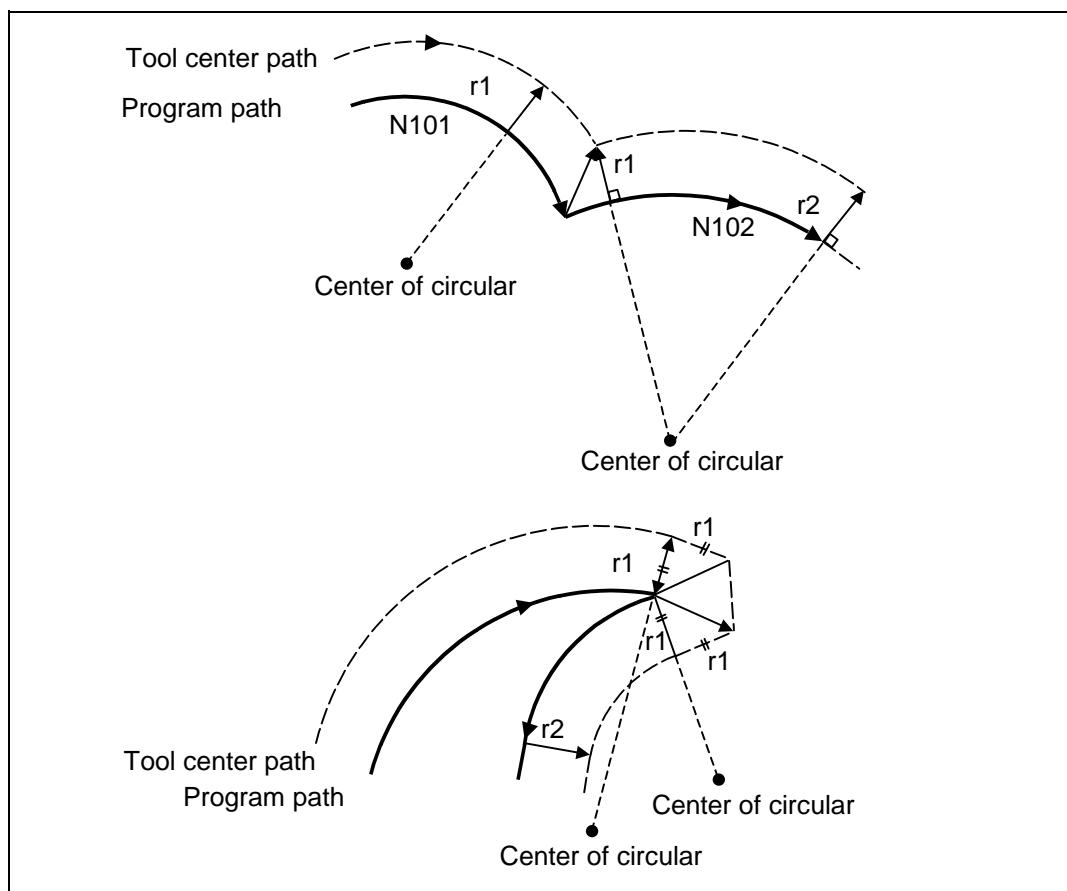
12.3.7 Start of tool radius compensation and Z axis cut in operation
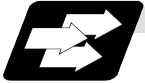
Function and purpose

Often when starting cutting, a method of applying a radius compensation (normally the XY plane) beforehand at a position separated for the workpiece, and then cutting in with the Z axis is often used. When using this method, create the program so that the Z axis movement is divided into the two steps of rapid traverse and cutting feed after nearing the workpiece.

Example of program

When the following type of program is created:

```
N1   G91 G00 G41 X 500. Y 500. D1 ;
N2   S1000   ;
N3   M3   ;
N4   G01   Z-300.   F1   ;
N6            Y 100.   F2   ;
         ·
         ·
         ·
         ·
```


Tool center path
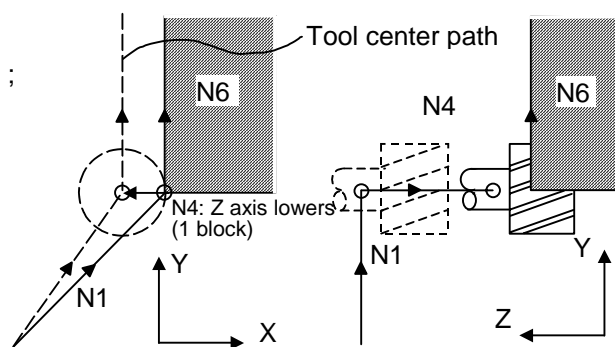
With this program, at the start of the N1 compensation the program will be read to the N6 block. The relation of N1 and N6 can be judged, and correct compensation can be executed as shown above.

If the above program's N4 block is divided into two

```
N1   G91 G00 G41 X 500. Y 500. D1;
N2   S1000 ;
N3   M3 ;
N4        Z-250. ;

N5   G01 Z-50.    F1 ;
N6            Y 100.   F2 ;
```


Cut in
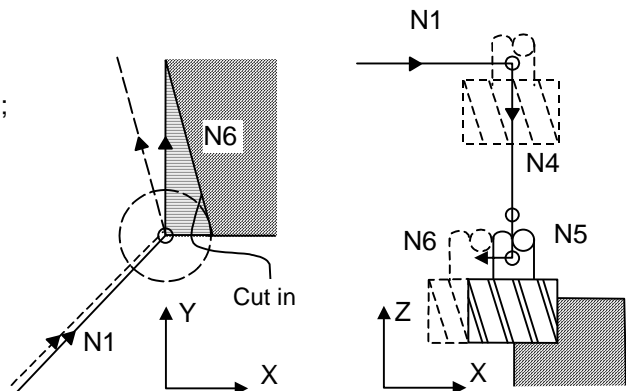
In this case, the four blocks N2 to N5 do not have a command in the XY plane, so when the N1 compensation is started, the program cannot be read to the N6 block.

As a result, the compensation is done based only on the information in the N1 block, and the compensation vector is not created at the start of compensation. Thus, an excessive cut in occurs as shown above.

In this case, consider the calculation of the inner side, and before the Z axis cutting, issue a command in the same direction as the direction that the Z axis advances in after lowering, to prevent excessive cutting.

```
N1   G91 G00 G41 X 500. Y 400. D1 ;
N2        Y100.   S1000 ;
N3   M3 ;
N4        Z-250. ;
N5   G01 Z-50.   F1 ;
N6        Y 100. F2 ;
```
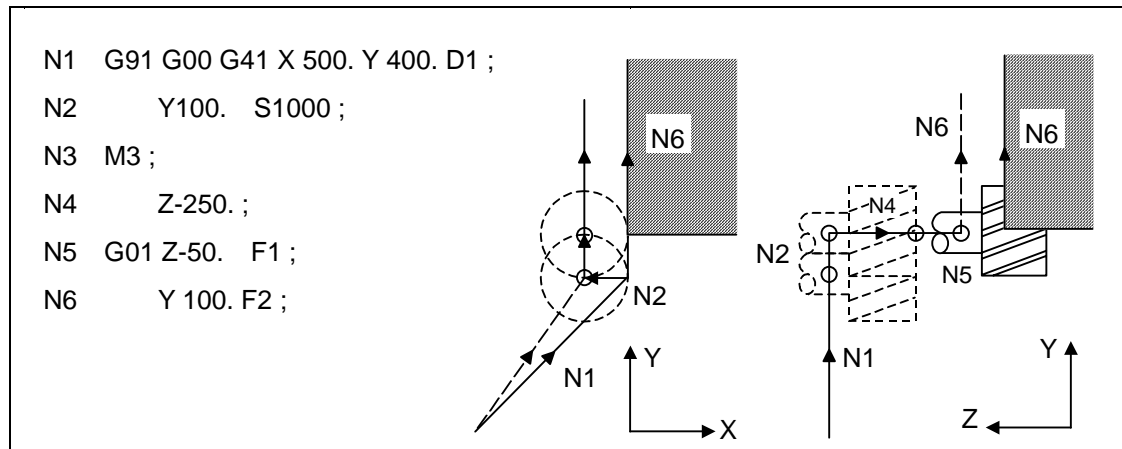
The movement is correctly compensated as the same direction as the N6 advance direction is commanded in N2.

12.3.8 Interference check
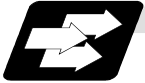
Function and purpose

(1) Outline

A tool, whose radius has been compensated with the tool radius compensation function by the usual 2-block pre-read, may sometimes cut into the workpiece. This is known as interference, and interference check is the function which prevents this from occurring.

There are three types of interference check, as indicated below, and each can be selected for use by parameter.

Function	Parameter	Operation
Interference check alarm function	#8102 : OFF #8103 : OFF	A program error results before the execution of the block in which the cut arises, and operation stops.
Interference check avoidance function	#8102 : ON #8103 : OFF	The tool path is changed so that workpiece is not cut into.
Interference check invalid function	#8103 : ON	Cutting proceeds unchanged even when it occurs. Use this for microscopic segment programs.

(Note) #8102 COLL. ALM OFF (interference check avoidance)
#8103 COLL. CHK OFF (interference check invalid)

Detailed description

(Example)


```
(G41)
N1   G90   G1   X50.   Y-100.;
N2   X70.   Y-100.;
N3   X120.   Y0;
```
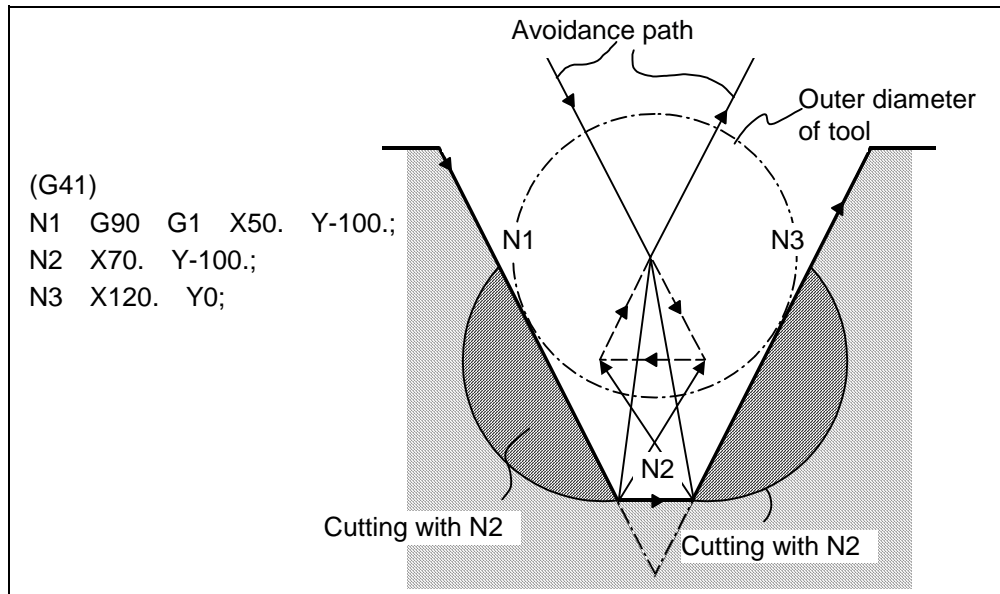
(1) With alarm function

The alarm occurs before N1 is executed and so, using the edit function, N1 can be changed as below and machining can be continued :

N1   G90   G1   X20.   Y−40. ;

(2) With avoidance function

The intersection point of N1 and N3 is calculated and the interference avoidance vectors are created.

(3)   With interference check invalid function

The tool passes while cutting the N1 and N3 line.
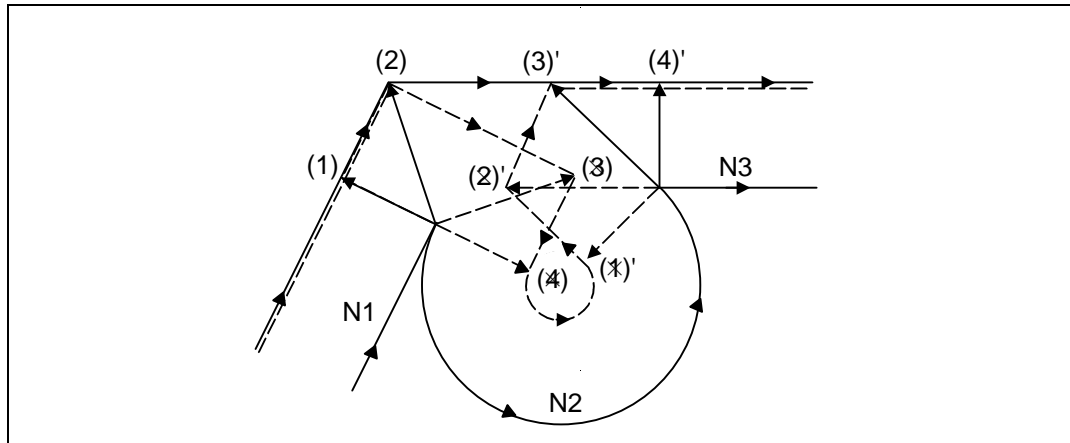
Example of interference check

Vectors (1) (4)' check   →   No interference
                  ↓
Vectors (2) (3)' check   →   No interference
                  ↓
Vectors (3) (2)' check   →   Interference →   Erase vectors (3) (2)'
                                                              ↓
                                               Erase vectors (4) (1)'

With the above process, the vectors (1), (2), (3)' and (4)' will remain as the valid vectors, and the path that connects these vectors will be executed as the interference avoidance path.

Conditions viewed as interference

If there is a movement command in three of the five pre-read blocks, and if the compensation calculation vectors created at the contacts of each movement command intersect, it will be viewed as an interference.
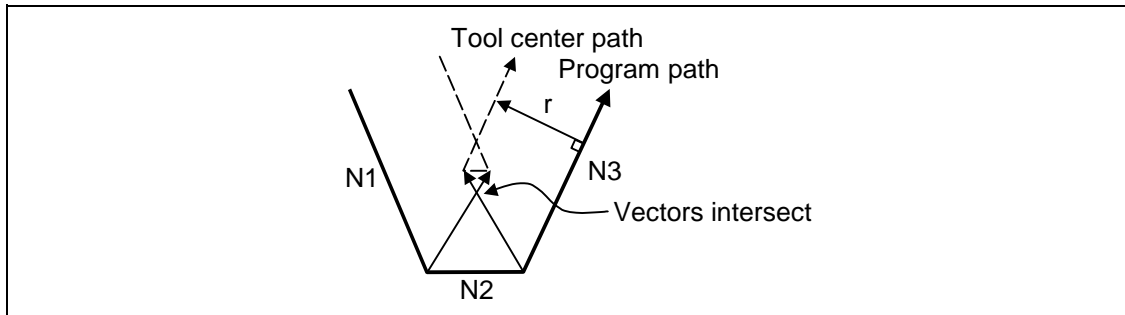
When interference check cannot be executed

(1) When three of the movement command blocks cannot be pre-read
    (When there are three or more blocks in the five pre-read blocks that do not have movement)
(2) When there is an interference following the fourth movement block
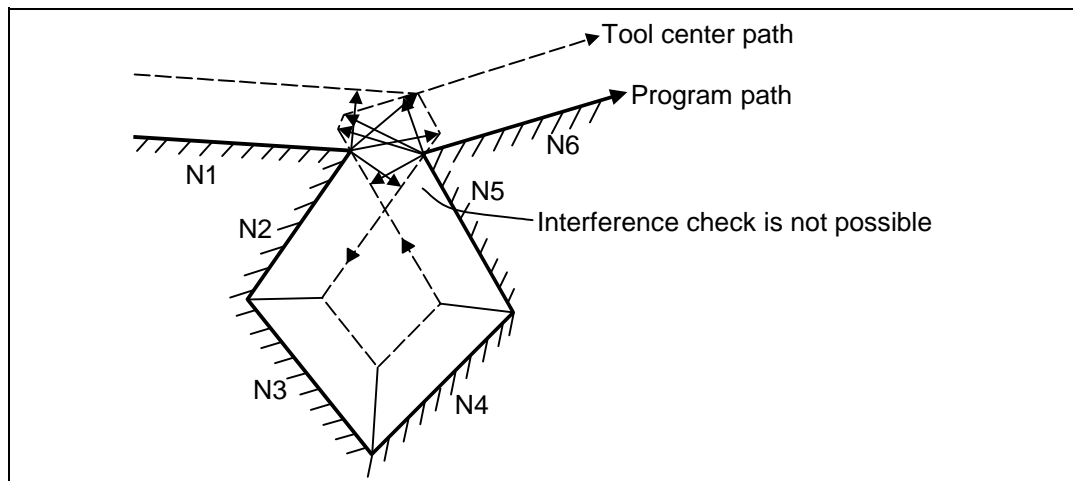
Operation during interference avoidance

The movement will be as shown below when the interference avoidance check is used.

Avoidance vector

N3

N2

Tool center path

N1

Avoidance vector

Program path

If all of the line vectors for the interference avoidance are deleted, create a new avoidance vector as shown on the right to avoid the interference.

N4

r2

r1

N3

Avoidance vector 1

Avoidance vector 2

N2

Tool center path 2

Tool center path 1

r2   r1

N1

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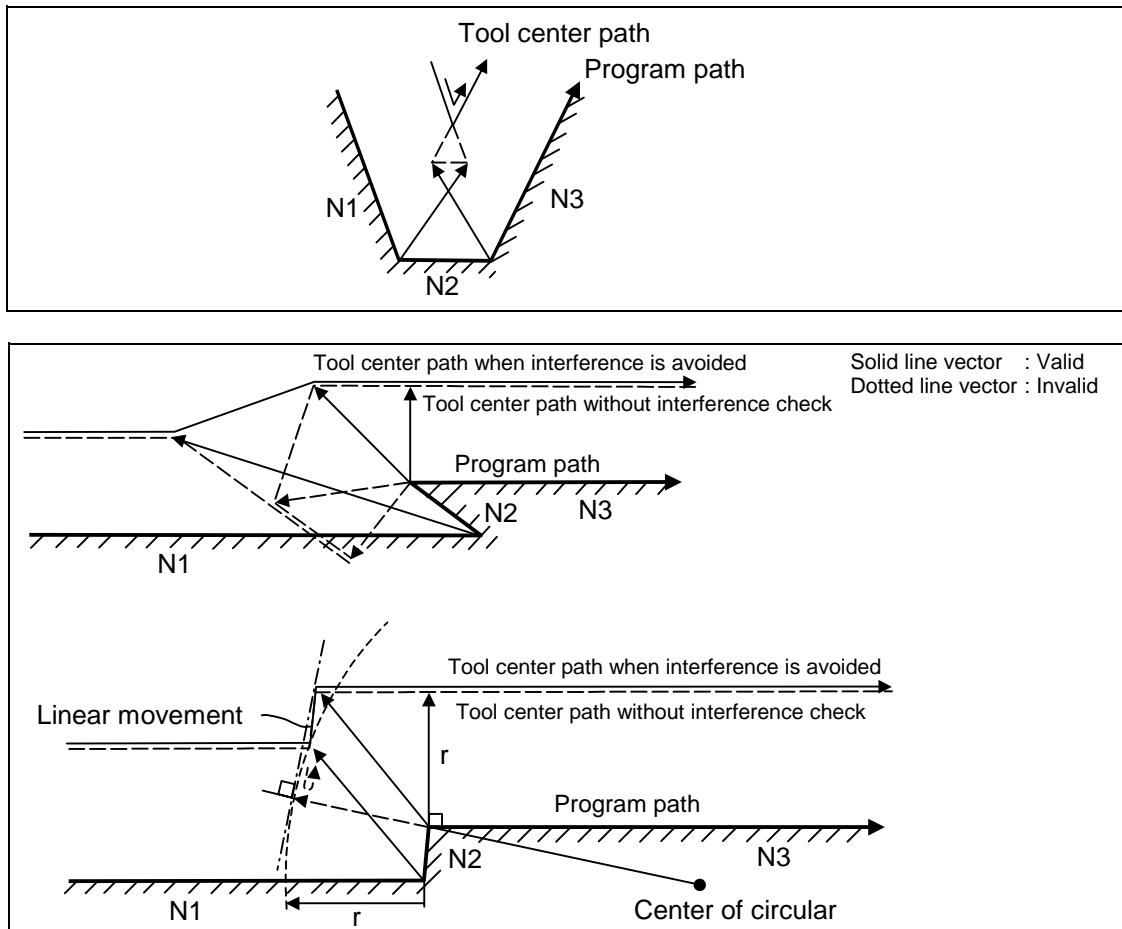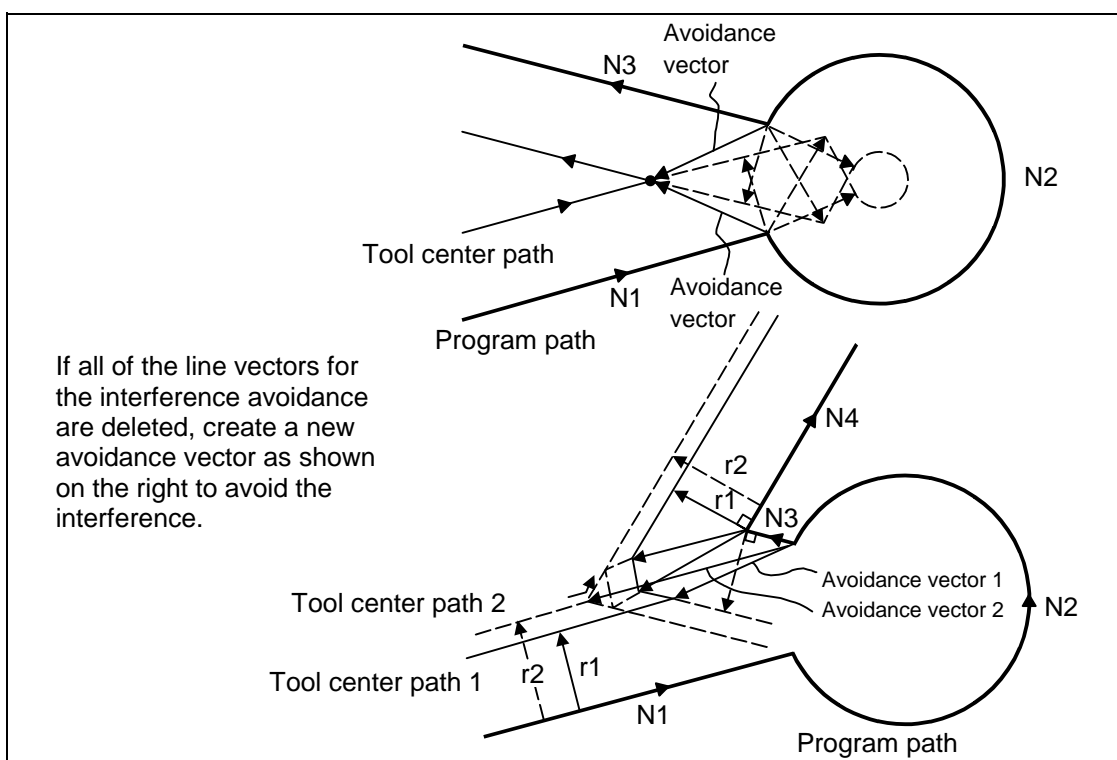
In the case of the figure below, the groove will be left uncut.

Interference avoidance path

Tool center path

Program path
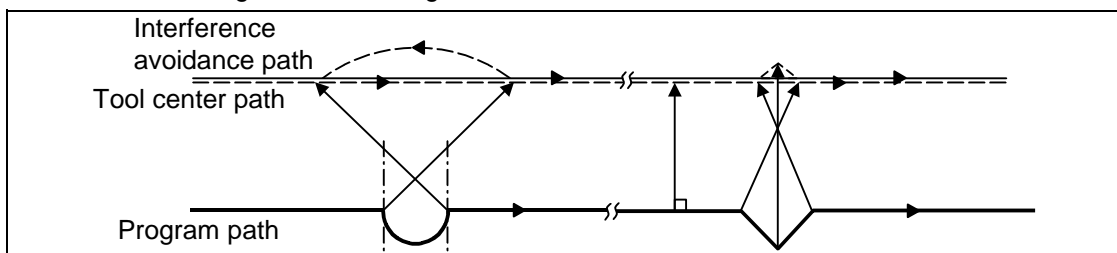
Interference check alarm

The interference check alarm occurs under the following conditions.

(1) When the interference check alarm function has been selected

(a) When all the vectors at the end block of its own block have been deleted.

When, as shown in the figure, vectors 1 through 4 at the end point of the N1 block have all been deleted, program error "P153" results prior to N1 execution.
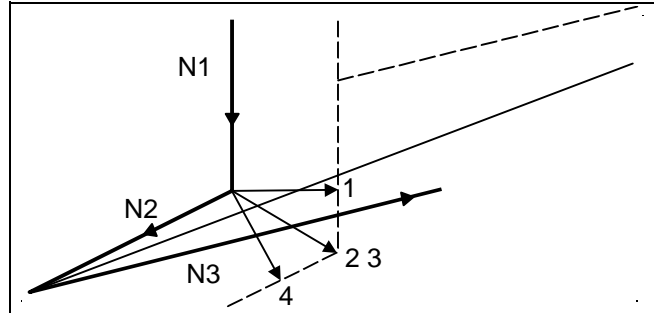
(2) When the interference check avoidance function has been selected

(a) When there are valid vectors at the end point of the following block even when all the vectors at the end point of its own block have been deleted.

(i) When, in the figure, the N2 interference check is conducted, the N2 end point vectors are all deleted but the N3 end point vectors are regarded as valid. Program error "P153" now occurs at the N1 end point.
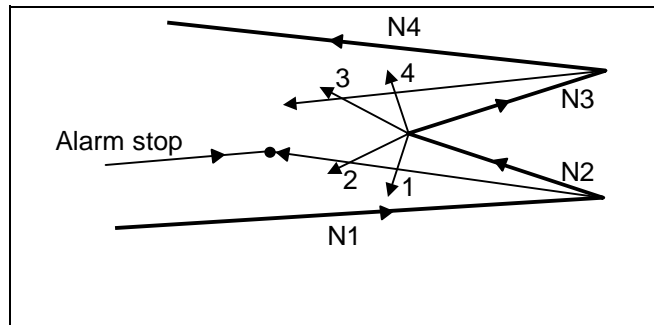
(ii) In a case such as that shown in the figure, the tool will move in the reverse direction at N2. Program error "P153" occurs after N1 execution.

159
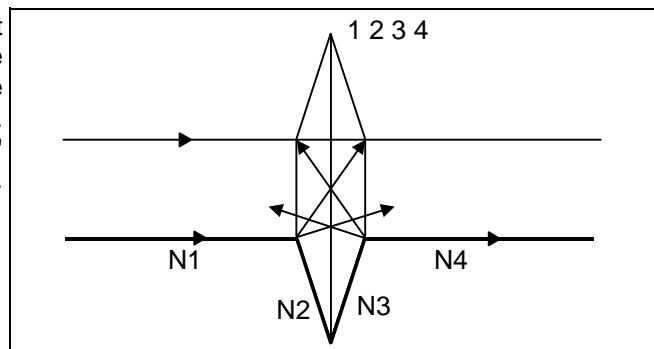
(b) When avoidance vectors cannot be created

(i) Even when, as in the figure, the conditions for creating the avoidance vectors are met, it may still be impossible to create these vectors or the interference vectors may interfere with N3. Program error "P153" will occur at the N1 end point when the vector intersecting angle is more than 90°.
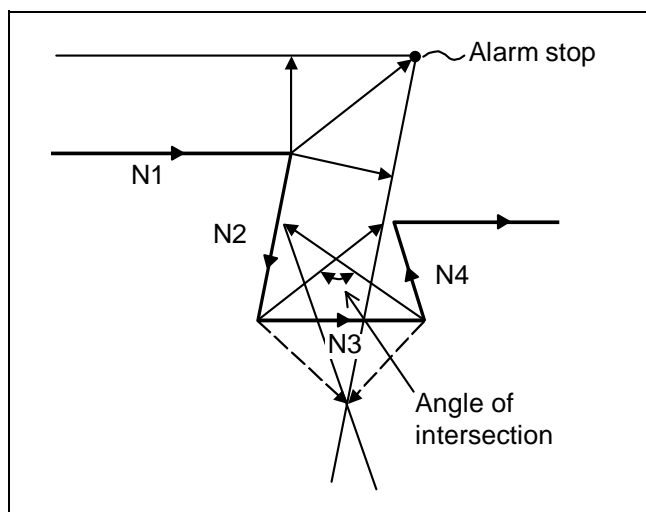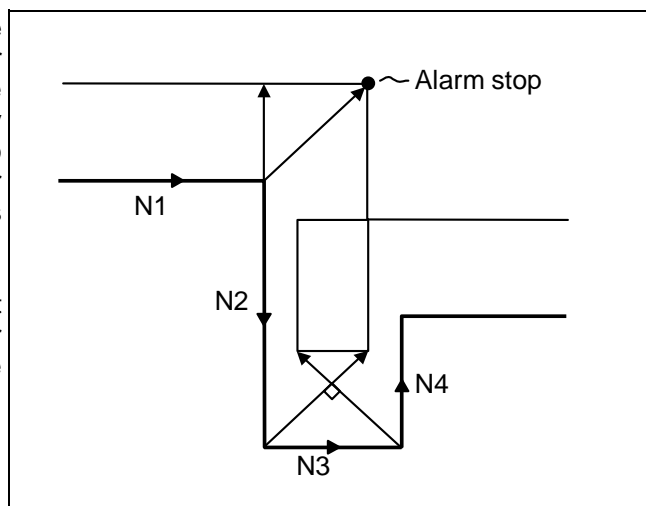
(c) When the program advance direction and the advance direction after compensation are reversed
In the following case, interference is still regarded as occurring even when there is actually no interference.
When grooves which are narrower than the tool radius or which have parallel or widening walls are programmed
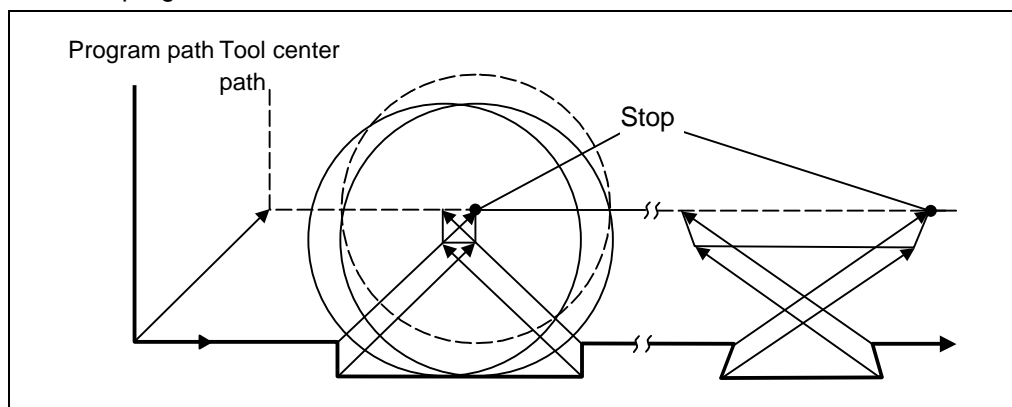
12.4 Programmed offset input; G10, G11

Function and purpose

The tool offset and workpiece offset can be set or changed on the tape using the G10 command. During the absolute value (G90) mode, the commanded offset amount will become the new offset amount, and during the incremental value (G91) mode, the commanded offset amount will be added to the currently set offset amount to create the new offset amount.

Command format

(1) Workpiece offset input

```
G90  G10  L2  P__Xp__Yp__Zp__ ;
G91
P                   : 0  External workpiece
                      1  G54
                      2  G55
                      3  G56
                      4  G57
                      5  G58
                      6  G59
                      If the P command is omitted, the currently selected workpiece offset
                       will be handled as the input.
```

(Note)   The offset amount in the G91 will be an incremental value and will be cumulated each time the program is executed. Command G90 or G91 before the G10 as a cautionary means to prevent this type of error.

(2) Tool offset input

(a) For tool offset memory I

```
G10  L10  P__R__ ;
P                   : Offset No.
R                   : Offset amount
```

(b) For tool offset memory II

G10 L10 P__R__ ;	**Long compensation shape offset**
G10 L11 P__R__ ;	**Long compensation wear compensation**
G10 L12 P__R__ ;	**Diameter shape offset**
G10 L13 P__R__ ;	**Diameter wear compensation**

(3) Offset input cancel

```
G11 ;
```

12. Tool Offset Functions

Detailed description

(1) Program error P171 will occur if this command is input when the specifications are not available.
(2) G10 is an unmodal command   and is valid only in the commanded block.
(3) The G10 command does not contain movement, but must not be used with G commands other than G21, G22, G54 to G59, G90 or G91.
(4) If an illegal L No. or offset No. is commanded, the program errors P172 and P170 will occur respectively.
  If the offset amount exceeds the maximum command value, the program error P35 will occur.
(5) Decimal point inputs can be used for the offset amount.
(6) The offset amounts for the external workpiece coordinate system and the workpiece coordinate system are commanded as distances from the basic machine coordinate system zero point.
(7) The workpiece coordinate system updated by inputting the workpiece coordinate system will follow the previous modal (G54 to G59) or the modal (G54 to G59) in the same block.
(8) L2 can be omitted when the workpiece offset is input.
(9) Do not command G10 in the same block as the fixed cycle and sub-program call command. This will cause malfunctioning and program errors.

Example of program

(1) Input the offset amount from the tape.

….. ; G10L10P10R−12345   ; G10L10P05R98765   ; G10L10P30R2468 ; …

H10=−12345 H05=98765 H30=2468

(2) Updating of offset amount

(Example 1) Assume that H10 = -1000 is already set.

N1	G01 G90 G43 Z − 100000 H10;	(Z = −101000)
N2	G28 Z0;	
N3	G91 G10 L10 P10R − 500 ;	(The mode is the G91 mode, so −500 is added.)
N4	G01 G90 G43 Z − 100000 H10 ;	(Z = −101500)

(Example 2) Assume that H10 = −1000 is already set.

Main program

N1 G00 X100000 ;	a
N2 #1 = −1000 ;	
N3 M98 P1111 L4 ;	b_1, b_2, b_3, b_4

Subprogram O1111

N1 G01 G91 G43 Z0 H10 F100 ;	c_1, c_2, c_3, c_4
G01 X1000 ;	d_1, d_2, d_3, d_4
#1 = #1 − 1000 ;	
G90 G10 L10 P10 R#1 ;	
M99;	

(Note)

Final offset amount will be H10= −5000.
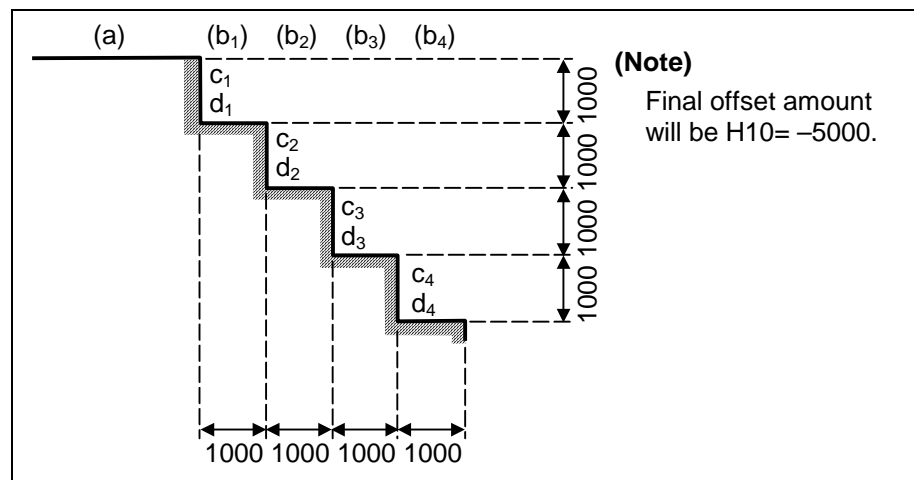
(Example 3) The program for Example 2 can also be written as follows.

Main program

N1 G00 X100000 ;
N2 M98 P1111 L4 ;

Subprogram O1111

N1 G01 G91 G43 Z0 H10 F100 ;
N2 G01 X1000 ;
N3 G10 L10 P10 R−1000 ;
N4 M99 ;

(3) When updating the workpiece coordinate system offset amount

Assume that the previous workpiece coordinate system offset amount is as follows.

X = −10.000   Y = −10.000

N100	G00 G90 G54 X0 Y0 ;
N101	G90 G10 L2 P1 X−15.000 Y−15.000 ;
N102	X0 Y0 ;
M02 ;	
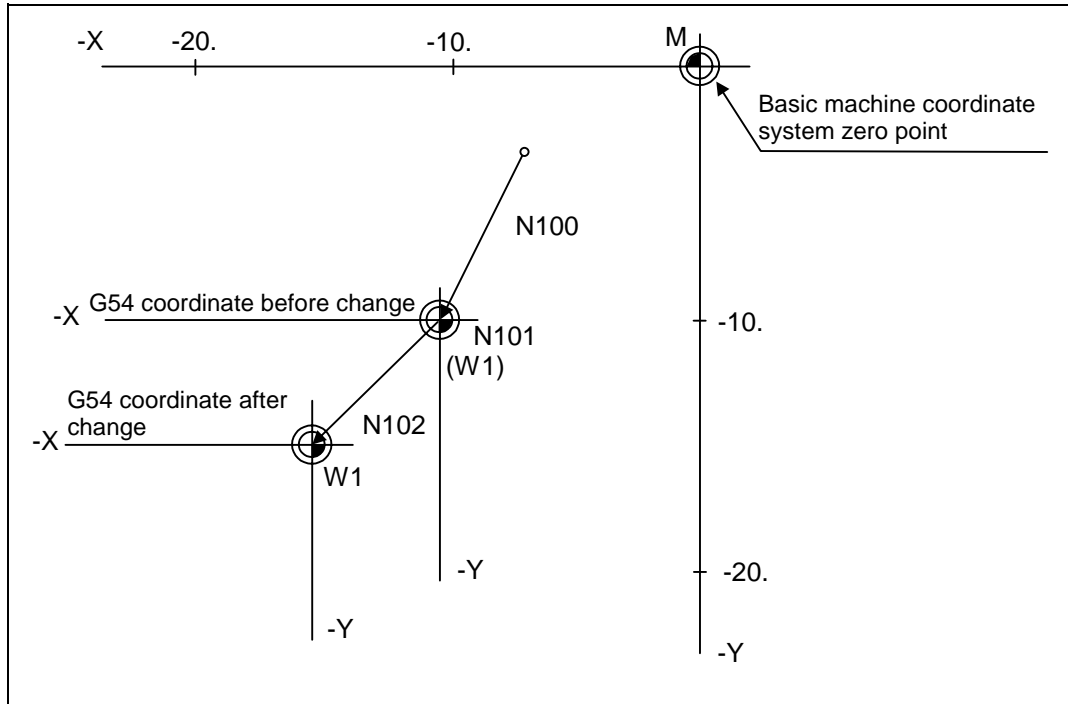
(Note 1) Changes of workpiece position display at N101
At N101, the G54 workpiece position display data will change before and after the workpiece coordinate system is changed with G10.

$$X = 0 \qquad X = +5.000$$
$$\rightarrow$$
$$Y = 0 \qquad Y = +5.000$$

When workpiece coordinate system offset amount is set in G54 to G59

G90 G10 L2 P1 X−10.000 Y−10.000 ;
G90 G10 L2 P2 X−20.000 Y−20.000 ;
G90 G10 L2 P3 X−30.000 Y−30.000 ;
G90 G10 L2 P4 X−40.000 Y−40.000 ;
G90 G10 L2 P5 X−50.000 Y−50.000 ;
G90 G10 L2 P6 X−60.000 Y−60.000 ;

(4) When using one workpiece coordinate system as multiple workpiece coordinate systems

⋮

Main program

| #1 = -50. #2 = 10. ; |
| M98   P200 L5 ; |
| M02 ; |
| % |

Subprogram O200

| N1   G90 G54 G10 L2 P1 X#1 Y#1 ; |
| N2   G00 X0 Y0 ; |
| N3   X-5. F100 ; |
| N4   X0 Y-5. ; |
| N5   Y0 ; |
| N6   #1 = #1 + #2 ; |
| N7   M99 ; |
| % |
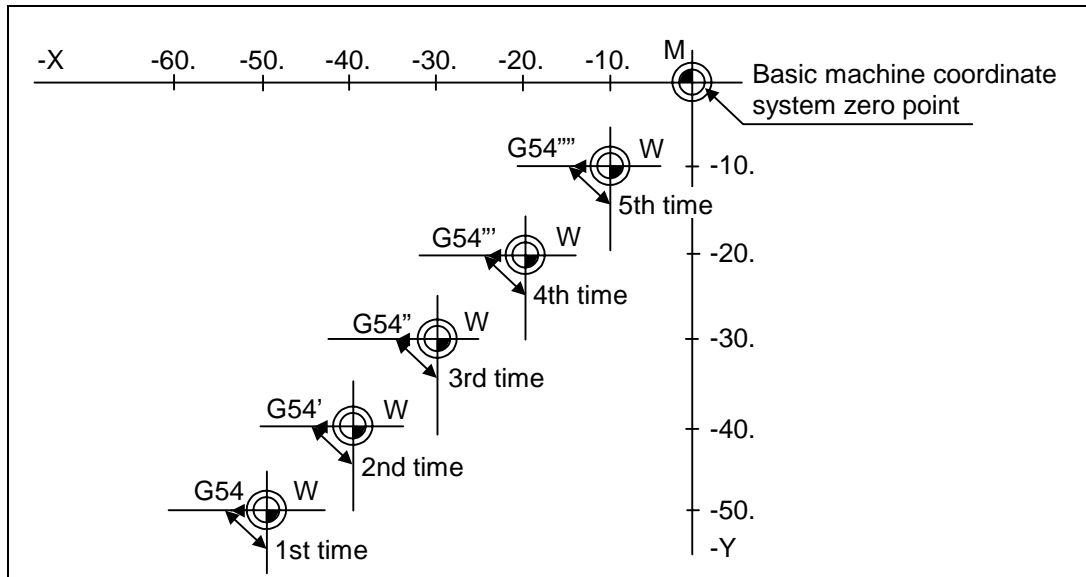
Precautions

(1) Even if this command is displayed on the screen, the offset No. and variable details will not be updated until actually executed.

    N1   G90 G10 L10 P10R−100 ;
    N2   G43 Z-10000 H10 ;
    N3   G0 X-10000 Y-10000 ;
    N4   G90 G10 L10 P10 R-200 ; ... The H10 offset amount is updated when the N4 block is
                                                     executed.

12.5 Tool position offset; G45 to G48 [E68]
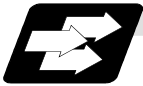
Function and purpose

Using the G45 to G46 commands, the movement distance of the axes specified in the same block can be extended or reduced by a preset offset length. Furthermore, the offset amount can be similarly doubled (✕ 2 expansion) or halved (✕ 2 reduction) with commands G47 and G48.
The number of sets for the offset differ according to machine specification. Refer to Specifications Manual.

     D01 to Dn

     (The numbers given are the total number of sets for tool length offset, tool position offset and cutter compensation.)

G45 command	G46 command	G47 command	G48 command
Expansion by offset amount only	Reduction by offset amount only	✕ 2 expansion by offset amount	✕ 2 reduction by offset amount

(Program command value) (Offset amount) (Movement amount after offset)

Command format

> **G45 Xx Yy Zz Dd ; Expansion of movement amount by offset amount set in offset memory**
>
> **G46 Xx Yy Zz Dd ; Reduction of movement amount by offset amount set in offset memory**
>
> **G47 Xx Yy Zz Dd ; Expansion of movement amount by double the offset amount set in offset memory**
>
> **G48 Xx Yy Zz Dd ; Reduction of movement amount by double the offset amount set in offset memory**
>
> Xx Yy Zz : Movement amount of each axis
> Dd   : Tool compensation No.

Detailed description

Details for incremental values are given below.

Command	Movement amount of equivalent command (assigned offset amount = I)	Example (when X = 1000)	
G45Xx Dd	X (x + I)	I = 10 I = -10	X = 1010 X = 990
G45X-x Dd	X - (x + I)	I = 10 I = -10	X = -1010 X = -990
G46Xx Dd	X (x - I)	I = 10 I = -10	X = 990 X = 1010
G46X-x Dd	X - (x - I)	I = 10 I = -10	X = -990 X = -1010
G47Xx Dd	X (x + 2 * I)	I = 10 I = -10	X = 1020 X = 980
G47X-x Dd	X - (x + 2 * I)	I = 10 I = -10	X = -1020 X = -980
G48Xx Dd	X (x - 2 * I)	I = 10 I = -10	X = 980 X = 1020
G48X-x Dd	X - (x - 2 * I)	I = 10 I = -10	X = -980 X = -1020

Checkpoints

(1) These commands should be used when operation is not in a fixed cycle mode.
(They are ignored even if they are assigned during a fixed cycle.)

(2) As a result of the internal arithmetic processing based on the expansion or reduction, the tool will proceed to move in the opposite direction when the command direction is reversed.
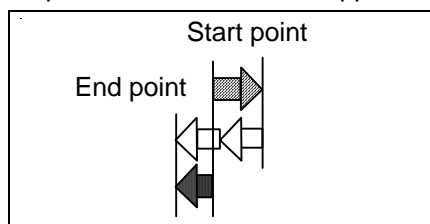
Start point End point	Programmed command G48X20.000D01; Offset Offset amount = +15.000 Tool movement Actual movement = X - 10.000

(3) When a zero movement amount has been specified in the incremental value command (G91) mode, the result is as follows.

Offset number : D01
Offset amount corresponding to D01: 1234

NC command	G45X0 D01;	G45X - 0 D01;	G46X0 D01;	G46X - 0 D01;
Equivalent command	X1234;	X - 1234;	X - 1234;	X1234;

When a zero movement amount has been specified with an absolute value command, the operation is completed immediately and the tool does not move for the offset amount.

(4) In the case of circular interpolation, cutter compensation is possible using the G45 to G48 commands only for one quadrant, two quadrants (semisphere) or three quadrants when the start and end points are on the axis.

The commands are assigned as follows depending on whether the compensation is applied for outside or inside the arc programmed path.

However, in this case, offset must already be provided in the desired direction at the arc start point. (If an offset command is assigned for the arc independently, the arc start point and end point radii will shift by an amount equivalent to the offset amount.)
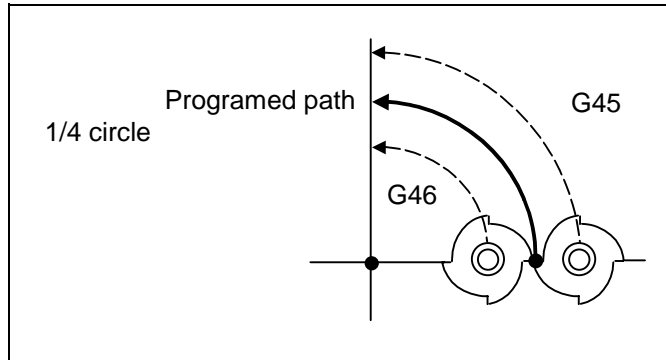
G45 for compensation outside the circle
G46 for compensation inside the circle
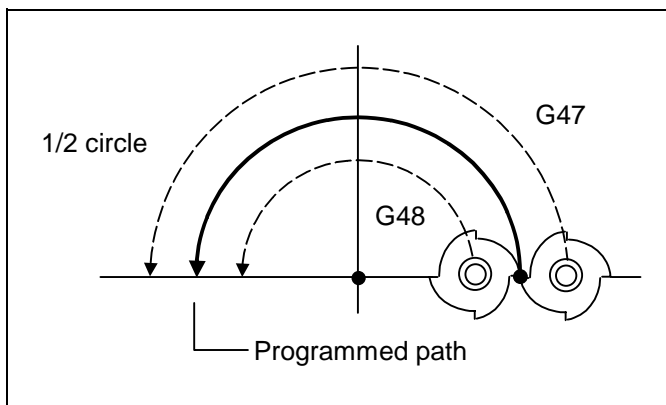
G47 for compensation outside the circle
G48 for compensation inside the circle
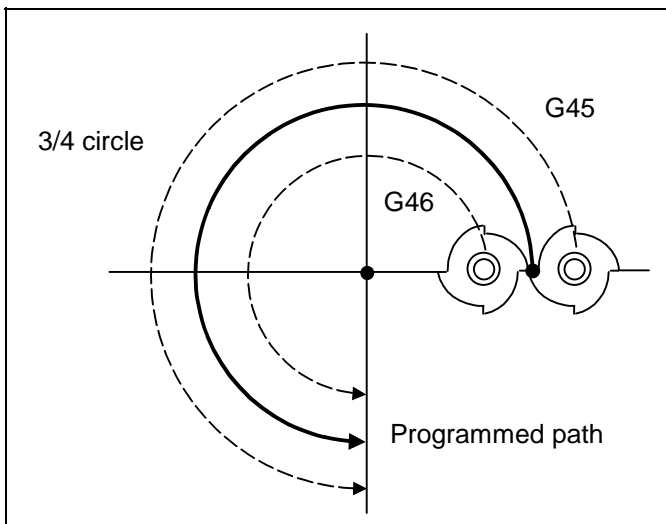
G45 for compensation outside the circle
G46 for compensation inside the circle

168

Example of program

(Example 1)

Tool position offset with 1/4 arc command
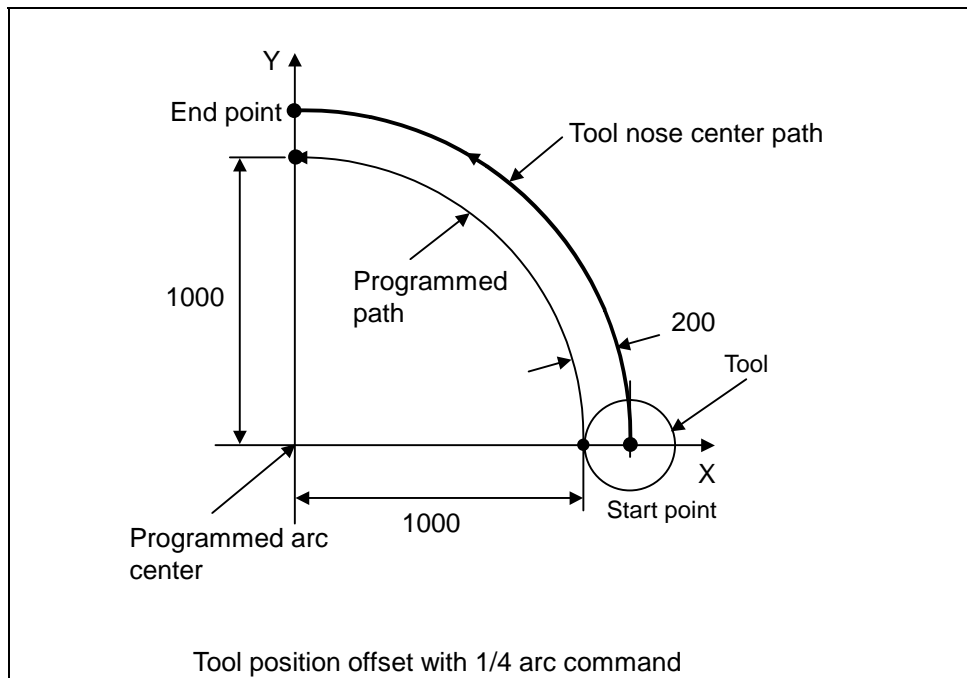
It is assumed that offset has already been provided in the + X direction by D01 = 200.

```
G91  G45  G03  X-1000  Y1000  I-1000  F1000  D01;
```

Even if the offset numbers are not assigned in the same block as the G45 to G48 commands, offset is provided with the tool position offset number previously stored in the memory.

Program error "P170" results when the specified offset number has exceeded the specification range.

These G codes are unmodal and are effective only in the command block.

Even with an absolute value command, the amount of the movement is extended or reduced for each axis with respect to the direction of movement from the end point of the preceding block to the position assigned by the G45 to G48 block.

In other words, even for an absolute value command, offset can be applied to movement amounts (incremental values) in the same block.

When a command for "n" number of simultaneous axes is given, the same offset will be applied to all axes. It is valid even for the additional axes (but it must be within the range of the number of axes which can be controlled simultaneously.)
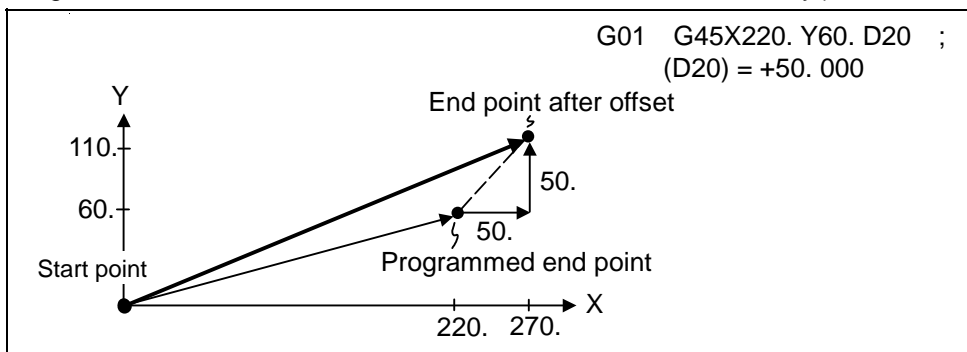
(Note 1) If offset is applied to two axes, over-cutting or under-cutting will result, as shown in the figures below. In cases like this, use the cutter compensation commands (G40 to G42).

G01　G45　Xx_1　Dd_1　;
　　　　　Xx_2　Yy_2　;
　　G45　Yy_3　;

"I":offset amount setting

G01　Xx_1　;
G45　Xx_2　Yy_2　Dd_2　;
　　　Yy_3　;

"*I*":offset amount setting

(Example 2)


```
N1  G46  G00  Xx1  Yy1  Dd1  ;
N2  G45  G01  Yy2  Ff2  ;
N3  G45  G03  Xx3  Yy3  Ii3  ;
N4  G01  Xx4  ;
```
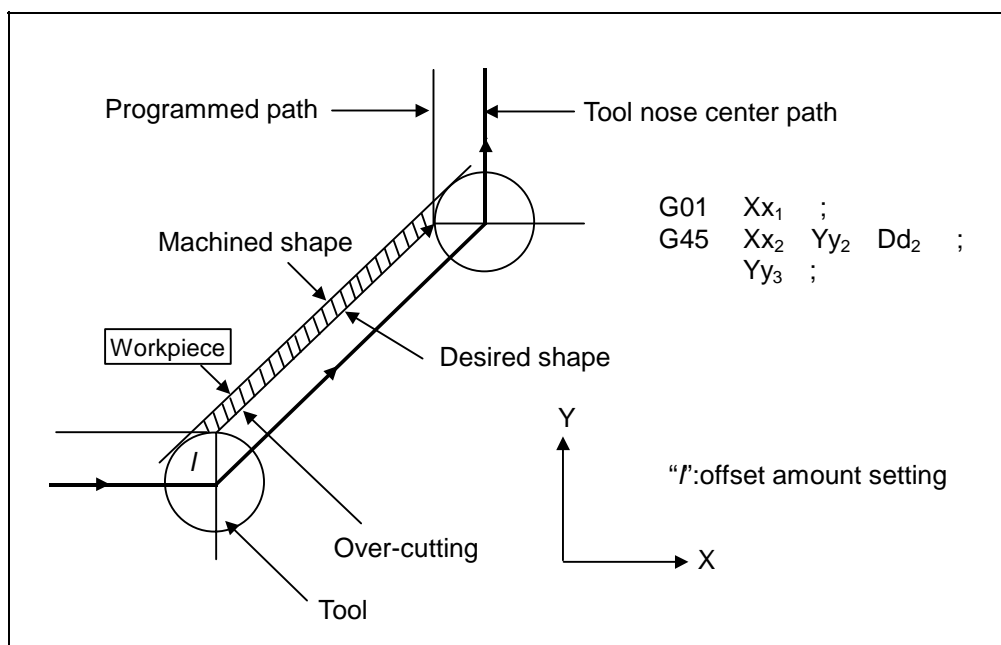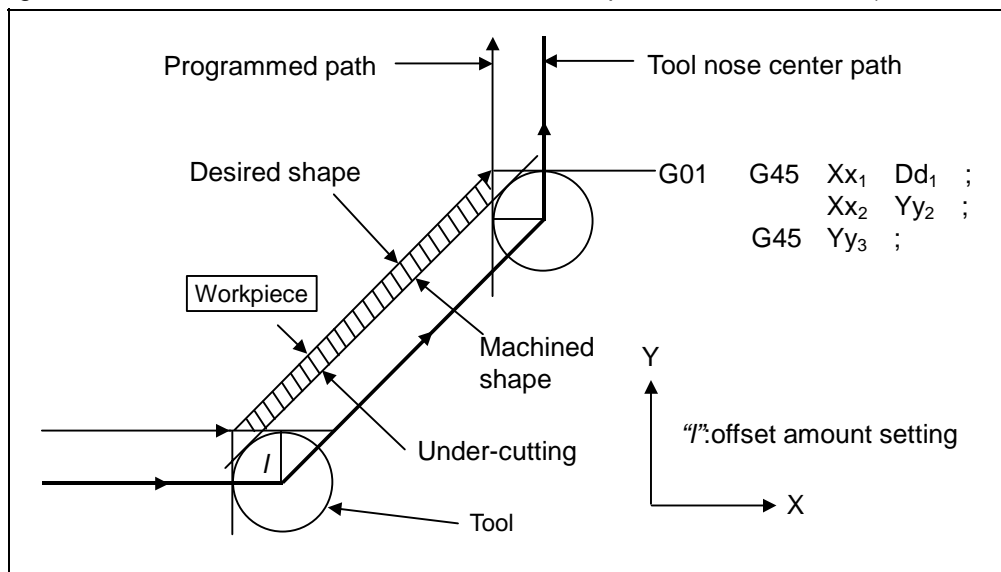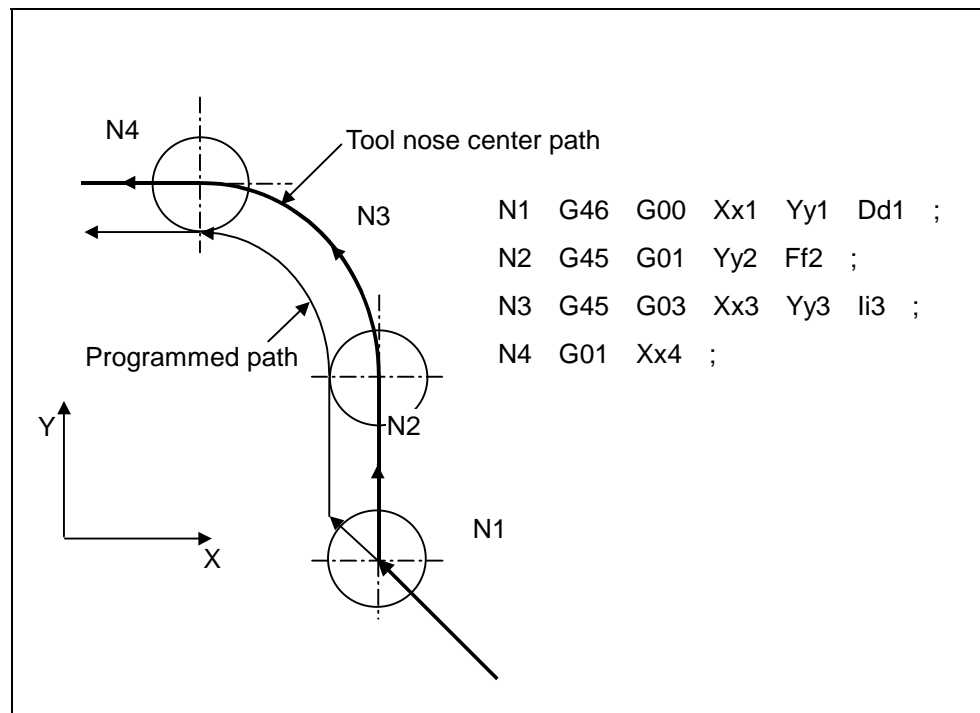
(Example 3)

When the G45 to G48 command is assigned, the offset amount for each pass is the amount assigned by the offset number, and the tool does not move for the difference from the previous offset as it would do with the tool length offset command (G43).
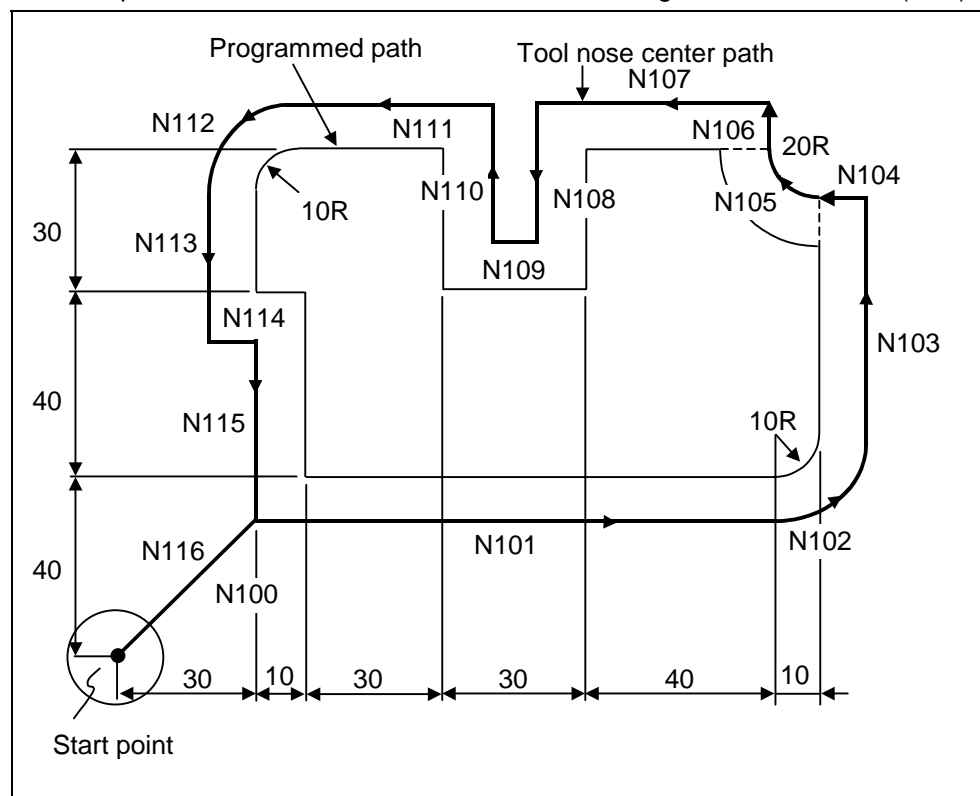
Offset amount D01 = 10.000mm (Offset amount of tool radius)

```
N100   G91   G46   G00    X40.0  Y40.0  D01 ;
N101   G45   G01   X100.0  F200   ;
N102   G45   G03   X10.0   Y10.0  J10.0 ;
N103   G45   G01   Y40.0   ;
N104   G46   X0    ;
N105   G46   G02   X-20.0  Y20.0  J20.0 ;
N106   G45   G01   Y0  ;
N107   G47   X-30.0  ;
N108         Y-30.0  ;
N109   G48   X-30.0  ;
N110         Y 30.0   ;
N111   G45   X-30.0  ;
N112   G45   G03   X-10.0  Y-10.0  J-10.0 ;
N113   G45   G01   Y-20.0  ;
N114         X10.0  ;
N115         Y-40.0 ;
N116   G46   X-40.0 Y-40.0  ;
N117   M02   ;
%
```

12.6 Inputting the tool life management data
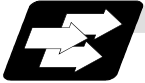
Function and purpose

Using the G10 command, the tool life management data can be registered, changed and added to, and preregistered groups can be deleted.

Command format

(1) Registering data

G10 L3 ;	**Start of life management data registration**
P_L_Q_ ;	**Registration of the group No., the life, the control method**
T_H_D_ ;	**Registration of the tool No., the length compensation No., the radius compensation No.**
T_H_D_ ;	
P_L_Q_ ;	**Registration of next group No., the life, the control method**
T_H_D_ ;	**Registration of the tool No., the length compensation No., the radius compensation No.**
G11 ;	**End of life management data registration**
P	: Group No.
L	: Life
Q	: Control method
T	: Tool No., The spare tools are selected in the order of the tool numbers registered here.
H	: Length compensation No.
D	: Radius compensation No.

First group — applies to the P_L_Q_, T_H_D_, T_H_D_ rows.
Next group — applies to the P_L_Q_, T_H_D_ rows.

(2) Changing and adding groups

G10 L3 P1;	**Start of life management data change or addition**
P_L_Q_ ;	**Change or addition of the group No., the life, the control method**
T_H_D_ ;	**Change or addition of the tool No., the length compensation No., the radius compensation No.**
T_H_D_ ;	
P_L_Q_ ;	**Change or addition of next group No., the life, the control method**
T_H_D_ ;	**Change or addition of the tool No., the length compensation No., the radius compensation No.**
G11 ;	**End of life management data change or addition**
P	: Group No.
L	: Life
Q	: Control method
T	: Tool No.
H	: Length compensation No.
D	: Radius compensation No.

First group — applies to the P_L_Q_, T_H_D_, T_H_D_ rows.
Next group — applies to the P_L_Q_, T_H_D_ rows.

(3) Deleting a group

G10 L3 P2;	**Start of life management data deletion**
P_ ;	**Delete the group No.**
P_ ;	**Delete next group No.**
G11 ;	**End life management data deletion**
P	: Group No.

Example of operation

	Program example	Operation
Data registration	G10 L3 ; P10 L10 Q1 ; T10 H10 D10 ; G11 ; M02 ;	1. After deleting all group data, the registration starts. 2. Group No. 10 is registered. 3. Tool No. 10 is registered in group No. 10. 4. The registration ends. 5. The program ends.
Group change, addition	G10 L3 P1 ; P10 L10 Q1 ; T10 H10 D10 ; G11 ; M02 ;	1. Changing and addition of the group and tool starts. 2. The change and addition operation takes place in the following manner. 　(1) When group No. 10 has not been registered. 　　1) Group No. 10 is additionally registered. 　　2) Tool No. 10 is registered in group No. 10. 　(2) When group No. 10 has been registered, but tool No. 10 has not been registered. 　　1) Tool No. 10 is additionally registered in group No. 10. 　(3) When group No. 10 and tool No. 10 have been both registered. 　　1) The tool No. 10 data is changed. 3. The group and tool change and addition ends. 4. The program ends.
Group deletion	G10 L3 P2 ; P10 ; G11 ; M02 ;	1. The group deletion starts. 2. The group No. 10 data is deleted. 3. The group deletion ends. 4. The program ends.

Command range

Item		Command range
Group No.	(Pn)	1 to 99999999
Life	(Ln)	0 to 9999 times (No. of times control method) 0 to 4000 minutes (time control method)
Control method	(Qn)	1 to 3 　1: Number of mounts control 　2: Time control 　3: Number of cutting times control
Tool No.	(Tn)	1 to 99999999
Length compensation No.	(Hn)	0 to 200
Radius compensation No.	(Dn)	0 to 200

Precautions

(1)   The tool life data is registered, changed, added to or deleted by executing the program in the memory or MDI mode.

(2)   The group No. and tool No. cannot be commanded in duplicate.

(3)   When two or more addresses are commanded in one block, the latter address will be valid.

(4)   If the life data (L_) is omitted, the life data for that group will be "0".

(5)   If the control method (Q_) is omitted, the control method for that group will follow the base specification parameter "#1106 Tcount".
      Note that when carrying out the No. of cutting times control method, command the method from the program.

(6)   If the length compensation No. (H_) is omitted, the length compensation No. for that group will be "0".

(7)   If the radius compensation No. (D_) is omitted, the radius compensation No. for that group will be "0".

(8)   Programming with a sequence No. is not possible between G10 L3 and G11.

(9)   If the usage data count valid signal (Y2CA) is ON, G10 L3 cannot be commanded.

(10) The registered data is held even if the power is turned OFF.

(11) When G10 L3 is commanded, the commanded group and tool will be registered after all of the registered data is erased.

(12) The change and addition conditions in the G10L3P1 command are as follows.

    (a)   Change conditions

        1) Both the commanded group No. and tool No. are registered.
          →Change the commanded tool No. data.

    (b)   Additional conditions

        1) Neither the commanded group No. nor tool No. is registered.
          →Additionally register the commanded group No. and tool No. data.

        2) The commanded group No. is registered, but the commanded tool No. is not registered.
          →Additionally register the commanded tool No. data to the commanded group No.

175

13. Program Support Functions

13.1 Fixed cycles

13.1.1 Standard fixed cycles; G80 to G89, G73, G74, G76

Function and purpose

These standard fixed cycles are used for predetermined sequences of machining operations such as positioning, hole drilling, boring, tapping, etc. which are specified in a block. The various sequences in the table below are provided for the standard fixed cycles.

By editing the standard fixed cycle subprogram, the fixed cycle sequence can be changed by the user. The user can also register and edit an original fixed cycle program. For the standard fixed cycle subprogram, refer to the list of the fixed cycle subprogram in the appendix of the operation manual. The list of fixed cycle functions for this control unit is shown below.

G code	Hole machining start (-Z direction)	Operation at hole bottom		Return operation (+Z direction)	Application
		Dwell	**Spindle**		
G80	-	-	-	-	Cancel
G81	Cutting feed	-	-	Rapid feed	Drill, spot drilling cycle
G82	Cutting feed	Yes	-	Rapid feed	Drill, counter boring cycle
G83	Intermittent feed	-	-	Rapid feed	Deep hole drilling cycle
G84	Cutting feed	Yes	Reverse rotation	Cutting feed	Tapping cycle
G85	Cutting feed	-	-	Cutting feed	Boring cycle
G86	Cutting feed	Yes	Stop	Rapid feed	Boring cycle
G87	Cutting feed	-	Forward rotation	Cutting feed	Back boring cycle
G88	Rapid traverse	Yes	Stop	Rapid feed	Boring cycle
G89	Cutting feed	Yes	-	Cutting feed	Boring cycle
G73	Cutting feed	Yes	-	Rapid feed	Stepping cycle
G74	Intermittent feed	Yes	Forward rotation	Cutting feed	Reverse tapping cycle
G76	Cutting feed	-	Oriented spindle stop	Rapid feed	Fine boring cycle

A fixed cycle mode is canceled when the G80 or any G command in (G00, G01, G02, G03) is issued. The various data will also be cleared simultaneously to zero.

13. Program Support Functions

Command format

G8Δ (G7Δ)   X__ Y__ Z__ R__ Q__ P__ F__ L__ S__ , S __ ,I__ ,J__ ;

G8Δ (G7Δ)   X__ Y__ Z__ R__ Q__ P__ F__ L__ S__ , R __ ,I__ ,J__ ;

G8Δ (G7Δ)	: Hole machining mode
X__ Y__	: Hole positioning data
Z__ R__ Q__ P__ F__	: Hole machining data
L__	: Number of repetitions (When "0" is set, the program is not executed.)
S__	: Spindle rotation speed
, S__	: Spindle rotation speed during return
, R_	: Synchronization changeover
, I__	: Positioning axis in-position width
,J__	: Drilling axis in-position width

As shown above, the format is divided into the hole machining mode, hole positioning data, hole machining data, No. of repetitions, spindle rotation speed, spindle rotation speed at synchronization changeover (or during return), positioning axis in-position width and drilling axis in-position width.

(Note) The programmable in-position check by ",I" and ",J" address is a specification of the E68 system.

Detailed description

(1) Outline of data and corresponding addresses
   (a) Hole machining mode: Fixed cycle modes such as drilling, counter boring, tapping and boring.
   (b) Hole position data: Data used to position the X and Y axes. (Unmodal)
   (c) Hole machining data: Actual machining data used for machining. (Modal)
   (d) Number of repetitions: Number of times to carry out hole machining. (Unmodal)
   (e) Synchronization changeover: Synchronous/asynchronous tapping selection command is issued during G84/G74 tapping. (Modal)

(2) If M00 or M01 is commanded in the same block as the fixed cycle or during the fixed cycle mode, the fixed cycle will be ignored. Instead, M00 and M01 will be output after positioning. The fixed cycle will be executed if X, Y, Z or R is commanded.

(3) There are 7 actual operations which are each described in turn below.
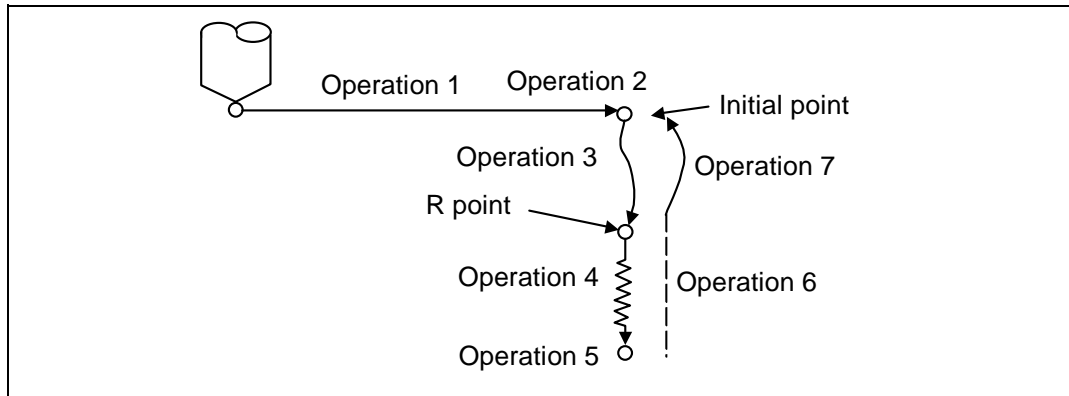
Operation 1 : This indicates the X and Y axes positioning, and executes positioning with G00.

Operation 2 : This is an operation done after positioning is completed (at the initial hole), and when G87 is commanded, the M10 command is output from the control unit to the machine. When this M command is executed and the finish signal (FIN) is received by the control unit, the next operation will start. If the single block stop switch is ON, the block will stop after positioning.

Operation 3 : The tool is positioned to the R point by rapid traverse.

Operation 4 : Hole machining is conducted by cutting feed.

Operation 5 : This operation takes place at the hole bottom position and it differs according to the fixed cycle mode. Possible actions include spindle stop (M05) spindle reverse rotation (M04), spindle forward rotation (M03), dwell and tool shift.

Operation 6 : The tool is retracted to the R point.

Operation 7 : The tool is returned to the initial pint at the rapid traverse rate.

Whether the fixed cycle is to be completed at operation 6 or 7 can be selected by the following G commands.

G98 .... Initial level return
G99 .... R point level return

These are modal commands, and for example, if G98 is commanded once, the G98 mode will be entered until G99 is designated. The initial state when the NC is ready is the G98 mode.
The hole machining data will be ignored if X, Y, Z or R is not commanded. This function is mainly used with the special fixed cycled.

(4) Fixed cycle addresses and meanings

Address	Significance
G	Selection of hole machining cycle sequence (G80 to G89, G73, G74, G76)
X	Designation of hole drilling position (absolute value or incremental value)
Y	Designation of hole drilling position (absolute value or incremental value)
Z	Designation of hole bottom position (absolute value or incremental value)
P	Designation of dwell time at hole bottom position (decimal points will be ignored)
Q	Designation of cut amount for each cutting pass with G73 or G83, or designation of the shift amount at G76 or G87 (incremental value)
R	Designation of R point position (absolute value or incremental value)
F	Designation of feed rate for cutting feed (asynchronous tapping) or number of Z axis pitch (synchronous tap)
L	Designation of number of repetitions. 0 to 9999 (When "0" is set, the program is not executed.)
I,J,K	Designation of shift amount at G76 or G87 (incremental value) (The shift amount is designated with the Q address depending on the parameter setting.)
S	Spindle rotation speed command
,S	Spindle rotation speed designation for synchronous tap return
,R	Synchronous/asynchronous tap cycle selection (If ",R" setting is omitted, that depends on the parameter setting.)
M	Designation of miscellaneous command **(Note)** On the asynchronous tapping with normal tapping, spindle forward rotation will be applied for Mm1 basically. Note that the spindle reverse rotation should be commanded with M(m1 + 1), that is, the M code of spindle forward rotation command +1, or that could lead an incorrect operation.
,I	Designation of positioning axis in-position width
,J	Designation of drilling axis in-position width

(5) Difference between absolute value command and incremental value command

179
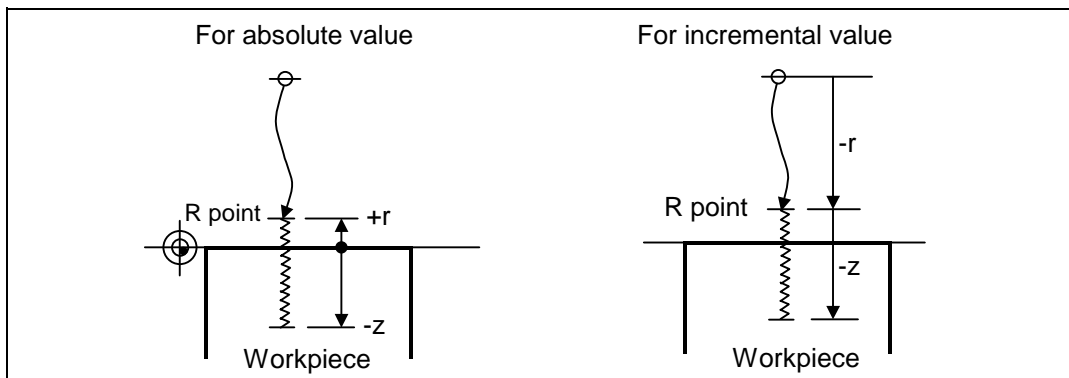
(6) Feed rate for tapping cycle and tapping return
The feed rates for the tapping cycle and tapping return are as shown below.

(a) Selection of synchronous tapping cycle/asynchronous tapping cycle

Program G84×××, Rxx	Control parameter Synchronous tapping	Synchronous/ asynchronous
, R00	-	Asynchronous
, Rxx	OFF	
No designation	ON	Synchronous
, R01	-	

- is irrelevant to the setting

(b) Selection of asynchronous tapping cycle feed rate

G94/G95	Control parameter F1-digit value	F command value	Feed designation
G94	OFF	F designation	Per-minute feed
	ON	Other than F0 to F8	
		F0 to F8 (no decimal point)	F1-digit feed
G95	-	F designation	Per-revolution feed

- is irrelevant to the setting

(c) Spindle rotation speed during return of synchronous tapping cycle

Address	Meaning of address	Command range (unit)	Remarks
,S	Spindle rotation speed during return	0 to 99999 (r/min)	The data is held as modal information. If the value is smaller than the speed rotation speed, the speed rotation speed value will be valid even during return. If the spindle rotation speed is not 0 during return, the tap return override value will be invalid.

(7) M code for forward/reverse rotation command.
The M code set with the parameter "#3028 sprcmn" is output as the M code for spindle forward/reverse rotation that is output at "hole bottom" or at "R point" during asynchronous tapping cycle.
Note that the M code for forward rotation is output as "M3" and that for reverse rotation is as "M4" if the parameter #3208) is set to "0".

Positioning plane and hole drilling axis

The fixed cycle has basic control elements for the positioning plane and hole drilling axis. The positioning plane is determined by the G17, G18 and G19 plane selection command, and the hole drilling axis is the axis perpendicular (X, Y, Z or parallel axis) to the above plane.

Plane selection	Positioning plane	Hole drilling axis
G17 (X - Y)	Xp - Yp	Zp
G18 (Z - X)	Zp - Xp	Yp
G19 (Y - Z)	Yp - Zp	Xp

Xp, Yp and Zp indicate the basic axes X, Y and Z or an axis parallel to the basic axis.
A random axis other than the hole drilling axis can be commanded for positioning.
The hole drilling axis is determined by the axis address of the hole drilling axis commanded in the same block as G81 to G89, G73, G74 or G76. The basic axis will be the hole drilling axis if there is no designation.

(Example 1) When G17 (XY plane) is selected, and the axis parallel to the Z axis is set as the W axis.
      G81 ... W__;   The W axis is used as the hole drilling axis.
      G81 ... Z __;   The Z axis is used as the hole drilling axis.
      G81 ... ;   (No Z or W) The Z axis is used as the hole drilling axis.

(Note 1) The hole drilling axis can be fixed to the Z axis with parameter #1080 Dril_Z.
(Note 2) Change over the hole drilling axis in the fixed cycle canceled state.

In the following explanations on the movement in each fixed cycle mode, the XY plane is used for the positioning plane and the Z axis for the hole drilling axis.
Note that all command values will be incremental values, the positioning plane will be the XY plane and the hole drilling axis will be the Z axis.

Programmable in-position width command in fixed cycle   **[E68]**

This command commands the in-position width for the fixed cycle from the machining program.
The commanded in-position width is valid only in the G81 (drill, spot drill), G82 (drill, counter boring), G83 (deep drill cycle), G84 (tap cycle), G85 (boring), G89 (boring), G73 (step cycle) and G74 (reverse tap cycle) fixed cycles. The ", I" address is commanded in respect to the positioning axis, and the ",J" address is commanded in respect to the drilling axis.

Address	Meaning of address	Command range (unit)	Remarks
,I	In-position width (position error amount)	0.001 to 999.999 (mm)	When a value exceeding the command range is issued, a program error (P35) will occur.
,J	In-position width for drilling axis in fixed cycle (position error amount)		

13. Program Support Functions

In-position check in fixed cycle [E68]

When the number of repetitions L is designated as two or more times in the fixed cycle, the commanded in-position width will be valid in the repetition block (operation 5 to operation 8).

```
:
G81  X-50.  Z-50.  R-50.  L2  F2000, I0.2, J0.3 ;
:
```


Operation pattern	,I	,J
Operation 1	Valid	-
Operation 2	-	Invalid
Operation 3	-	Invalid
Operation 4	-	Valid
Operation 5	Valid	-
Operation 6	-	Invalid
Operation 7	-	Invalid
Operation 8	-	Valid
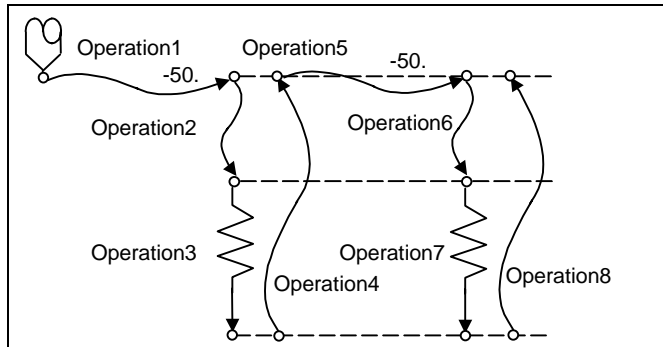
**Fig. 1 Operation when number of repetitions
        L is designated**

In the following machining program, the commanded in-position width is valid for the Fig. 2 block. In the (B) block, the in-position width (, I) commanded for positioning in the previous block (A) is invalid (operation 5). However, when returning from the hole bottom, the in-position width (, J) commanded in the previous block (A) is valid (operation 8).
To validate the in-position width for positioning, command again as shown in block (C) (Operation 9).

```
:
G81  X-50.  Z-50.  R-50.  F2000, I0.2,  J0.3 ; .......(A)
X-10. ; ...................................................................(B)
X-10., I0.2 ; ...........................................................(C)
:
```


Operation pattern	,I	,J
Operation 1	Valid	-
Operation 2	-	Invalid
Operation 3	-	Invalid
Operation 4	-	Valid
Operation 5	Invalid	-
Operation 6	-	Invalid
Operation 7	-	Invalid
Operation 8	-	Valid
Operation 9	Valid	-
Operation 10	-	Invalid
Operation 11	-	Invalid
Operation 12	-	Valid
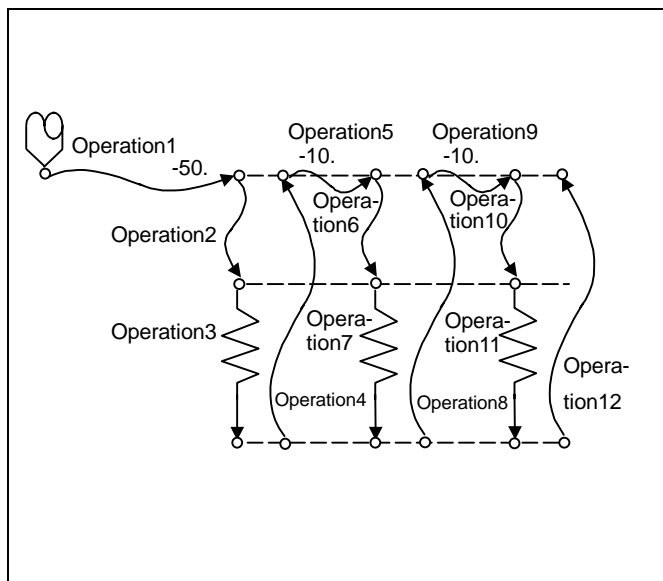
Fig. 2 Operation in fixed cycle modal

Setting values of synchronous tap in-position check parameters and tap axis movement

(1)   Synchronous tap in-position check settings

#1223 aux07				"P" designation of G84/G74 command	In-position check during synchronous tap
bit3	bit4	bit5	bit2		
Synchronous tapping in-position check	At hole bottom	At R point	I point → R point		
0	-	-	-	-	Carry out in-position check at I point → R point/R point/hole bottom.
1	-	-	-	No "P" designation **Example**: G84 F1. Z-5. S1000 R-5.	Hole bottom: Do not carry out in-position check. R point: Do not carry out in-position check. I point → R point: Do not carry out in-position check.
1	1	1	1	"P" designation **Example**: G84 F1. Z-5. S1000 **PO** R-5.	Hole bottom: Carry out in-position check by tap in-position width. R point: Carry out in-position check. I point → R point: Carry out in-position check.
1	1	0	1	"P" designation **Example**: G84 F1. Z-5. S1000 **PO** R-5.	Hole bottom: Carry out in-position check by tap in-position width. R point: Do not carry out in-position check. I point → R point: Carry out in-position check.
1	0	1	1	"P" designation **Example**: G84 F1. Z-5. S1000 **PO** R-5.	Hole bottom: Do not carry out in-position check. R point: Carry out in-position check. I point → R point: Carry out in-position check.
1	0	0	1	"P" designation **Example**: G84 F1. Z-5. S1000 **PO** R-5.	Hole bottom: Do not carry out in-position check. R point: Do not carry out in-position check. I point → R point: Carry out in-position check.
1	1	1	0	"P" designation **Example**: G84 F1. Z-5. S1000 **PO** R-5.	Hole bottom: Carry out in-position check by tap in-position width. R point: Carry out in-position check. I point → R point: Do not carry out in-position check.
1	1	0	0	"P" designation **Example**: G84 F1. Z-5. S1000 **PO** R-5.	Hole bottom: Carry out in-position check by tap in-position width. R point: Do not carry out in-position check. I point → R point: Do not carry out in-position check.
1	0	1	0	"P" designation **Example**: G84 F1. Z-5. S1000 **PO** R-5.	Hole bottom: Do not carry out in-position check. R point: Carry out in-position check. I point → R point: Do not carry out in-position check.
1	0	0	0	"P" designation **Example**: G84 F1. Z-5. S1000 **PO** R-5.	Hole bottom: Do not carry out in-position check. R point: Do not carry out in-position check. I point → R point: Do not carry out in-position check.

(Note 1)   The I point refers to the initial point.

(2) Relation between the in-position width and tap axis movement for a synchronous tap in-position check

R point : In-position check by the G1inps value.
I point : In-position check by the G0inps value.
Hole bottom : In-position check by the Tapinps value.
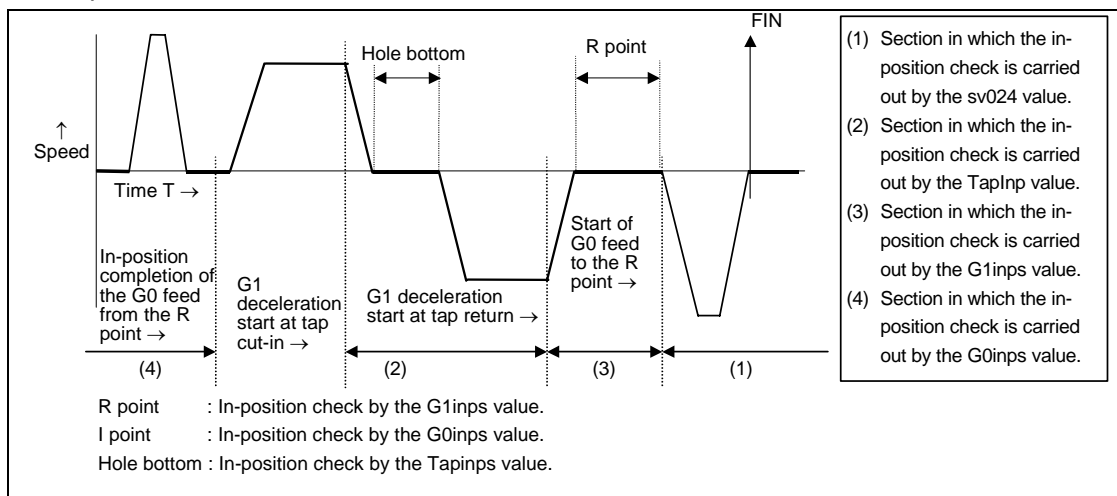
(3) Relation between the parameter setting values and tap axis movement for a synchronous tap in-position check

#1223 aux07				Hole bottom wait time	Operation at hole bottom	Operation at R point	Operation at I point → R point
bit3	bit4	bit5	bit2				
Synchronous tapping in-position check	At hole bottom	At R point	I point → R point				
0	-	-	-	Time designated by "P". Processing time is several 10ms when no "P".	Operation determined by setting of inpos (#1193) and aux 07 (#1223): bit 1 parameters.	Operation determined by setting of inpos (#1193) and aux 07 (#1223): bit 1 parameters.	Operation determined by setting of inpos (#1193) and aux 07 (#1223): bit 1 parameters.
1	0	0	1	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until time in left column has elapsed.		Wait until completion of in-position check by G0inps value.
1	0	1	1	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until time in left column has elapsed.	Wait until completion of in-position check by G1inps value.	Wait until completion of in-position check by G0inps value.
1	1	0	1	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until dwell time in left column has elapsed after completion of in-position check.		Wait until completion of in-position check by G0inps value.
1	1	1	1	Larger value of "P" and TapDwl (#1313) is valid. Processing time is several 10ms if both values are 0.	Wait until dwell time in left column has elapsed after completion of in-position check.	Wait until completion of in-position check by G1inps value.	Wait until completion of in-position check by G0inps value.
1	0	0	0	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until time in left column has elapsed.		
1	0	1	0	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until time in left column has elapsed.	Wait until completion of in-position check by G1inps value.	
1	1	0	0	Larger value of "P" and TapDwl (#1313) is valid. No dwell executed if both values are 0.	Wait until dwell time in left column has elapsed after completion of in-position check.		
1	1	1	0	Larger value of "P" and TapDwl (#1313) is valid. Processing time is several 10ms if both values are 0.	Wait until dwell time in left column has elapsed after completion of in-position check.	Wait until completion of in-position check by G1inps value.	

(Note 1) The I point refers to the initial point.

(Note 2) Keep in mind that vibration or deterioration in accuracy may occur when having the in-position check invalidated at R point.

Operations when each fixed cycle is executed

(a) G81 (Drilling, spot drilling)
Program
G81 Xx_1 Yy_1 Zz_1 Rr_1 Ff_1 ,Ii_1 ,Jj_1;

(1) G0 Xx_1 Yy_1
(2) G0 Zr_1
(3) G1 Zz_1 Ff_1
(4) G98 mode G0Z - (z_1+r_1)
    G99 mode G0Z - z_1
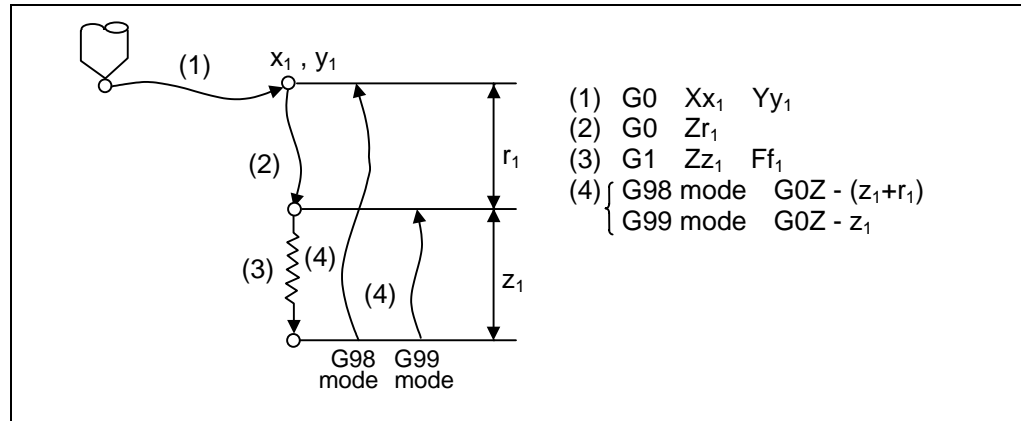
The operation stops at after the (1), (2) and (4) commands during single block operation.

Operation pattern	i1	j1
(1)	Valid	-
(2)	-	Invalid
(3)	-	Invalid
(4)	-	Valid

(b) G82 (Drilling, counter boring)
Program
G82 Xx_1 Yy_1 Zz_1 Rf_1 Ff_1 Pp_1 ,Ii_1 ,Jj_1 ;
P : Dwell designation

(1) G0 Xx_1 Yy_1
(2) G0 Zr_1
(3) G1 Zz_1 Ff_1
(4) G4 Pp_1 (Dwell)
(5) G98 mode G0Z - (z_1+r_1)
    G99 mode G0Z - z_1
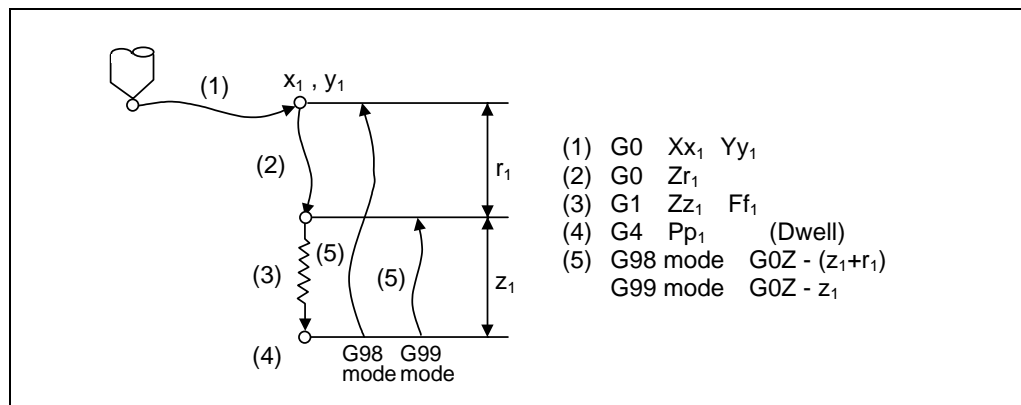
Operation pattern	i1	j1
(1)	Valid	-
(2)	-	Invalid
(3)	-	Invalid
(4)	-	-
(5)	-	Valid

The operation stops at after the (1), (2) and (5) commands during single block operation.

(c) G83 (Deep hole drilling cycle)
Program
G83 Xx_1 Yy_1 Zz_1 Rr_1 Qq_1 Ff_1 $,Ii_1$ $,Jj_1$;
Q : This designates the cutting amount per pass, and is always designated with an incremental value.

(1) G0 Xx_1 Yy_1
(2) G0 Zr_1
(3) G1 Zq_1 Ff_1
(4) G0 $Z - q_1$
(5) G0 $Z (q_1 - m)$
(6) G1 $Z (q_1 + m)$ Ff_1
(7) G0 $Z - 2 \cdot q_1$
(8) G0 $Z (2 \cdot q_1 - m)$
(9) G1 $Z (q_1 + m)$ Ff_1
(10) G0 $Z - 3 \cdot q_1$
⋮
⋮
(n) G98 mode $G0Z - (z_1 + r_1)$
G99 mode $G0Z - z_1$
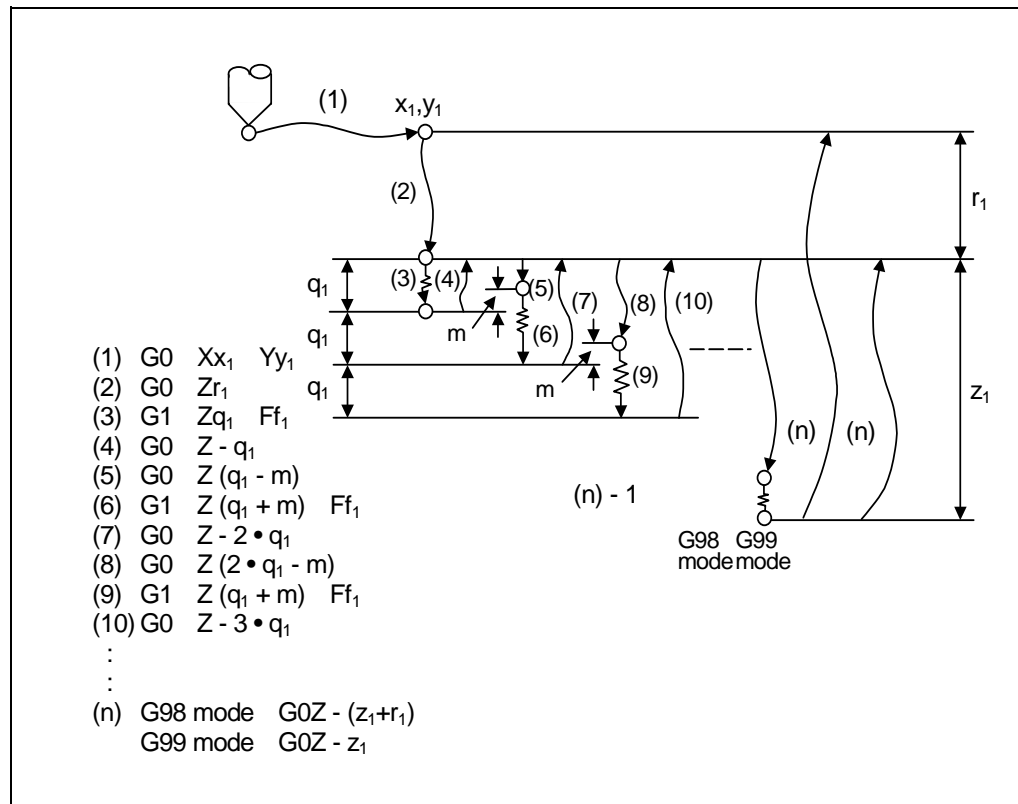
Operation pattern	i1	j1
(1)	Valid	-
(2)	-	Invalid
(3)	-	Invalid
(4)	-	Invalid
(5)	-	Invalid
(6)	-	Invalid
(7)	-	Invalid
(8)	-	Invalid
(9)	-	Invalid
(10)	-	Invalid
⋮		
n-1	-	Invalid
n	-	Valid

When executing a second and following cutting in the G83 as shown above, the movement will change from rapid traverse to cutting feed several mm before the position machined last. When the hole bottom is reached, the axis will return according to the G98 or G99 mode.
m will differ according to the parameter #8013 G83 return. Program so that $q_1 > m$.
The operation stops at after the (1), (2) and (n) commands during single block operation.

(d) G84 (Tapping cycle)
Program
G84 Xx_1 Yy_1 Zz_1 Rr_1 Ff_1 Pp_1 Rr_2 (or S_1, S_2) ,Ii_1 ,Jj_1;
P : Dwell designation
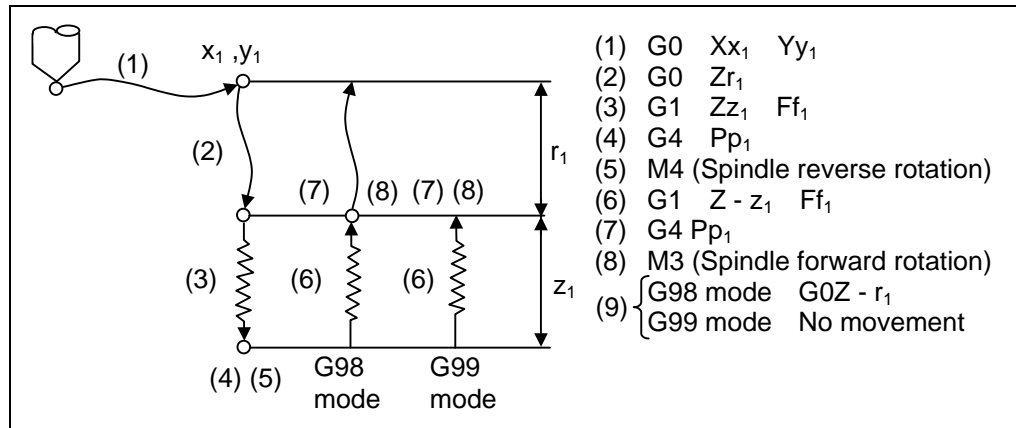
(1) G0 Xx_1 Yy_1
(2) G0 Zr_1
(3) G1 Zz_1 Ff_1
(4) G4 Pp_1
(5) M4 (Spindle reverse rotation)
(6) G1 $Z - z_1$ Ff_1
(7) G4 Pp_1
(8) M3 (Spindle forward rotation)
(9) G98 mode G0Z - r_1
G99 mode No movement

Operation pattern	i1	j1
(1)	Valid	-
(2)	-	Invalid
(3)	-	Invalid
(4)	-	-
(5)	-	-
(6)	-	Invalid
(7)	-	-
(8)	-	-
(9)	-	Valid

When $r_2 = 1$, the synchronous tapping mode will be entered, and when $r_2 = 0$, the asynchronous tapping mode will be entered.
When G84 is executed, the override will be canceled and the override will automatically be set to 100%. Dry run is valid when the control parameter "G00 DRY RUN" is on and is valid for the positioning command. If the feed hold button is pressed during G84 execution, and the sequence is at (3) to (6), the movement will not stop immediately, and instead will stop after (6). During the rapid traverse in sequence (1), (2) and (9), the movement will stop immediately.
The operation stops at after the (1), (2) and (9) commands during single block operation.
During the G84 modal, the "Tapping" NC output signal will be output.
During the G84 synchronous tapping modal, the M3, M4, M5 and S code will not be output.

This function allows spindle acceleration/deceleration pattern to be approached to the speed loop acceleration/deceleration pattern by dividing the spindle and drilling axis acceleration/deceleration pattern into up to three stages during synchronous tapping.
The acceleration/deceleration pattern can be set up to three stages for each gear.
When returning from the hole bottom, rapid return is possible depending on the spindle rotation speed during return. The spindle rotation speed during return is held as modal information.

(i) When tap rotation speed < spindle rotation speed during return ≤ synchronous tap changeover spindle rotation speed 2

S	: Command spindle rotation speed
S'	: Spindle rotation speed during return
S1	: Tap rotation speed (spindle base specification parameters #3013 to #3016)
S2	: Synchronous tap changeover spindle rotation speed 2 (spindle base specification parameters #3037 to #3040)
Smax	: Maximum rotation speed (spindle base specification parameters #3005 to #3008)
T1	: Tap time constant (spindle base specification parameters #3017 to #3020)
T2	: Synchronous tap changeover time constant 2 (spindle base specification parameters #3041 to #3044)

(ii) When synchronous tap changeover spindle rotation speed 2 < spindle rotation speed during return

S   : Command spindle rotation speed

S'   : Spindle rotation speed during return

S1   : Tap rotation speed (spindle base specification parameters #3013 to #3016)

S2   : Synchronous tap changeover spindle rotation speed 2
         (spindle base specification parameters #3037 to #3040)

Smax : Maximum rotation speed (spindle base specification parameters #3005 to #3008)

T1   : Tap time constant (spindle base specification parameters #3017 to #3020)

T2   : Synchronous tap changeover time constant 2
         (spindle base specification parameters #3041 to #3044)

T3   : Synchronous tap changeover time constant 3
         (spindle base specification parameters #3045 to #3048)

(iii) Pecking tapping cycle [E68]

The load applied to the tool can be reduced by designating the depth of cut per pass (Q) and cutting the workpiece to the hole bottom for a multiple number of passes.
The amount retracted from the hole bottom is set to the parameter "#8018 G84/G74 return".
Select either the pecking tapping cycle or the deep-hole tapping cycle by parameter. ("#1272 ext08/bit4")
When "depth of cut per pass Q" is designated in the block containing the G84 or G74 command in the state where the pecking tapping cycle is selected, the pecking tapping cycle is executed.
In the following cases, the normal tapping cycle is established.
- When Q is not designated
- When the command value of Q is zero

G84 Xx1 Yy1 Zz1 Rr1 Qq1 Ff1 Ee1 Pp1 Ss1 ,Ss2 ,Ii1 ,Jj1 ,Rr2 ;	
X, Yy	: Hole drilling position
Z	: Hole bottom position
R	: Point R position
Q	: Depth of cut per pass (designated as an incremental position)
F	: Z-axis feed amount (tapping pitch) per spindle rotation
E	: Tap thread number per 1-inch feed of Z axis
P	: Dwell time at hole bottom position
S	: Rotation speed of spindle
, S	: Rotation speed of spindle during retract
, I	: In-position width of positioning axis
, J	: In-position width of hole drilling axis
, R	: Synchronization method selection (r2=1 synchronous, r2=0 asynchronous)

(Note) When ",R0" is commanded, F address is regarded as cutting federate.

Refer to the section "Fixed cycle" for details.

(1) G0   Xx1   Yy1   , Ii1
(2) G0   Zr1
(3) G1   Zq1   Ff1
(4) M4 (Spindle reverse rotation)
(5) G1   Z-m   Ff1
(6) M3 (Spindle forward rotation)
(7) G1   Z(q1+m)   Ff1
(8) M4 (Spindle reverse rotation)
(9) G1   Z-m   Ff1
(10) M3 (Spindle forward rotation)
(11) G1   Z(q1+m)   Ff1
    :
    :
(n1) G1   Z(z1-q1*n)   Ff1
(n2) G4   Pp1
(n3) M4 (Spindle reverse rotation)
(n4) G1   Z-z1   Ff1   Ss2
(n5) G4   Pp1
(n6) M3 (Spindle forward rotation)
(n7) G98 mode   G0   Z-r1   , Ij1
      G99 mode   No movement

* 1. m: Parameter "#8018 G84/G74 return"
  2. This program is for the G84 command.   The spindle forward rotation (M3) and reverse rotation (M4) are reversed with the G74 command.

(iv) Deep-hole tapping cycle [E68]

In the deep-hole tapping, the load applied to the tool can be reduced by designating the depth of cut per pass and cutting the workpiece to the hole bottom for a multiple number of passes.

Under the deep-hole tapping cycle, the tool is retracted to the R-point every time.

Select either the pecking tapping cycle or the deep-hole tapping cycle by parameter. ("#1272 ext08/bit4")

When "depth of cut per pass Q" is designated in the block containing the G84 or G74 command in the state where the deep-hole tapping cycle is selected by parameter, the deep-hole tapping cycle is executed.

In the following cases, the normal tapping cycle is established.

- When Q is not designated
- When the command value of Q is zero

G84 Xx1 Yy1 Zz1 Rr1 Qq1 Ff1 Ee1 Pp1 Ss1 ,Ss2 ,Ii ,Jj ,Rr2 ;

X,Y	: Hole drilling position
Z	: Hole bottom position
R	: Point R position
Q	: Depth of cut per pass (designated as an incremental position)
F	: Z-axis feed amount (tapping pitch) per spindle rotation
E	: Tap thread number per 1-inch feed of Z axis
P	: Dwell time at hole bottom and point R return
S	: Rotation speed of spindle
,S	: Rotation speed of spindle during retract
,I	: In-position width of positioning axis
,J	: In-position width of hole drilling axis
,R	: Synchronization method selection (r2=1 synchronous, r2=0 asynchronous)

(Note) When ",R0" is commanded, F address is regarded as cutting federate.

Refer to the section "Fixed cycle" for details.

(1) G0 Xx1 Yy1
(2) G0 Zr1
(3) G9 G1 Zq1 Ff1
(4) M4 (Spindle reverse rotaion)
(5) G9 G1 Z-q1 Ff1
(6) G4 Pp1
(7) M3 (Spindle forward rotation)
(8) G1 Z(q1-c) Ff1
(9) G9 G1 Z(q1+c) Ff1
(10) M4 (Spindle reverse rotaion)
(11) G9 G1 Z-(2 q1) Ff1
(12) G4 Pp1
(13) M3 (Spindle forward rotation)
(14) G1 Z(2 q1-c) Ff1
(15) G9 G1 Z(q1+c) Ff1
     :
(n1) G9 G1 Z(z1-q1*n+c) Ff1
(n2) G4 Pp1
(n3) M4 (Spindle reverse rotaion)
(n4) G9 G1 Z-z1 Ff1
(n5) G4 Pp1
(n6) M3 (Spindle forward rotation)
(n7) G98 mode G0 Z-r1
        G99 mode No movement

*1. Clearance amount c : Parameter "#8018 G84/G74 return"
2. This program is for the G84 command. The spindle forward rotation (M3) and reverse rotation (M4) are reversed with the G74 command.

(e) G85 (Boring)
Program
G85 Xx_1 Yy_1 Zz_1 Rr_1 Ff_1 ,Ii_1 ,Jj_1;

(1) G0 Xx_1 Yy_1
(2) G0 Zr_1
(3) G1 Zz_1 Ff_1
(4) G1 $Z - z_1$ Ff_1
(5) $\begin{cases} \text{G98 mode} & \text{G0Z} - r_1 \\ \text{G99 mode} & \text{No movement} \end{cases}$
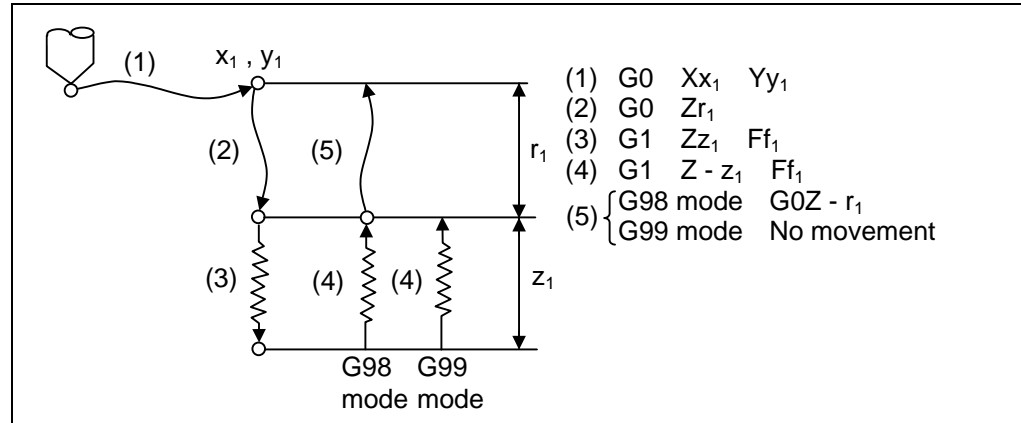
Operation pattern	i1	j1
(1)	Valid	-
(2)	-	Invalid
(3)	-	Invalid
(4)	-	Invalid
(5)	-	Valid

The operation stops at after the (1), (2), and (4) or (5) commands during single block operation.

(f) G86 (Boring)
Program
G86 Xx_1 Yy_1 Zz_1 Rr_1 Ff_1 Pp_1 ;
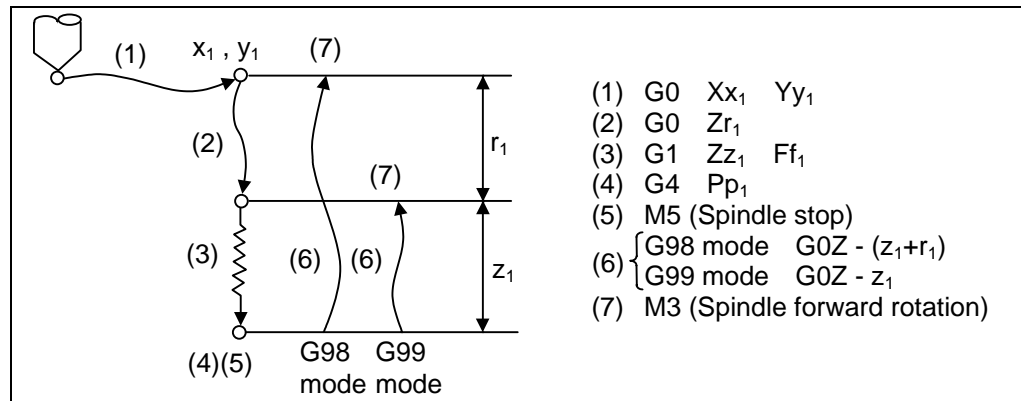
(1) G0 Xx_1 Yy_1
(2) G0 Zr_1
(3) G1 Zz_1 Ff_1
(4) G4 Pp_1
(5) M5 (Spindle stop)
(6) $\begin{cases} \text{G98 mode} & \text{G0Z} - (z_1 + r_1) \\ \text{G99 mode} & \text{G0Z} - z_1 \end{cases}$
(7) M3 (Spindle forward rotation)

The operation stops at after the (1), (2) and (7) commands during single block operation.

(g) G87 (Back boring)
Program
$G87 \; Xx_1 \; Yy_1 \; Zz_1 \; Rr_1 \; Iq_1 \; Jq_2 \; Ff_1$;
(Note) Take care to the z_1 and r_1 designations.
(The z_1 and r_1 symbols are reversed).
There is no R point return.

The operation stops at after the (1), (4), (6) and (11) commands during single block operation.
When this command is used, high precision drilling machining that does not scratch the machining surface can be done.
(Positioning to the hole bottom and the escape (return) after cutting is executed in the state shifted to the direction opposite of the cutter.)
The shift amount is designated as shown below with addresses I, J and K.

The shift amount is executed with linear interpolation, and the feed rate follows the F command.
Command I, J, and K with incremental values in the same block as the hole position data.
I, J and K will be handled as modals during the fixed cycle.
(Note)   If the parameter (#1080 Dril_Z) which fixes the hole drilling axis to the Z axis is set, the shift amount can be designated with address Q instead of I and j. In this case, whether to shift or not and the shift direction are set with parameter #8207 G76/87 No shift and #8208 G76/87 Shift (-). The symbol for the Q value is ignored and the value is handled as a positive value.
The Q value is a modal during the fixed cycle, and will also be used as the G83, G73 and G76 cutting amount.

(h) G88 (Boring)
Program
G88 Xx_1 Yy_1 Zz_1 Rr_1 Ff_1 Pp_1 ;

(1) G0 Xx_1 Yy_1
(2) G0 Zr_1
(3) G1 Zz_1 Ff_1
(4) G4 Pp_1
(5) M5 (Spindle stop)
(6) Stop when single block stop switch is ON.
(7) Automatic start switch ON
(8) $\begin{cases} \text{G98 mode} \quad \text{G0Z - } (z_1+r_1) \\ \text{G99 mode} \quad \text{G0Z - } z_1 \end{cases}$
(9) M3 (Spindle forward rotation)
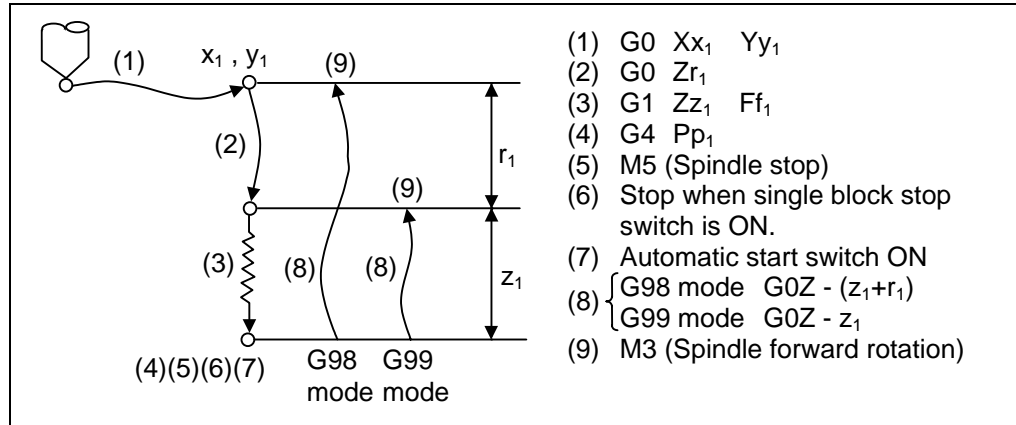
The operation stops at after the (1), (2), (6) and (9) commands during single block operation.

(i) G89 (Boring)
Program
G89 Xx_1 Yy_1 Zz_1 Rr_1 Ff_1 Pp_1 ,Ii_1 ,Jj_1 ;

(1) G0 Xx_1 Yy_1
(2) G0 Zr_1
(3) G1 Zz_1 Ff_1
(4) G4 Pp_1
(5) G1 $Z - z_1$ Ff_1
(6) $\begin{cases} \text{G98 mode} \quad \text{G0Z - } r_1 \\ \text{G99 mode} \quad \text{No movement} \end{cases}$
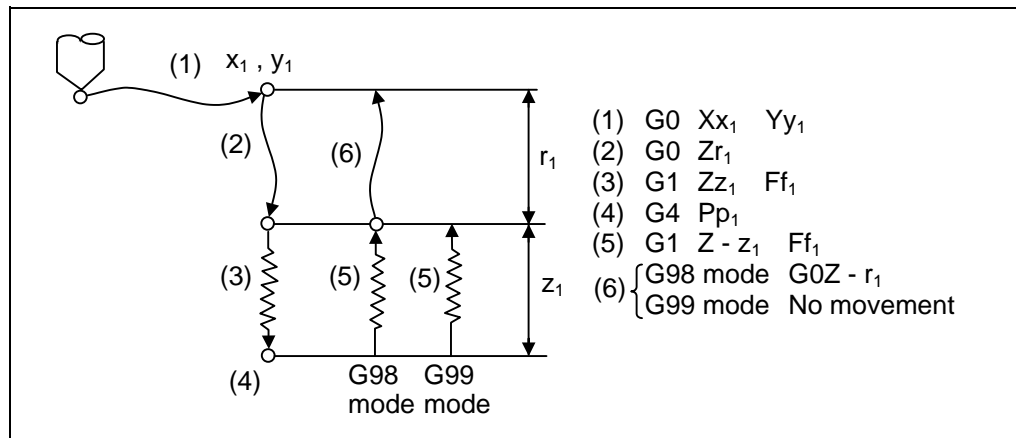
Operation pattern	i1	j1
(1)	Valid	-
(2)	-	Invalid
(3)	-	Invalid
(4)	-	-
(5)	-	Invalid
(6)	-	Valid

The operation stops at after the (1), (2) and (5) or (6) commands during single block operation.

(j) G73 (Step cycle)
Program
G73 Xx_1 Yy_1 Zz_1 Qq_1 Rr_1 Ff_1 Pp_1 ,Ii_1 ,Jj_1;
P : Dwell designation

(1) G0 Xx_1 Yy_1
(2) G0 Zr_1
(3) G1 Zq_1 Ff_1
(4) G4 Pp_1
(5) G0 Z - m
(6) G1 Z $(q_1 + m)$ Ff_1
 :
(n) G98 mode G0Z - $(z_1 + r_1)$
    G99 mode G0Z - z_1
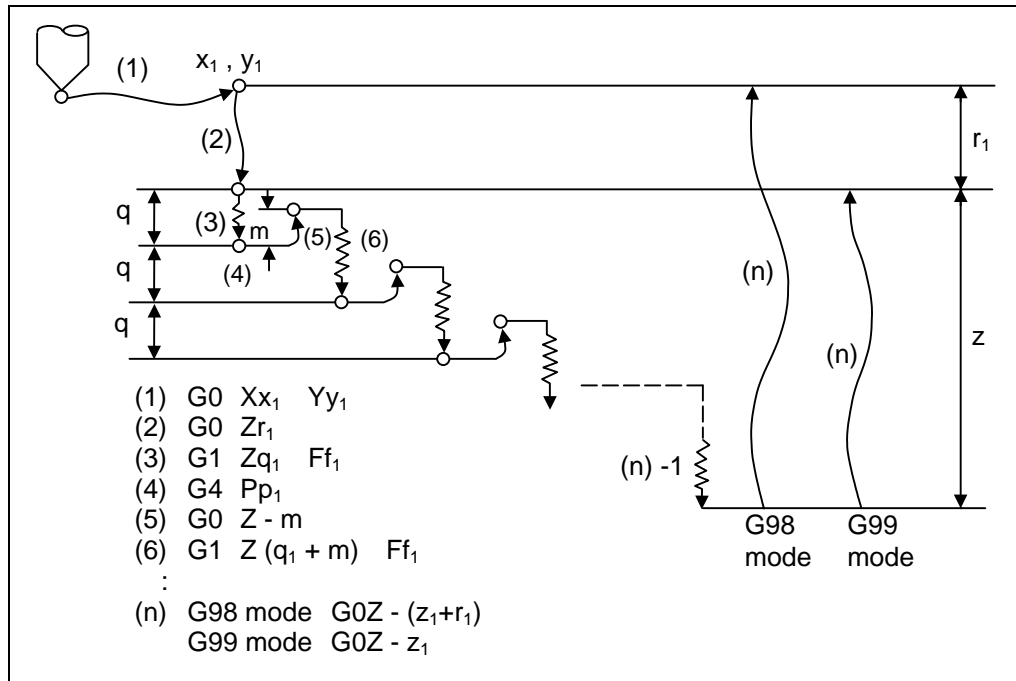
Operation pattern	i1	j1
(1)	Valid	-
(2)	-	Invalid
(3)	-	Invalid
(4)	-	-
(5)	-	Invalid
(6)	-	Invalid
:		
:		
(n) -1	-	Invalid
(n)	-	Valid

When executing a second and following cutting in the G73 as shown above, the movement will return several m mm with rapid traverse and then will change to cutting feed.
The return amount m will differ according to the parameter #8012 G73 return.
The operation stops at after the (1), (2) and (n) commands during single block operation.

(k)　G74 (Reverse tapping cycle)
Program
G74　Xx_1　Yy_1　Zz_1　Rr_1　Pp_1　Rr_2(or S_1,S_2) ,Ii_1 ,Jj_1 ;
P : Dwell designation
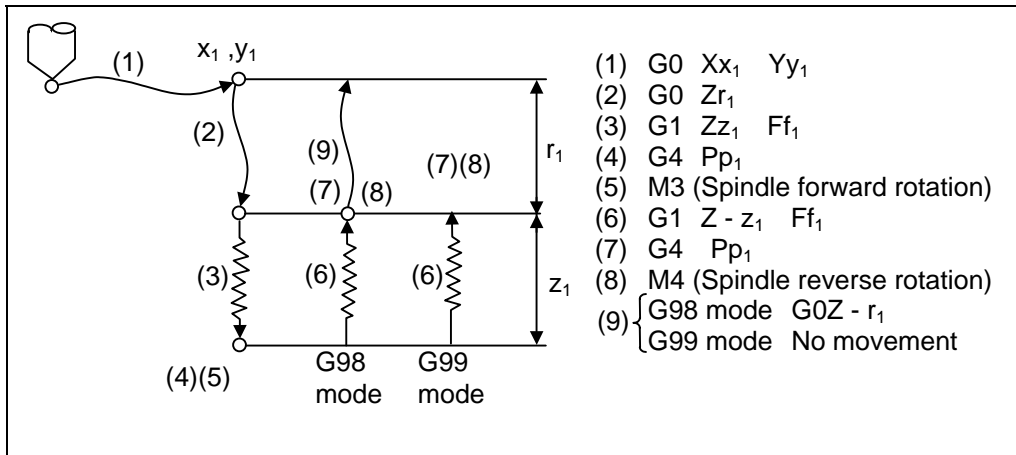
(1)　G0　Xx_1　Yy_1
(2)　G0　Zr_1
(3)　G1　Zz_1　Ff_1
(4)　G4　Pp_1
(5)　M3 (Spindle forward rotation)
(6)　G1　$Z - z_1$　Ff_1
(7)　G4　Pp_1
(8)　M4 (Spindle reverse rotation)
(9) $\begin{cases} \text{G98 mode   G0Z} - r_1 \\ \text{G99 mode   No movement} \end{cases}$

Operation pattern	i1	j1
(1)	Valid	-
(2)	-	Invalid
(3)	-	Invalid
(4)	-	-
(5)	-	-
(6)	-	Invalid
(7)	-	-
(8)	-	-
(9)	-	Valid

When $r_2 = 1$, the synchronous tapping mode will be entered, and when $r_2 = 0$, the asynchronous tapping mode will be entered.
When G74 is executed, the override will be canceled and the override will automatically be set to 100%. Dry run is valid when the control parameter #1085 G00Drn is set to "1" and is valid for the positioning command. If the feed hold button is pressed during G74 execution, and the sequence is at (3) to (6), the movement will not stop immediately, and instead will stop after (6). During the rapid traverse in sequence (1), (2) and (9), the movement will stop immediately.
The operation stops at after the (1), (2) and (9) commands during single block operation.
During the G74 and G84 modal, the "Tapping" NC output signal will be output.
During the G74 synchronous tapping modal, the M3, M4, M5 and S code will not be output.

This function allows spindle acceleration/deceleration pattern to be approached to the speed loop acceleration/deceleration pattern by dividing the spindle and drilling axis acceleration/deceleration pattern into up to three stages during synchronous tap.
Refer to the item "d) G84 (Tapping cycle)" for details of multi-stages of the spindle acceleration/deceleration pattern.

(i) When tap rotation speed < spindle rotation speed during return ≤ synchronous tap changeover spindle rotation speed 2

S	: Command spindle rotation speed
S'	: Spindle rotation speed during return
S1	: Tap rotation speed (spindle base specification parameters #3013 to #3016)
S2	: Synchronous tap changeover spindle rotation speed 2 (spindle base specification parameters #3037 to #3040)
Smax	: Maximum rotation speed (spindle base specification parameters #3005 to #3008)
T1	: Tap time constant (spindle base specification parameters #3017 to #3020)
T2	: Synchronous tap changeover time constant 2 (spindle base specification parameters #3041 to #3044)

(ii) When synchronous tap changeover spindle rotation speed 2 < spindle rotation speed during return

S	: Command spindle rotation speed
S'	: Spindle rotation speed during return
S1	: Tap rotation speed (spindle base specification parameters #3013 to #3016)
S2	: Synchronous tap changeover spindle rotation speed 2 (spindle base specification parameters #3037 to #3040)
Smax	: Maximum rotation speed (spindle base specification parameters #3005 to #3008)
T1	: Tap time constant (spindle base specification parameters #3017 to #3020)
T2	: Synchronous tap changeover time constant 2 (spindle base specification parameters #3041 to #3044)
T3	: Synchronous tap changeover time constant 3 (spindle base specification parameters #3045 to #3048)

(iii) Pecking reverse tapping cycle [E68]

The items other than the spindle rotation direction are the same with the pecking tapping cycle (G84). Refer to "(iii) Pecking tapping cycle" in "(d) G84 (Tapping cycle)" for details.

```
G74 Xx1 Yy1 Zz1 Rr1 Qq1 Ff1 Ee1 Pp1 Ss1 ,Ss2 ,Ii1 ,Jj1 ,Rr2 ;
```

X, Yy	: Hole drilling position
Z	: Hole bottom position
R	: Point R position
Q	: Depth of cut per pass (designated as an incremental position)
F	: Z-axis feed amount (tapping pitch) per spindle rotation
E	: Tap thread number per 1-inch feed of Z axis
P	: Dwell time at hole bottom position
S	: Rotation speed of spindle
, S	: Rotation speed of spindle during retract
, I	: In-position width of positioning axis
, J	: In-position width of hole drilling axis
, R	: Synchronization method selection (r2=1 synchronous, r2=0 asynchronous)

(Note) When ",R0" is commanded, F address is regarded as cutting federate.

(iv) Deep-hole reverse tapping cycle [E68]

The items other than the spindle rotation direction are the same with the deep-hole tapping cycle (G84). Refer to "(iv) Deep-hole tapping cycle" in "(d) G84 (Tapping cycle)" for details.

```
G74 Xx1 Yy1 Zz1 Rr1 Qq1 Ff1 Ee1 Pp1 Ss1 ,Ss2 ,Ii ,Jj ,Rr2 ;
```

X,Y	: Hole drilling position
Z	: Hole bottom position
R	: Point R position
Q	: Depth of cut per pass (designated as an incremental position)
F	: Z-axis feed amount (tapping pitch) per spindle rotation
E	: Tap thread number per 1-inch feed of Z axis
P	: Dwell time at hole bottom and point R return
S	: Rotation speed of spindle
,S	: Rotation speed of spindle during retract
,I	: In-position width of positioning axis
,J	: In-position width of hole drilling axis
,R	: Synchronization method selection (r2=1 synchronous, r2=0 asynchronous)

(Note) When ",R0" is commanded, F address is regarded as cutting federate.

(l) G76 (Fine boring)
Program
G76 Xx_1 Yy_1 Zz_1 Rr_1 Iq_1 Jq_2 Ff_1 ;

(1) G0 Xx_1 Yy_1
(2) G0 Zr_1
(3) G1 Zz_1 Ff_1
(4) M19 (Spindle orient)
(5) G1 Xq_1 (Yq_2) Ff_1 (Shift)
(6) $\begin{cases} \text{G98 mode} & \text{G0Z} - (z_1+r_1) \\ \text{G99 mode} & \text{G0Z} - z_1 \end{cases}$
(7) G0 X - q_1 (Y - q_2) Ff_1
    (Shift)
(8) M3 (Spindle forward rotation)

The operation stops at after the (1), (2) and (7) commands during single block operation.
When this command is used, high precision drilling machining that does not scratch the machining surface can be done.
(Positioning to the hole bottom and the escape (return) after cutting is executed in the state shifted to the direction opposite of the cutter.)

The shift amount is designated as shown below with addresses I, J and K.
   For G17 : I, J
   For G18 : K, I
   For G19 : J, K
The shift amount is executed with linear interpolation, and the feed rate follows the F command.

Command I, J, and K with incremental values in the same block as the hole position data.
I, J and K will be handled as modals during the fixed cycle.

(Note) If the parameter (#1080 Dril_z) which fixes the hole drilling axis to the Z axis is set, the shift amount can be designated with address Q instead of I and J. In this case, whether to shift or not and the shift direction are set with parameter "#8207 G76/87 IGNR" and "#8208 G76/87 (-)". The symbol for the Q value is ignored and the value is handled as a positive value.
The Q value is a modal during the fixed cycle, and will also be used as the G83, G87 and G73 cutting amount.

✋ Precautions for using fixed cycle

(1) Before the fixed cycle is commanded, the spindle must be rotating in a specific direction with an M command (M3 ; or M4 ;).
Note that for the G87 (back boring) command, the spindle rotation command is included in the fixed cycle so only the rotation speed command needs to be commanded beforehand.

(2) If there is a basic axis, additional axis or R data in the block during the fixed cycle mode, the hole drilling operation will be executed. If there is not data, the hole will not be drilled.
Note that in the X axis data, if the data is a dwell (G04) time command, the hole will not be drilled.

(3) Command the hole machining data (Q, P, I, J, K) in a block where hole drilling is executed. (Block containing a basic axis, additional axis or R data.)

(4) The fixed cycle can be canceled by the G00 to G03 or G33 command besides the G80 command. If these are designated in the same block as the fixed cycle, the following will occur.

> (Where, 00 to 03 and 33 are m, and the fixed cycle code is n)
>
> Gm   Gn   X___Y___Z___R___Q___P___L___F___;
> Execute Ignore Execute   Ignore   Record
> Gm   Gn   X___Y___Z___R___Q___P___L___F___;
> Ignore   Execute   Ignore   Record

Note that for the G02 and G03 commands, R will be handled as the arc radius.

(5) If an M function is commanded in the same block as the fixed cycle command, the M code and MF will be output during the initial positioning. The next operation will be moved to with FIN (finish signal).
If there is a No. of times designated, the above control will be executed only on the first time.

(6) If another control axis (ex., rotary axis, additional axis) is commanded in the same block as the fixed cycle control axis, the fixed cycle will be executed after the other control axis is moved first.

(7) If the No. of repetitions L is not designated, L1 will be set. If L0 is designated in the same block as the fixed cycle G code command, the hole machining data will be recorded, but the hole machining will not be executed.
(Example) G73X___Y___Z___R___Q___P___F___L0___;
                Execute   Record only code having an address

(8) When the fixed cycle is executed, only the modal command commanded in the fixed cycle program will be valid in the fixed cycle subprogram. The modal of the program that called out the fixed cycle will not be affected.

(9) Other subprograms cannot be called from the fixed cycle subprogram.

(10) Decimal points in the movement command will be ignored during the fixed cycle subprogram.

(11) If the No. of repetitions L is 2 or more during the incremental value mode, the positioning will also be incremented each time.
(Example) G91G81X10. Z−50.R−20.F100.L3 ;
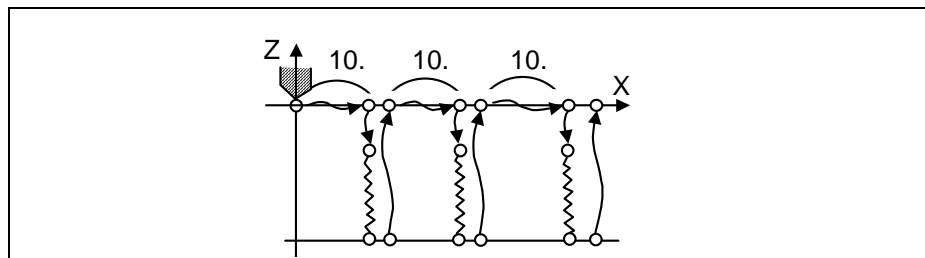
(12) If the spindle rotation speed value during return is smaller than the spindle speed, the spindle rotation speed value is valid even during return.

(13) If the 2nd and 3rd acceleration/deceleration stage inclinations following the spindle rotation speed and time constants set in the parameters are each steeper than the previous stage's inclination, the previous stage's inclination will be valid.

(14) If the values set in the spindle base specification parameter "stap1-4" (tap rotation speed) and "taps21-24" (synchronous tap changeover spindle rotation speed 2) exceed the maximum rotation speed, the spindle rotation speed will be clamped at the maximum rotation speed.

(15) If the spindle rotation speed during return is not 0, the tap return override value will be invalid.

(16) In a block where the movement direction of any axis reverses as shown below, the servo system load will greatly increase so do not command the in-position width in the machining program.
    G0 X100., I10.0 ;
    X–200. ;

(17) If the in-position width commanded with the programmable in-position width command is increased, the positioning time and linear interpolation time can be reduced. However, the position error amount of the previous block will also increase before the next block starts, so the actual machining could be obstructed.

(18) The in-position width and the position error amount are compared at a set time, so the position error amount at the point to be judged as in-position will be smaller than the commanded in-position width.

(19) If the in-position width commanded with the programmable in-position command is small, the commanded deceleration check or in-position check following the parameters may be carried out first.

(20) Synchronous and asynchronous tap can be selected with the M function.

Base specifications parameters

#	Items		Details	Setting range
1272 (PR)	ext08	bit1	M-function synchronous tap cycle valid.	0:Invalid 1:Valid

Synchronous tap cannot be selected with the M function when this parameter is OFF.

Base specifications parameters

#	Items	Details	Setting range
1513	stapM	M code for synchronous tap selection	0 to 99999999

The synchronous tap mode is selected with the miscellaneous function code set with this parameter.
The M function can be commanded in the same block before the tap command.

The synchronous and asynchronous tap will follow the combination shown below.

	Combination											
Program command (,R0/1)	0	0	0	0	1	1	1	1	No command			
#1229 (bit4) (Synchronous tap valid)	0	0	1	1	0	0	1	1	0	0	1	1
M function code (M**)	×	O	×	O	×	O	×	O	×	O	×	O
Synchronous/ asynchronous selection	A	A	A	A	S	S	S	S	A	S	S	S

× : Does not command   A : Asynchronous tap
O : Commands   S : Synchronous tap

(Note1) Do not use M00, 01, 02, 30, 96, 97, 98 or 99.
(Note2) Depending on the model, selection with the M function may not be possible.

13.1.2 Initial point and R point level return; G98, G99

Function and purpose

Whether to use R point or initial level for the return level in the final sequence of the fixed cycle can be selected.

Command format

> **G98 ;**
> **G99 ;**
> G98 ; Initial level return
> G99 ; R point level return

Detailed description

The relation of the G98/G99 mode and No. of repetition designation is as shown below.

No. of hole drilling	Program example	G98 At power ON, at cancel with M02, M30, and reset button	G99
Only one execution	G81X100. Y100. Z−50. R25. F1000;	 Initial level return is executed.	 R point level return is executed.
Second and following executions	G81X100. Y100. Z−50. R25. L5F1000;	 First time Second time Final time Initial level return is executed for all times.	 First time Second time Final time

Example of program

(Example 1)

G82 Zz$_1$ Rr$_1$ Pp$_1$ Ff$_1$ L0 ;	Record only the hold machining data (Do not execute)
Xx$_1$ Yy$_1$;	Execute hole drilling operation with G82 mode

The No. of fixed cycle repetitions is designated with L. If L1 is designated or L not designated, the fixed cycle will be executed once. The setting range is 1 to 9999.
If L0 is commanded, only the hole machining data will be recorded.
G8Δ (7Δ) Xx$_1$ Yy$_1$ Zz$_1$ Rr$_1$ Pp$_1$ Qq$_1$ Ff$_1$ Ll$_1$;

The ideology of the data differs between the absolute value mode (G90) and incremental value mode (G91) as shown below.
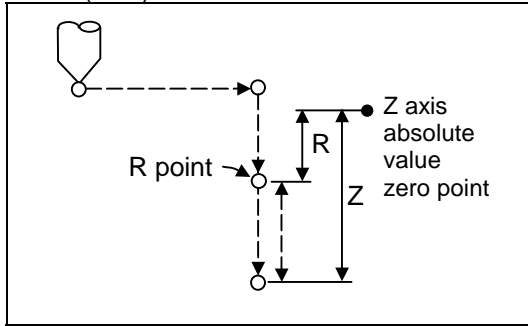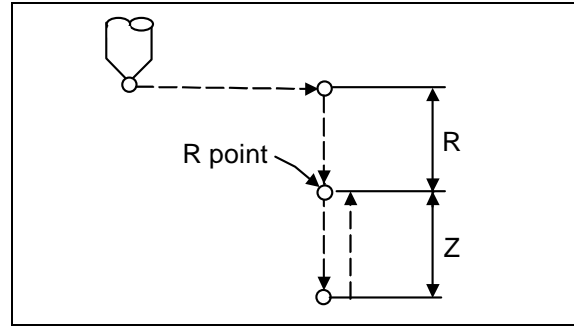
Absolute value mode (G90)   **Incremental value mode (G91)**

Designate a command value with a symbol for X, Y and Z. R indicates the coordinate value from the zero point in the absolute value mode, so a symbol must always be added. However, in the incremental value the symbol will be ignored and will be viewed as the same symbol as for Z. Note that the symbols will be viewed in reverse for G87.

The hole machining data is held as shown below in the fixed cycle. The hole machining data is canceled when the G80 command or G commands (G00, G01, G02, G03, G2.1, G3.1, G33) in the 01 group are reached.

(Example 2)

N001 G81 Xx_1 Yy_1 Zz_1 Rr_1 Ff_1 ;	
N002 G81 ;	Only selection of fixed cycle sequence
N003 Xx_2 Yy_2 ;	Change of positioning point, and execution of fixed cycle
N004 M22 ;	Execution of only M22
N005 G04 Xx_3 ;	Execution of only dwell
N006 G92 Xx_4 Yy_4 ;	Execution of only coordinate system setting
N007 G28 (G30) Z0 ;	Execution of only reference point (zero point) return
N008 ;	No work
N009 G99 Zz_2 Rr_2 Ff_2 L0 ;	Execution of only hole machining data recording
N010 Xx_5 Yy_5 Ll_5 ;	Change of positioning point, and execution of R point return fixed cycle for l_5 times
N011 G98 Xx_6 Yy_6 Zz_6 Rr_6 ;	Change of positioning point, and execution of fixed cycle
N012 Ww_1 ;	Execute W axis according to 01 group modal before N001, and then execute fixed cycle

13.1.3 Setting of workpiece coordinates in fixed cycle mode

The designated axis moves with the workpiece coordinate system set for the axis.
The Z axis is valid after the R point positioning after positioning or from Z axis movement.

(Note) When the workpiece coordinates are changed over for address Z and R, re-program even if the values are the same.

(Example)

G54 Xx_1 Yy_1 Zz_1 ;	
G81 Xx_2 Yy_2 Zz_2 Rr_2 ;	
G55 Xx_3 Yy_3 Zz_2 Rr_2 ;	Re-command even if Z and R are the same as the previous value.
Xx_4 Yy_4 ;	
Xx_5 Yy_5 ;	

13.2 Special fixed cycle; G34, G35, G36, G37.1

Function and purpose

The special fixed cycle is used with the standard fixed cycle. Before using the special fixed cycle, program the fixed cycle sequence selection G code and hole machining data to record the hole machining data. (If there is no positioning data, the fixed cycle will not be executed, and only the data will be recorded.)

The axis is positioned to the hole machining position when the special fixed cycle is executed.

The hole machining operation is executed with the fixed cycle for hole machining.

Even after the special fixed cycle is executed, the recorded standard fixed cycled will be held until canceled.

If the special fixed cycle is designated when not in the fixed cycle mode, only positioning will be executed, and the hole drilling operation will not be done.

If the special fixed cycle is commanded without commanding the fixed cycle for hole machining, positioning will be executed following the current 01 group modal G code.

Bolt hole circle (G34)

G34 X x₁ Y y₁ I r J θ K n ;

X, Y	:Positioning of bolt hole cycle center. This will be affected by G90/G91.
I	:Radius r of the circle. The unit follows the input setting unit, and is given with a positive number.
J	:Angle θ of the point to be drilled first. The CCW direction is positive. (The decimal point position will be the degree class. If there is no decimal point, the unit will be 0.001 ˚.)
K	:No. of holes n to be drilled. 1 to 9999 can be designated, but 0 cannot be designated. When the value is positive, positioning will take place in the CCW direction, and when negative, will take place in the CW direction. If 0 is designated, the alarm P221 Special Canned Holes Zero will occur.

Drilling of n obtained by dividing the circumference by n will start at point created by the Z axis and angle θ. The circumference is that of the radius R centering on the coordinates designated with XX and Y. The hole drilling operation at each hole will hold the drilling data for the standard fixed cycle such as G81.

The movement between hole positions will all be done in the G00 mode. G34 will not hold the data even when the command is completed.

(Example)

When input setting unit is 0.001mm

```
N001   G91   ;
N002   G81   Z – 10000   R5000   L0     F200   ;
N003   G90   G34   X200000   Y100000   I100000   J20000   K6   ;
N004   G80   ;  ...............   (Cancel of G81)
N005   G90   G0   X500000   Y100000   ;
```


x₁=200mm
n = 6 holes
20°
I=100mm
y₁=100mm
(500mm, 100mm)
Position before G34 is executed
N005 G0 command
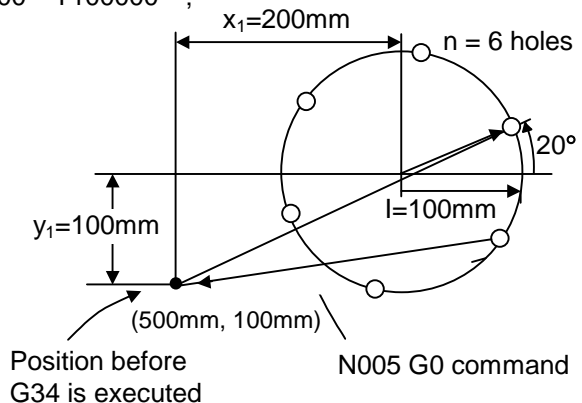
As shown in the example, the tool position after the G34 command is completed is over the final hole. When moving to the next position, the coordinate value must be calculated to issue the command with an incremental value. Thus, use of the absolute value mode is handy.

(Note1) If an address other than the selected plane's vertical axis, horizontal axis, G, N, I, J, K, H, O, P, F, M, S or 2nd auxiliary function is issued in the same block as the G34 command, a program error (P32) will occur.

Line at angle (G35)

G35 X x1 Y y1 I d J θ K n ;
   X, Y   :Designation of start point coordinates. This will be affected by G90/G91.
   I   :Interval d. The unit follows the input setting unit. If d is negative, the drilling will
             take place in the direction symmetrical to the point that is the center of the start
             point.
   J   :Angle θ. The CCW direction is positive.
             (The decimal point position will be the degree class. If there is no decimal point,
             the unit will be 0.001 ˚ .)
   K   :No. of holes n to be drilled. 1 to 9999 can be designated, and the start point is
             included.

Using the position designated by X and Y as the start point, the Zn holes will be drilled with interval d in the direction created by X axis and angle θ. The hole drilling operation at each hole position will be determined by the standard fixed cycle, so the hole drilling data (hole machining mode and hole machining data) must be held beforehand. The movement between hole positions will all be done in the G00 mode. G35 will not hold the data even when the command is completed.

(Example)

When input setting unit is 0.001mm

   G91   ;
   G81   Z − 10000   R5000   L0   F100   ;
   G35   X200000   Y100000   I100000   J30000   K5   ;

d=100mm
n = 5 holes
θ=30°
y_1=100mm
x_1=200mm
Position before
G35 is executed
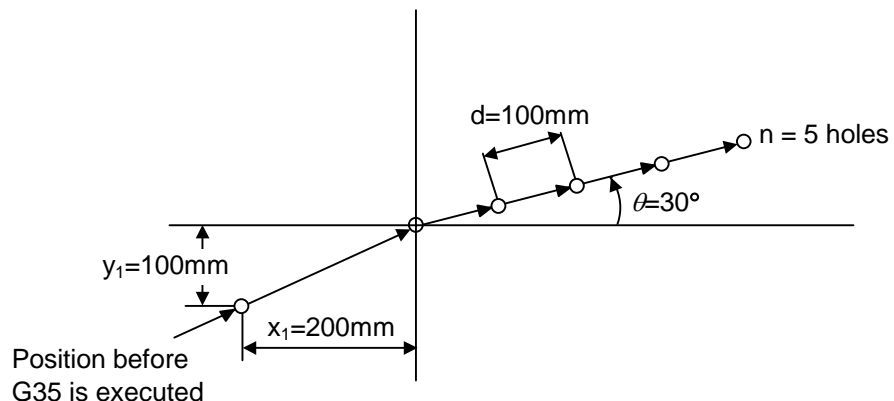
(Note 1) If the K command is K0 or if there is no K command, the program error "P221" will occur.
(Note 2) If the K value is more than four digits, the last four digits will be valid.
(Note 3) If a group 0 G command is issued in the same block as the G35 command, the command issued later is the priority.
         (Example) G35 G28 Xx_1 Yy_1 Ii_1 Jj_1 Kk_1 ;
                  G35 is ignored G 28 is executed as Xx1 Yy1
(Note 4) If there is a G72 to G89 command in the same block as the G35 command, the fixed cycle will be ignored, and the G35 command will be executed.
(Note 5) If there is a G72 to G89 command in the same block as the G35 command, the fixed cycle will be ignored, and the G35 command will be executed.

Arc (G36)

G36 X x1 Y y1 I r J θ P Δθ K n ;

X, Y	:Center coordinates of arc. This will be affected by G90/G91.
I	:Radius r of arc. The unit follows the input setting unit, and is given with a positive No.
J	:Angle θ of the point to be drilled first. The CCW direction is positive. (The decimal point position will be the degree class. If there is no decimal point, the unit will be 0.001 ˚.)
P	:Angle interval Δθ. When the value is positive, the drilling will take place in the CCW direction, and in the CW direction when negative. (The decimal point position will be the degree class. If there is no decimal point, the unit will be 0.001 ˚.)
K	:No. of holes n to be drilled. 1 to 9999 can be designated.

The n holes aligned at the angle interval Δθ will be drilled starting at point created by the X axis and angle θ. The circumference is that of the radius R centering on the coordinates designated with XX and Y. As with the bolt hole circle, the hole drilling operation at each hole will depend on the standard fixed cycle.

The movement between hole positions will all be done in the G00 mode. G36 will not hold the data even when the command is completed.

(Example)

When input setting unit is 0.001mm

```
N001  G91  ;
N002  G81  Z − 10000   R5000   F100   ;
N003  G36  X300000   Y100000   I300000   J10000   P15000   K6 ;
```
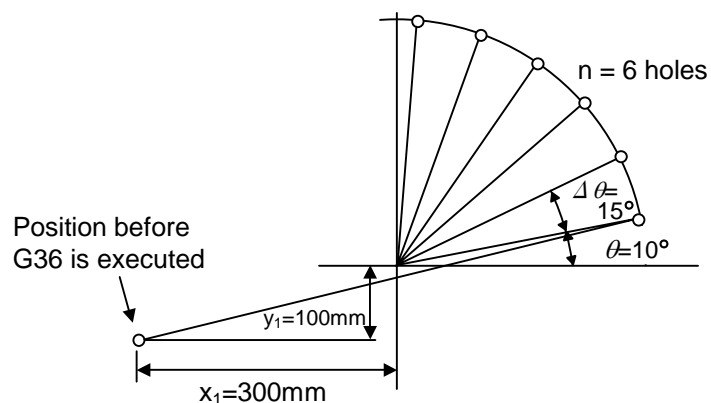
n = 6 holes

Δθ=15°

θ=10°

Position before G36 is executed

y₁=100mm

x₁=300mm

(Note 1) If an address other than the selected plane's vertical axis, horizontal axis, G, N, I, J, K, H, O, P, F, M, S or 2nd auxiliary function is issued in the same block as the G36 command, a program error (P32) will occur.

Grid (G37.1)

G37.1 X x1 Y y1 I Dx P nx J Dy K ny ;
X, Y	:Designation of start point coordinates. This will be affected by G90/G91.
I	:Interval Dx of the X axis. The unit will follow the input setting unit. If Dx is positive, the interval will be in the forward direction looking from the start point, and when negative, will be in the reverse direction looking from the start point.
P	:No. of holes nx in the X axis direction. The setting range is 1 to 9999.
J	:Interval Dy of the Y axis. The unit will follow the input setting unit. If Dy is positive, the interval will be in the forward direction looking from the start point, and when negative, will be in the reverse direction looking from the start point.
K	:No. of holes ny in the Y axis direction. The setting range is 1 to 9999.

The nx points on a grid are drilled with an interval Δx parallel to the X axis, starting at the position designated with X, Y. The drilling operation at each hole position will depend on the standard fixed cycle, so the hole drilling data (hole machining mode and hole machining data) must be held beforehand.

The movement between hole positions will all be done in the G00 mode. G37.1 will not hold the data even when the command is completed.

(Example)

When input setting unit is 0.01mm

  G91   ;
  G81   Z − 10000   R5000   F20   ;
  G37.1   X300000   Y−100000   I50000   P10   J100000   K8   ;

Position before
G37 is executed

y_1=100mm

ny = 8 holes

Δy= 100mm

Δx=50mm

x_1=300mm

nx = 10 holes
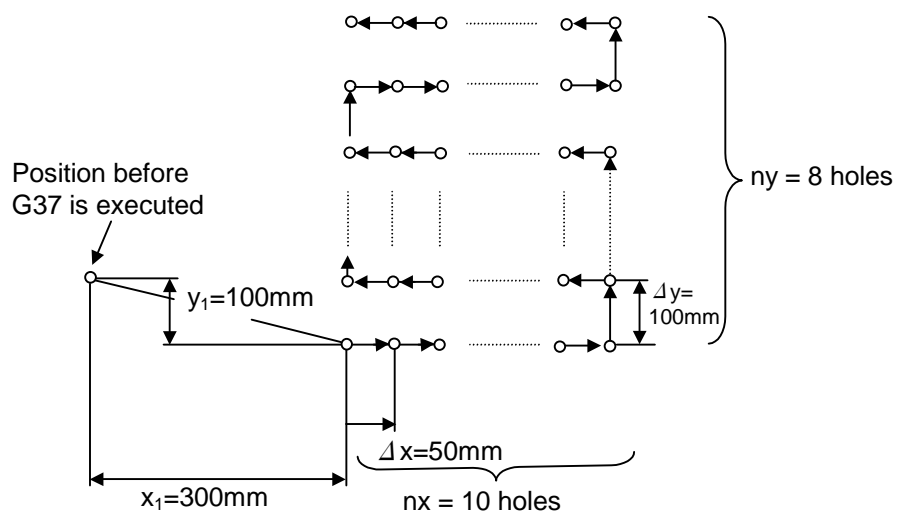
(Note 1) If the P and K commands are P0 or K0, or if there is no P or K command, the program error "P221" will occur.
If the P or K value is more than four digits, the last four digits will be valid.

(Note 2) If an address other than the selected plane's vertical axis, horizontal axis, G, N, I, J, K, H, O, P, F, M, S or 2nd auxiliary function is issued in the same block as the G37.1 command, a program error (P32) will occur.

(Note 3) If a group 0 G command is issued in the same block as the G37.1 command, the command issued later is the priority.

(Note 4) If there is a G72 to G89 command in the same block as the G37.1 command, the fixed cycle will be ignored, and the G37.1 command will be executed.

13.3 Subprogram control; M98, M99, M198

13.3.1 Calling subprogram with M98 and M99 commands

Function and purpose

Fixed sequences or repeatedly used patterns can be stored in the memory as subprograms which can then be called from the main program when required. M98 serves to call subprograms and M99 serves to return operation from the subprogram to the main program. Furthermore, it is possible to call other subprograms from particular subprograms and the nesting depth can include as many as 8 levels.

Main program	Subprogram	Subprogram	Subprogram	Subprogram
O0010 ;	O1000 ;	O1200 ;	O2000 ;	O3000 ;
		N20 ;		
M98 P1000 ;	M98 P1200 H20 ;	M98 P2000 ;	M98 P2500 ;	
		N60 ;		
M02 ;	M99 ;	M99 ;	M99 P60 ;	M99 ;
	(Level 1)	(Level 2)	(Level 3)	(Level 8)

Nesting depth
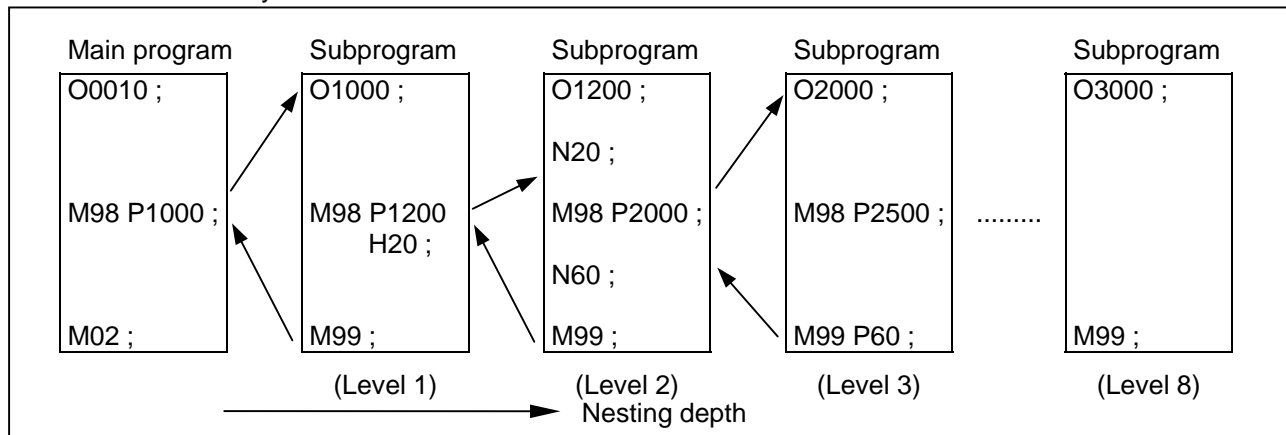
The table below shows the functions which can be executed by adding and combining the tape storing and editing functions, subprogram control functions and fixed cycle functions.

	Case 1	Case 2	Case 3	Case 4
1. Tape storing and editing	Yes	Yes	Yes	Yes
2. Subprogram control	No	Yes	Yes	No
3. Fixed cycles	No	No	Yes	Yes
Function				
1. Memory operation	◯	◯	◯	◯
2. Tape editing (main memory)	◯	◯	◯	◯
3. Subprogram call	×	◯	◯	×
4. Subprogram variable designation (Note 2)	×	◯	◯	×
5. Subprogram nesting level call (Note 3)	×	◯	◯	×
6. Fixed cycles	×	×	◯	◯
7. Fixed cycle subprogram editing	×	×	◯	◯

(Note 1) " ◯ " denotes function which can be used and "×" a function which cannot be used.

(Note 2) Variables cannot be transferred with the M98 command but variable commands in subprograms can be used provided that the variable command specifications are available.

(Note 3) A maximum of 8 nesting levels can be possible.

13. Program Support Functions

Command format

Subprogram call

M98 P_ H_ L_ ;	
P	: Program number of subprogram to be called (own program if omitted)
	P can only be omitted during memory operation and MDI operation.
	(Numerical value with up to 8 digits)
H	: Sequence number in subprogram to be called (head block if omitted)
	(Numerical value with up to 5 digits)
L	: Number of subprogram repetitions (When omitted, this is interpreted at
	L1, and is not excuted when L0)
	(1 to 9999 with numerical value up to 4 digits)
	For instance
	M98 P1 L3 ; is equivalent to the following:
	M98 P1 ;
	M98 P1 ;
	M98 P1 ;

Return to main program from subprogram

M99 P__ ;	
P__ :	Sequence number of return destination (returned to block following block to be called if omitted)

Creating and entering subprograms

Subprograms have the same format as machining programs for normal memory operation except that the subprogram completion instruction M99 (P__) is entered as an independent block at the last block.

O△△△△△△△△　　　　;	Program number as subprogram
....　;	
....　;	
:　　　　　;	Main body of subprogram
:	
....　;	
M99 ;	Subprogram return command
% (EOR)	Entry completion code

(1) The above program is entered by editing operations at the setting and display unit. For further details, refer to the section on program editing in the Control Instructions.

(2) Only those subprogram numbers ranging from 1 through 99999999 designated by the optional specifications can be used. When there are no program numbers on the tape, they are entered as the setting number for "program input."

(3) Up to 8 nesting levels can be used for calling subprograms from programs, and program error (P230) results if this number is exceeded.

(4) No distinction between main programs and subprograms is made since they are entered in the sequence in which they were read. This means that main programs and subprograms should not be given the same numbers. (If they are, error "E11" appears during entry.)

Registration example

```
     ;
O○○○○○     ;
.............................   ;
    :                          } Subprogram A
M99     ;
%
```

```
O△△△△        ;
.............................   ;
    :                          } Subprogram B
M99     ;
%
```

```
O * * * *        ;
.............................   ;
    :                          } Subprogram C
M99     ;
%
```

(5) Main programs can be entered during memory, tape, MDI operation but subprograms must be entered in the memory.

(6) Besides the M98 command, subprogram nesting is subject to the following commands:
   • G65 Macro call
   • G66 Modal call
   • G66.1   Modal call
   • G code call
   • Miscellaneous function call (M, S, T, etc.)
   • Macro interrupt
   • MDI interrupt
   • Automatic tool length measurement
   • Multi-stage skip function

(7) Subprogram nesting is not subject to the following commands which can be called even beyond the 8th nesting level.
   • Fixed cycles

(8) To repeatedly use the subprogram, it can be repeated $l1$ times by programming M98 Pp_1 Ll_1;.

Example of program

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

Main program	Sub program 1	Sub program 2	Sub program 3
	O1;	O10;	O20;

Sequence of execution :   (1) → (2) → (3) → (3)' → (2)' → (1)'
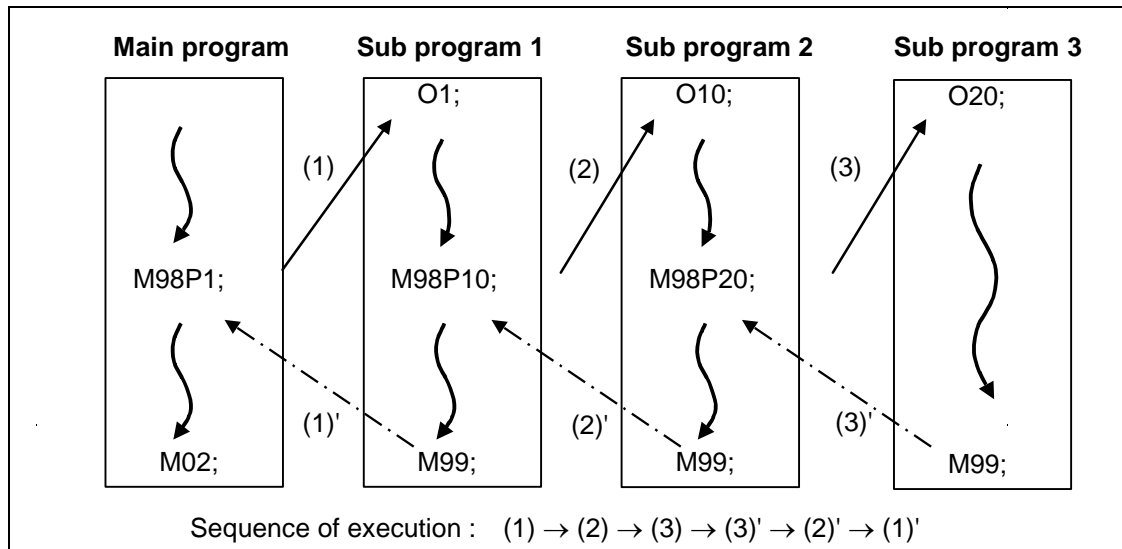
(1) For nesting, the M98 and M99 commands should always be paired off on a 1:1 basis, (1)' for (1), (2)' for (2), etc.
(2) Modal information can be rewritten according to the execution sequence without distinction between main programs and subprograms. This means that after calling a subprogram, attention must be paid to the modal data status when programming.

Example of program 2

The M98H__; M99P__; commands designate the sequence numbers in a program with a call instruction.

<table>
<tr><td>

For M98H__ ;

```
┌─ M98H3;
│   )
│
└─► N3___;
    )
  ─ M99;
```

</td><td>

For M99P__ ;

```
Search
┌─► N100___;
│   M98P123;  ────────►  O123;
│  ┐N200_;                )
│  │N300___;              )
│  ┘N400___;              )
                       M99P200;
```

</td></tr>
</table>
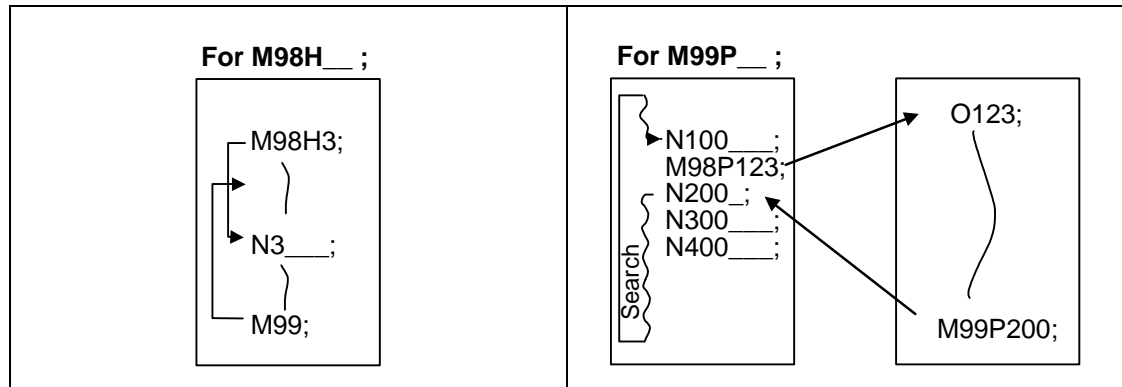
Other cautions

(1) Program error "P232" results when the designated program number (P) is not located.
(2) Single block stop does not occur with the M98P__; M99; block. If any address except O, N, P, L or H is used, single block stop can be executed. (With X100. M98 P100;, operation branches to O100 after X100. Is executed.)
(3) When M99 is commanded by the main program, operation returns to the head. (This is same for MDI.)
(4) Operation can branch from tape operation to a subprogram by M98P__ but the sequence number of the return destination cannot be designated with M99P__;, (P__ is ignored.)
(5) Bear in mind that the search operation will take time when the sequence number is designated by M99P__; .

13.3.2 Calling Subprogram with M198 Commands   [E68]

Function and purpose

Programs saved in the IC card can be called as subprograms. To call a program in the IC card as a subprogram, command in the main program as shown below.

Command format

Calling Subprogram

M198 P_ H_ L_ ;
P   Number of program in IC card to be called as subprogram. (Max. 8 digits)
H   Number of sequence in program to be called out. (Head block when omitted) (Max. 5 digits)
L   Number of subprogram repetitions. (Max. 4 digits)
• This can be omitted. (In this case, the subprogram will be called once.) When "L0" is designated, the subprogram call will not be executed.

Return to main program from subprogram

M99 ;

Detailed description

The subprograms that can be called from the main program are shown below. The subprogram can be called with the M198 command once in the subprogram nest.

Operation mode	**Sub** / **Main**	**Memory**	**IC card**	**Tape**
Memory and MDI mode	Memory	M98	M198	×
Tape mode	IC card	M98	×	×
	Tape	M98	×	×

(1) The section from the head of the program to the first LF (carriage return code, 0x0A hexadecimal) is invalid, and is not run or displayed. Note that if the head starts with a 0 number, the program will be valid from the head.

(2) To run a program saved on an IC card with the tape mode, set the number of characters in the program to 250 characters or more.

(3) It is necessary to operate in the memory mode or the MDI mode to call the program in IC card as a subprogram.

(4) When the subprogram is called by M198 command, which is the tape mode, an error will occur if there is the following macro statement in IC card.

   • Control commands ("IF-GOTO-" command and "WHILE-DO-" command) (When the macro statement is described, the program error (P295) will occur.)

   • External output commands (When the macro statement is described, the program error (P460) will occur.)

(5) The subprogram can be called once in the subprogram nest because the M198 command is IC card operation (tape mode). If the subprogram is called twice in nest, the program error (P230) will occur.

(6)   When the M198 is commanded without mounting an IC card, the program error (P460) will occur.

(7)   When the additional H/W HR551 • HR253 is not mounted, the program error (P460) will occur if M198 is commanded.

(8)   When the additional HR751 is not mounted, the program error (P460) will occur if M198 is commanded.

13.4 Variable commands

Function and purpose

Programming can be endowed with flexibility and general-purpose capabilities by designating variables, instead of giving direct numerical values to particular addresses in a program, and by assigning the values of those variables as required when executing a program.

Command format

#ΔΔΔ = ○○○○○○○○ or #ΔΔΔ = [formula]

Detailed description

(1) Variable expressions   Example

(a) #m	m = value consisting of 0 to 9	#100
(b) #[f]	f = one of the following in the formula	#[-#120[
	Numerical value m	123
	Variable	#543
	Formula operator formula	#110+#119
	- (minus) formula	-#120
	[Formula]	[#119[
	function [formula]	SIN[#110[

(Note 1) The 4 standard operators are +, -, ∗ and /.
(Note 2) Functions cannot be used unless the user macro specifications are available.
(Note 3) Error "P241" results when a variable number is negative.
(Note 4) Examples of incorrect variable expressions are given below.

Incorrect		Correct
#6/2	→	#[6/2] (Note that expression such as "#6/2" is regarded as "[#6] /2")
#- -5	→	#[- [-5]]
#- [#1]	→	#[-#1]

(2) Type of variables

The following table gives the types of variables.

Type of variable	Number		Function
Common variables 200 sets	Common variables 1	Common variables 2	Can be used in common throughout main, sub and macro programs.
	500 to 599	100 to 199	
Local variables	1 to 33		Can be used for local variables in macro programs.
System variables	1000 to		Application is fixed by system.
Fixed cycle variables	1 to 32		Local variables in fixed cycle programs.

(Note 1) All common variables are retained even when the power is switched off.

(Note 2) When the power is turned off or reset, the common variables can be set to <null> by setting the parameter (#1128 RstVC1, #1129 PwrVC1).

(3) Variable quotations

Variables can be used for all addresses except O, N and / (slash).

(a) When the variable value is used directly:

X#1 ....Value of #1 is used as the X value.

(b) When the complement of the variable value is used:

X - #2................................ Value with the #2 sign changed is used as the X value.

(c) When defining variables:

#3 = #5 ....Variable #3 uses the equivalent value of variable #5.

#1 = 1000 ....Variable #1 uses the equivalent value 1000 (which is treated as 1000.)

(d) When defining variables:

#1 = #3 + #2 - 100 ....The value of the arithmetic result of #3 + #2 - 100. Is used as the #1 value.

X[#1 + #3 + 1000].............. The value of the arithmetic result of #1 + #3 + 1000. Is used as the X value.

(Note 1) A variable cannot be defined in the same block as an address. It must be defined in a separate block.

      Incorrect   Correct

      X#1 = #3 + 100 ;   →   #1 = #3 + 100 ;

                                      X#1 ;

(Note 2) Up to five sets of square parentheses [ ] may be used.

     #543 = - [[[[[#120]/2+15.]∗3 - #100]/#520 + #125 + #128] ∗#130 + #132]

(Note 3) There are no restrictions on the number of characters and number of variables for variable definition.

(Note 4) The variable values should be within a range form 0 to ±99999999. If this range is exceeded, the arithmetic operations may not be conducted properly.

(Note 5) The variable definitions are valid from the moment that the variables are actually defined.

    #1 = 100 ;.............................. #1 = 100 Valid from the next command

    #1 = 200 #2 = #1 + 200 ; ..... #1 = 200, #2 = 400 Valid from the next command

    #3 = #1 + 300 ; .... #3 = 500 Valid from the next command

(Note 6) Variable quotations are always regarded as having a decimal point at the end. When #100 = 10, then X#100 ; is treated as X10.

13.5 User macro specifications

13.5.1 User macro commands ;G65,G66,G66.1,G67
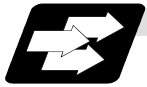
Function and purpose

By combining the user macros with variable commands, it is possible to use macro program call, arithmetic operation, data input/output with PLC, control, decision, branch and many other instructions for measurement and other such applications.

```
O Main program                          O Macro program

....... ;                               ....... ;


Macro call instruction


M30 ;                                   M99 ;
```
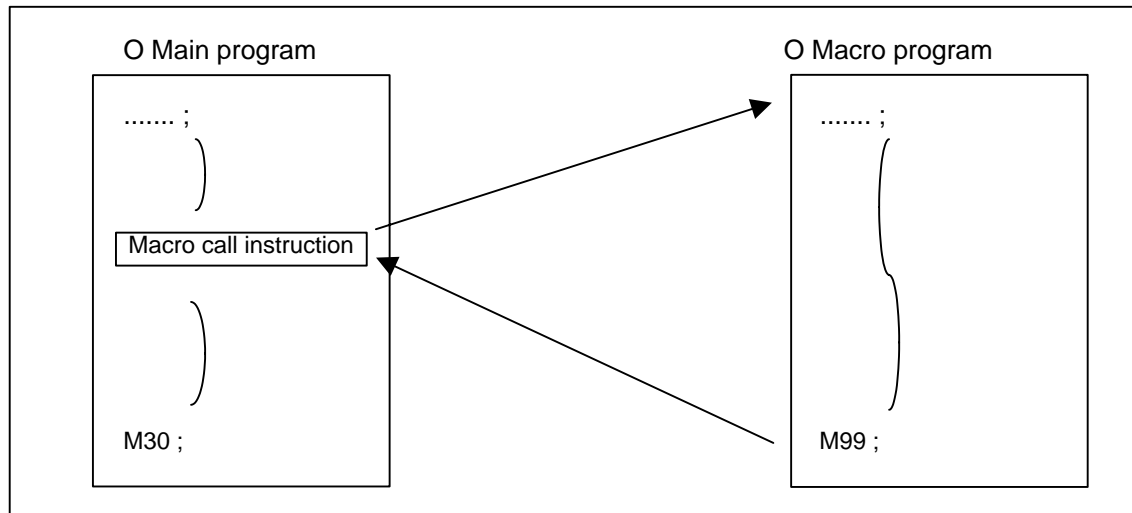
Macro programs use variables, arithmetic instructions and control instructions to create subprograms which function to provide special-purpose control.
These special-purpose control functions (macro programs) are called by the macro call instructions exactly when required from the main program.
The following G codes are available for the macro call commands.

G code	Function
G65	User macro Simple call
G66	User macro Modal call A (Movement command call)
G66.1	User macro Modal call B (Per-block call)
G67	User macro Modal call cancel

Detailed description

(1) When the G66 (or 66.1) command is entered, the specified user macro subprogram will be called after each block has been executed (or after the movement command in the block) with the movement commands has been executed until the G67 (cancel) command is entered.

(2) The G66 (or G66.1) and G67 commands must be paired in the same program.

13.5.2 Macro call instruction
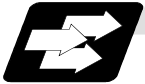
Function and purpose

Included among the macro call commands are the simple calls which apply only to the instructed block and also modal calls (types A and B) which apply to each block in the call modal.

Simple macro calls

Main program	To subprogram	Subprogram (O_{o1})

O_{o1}

G65Pp1L*l*   <argument>;

M99

To main program

M99 is used to conclude the user macro subprogram.
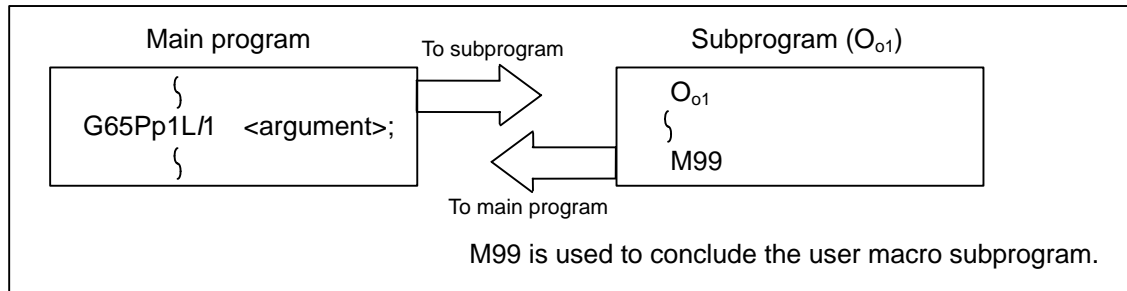
Format

```
G65  P___  L___   <argument> ;
P___                          : Program No.
L___                          : No. of repetitions
```

When the <argument> must be transferred as a local variable to a user macro subprogram, the actual value should be designated after the address.
Regardless of the address, a sign and decimal point can be used in the argument. There are 2 ways in which arguments are designated.

(1)　Argument designation I

Format : A__ B__ C__ ..... X__ Y__ Z__

Detailed description

(a)　Arguments can be designated using any address except G, L, N, O and P.

(b)　Except for I, J and K, there is no need for designation in alphabetical order.

(c)　I, J and K must be designated in alphabetical order.

　I__ J__ K__ .... Correct
　J__ I__ K__ .... Incorrect

(d)　Address which do not need to be designated can be omitted.

(e)　The following table shows the correspondence between the addresses which can be designated by argument designation I and the variable numbers in the user macro main body.

Address and variable number correspondence		Call instructions and usable address	
Argument designation I address	Variable in macro	G65, G66	G66.1
A	#1	◯	◯
B	#2	◯	◯
C	#3	◯	◯
D	#7	◯	◯
E	#8	◯	◯
F	#9	◯	◯
G	#10	✕	✕*
H	#11	◯	◯
I	#4	◯	◯
J	#5	◯	◯
K	#6	◯	◯
L	#12	✕	✕*
M	#13	◯	◯
N	#14	✕	✕*
O	#15	✕	✕
P	#16	✕	✕*
Q	#17	◯	◯
R	#18	◯	◯
S	#19	◯	◯
T	#20	◯	◯
U	#21	◯	◯
V	#22	◯	◯
W	#23	◯	◯
X	#24	◯	◯
Y	#25	◯	◯
Z	#26	◯	◯

◯ : Can be used.

✕ : Cannot be used.

∗ : Can be used while G66.1 command is modal.

(2) Argument designation II

Format : A__ B__ C__ I__ J__ K__ I__ J__ K__ ....

Detailed description
(a) In addition to address A, B and C, up to 10 groups of arguments with I, J, K serving as 1 group can be designated.
(b) When the same address is duplicated, designate the addresses in the specified order.
(c) Addresses which do not need to be designated can be omitted.
(d) The following table shows the correspondence between the addresses which can be designated by argument designation II and the variable numbers in the user macro main body.

Argument designation II address	Variable within macro	Argument designation II address	Variable within macro
A	# 1	J5	#17
B	# 2	K5	#18
C	# 3	I6	#19
I1	# 4	J6	#20
J1	# 5	K6	#21
K1	# 6	I7	#22
I2	# 7	J7	#23
J2	# 8	K7	#24
K2	# 9	I8	#25
I3	#10	J8	#26
J3	#11	K8	#27
K3	#12	I9	#28
I4	#13	J9	#29
J4	#14	K9	#30
K4	#15	I10	#31
I5	#16	J10	#32
		K10	#33

(Note 1) The numbers 1 through 10 accompanying I, J and K denote the sequence of the commanded groups and they are not required for the actual instructions.

(3) Using arguments designations I and II together

If addresses corresponding to the same variable are commanded when both types I and II are used to designate arguments, the latter address is valid.

(Example 1)

```
Call instruction    G65    A1.1    B-2.2    D3.3    I4.4    I7.7   ;
  Variable
#1 :   1.1
#2 :  -2.2
#4 :   4.4
#5 :
#6 :
#7 :   ✗                                              7.7
```
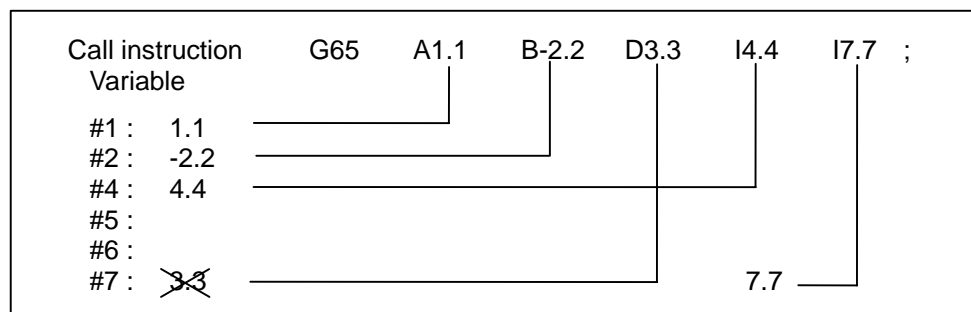
In the above example, the last I7.7 argument is valid when both arguments D3.3 and I7.7 are commanded for the #7 variable.

Modal call A (movement command call)
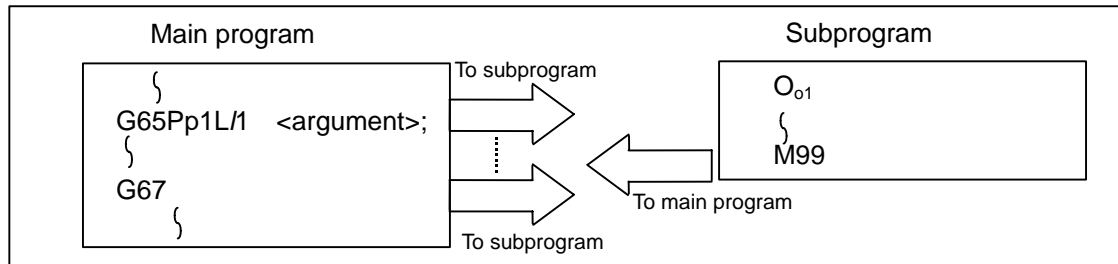
When the block with a movement command is commanded between G66 and G67, the movement command is first executed and then the designated user macro subprogram is executed. The number of times the subprogram is executed is $\lambda 1$ times with each call.
The <argument> is the same as for a simple call.

Format

G66 P___ L___ <argument> ;	
P___	: Program No.
L___	: No. of repetitions

Detailed description
(1) When the G66 command is entered, the specified user macro subprogram will be called after the movement command in the block with the movement commands has been executed until the G67 (cancel) command is entered.
(2) The G66 and G67 commands must be paired in the same program.
A program error will result when G67 is issued without the G66 command.

(Example) Drill cycle
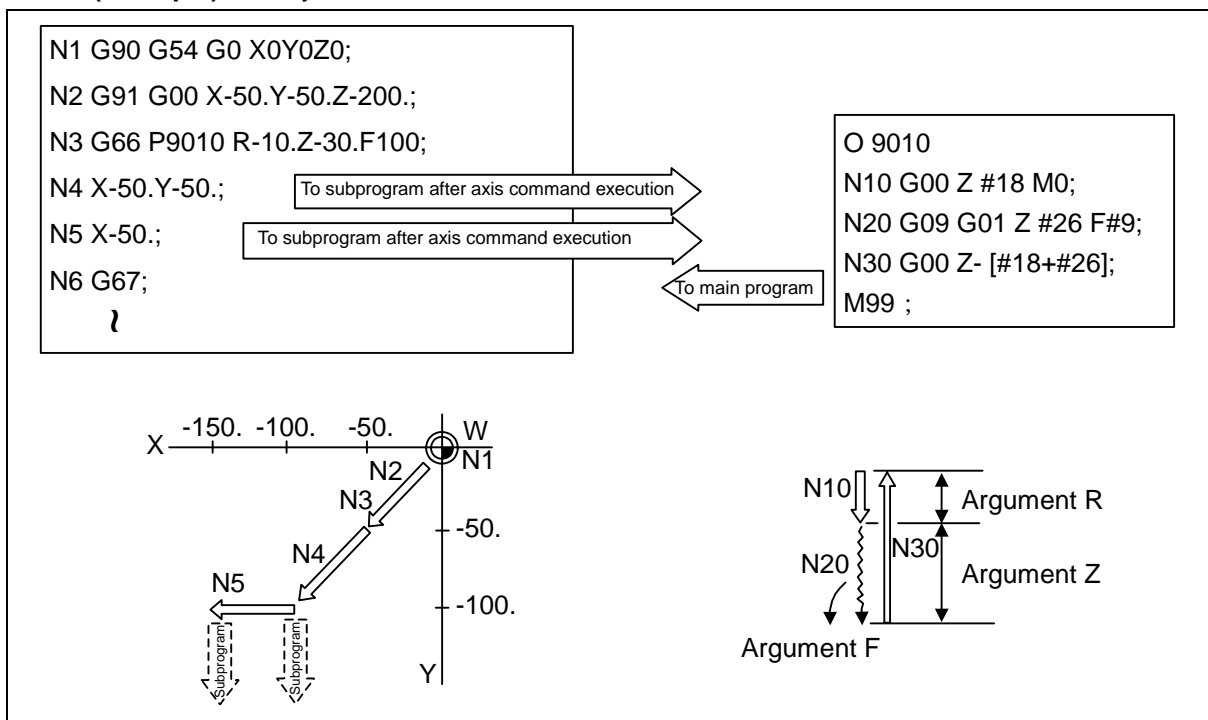
(Note 1) After the axis command is executed in the main program, the subprogram is executed.
(Note 2) The subprogram is not executed in the blocks following G67.

13. Program Support Functions

Modal call B (for each block)

The specified user macro subprogram is called unconditionally for each command block which is assigned between G66.1 and G67 and the subprogram is executed the specified number of times.
Format

G66.1 P___ L___ <argument> ;	
P___	: Program No.
L___	: No. of repetitions

Detailed description
(1) In the G66.1 mode, everything except the O, N and G codes in the various command blocks which are read are handled as the argument without being executed. Any G code designated last or any N code commanded after anything except O and N will function as the argument.
(2) The same applies as when G65P__ is assigned at the head of a block for all significant blocks in the G66.1 mode.

(Example 1)

N100 G01 G90 X100. Y200. F400 R1000; in the G66.1 P1000; mode is the same as: N100 G65 P1000 G01 G90 X100. Y200. F400 R1000;

(Note 1)   The Call is performed even in the G66.1 command block in the G66.1 mode and the correspondence between the argument address and the variable number is the same as for G65 (simple call).

(3) The range of the G and N command values which can be used anew as variables in the G66.1 mode is subject to the restrictions applying to values as normal NC command values.
(4) Program number O, sequence numbers N and modal G codes are updated as modal information.

G code macro call

User macro subprogram with prescribed program numbers can be called merely by issuing the G code command.

Format

G ** **argument ;**	
G**	:G code for macro call

Detailed description
(1) The above instruction functions in the same way as the instructions below, and parameters are set for each G code to determine the correspondence with the instructions.
   a. M98PΔΔΔΔ ;
   b. G65PΔΔΔΔΔ <argument> ;
   c. G66P ΔΔΔΔΔ <argument> ;
   d. G66.1PΔΔΔΔΔ <argument> ;
When the parameters corresponding to c and d above are set, issue the cancel command (G67) either in the user macro or after the call code has been commanded so as to cancel the modal call.

(2) The correspondence between the "**" which conducts the macro call and the program number PΔΔΔΔ of the macro to be called is set by parameter.

(3) Up to 10 G codes from G100 to G255 can be used with this instruction. (G01 to 99 can also be used with parameter #1081 Gmac_P).

(Note 1)   G101 to G110 and G200 to G202 are user macro I codes, but if the parameters are set as the G code call codes, the G code call will be the priority, and these codes cannot be used for user macro I.

(4) These commands cannot be issued during a user macro subprogram which has been called by a G code.

```
Program example                                              O9016

    G16X100. Y100. Z100. F500 ; ————————————→    ⎞
                 ⎰                                            ⎛
                 ⎱              ←————————————                 ⎝
                                                       M99 ;
```
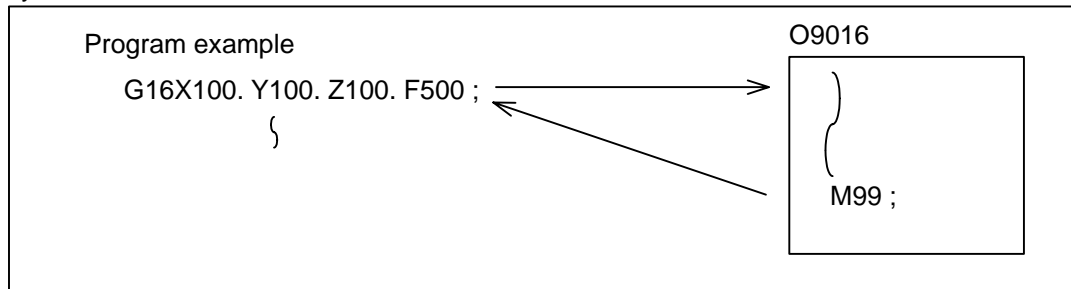
Miscellaneous command macro call (for M, S, T, B code macro call)

The user macro subprogram of the specified program number can be called merely by issuing an M (or S, T, B) code. (Only entered codes apply for M but all S, T and B codes apply.)

Format

M ; (or S** ;, T** ;, B** ;)
M**   M code for macro call (or S, T, B code)

Detailed description

(1) The above instruction functions in the same way as the instructions below, and parameters are set for each M code to determine the correspondence with the instructions. (Same for S, T and B codes)

a :   M98 PΔΔΔΔ ;   ⎫
b :   G65 PΔΔΔΔ M** ;   ⎬ M98, M** are not output
c :   G66 P ΔΔΔΔ M** ;
d :   G66. 1PΔΔΔΔ M** ;

When the parameters corresponding to c and d above are set, issue the cancel command (G67) either in the user macro or after the call code has been commanded so as to cancel the modal call.

(2) The correspondence between the "M**" which conducts the macro call and the program number PΔΔΔΔ of the macro to be called is set by parameter. Up to 10 M codes from M00 to M95 can be entered. Note that the codes to be registered are the codes basically required for the machine, and codes excluding M0, M1, M2, M30 and M96 to M99.

(3) As with M98, it is displayed on the screen display of the setting and display unit but the M codes and MF are not output.

(4) Even if the miscellaneous command entered above is issued during a user macro subprogram called by the M code, macro call will not result and it will be handled as an ordinary miscellaneous command.

(5) All S, T and B codes call the subprograms in the prescribed program numbers of the corresponding S, T and B functions.

(6) A maximum of 10 M codes can be set. However when not setting all 10. Set the parameters as shown below.

```
[ MACRO ]

          <Code> <Type> <Program No.>
   M [01]    20      0        8000      ········ Setting to call O8000 with type 0
   M [02]    21      0        8001               (M98 type) during M20 command
   M [03]  9999      0    199999999  ┐
   M [04]  9999      0    199999999  │           Setting to call O8001 with type 0
   M [05]  9999      0    199999999  │           (M98 type) during M21 command
            :        :        :      │
            :        :        :      │           Set parameters not being used as
   M [10]  9999      0    199999999  ┘           shown on left.
```

Differences between M98 and G65 commands

(1) The argument can be designated for G65 but not for M98.
(2) The sequence number can be designated for M98 but no for G65, G66 and G66.1.
(3) M98 executes a subprogram after all the commands except M, P, H and L in the M98 block have been executed, but G65 branches to the subprogram without any further operation.
(4) When any address except O, N, P, H or L is included in the M98 block, single block stop results. This is not the case with G65.
(5) The level of the M98 local variables is fixed but it can be varied in accordance with the nesting depth for G65. (#1, for instance, has the same significance either before or after M98 but a different significance in each case with G65.)
(6) The M98 nesting depth extends up to 8 levels in combination with G65, G66 and G66.1. The G65 nesting depth extends up to only 4 levels in combination with G66 and G66.1.

Macro call command nesting depth

Up to 4 nesting levels are available for macro subprogram calls based on simple call or modal call. The argument with a macro call instruction is valid only on the called macro level. Since the nesting depth for macro calls extends up to 4 levels, the argument can be used as a local variable for the program with each respective macro call.

(Note 1)   When a G65, G66, G66.1 G code macro call or miscellaneous command macro call is conducted, this is regarded as nesting level 1 and the level of the local variables is also incremented by one.

(Note 2)   The designated user macro subprogram is called every time the movement command is executed with modal call A. However, when the G66 command has been duplicated, the next user macro subprogram is called every time an axis is moved even with movement commands in the macro.
                 User macro subprograms are called in sequence from the subprogram commanded last.

(Example 1)

Main program

User macro operation

G66Pp₁; (p₁ call) — Macro p₁
Zz1 ; After Z1 execution → x₁ → y₁ → x₂ → M99

G66Pp₂; (p₂ call) — Macro p₂
Zz2 ; After Z2 execution → x₁ → y₁ → x₂ → M99
G67 ; (p₂ cancel) — Macro p₁, Macro p₁, Macro p₁

Zz3 ; After Z3 execution — Macro p₁ → x₁ → y₁ → x₂ → M99
G67 ; (p₁ cancel)

Zz4 ;
Zz5 ;

228
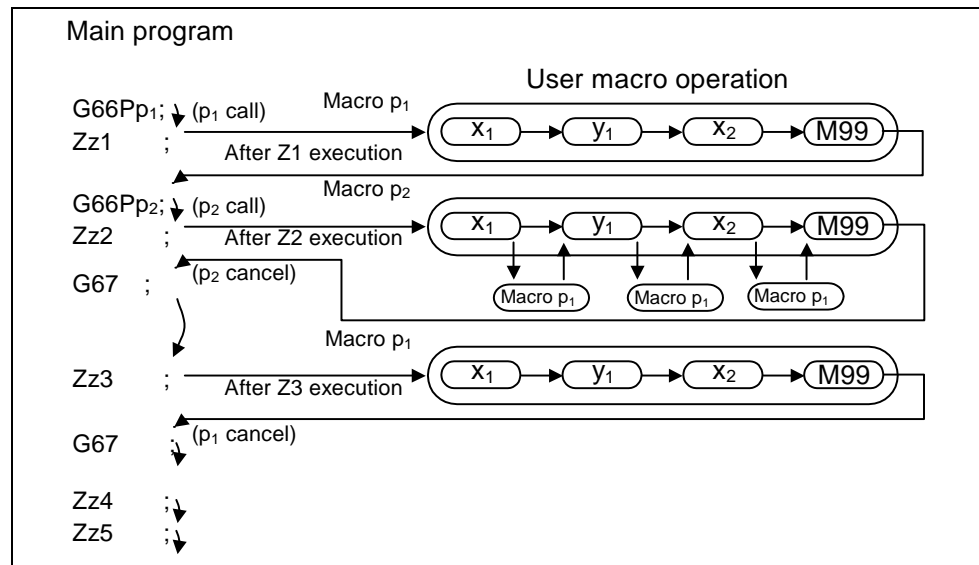
13.5.3　ASCII Code Macro
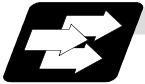
Function and purpose

A macro program can be called out by setting the correspondence of a subprogram (macro program) preregistered with the parameters to codes, and then commanding the ASCII code in the machining program.
This function can be used in addition to the G, M, S, T and B miscellaneous command macro call function.

(Execution example 1) M98 type

```
        Main program              Subprogram

     ┌──────────────┐          ┌──────────────┐
     │ O0002 ;      │      ┌──▶│ O200         │
     │    :         │      │   │   :          │
     │ D2000 ;  ◀───┼──────┤   │   :          │
     │    :         │      │   │   :          │
     │ M30 ;        │      └───┤ M99 ;        │
     └──────────────┘          └──────────────┘
```

After outputting 2000 to common variable #146, the
program No. 200 subprogram is called with the
M98 subprogram call type.

```
┌─ Parameter ─────────────────────────────────────────┐
│ #7401 (ASCII [01] Valid/Invalid)      1 (Valid)      │
│ #7402 (ASCII [01] Code)               D              │
│ #7403 (ASCII [01] Type)               0 (M98 type)   │
│ #7404 (ASCII [01] Program No.)        200            │
│ #7405 (ASCII [01] Variable)           146            │
└──────────────────────────────────────────────────────┘
```
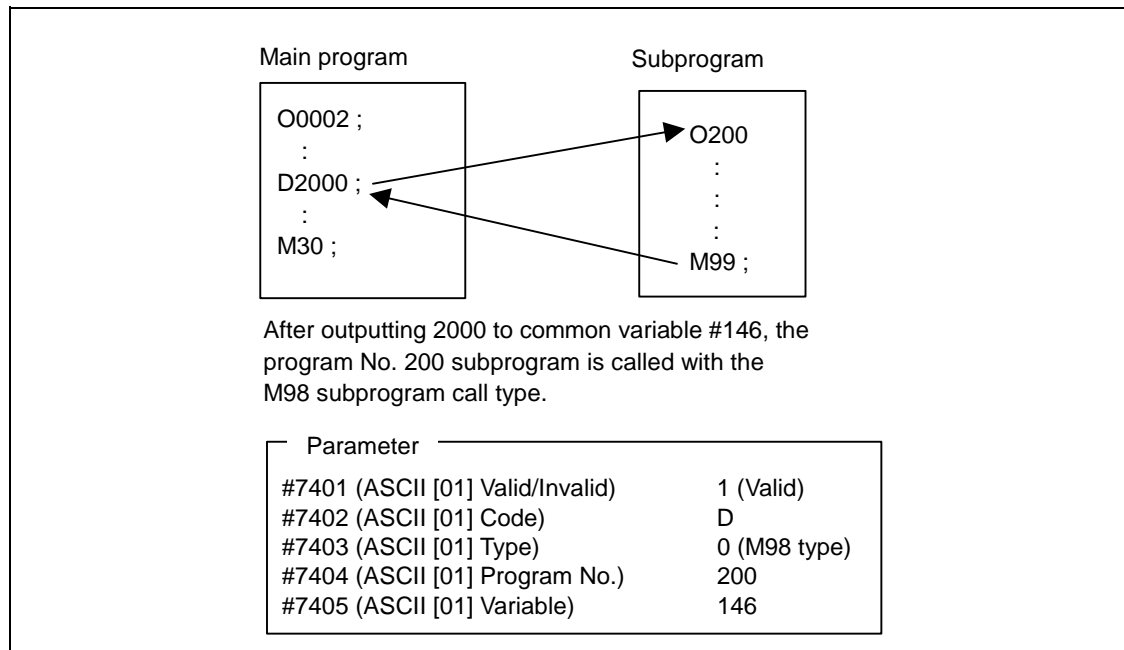
(Execution example 2) G65 type

```
        Main program              Subprogram

     ┌──────────────┐          ┌──────────────┐
     │ O0003 ;      │      ┌──▶│ O3000        │
     │    :         │      │   │   :          │
     │ A500 ;  ◀────┼──────┤   │   :          │
     │    :         │      │   │   :          │
     │ M30 ;        │      └───┤ M99 ;        │
     └──────────────┘          └──────────────┘
```

After outputting 500 to local variable #1, the
program No. 3000 subprogram is called out with
the G65 macro call type.

```
┌─ Parameter ─────────────────────────────────────────┐
│ #7411 (ASCII [02] Valid/Invalid)      1 (Valid)      │
│ #7412 (ASCII [02] Code)               A              │
│ #7413 (ASCII [02] Type)               0 (G65 type)   │
│ #7414 (ASCII [02] Program No.)        3000           │
│ #7415 (ASCII [02] Variable)           100 (Not used) │
└──────────────────────────────────────────────────────┘
```
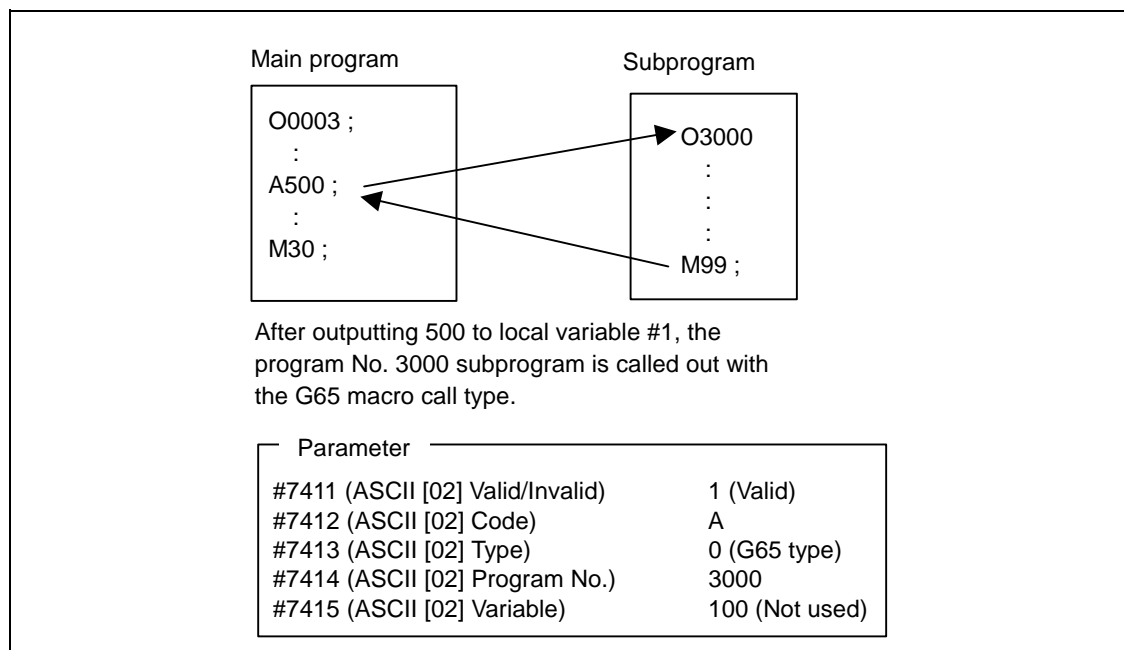
13. Program Support Functions

Command format

□****;	**Designates the address and code**
□	ASCII code for calling out macro (one character)
****	Value or expression output to variable
	(Setting range: ±999999.9999)

Detailed description

(1) The command above functions in the same way as that below. The correspondence of commands is set for each ASCII code with the parameters.

　0: M98 P△△△△ ;
　1: G65 P△△△△ <Argument> ;
　2: G66 P△△△△ <Argument> ;
　3: G66.1 P△△△△ <Argument> ;

When parameters are set for 2 and 3 above, the modal call will be canceled. Thus, command the cancel command (G67) after commanding the call code or the during the user macro.

(2) The ASCII code for calling the macro with the ASCII codes and the program No. P△△△△ to be called are set with the parameters.
Up to two ASCII codes can be registered.

(3) The code section is output to the variables, but the output destination differs according to the call type and address.

(a) For M98 type
The code section is output to the common variable, and the variable No. is set with the parameters.
When corresponding to the first address (parameter #7401), the section is output to the common variable that indicates the first variable No. (parameter #7404).

(b) For G65/G66/G66.1 type
The code section is output to the local variable. The variable No. differs according to the address, and corresponds to the following table.

Address	#	Address	#	Address	#
A	1	K	6	U	21
B	2	L	12	V	22
C	3	M	13	W	23
D	7	N	14	X	24
E	8	O	15	Y	25
F	9	P	16	Z	26
G	10	Q	17		
H	11	R	18		
I	4	S	19		
J	5	T	20		

(Note) The following addresses can be used.

A, B, D, F, H, I, J, K, M, Q, R, S, T

Restrictions

(1) Calling a macro with an ASCII code from a program macro-called with an ASCII code

A macro cannot be called with an ASCII code from a program macro-called with an ASCII code.
The other patterns are shown below.
If it is determined that the macro cannot be called, the command will be handled as a normal command.

		Called side			
		ASCII	**GMSTB macro**	**G65/66/66.1**	**M98**
Calling side	**ASCII**	×	×	○	○
	GMSTB macro	×	×	○	○
	G65/66/66.1	○	○	○	○
	M98	○	○	○	○

(2) Nest level of macro call command

The macro subprogram can be called in up to four levels using simple call (G65) and modal call (G66/G66.1).
The macro call command's argument is valid only in the called macro level.
Since the macro call nest level is four levels, the argument for each macro call can be used in the program as a local variable.

(3) Nest level of subprogram call

Counting the main program as 0, up to eight levels of subprograms can be called (M98) from a subprogram.
The following commands are used for subprogram nesting.

(a) M98

(b) G65 G66 G66.1

(c) G code call Miscellaneous function call (M/S/T/B)

(d) MDI interruption

(e) Automatic tool length measurement

(f) Multiple-step skip function

The following commands can be commanded regardless of nesting.

(g) Fixed cycle

(h) Macro interruption

(4) Order of command priority

If "M" is designated for the ASCII code address, the codes basically necessary for that machine will be overlapped. In this case, commands will be identified with the following priority using code values.

- (a) M98, M99 (subprogram call command)
       M00 (program stop command), M01 (optional stop command)
       M02, M30, M198, M199 (end command)
       M96, M97 (macro interruption command)
- (b) When corresponding to miscellaneous code (M) call macro command
- (c) When corresponding to ASCII code macro command
- (d) Used as normal command

If "S", "T" and "B" are designated for the ASCII code address, commands will be identified with the following priority using code values.

- (a) When corresponding to miscellaneous code (S, T, B) call macro command
- (b) When corresponding to ASCII code macro command
- (c) Used as normal command

If the other addresses do not correspond to the ASCII code macro command, they will be identified as normal commands. If the command to be used overlaps with an ASCII code macro command, it must be commanded in the program macro-called with the ASCII code. Note that there are cases when the command will be unconditionally handled as a normal command, as explained in (5) below.

(5) Conditions for handling address set in ASCII code macro command as normal command

- (a) When there is a data setting command (G10) in the same block.
- (b) When ASCII code macro call is executed after the G code macro call command in the same block (also applies for M, S, T, B and ASCII)

    Example) When address "D" (G65 type) is set in the ASCII code macro, and M50 is set in the macro call (G65 type).

    | M50 D200 ;   Execute M code macro with argument (200 set in #7) |

- (c) When inputting parameters
- (d) When there is a comma (,) before the address. **Example)** ,D ,R, etc.
- (e) When commanded in fixed cycle
- (f) When commanded in macro subprogram called with G code macro call
       (Also applies when macro is called with M, S, T, B or ASCII)

13.5.4 Variables
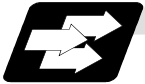
Function and purpose

Both the variable specifications and user macro specifications are required for the variables which are used with the user macros.

The offset amounts of the local, common and system variables among the variables for this MELDAS NC system except #33 are retained even when the unit's power is switched off. (Common variables can also be cleared by parameter #1129 PwrVC1.)

Use of multiple variables

When the user macro specifications applied, variable numbers can be turned into variables (multiple use of variables) or replaced by <formula>. Only one of the four basic arithmetic rule (+, -, ×, ÷) operations can be conducted with <formula>.

(Example 1) Multiple use of variables

#1 = 10 #10 = 20 #20 = 30 ; #5 = #[#[#1]] ;	#[#[#1]] = #[#10] from #1 = 10. #[#10] = #20 from #10 = 20. Therefore, #5 = #20 or #5 = 30.

#1 = 20 #10 = 20 #20 = 30 #5 = 1000 ; #[#[#1]] = #5 ;	#[#[#1]] = #[#10] from #1 = 10. #[#10] = #20 from #10 = 20. Therefore, #20 = #5 or #20 = 1000.

(Example 2) Example of multiple designation of variables

#10 = 5 In which case ##10 = 100 ; #5 = 100	<Formula>##10 = 100; is handled in the same manner as # [#10] = 100.

(Example 3) Replacing variable numbers with <formula>

#10 = 5 ; #[#10 + 1] = 1000 ;	In which case, #6 = 1000.
#[#10 - 1] = -1000 ;	In which case, #4 = -1000.
#[#10∗3] = 100 ;	In which case, #15 = 100.
#[#10/2] = -100 ;	In which case, #3 = -100. (Rounded off.)

Undefined variables

Variables applying with the user macro specifications such as variables which have not been used even once after the power was switched on or local variables not quoted by the G65, G66 or G66.1 commands can be used as <vacant>. Also, variables can forcibly be set to <vacant>. Variable #0 is always used as the <vacant> variable and cannot be defined in the left-side member.

(1)   Arithmetic expressions

#1 = #0 ; · · · · · · · · · · #1 = <vacant>
#2 = #0 + 1 ; · · · · · · · #2 = 1
#3 = 1 + #0 ; · · · · · · · #3 = 1
#4 = #0∗10 ; · · · · · · · #4 = 0
#5 = #0 + #0 ; · · · · · · #5 = 0

> It should be borne in mind that <vacant> in an arithmetic expression is handled in the same way as 0.
>
> <Vacant> + <Vacant> = 0
>
> <Vacant> + <Constant> = Constant
>
> <Constant> + <Vacant> = Constant

(2)   Variable quotations

When undefined variables only are quoted, they are ignored up to the address.
When #1 = <Vacant>
G0 X#1 Y1000 ; · · · · · · · · Equivalent to G0 Y1000 ;
G0 X#1 + 10 Y1000 ; · · · Equivalent to G0 X10 Y1000 ;

(3)   Conditional expressions

<Vacant> and 0 are not equivalent for EQ and NE only. (#0 means <vacant>.)

When #101 = <Vacant>	When #101 = 0
#101 EQ #0 <Vacant> = <Vacant> established	#101 EQ #0 0 = <Vacant> not established
#101 NE 0 <Vacant> ≠ 0 established	#101 NE 0 0 ≠ 0 not established
#101 GE #0 <Vacant> ≥ <Vacant> established	#101 GE #0 0 ≥ <Vacant> established
#101 GT 0 <Vacant> > 0 not established	#101 GT 0 0 > 0 not established
#101 LE #0 <Vacant> ≤ <Vacant> established	#101 LE #0 0 ≤ <Vacant> established
#101 LT 0 <Vacant> < 0 not established	#101 LT 0 0 < 0 not established

(Note 1) EQ and NE should be used only for integers. For comparison of numeric values with decimals, GE, GT, LE, and LT should be used.

13.5.5 Types of variables

Common variables

Common variables can be used commonly from any position. Number of the common variables sets depends on the specifications. Refer to "13.4 Variable commands" for details.

Local variables (#1 to #33)

These can be defined as an <argument> when a macro subprogram is called or used locally within main programs and subprograms. They can be duplicated regardless of the relationship existing between macros (up to 4 levels).

G65 Pp$_1$ Ll_1 <argument> ;	
P$_1$	: Program number
l_1	: Number of repetitions

The <argument> is assumed to be Aa1 Bb1 Cc1 .... Zz1.

The following table shows the correspondences between the addresses designated by <argument> and the local variable numbers used in the user macro main bodies.

[Argument specification I]

Call command		Argument address	Local variable number		Call command		Argument address	Local variable number
G65 G66	G66.1				G65 G66	G66.1		
○	○	A	#1		○	○	Q	#17
○	○	B	#2		○	○	R	#18
○	○	C	#3		○	○	S	#19
○	○	D	#7		○	○	T	#20
○	○	E	#8		○	○	U	#21
○	○	F	#9		○	○	V	#22
×	×*	G	#10		○	○	W	#23
○	○	H	#11		○	○	X	#24
○	○	I	#4		○	○	Y	#25
○	○	J	#5		○	○	Z	#26
○	○	K	#6				-	#27
×	×*	L	#12				-	#28
○	○	M	#13				-	#29
×	×*	N	#14				-	#30
×	×	O	#15				-	#31
×	×*	P	#16				-	#32
							-	#33

"×" in the above table denotes an argument address which cannot be used. However, provided that the G66.1 mode has been established, an argument address denoted by the asterisk can be added for use.

"-" denotes that a corresponding address is not available.

[Argument specification II]

Argument specification II address	Variable in macro	Argument specification II address	Variable in macro
A	# 1	I6	#19
B	# 2	J6	#20
C	# 3	K6	#21
I1	# 4	I7	#22
J1	# 5	J7	#23
K1	# 6	K7	#24
I2	# 7	I8	#25
J2	# 8	J8	#26
K2	# 9	K8	#27
I3	#10	I9	#28
J3	#11	J9	#29
K3	#12	K9	#30
I4	#13	I10	#31
J4	#14	J10	#32
K4	#15	K10	#33
I5	#16		
J5	#17		
K5	#18		

(Note 1) Subscripts 1 to 10 for I, J, and K indicate the order of the specified command sets. They are not required to specify instructions.

(1) Local variables in subprograms can be defined by means of the <argument> designation during macro call.

Main program

```
G65 P9900 A60. S100. F800;

M02;
```

To subprogram

Subprogram (9900)

```
G91 G01 X [#19∗COS [#1] ]
        Y [#19∗SIN [#1] ] F#9;

M99;
```

Refer to the local variables and control the movement, etc.

Local variables set by argument

Local variable data table

```
A(#1)=    60.000

F(#9)=       800

S(#19)= 100.000
```
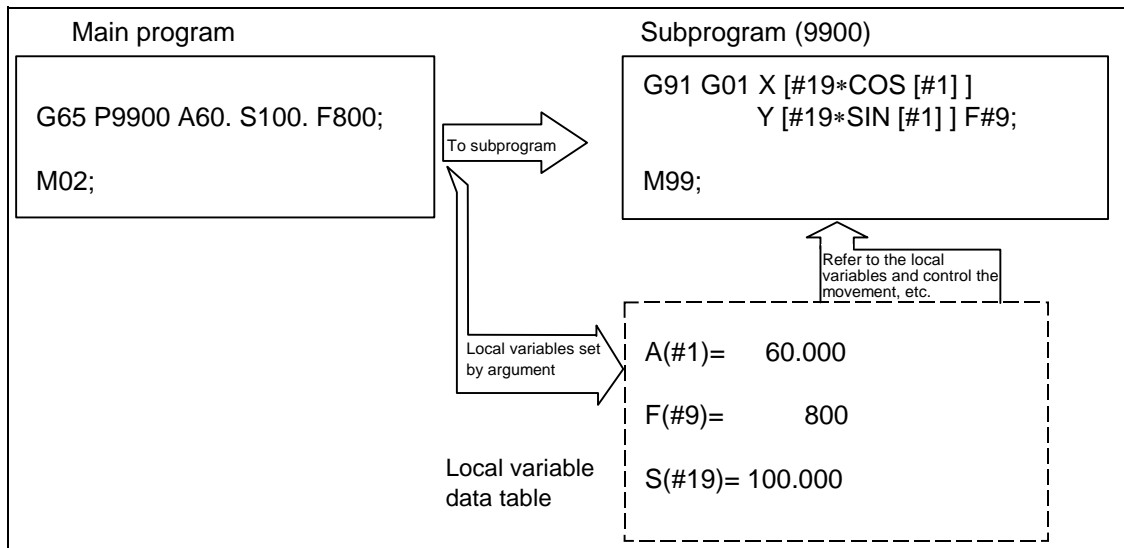
(2)   The local variables can be used freely in that subprogram.

Main program

G65 P1 A100. B50. J10. F500;

To subprogram

Subprogram (1)

#30=FUP [#2/#5/2]   ;

#5=#2/#30/2 ;

M98 H100 L#30 ;

X#1 ;

M99 ;

N100 G1 X#1 F#9 ;

Y#5 ;

X-#1 ;

Y#5 ;

M99 ;

Example of front surface milling

Local variables set by argument

The local variables can be changed in the subprogram.

Local variable data table

```
A   (#1)   100.000
B   (#2)    50.000
F   (#9)       500
J   (#5)    10.000 → 8.333
    (#30)          →     3
```
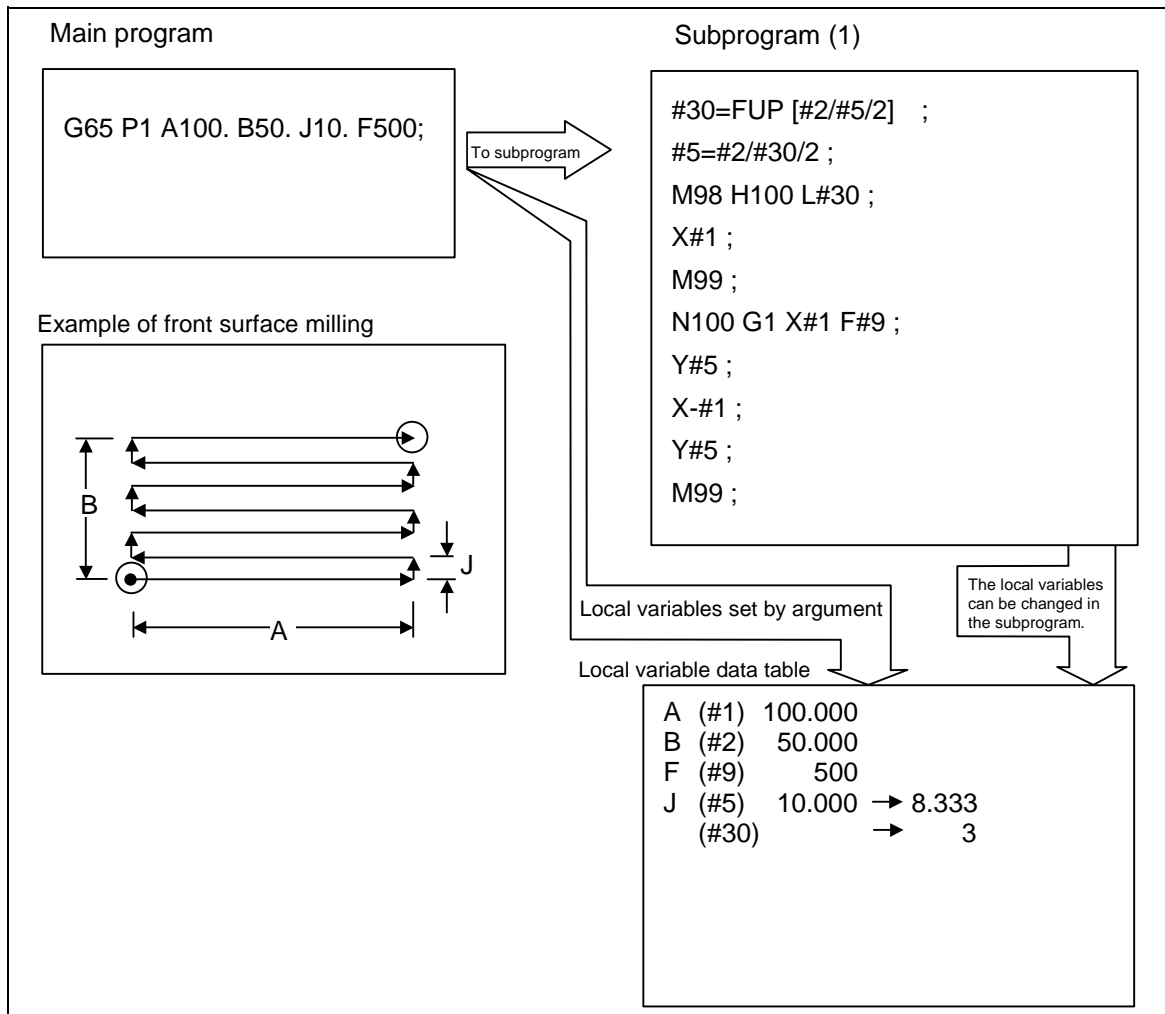
In the front surface milling example, argument J is programmed as the milling pitch 10.mm.
However, this is changed to 8.333mm to create an equal interval pitch.
The results of the No. of reciprocation data calculation is set in local variable #30.

(3) Local variables can be used independently on each of the macro call levels (4 levels).
Local variables are also provided independently for the main program (macro level 0).
Arguments cannot be used for the level 0 local variables.

Main (level 0)	O1 (macro level 1)	O10 (macro level 2)	O100 (macro level 3)
#1=0.1 #2=0.2 #3=0.3; G65 P1A1. B2. C3.; M02;	G65 P10A10. B20. C30.; M99;	G65 P100A100. B200.; M99;	M99;
Local variables (0) #1 0.100 #2 0.200 #3 0.300 #32	**Local variables (1)** A (#1) 1.000 B (#2) 2.000 C (#3) 3.000 D (#7) Z(#26) #32	**Local variables (2)** A (#1) 10.000 B (#2) 20.000 C (#3) 30.000 D (#7) Z(#26) #32	**Local variables (3)** A (#1) 100.000 B (#2) 200.000 C (#3) Z(#26) #32
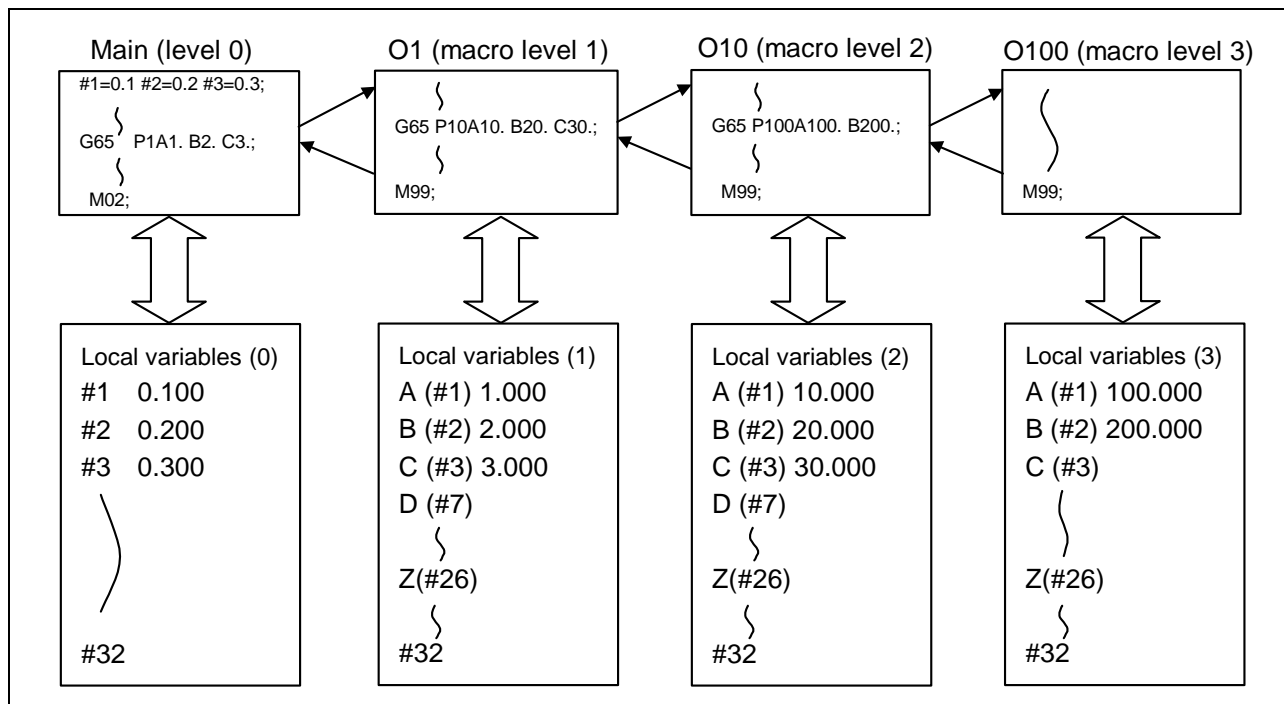
The status of the local variables appear on the setting and display unit.
Refer to the Operation Manual for details.

Macro interface inputs (#1000 to #1035, #1200 to #1295) : PLC → NC

The status of the interface input signals can be ascertained by reading out the values of variable numbers #1000 to #1035, #1200 to #1295. A variable value which has been read out can be only one of 2 values: 1 or 0 (1: contact closed, 0: contact open). All the input signals from #1000 to #1031 can be read at once by reading out the value of variable number #1032.
Similarly, the input signals #1200 to #1231, #1232 to #1263, and #1264 to #1295 can be read by reading the values of the variable numbers #1033 to #1035.
Variable numbers #1000 to #1035, #1200 to #1295 are for readout only, and cannot be placed in the left side member of their arithmetic formula. Input here refers to input to the control unit.

System variable	No. of points	Interface input signal	System variable	No. of points	Interface input signal
#1000	1	Register R72 bit 0	#1016	1	Register R73 bit 0
#1001	1	Register R72 bit 1	#1017	1	Register R73 bit 1
#1002	1	Register R72 bit 2	#1018	1	Register R73 bit 2
#1003	1	Register R72 bit 3	#1019	1	Register R73 bit 3
#1004	1	Register R72 bit 4	#1020	1	Register R73 bit 4
#1005	1	Register R72 bit 5	#1021	1	Register R73 bit 5
#1006	1	Register R72 bit 6	#1022	1	Register R73 bit 6
#1007	1	Register R72 bit 7	#1023	1	Register R73 bit 7
#1008	1	Register R72 bit 8	#1024	1	Register R73 bit 8
#1009	1	Register R72 bit 9	#1025	1	Register R73 bit 9
#1010	1	Register R72 bit 10	#1026	1	Register R73 bit 10
#1011	1	Register R72 bit 11	#1027	1	Register R73 bit 11
#1012	1	Register R72 bit 12	#1028	1	Register R73 bit 12
#1013	1	Register R72 bit 13	#1029	1	Register R73 bit 13
#1014	1	Register R72 bit 14	#1030	1	Register R73 bit 14
#1015	1	Register R72 bit 15	#1031	1	Register R73 bit 15

System variable	No. of points	Interface input signal
#1032	32	Register R72, R73
#1033	32	Register R74, R75
#1034	32	Register R76, R77
#1035	32	Register R78, R79

System variable	No. of points	Interface input signal	System variable	No. of points	Interface input signal
#1200	1	Register R74 bit 0	#1216	1	Register R75 bit 0
#1201	1	Register R74 bit 1	#1217	1	Register R75 bit 1
#1202	1	Register R74 bit 2	#1218	1	Register R75 bit 2
#1203	1	Register R74 bit 3	#1219	1	Register R75 bit 3
#1204	1	Register R74 bit 4	#1220	1	Register R75 bit 4
#1205	1	Register R74 bit 5	#1221	1	Register R75 bit 5
#1206	1	Register R74 bit 6	#1222	1	Register R75 bit 6
#1207	1	Register R74 bit 7	#1223	1	Register R75 bit 7
#1208	1	Register R74 bit 8	#1224	1	Register R75 bit 8
#1209	1	Register R74 bit 9	#1225	1	Register R75 bit 9
#1210	1	Register R74 bit 10	#1226	1	Register R75 bit 10
#1211	1	Register R74 bit 11	#1227	1	Register R75 bit 11
#1212	1	Register R74 bit 12	#1228	1	Register R75 bit 12
#1213	1	Register R74 bit 13	#1229	1	Register R75 bit 13
#1214	1	Register R74 bit 14	#1230	1	Register R75 bit 14
#1215	1	Register R74 bit 15	#1231	1	Register R75 bit 15

System variable	No. of points	Interface input signal	System variable	No. of points	Interface input signal
#1232	1	Register R76 bit 0	#1248	1	Register R77 bit 0
#1233	1	Register R76 bit 1	#1249	1	Register R77 bit 1
#1234	1	Register R76 bit 2	#1250	1	Register R77 bit 2
#1235	1	Register R76 bit 3	#1251	1	Register R77 bit 3
#1236	1	Register R76 bit 4	#1252	1	Register R77 bit 4
#1237	1	Register R76 bit 5	#1253	1	Register R77 bit 5
#1238	1	Register R76 bit 6	#1254	1	Register R77 bit 6
#1239	1	Register R76 bit 7	#1255	1	Register R77 bit 7
#1240	1	Register R76 bit 8	#1256	1	Register R77 bit 8
#1241	1	Register R76 bit 9	#1257	1	Register R77 bit 9
#1242	1	Register R76 bit 10	#1258	1	Register R77 bit 10
#1243	1	Register R76 bit 11	#1259	1	Register R77 bit 11
#1244	1	Register R76 bit 12	#1260	1	Register R77 bit 12
#1245	1	Register R76 bit 13	#1261	1	Register R77 bit 13
#1246	1	Register R76 bit 14	#1262	1	Register R77 bit 14
#1247	1	Register R76 bit 15	#1263	1	Register R77 bit 15

System variable	No. of points	Interface input signal	System variable	No. of points	Interface input signal
#1264	1	Register R78 bit 0	#1280	1	Register R79 bit 0
#1265	1	Register R78 bit 1	#1281	1	Register R79 bit 1
#1266	1	Register R78 bit 2	#1282	1	Register R79 bit 2
#1267	1	Register R78 bit 3	#1283	1	Register R79 bit 3
#1268	1	Register R78 bit 4	#1284	1	Register R79 bit 4
#1269	1	Register R78 bit 5	#1285	1	Register R79 bit 5
#1270	1	Register R78 bit 6	#1286	1	Register R79 bit 6
#1271	1	Register R78 bit 7	#1287	1	Register R79 bit 7
#1272	1	Register R78 bit 8	#1288	1	Register R79 bit 8
#1273	1	Register R78 bit 9	#1289	1	Register R79 bit 9
#1274	1	Register R78 bit 10	#1290	1	Register R79 bit 10
#1275	1	Register R78 bit 11	#1291	1	Register R79 bit 11
#1276	1	Register R78 bit 12	#1292	1	Register R79 bit 12
#1277	1	Register R78 bit 13	#1293	1	Register R79 bit 13
#1278	1	Register R78 bit 14	#1294	1	Register R79 bit 14
#1279	1	Register R78 bit 15	#1295	1	Register R79 bit 15

Macro interface outputs (#1100 to #1135, #1300 to #1395) : NC → PLC

The interface output signals can be sent by substituting values in variable numbers #1100 to #1135, #1300 to #1395. An output signal can be only 0 or 1.
All the output signals from #1100 to #1131 can be sent at once by substituting a value in variable number #1132.
Similarly, the output signals #1300 to #1311, #1332 to #1363, and #1364 to #1395 can be sent by assigning values to the variable numbers #1133 to #1135.
The status of the writing and output signals can be read in order to offset the #1100 to #1135, #1300 to #1395 output signals. Output here refers to the output from the NC.

System variable	No. of points	Interface output signal	System variable	No. of points	Interface output signal
#1100	1	Register R172 bit 0	#1116	1	Register R173 bit 0
#1101	1	Register R172 bit 1	#1117	1	Register R173 bit 1
#1102	1	Register R172 bit 2	#1118	1	Register R173 bit 2
#1103	1	Register R172 bit 3	#1119	1	Register R173 bit 3
#1104	1	Register R172 bit 4	#1120	1	Register R173 bit 4
#1105	1	Register R172 bit 5	#1121	1	Register R173 bit 5
#1106	1	Register R172 bit 6	#1122	1	Register R173 bit 6
#1107	1	Register R172 bit 7	#1123	1	Register R173 bit 7
#1108	1	Register R172 bit 8	#1124	1	Register R173 bit 8
#1109	1	Register R172 bit 9	#1125	1	Register R173 bit 9
#1110	1	Register R172 bit 10	#1126	1	Register R173 bit 10
#1111	1	Register R172 bit 11	#1127	1	Register R173 bit 11
#1112	1	Register R172 bit 12	#1128	1	Register R173 bit 12
#1113	1	Register R172 bit 13	#1129	1	Register R173 bit 13
#1114	1	Register R172 bit 14	#1130	1	Register R173 bit 14
#1115	1	Register R172 bit 15	#1131	1	Register R173 bit 15

System variable	No. of points	Interface output signal
#1132	32	Register R172, R173
#1133	32	Register R174, R175
#1134	32	Register R176, R177
#1135	32	Register R178, R179

System variable	No. of points	Interface output signal	System variable	No. of points	Interface output signal
#1300	1	Register R174 bit 0	#1316	1	Register R175 bit 0
#1301	1	Register R174 bit 1	#1317	1	Register R175 bit 1
#1302	1	Register R174 bit 2	#1318	1	Register R175 bit 2
#1303	1	Register R174 bit 3	#1319	1	Register R175 bit 3
#1304	1	Register R174 bit 4	#1320	1	Register R175 bit 4
#1305	1	Register R174 bit 5	#1321	1	Register R175 bit 5
#1306	1	Register R174 bit 6	#1322	1	Register R175 bit 6
#1307	1	Register R174 bit 7	#1323	1	Register R175 bit 7
#1308	1	Register R174 bit 8	#1324	1	Register R175 bit 8
#1309	1	Register R174 bit 9	#1325	1	Register R175 bit 9
#1310	1	Register R174 bit 10	#1326	1	Register R175 bit 10
#1311	1	Register R174 bit 11	#1327	1	Register R175 bit 11
#1312	1	Register R174 bit 12	#1328	1	Register R175 bit 12
#1313	1	Register R174 bit 13	#1329	1	Register R175 bit 13
#1314	1	Register R174 bit 14	#1330	1	Register R175 bit 14
#1315	1	Register R174 bit 15	#1331	1	Register R175 bit 15

System variable	No. of points	Interface output signal	System variable	No. of points	Interface output signal
#1332	1	Register R176 bit 0	#1348	1	Register R177 bit 0
#1333	1	Register R176 bit 1	#1349	1	Register R177 bit 1
#1334	1	Register R176 bit 2	#1350	1	Register R177 bit 2
#1335	1	Register R176 bit 3	#1351	1	Register R177 bit 3
#1336	1	Register R176 bit 4	#1352	1	Register R177 bit 4
#1337	1	Register R176 bit 5	#1353	1	Register R177 bit 5
#1338	1	Register R176 bit 6	#1354	1	Register R177 bit 6
#1339	1	Register R176 bit 7	#1355	1	Register R177 bit 7
#1340	1	Register R176 bit 8	#1356	1	Register R177 bit 8
#1341	1	Register R176 bit 9	#1357	1	Register R177 bit 9
#1342	1	Register R176 bit 10	#1358	1	Register R177 bit 10
#1343	1	Register R176 bit 11	#1359	1	Register R177 bit 11
#1344	1	Register R176 bit 12	#1360	1	Register R177 bit 12
#1345	1	Register R176 bit 13	#1361	1	Register R177 bit 13
#1346	1	Register R176 bit 14	#1362	1	Register R177 bit 14
#1347	1	Register R176 bit 15	#1363	1	Register R177 bit 15

System variable	No. of points	Interface output signal	System variable	No. of points	Interface output signal
#1364	1	Register R178 bit 0	#1380	1	Register R179 bit 0
#1365	1	Register R178 bit 1	#1381	1	Register R179 bit 1
#1366	1	Register R178 bit 2	#1382	1	Register R179 bit 2
#1367	1	Register R178 bit 3	#1383	1	Register R179 bit 3
#1368	1	Register R178 bit 4	#1384	1	Register R179 bit 4
#1369	1	Register R178 bit 5	#1385	1	Register R179 bit 5
#1370	1	Register R178 bit 6	#1386	1	Register R179 bit 6
#1371	1	Register R178 bit 7	#1387	1	Register R179 bit 7
#1372	1	Register R178 bit 8	#1388	1	Register R179 bit 8
#1373	1	Register R178 bit 9	#1389	1	Register R179 bit 9
#1374	1	Register R178 bit 10	#1390	1	Register R179 bit 10
#1375	1	Register R178 bit 11	#1391	1	Register R179 bit 11
#1376	1	Register R178 bit 12	#1392	1	Register R179 bit 12
#1377	1	Register R178 bit 13	#1393	1	Register R179 bit 13
#1378	1	Register R178 bit 14	#1394	1	Register R179 bit 14
#1379	1	Register R178 bit 15	#1395	1	Register R179 bit 15

(Note 1)   The last values of the system variables #1100 to #1135, #1300 to #1395 sent are retained as 1 or 0. (They are not cleared even with resetting.)

(Note 2)   The following applies when any number except 1 or 0 is substituted into #1100 to #1131, #1300 to #1395.

<Vacant> is treated as 0.

Any number except 0 and <vacant> is treated as 1.

Any value less than 0.00000001 is indefinite.

Input signal

#1032 (R72,R73)

#1000
#1031

#1033 (R74,R75)

#1200
#1231

#1034 (R76,R77)

#1232
#1263

#1035 (R78,R79)

#1264
#1295

Macro instructions

#1132 (R172,R173)

#1100
#1131

#1133 (R174,R175)

#1300
#1331

#1134 (R176,R177)

#1332
#1363

#1135 (R178,R179)

#1364
#1395

Output signal
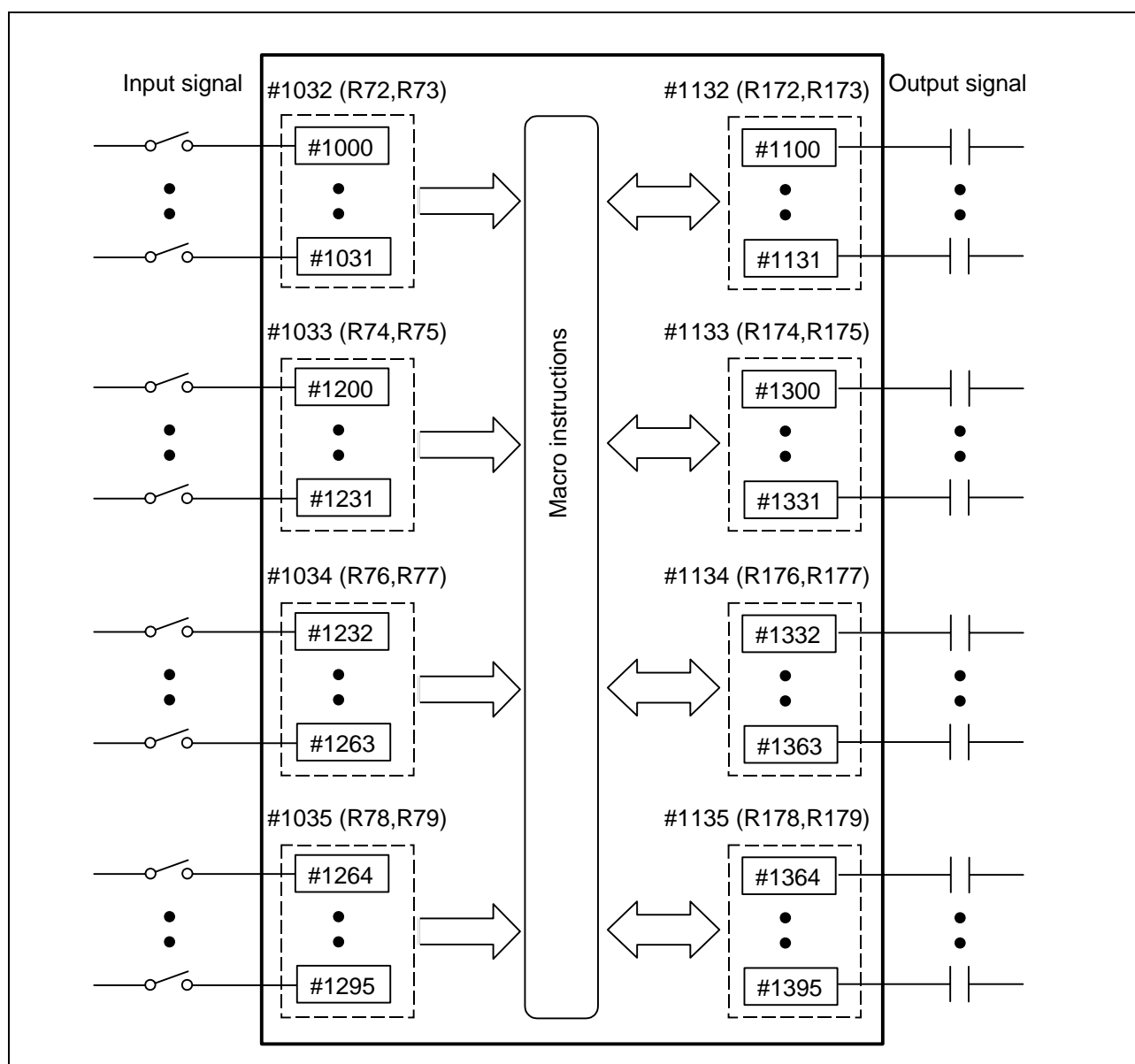
Tool offset

Tool data can be read and values substituted using the variable numbers.

Variable number range		Type 1	Type 2
#10001 to #10000 + n	#2001 to #2000 + n	○	○ (Length dimension)
#11001 to #11000 + n	#2201 to #2200 + n	×	○ (Length wear)
#16001 to #16000 + n	#2401 to #2400 + n	×	○ (Radius dimension)
#17001 to #17000 + n	#2601 to #2600 + n	×	○ (Radius wear)

"n" in the table indicates the tool No.
The maximum value of "n" matches the number of tool offset sets.
Numbers in the #10000 order have the equivalent function to that in the #2000 order, however, the maximum value of "n" for #2000 order is "200".
When the number of tool offset sets is larger than "200", use the variables of #10000 order.
The tool compensation data are configured as data with a decimal point in the same way as for other variables. Consequently, programming as "#10001 = 1000;" results the setting "1000.000" in the tool compensation data.

Programming example		Common variables	Tool offset data
#101=1000; #10001=#101; #102=#10001;	After execution →	#101=1000.0 #102=1000.0	H1=1000.000
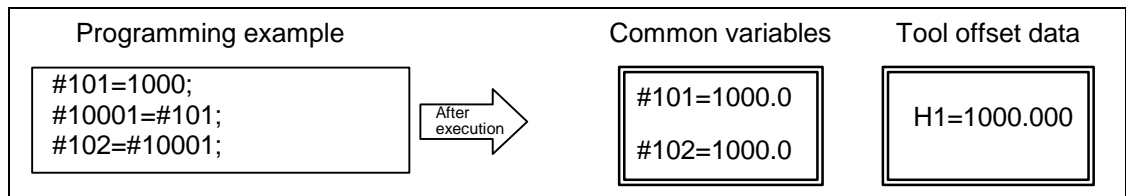
(Example 1) Calculation and tool offset data setting

G28 Z0 T01 ;	Zero point return
M06 ;	Tool change (spindle T01)
#1=#5003 ;	Start point memory
G00 Z-500. ;	Rapid traverse to safety position
G31 Z-100. F100;	Skip measurement
#10001=#5063-#1 ;	Measured distance calculation and tool offset data setting
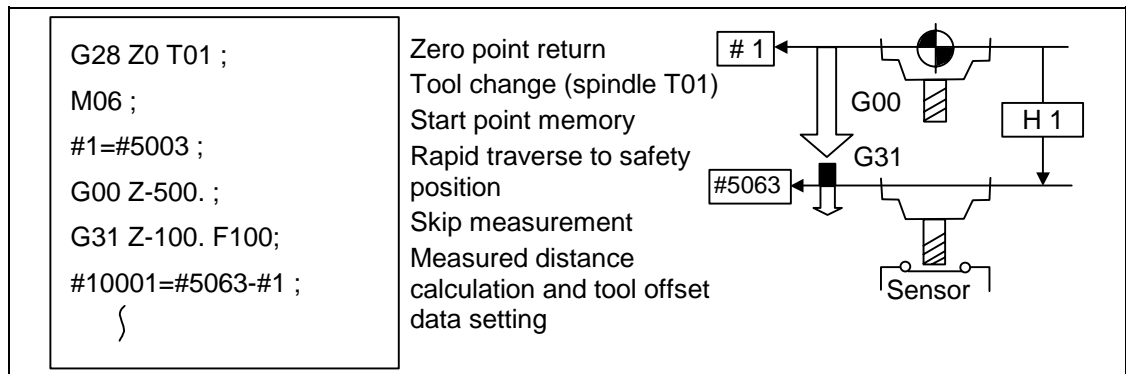
(Note) In this example, no consideration is given to the delay in the skip sensor signal.
#5003 is the Z-axis start point position and #5063 is the Z-axis skip coordinates, and indicated is the position at which the skip signal is input while G31 is being executed.

Work coordinate system offset

By using variable numbers #5201 to #532n, it is possible to read out the work coordinate system offset data or to substitute values.

(Note) The number of axes which can be controlled differs according to the specifications.

Axis name \ Axis No.	Axis 1	Axis 2	Axis 3	Remarks
External work offset	#5201	#5202	#5203	External workpiece offset specifications are required.
G54	#5221	#5222	#5223	
G55	#5241	#5242	#5243	
G56	#5261	#5262	#5263	
G57	#5281	#5282	#5283	
G58	#5301	#5302	#5303	
G59	#5321	#5322	#5323	

(Example 1)

```
N1 G28 X0 Y0 Z0 ;
N2 #5221=-20. #5222=-20. ;
N3 G90 G00 G54 X0 Y0 ;



N10 #5221=-90. #5222=-10. ;
N11 G90 G00 G54 X0Y0 ;



M02 ;
```

G54 work coordinate system defined by N10

G54 work coordinate system defined by N2
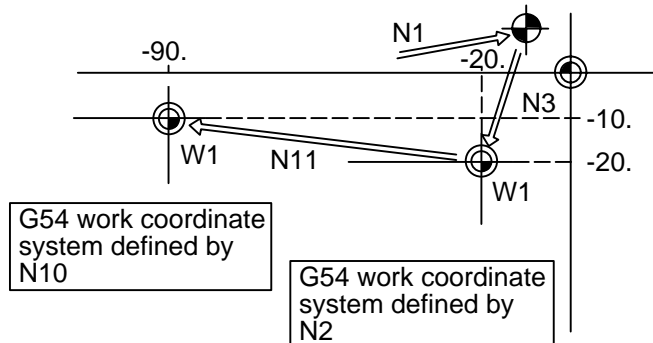
(Example 2)

```
N100 #5221=#5221+#5201 ;
     #5222=#5222+#5202 ;
     #5241=#5241+#5201 ;
     #5242=#5242+#5202 ;
     #5201=0 #5202=0;
```

Coordinate system before change

Base machine coordinate system
External workpiece offset

Coordinate system after change

Base machine coordinate system
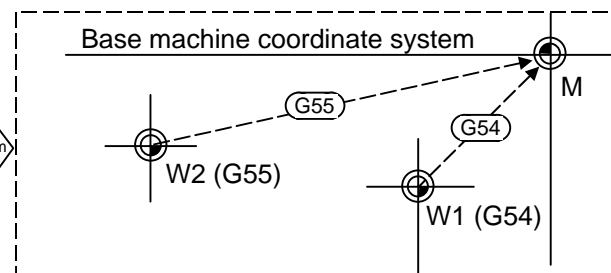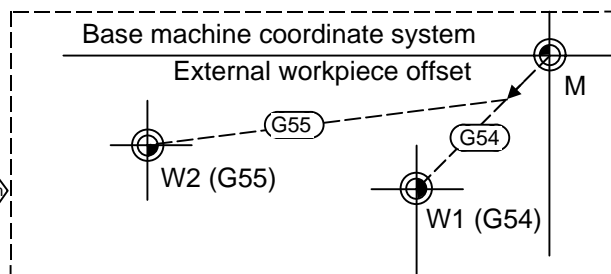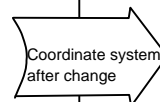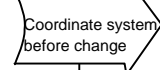
This is an example where the external workpiece offset values are added to the work coordinate (G54, G55) system offset values without changing the position of the work coordinate systems.

Alarm (#3000)

The NC system can be forcibly set to the alarm state by using variable number #3000.

Format

#3000 = 70 (CALL#PROGRAMMER#TEL#530) :
70   : Alarm number
CALL#PROGRAMMER#TEL#530   : Alarm message

Any alarm number from 1 to 9999 can be specified.
The alarm message must be less than 31 characters long.
"NC alarm 3 (program error)" signal will be output.
The "P277" user macro alarm message appears in the <alarm> column on diagnosis screen 1 while the alarm number and alarm message CALL #PROGRAMMER #TEL#530 is indicated in the <operator message>.

Example of program (alarm when #1 = 0)

```
IF [#1   NE   0] GOTO 100 ;
#3000=70
(CALL#PROGRAMMER#TEL#530) ;
N100
```

Stops with NC alarm

<Alarm>

    P277 : Macro alarm message

<Operator message>

    CALL#PROGRAMMER#TEL#530   70

(Note 1) Alarm number 0 is not displayed and any number exceeding 9999 cannot be indicated.
(Note 2) The characters following the first alphabet letter in the right member is treated as the alarm message. Therefore, a number cannot be designated as the first character of an alarm message. It is recommended that the alarm messages be enclosed in round parentheses.

Integrating (run-out) time (#3001, #3002)

The integrating (run-out) time can be read during automatic operation or automatic start or values can be substituted by using variable numbers #3001 and #3002.

Type	Variable number	Unit	Contents when power is switched on	Initialization of contents	Count condition
Integrating (run-out) time 1	#3001	ms	Same as when power is switched off	Value substituted for variable	At all times while power is ON
Integrating (run-out) time 2	#3002				In-automatic start

The integrating run time returns to zero in about 2.44×10^{11} ms (approximately 7.7 years).

```
                                                          O9010
   ⎧                                                ┌──────────────────────────────┐
   ⎰                                                │                              │
G65P9010 T (allowable time) ms ;    ═══════▷        │ #3001=0 ;                    │
   ⎱                                To              │ WHILE   [#3001LE#20] DO1 ;   │
   ⎰                                subprogram      │   ⎰                          │
                                                    │   ⎱                          │
                                                    │ END1 :                       │
                                                    │                              │
                                                    │ M99 ;                        │
                                                    └──────────────────────────────┘

                      Local variable
Entered in local    ┌──────────────┐    Allowable time portion :
variable #20   ───▶ │ T#20_____  │      DO1-END is repeated and when
                    └──────────────┘      allowable time is reached, operations
                                          jumps to M99.
```
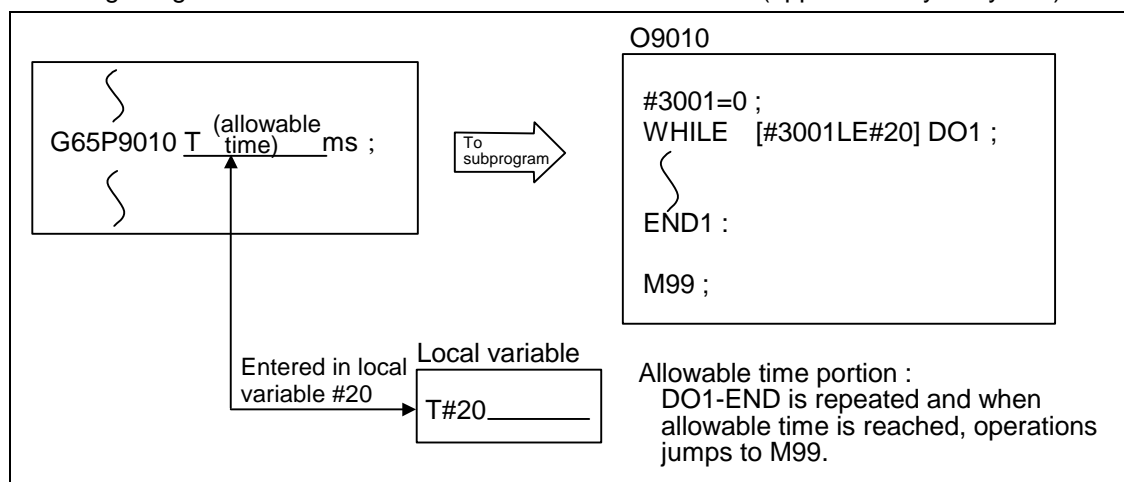
Suppression of single block stop and miscellaneous function finish signal waiting

By substituting the values below in variable number #3003, it is possible to suppress single block stop in the subsequent blocks or to advance to the next block without waiting for the miscellaneous function (M, S, T, B) finish (FIN) signal.

#3003	Single block stop	Miscellaneous function finish signal
0	Not suppressed	Awaited
1	Suppressed	Awaited
2	Not suppressed	Not awaited
3	Suppressed	Not awaited

(Note 1) #3003 is cleared to zero by NC reset.

Feed hold, feedrate override, G09 valid/invalid

By substituting the values below in variable number #3004, it is possible to make the feed hold, feedrate override and G09 functions either valid or invalid in the subsequent blocks.

#3004 Contents (value)	Bit 0 Feed hold	Bit 1 Feedrate override	Bit 2 G09 check
0	Valid	Valid	Valid
1	Invalid	Valid	Valid
2	Valid	Invalid	Valid
3	Invalid	Invalid	Valid
4	Valid	Valid	Invalid
5	Invalid	Valid	Invalid
6	Valid	Invalid	Invalid
7	Invalid	Invalid	Invalid

(Note 1) Variable number #3004 is set to zero by NC reset.
(Note 2) The functions are valid when the above bits are 0 and invalid when they are 1.
(Note 3) When the feed hold is set to invalid with #3004, the following will occur when the feed hold switch is pressed.
  - During thread cutting, block stop will be carried out at the end of the next block of the block where thread cutting is completed.
  - During tapping with tap cycle, block stop will be carried out after the operation of R point return.
  - In the case other than above, block stop will be carried out after termination of the currently executing block.

Message display and stop

By using variable number #3006, the execution is stopped after the previous block has been executed and, if message display data have been commanded, then the corresponding message will be indicated.

Format

#3006 = 1 (TAKE FIVE) :
TAKE FIVE   Message

The message should not be longer than 31 characters and it should be enclosed within round ( ) parentheses.

Mirror image

By reading variable number #3007, it is possible to ascertain the status of mirror image at a particular point in time for each axis.
The axes correspond to the bits of #3007.
When the bits are 0, it means that the mirror image function is not valid; when they are 1, it means that it is valid.

#3007

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
nth axis														3	2	1

G command modals

Using variable numbers #4001 to #4021, it is possible to read the G modal commands which have been issued up to the block immediately before.
Similarly, it is possible to read the modals in the block being executed with variable numbers #4201 to #4221.

Variable number		Function
Pre-read block	**Execution block**	
#4001	#4201	Interpolation mode　　　　　: G00:0, G01:1, G02:2, G03:3, G33:33
#4002	#4202	Plane selection　　　　　　　: G17:17, G18:18, G19:19
#4003	#4203	Absolute/incremental　　　　: G90:90, G91:91
#4004	#4204	No variable No.
#4005	#4205	Feed designation　　　　　　: G94:94, G95:95
#4006	#4206	Inch/metric　　　　　　　　　: G20:20, G21/21
#4007	#4207	Tool nose R compensation :　: G40:40, G41:41, G42:42
#4008	#4208	Tool length offset　　　　　　: G43:43, G44:44, G49:49
#4009	#4209	Fixed cycle　　　　　　　　　: G80:80, G73 to 74, G76:76, G81 to 89:81 to 89
#4010	#4210	Return level　　　　　　　　　: G98:98, G99:99
#4011	#4211	
#4012	#4212	Work coordinate system　　　: G54 to G59:54 to 59
#4013	#4213	Acceleration/deceleration　　: G61 to G64:61 to 64, G61.1:61.1
#4014	#4214	Macro modal call　　　　　　: G66:66, G66.1:66.1, G67:67
#4015	#4215	
#4016	#4216	
#4017	#4217	Constant surface speed control : G96:96, G97:97
#4018	#4218	No variable No.
#4019	#4219	Mirror image　　　　　　　　: G50.1:50.1, G51.1:51.1
#4020	#4220	
#4021	#4221	No variable No.

(Example)

```
        G28 X0 Y0 Z0 ;
        G90 G1 X100. F1000;
        G91 G65 P300 X100. Y100.;
        M02;
         O300;
        #1 = #4003;      → Group 3G modal (pre-read) #1 = 91.0
        #2 = #4203;      → Group 3G modal (now being executed) #2 = 90.0
        G#1 X#24 Y#25;
        M99;
        %
```

13. Program Support Functions

Other modals

Using variable numbers #4101 to #4120, it is possible to read the model commands assigned up to the block immediately before.
Similarly, it is possible to read the modals in the block being executed with variable numbers #4301 to #4320.

Variable number		Modal information	Variable number		Modal information
Pre-read	Execution		Pre-read	Execution	
#4101	#4301		#4111	#4311	Tool length offset No.H
#4102	#4302		#4112	#4312	
#4103	#4303		#4113	#4313	Miscellaneous function M
#4104	#4304		#4114	#4314	Sequence number N
#4105	#4305		#4115	#4315	Program number O
#4106	#4306		#4116	#4316	
#4107	#4307	Tool radius compensation No. D	#4117	#4317	
#4108	#4308		#4118	#4318	
#4109	#4309	Feedrate F	#4119	#4319	Spindle function S
#4110	#4310		#4120	#4320	Tool function T

Position information

Using variable numbers #5001 to #5104, it is possible to read the servo deviation amounts, tool position offset amount, skip coordinates, work coordinates, machine coordinates and end point coordinates in the block immediately before.

Position information / Axis No.	End point coordinate of block immediately before	Machine coordinate	Work coordinate	Skip coordinate	Tool position offset amount	Servo deviation amount
1	#5001	#5021	#5041	#5061	#5081	#5101
2	#5002	#5022	#5042	#5062	#5082	#5102
3	#5003	#5023	#5043	#5063	#5083	#5103
Remarks (reading during movement)	Yes	No	No	Yes	No	Yes

(Note1) The number of axes which can be controlled differs according to the specifications.

(Note2) The last digit of the variable No. corresponds to the control axis No.

251
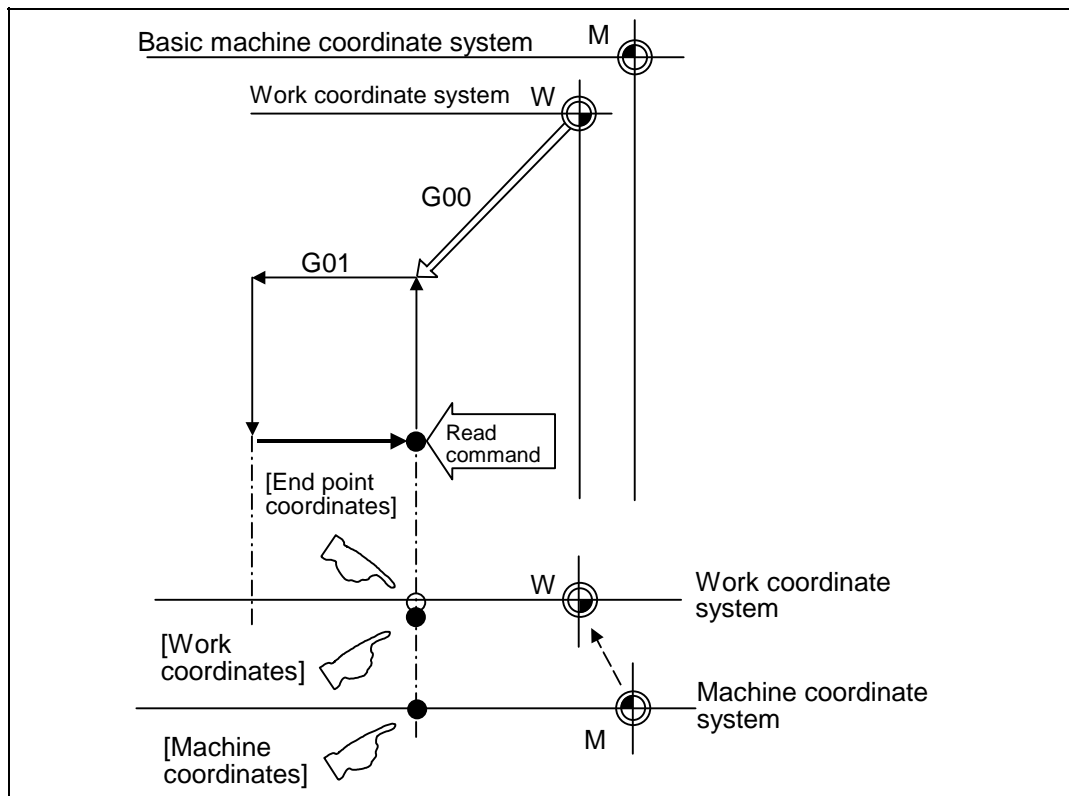
(1) The positions of the end point coordinates and skip coordinates are positions in the work coordinate system.

(2) The end point coordinates, skip coordinates and servo deviation amounts can be read even during movement. However, it must first be checked that movement has stopped before reading the machine coordinates and the work coordinates.

(3) The position where the skip signal is turned ON in the G31 block is indicated for the skip coordinates. The end point position is indicated when the skip signal has not been turned ON. (For further details, refer to the section on tool length measurement.)
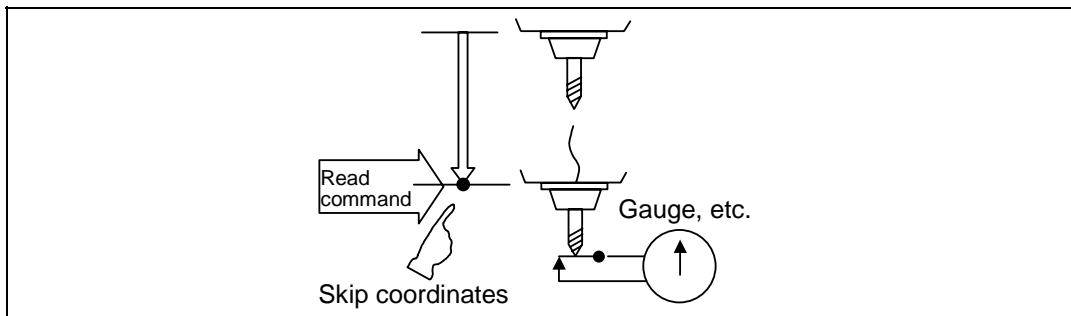
(4) The tool nose position where the tool offset and other such factors are not considered is indicated as the end point position. The tool reference point position with consideration given to tool offset is indicated for the machine coordinates, work coordinates and skip coordinates.
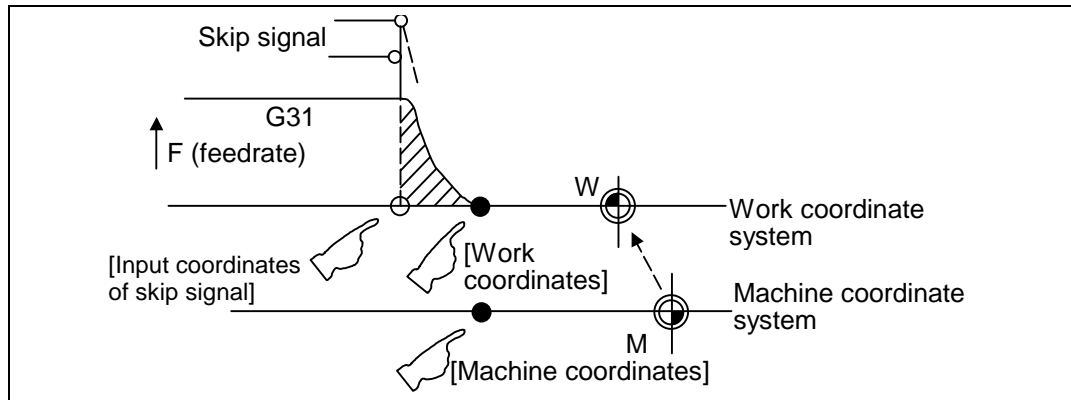
For "●", check stop and then proceed to read.
For "○", reading is possible during movement.

The position of the skip signal input coordinates is the position in the work coordinate system. The coordinates in variable numbers #5061 to #5064 memorize the moments when the skip input signal during movement was input and so they can be read at any subsequent time. For further details, reference should be made to the section on the skip function.

(Example 1)   Example of workpiece position measurement [E68]

An example to measure the distance from the measured reference point to the workpiece edge is shown below.

Argument
<Local variable>

F(#9)	200
X(#24)	100.000
Y(#25)	100.000
Z(#26)	-10.000

Main program

G65 P9031 X100. Y100. Z-10. F200;

To subprogram

<Common variable>

#101	87.245
#102	87.245
#103	123.383

O9031

```
N1 #180=#4003;
N2 #30=#5001 #31=#5002;
N3 G91 G01 Z#26 F#9;
N4 G31 X#24 Y#25 F#9;
N5 G90 G00 X#30 Y#31;
N6 #101=#30-#5061 #102=#31-#5062;
N7 #103=SQR [#101*#101+#102*#102] ;
N8 G91 G01Z-#26;
N9 IF [#180 EQ 91] GOTO 11;
N10 G90;
N11 M99;
```
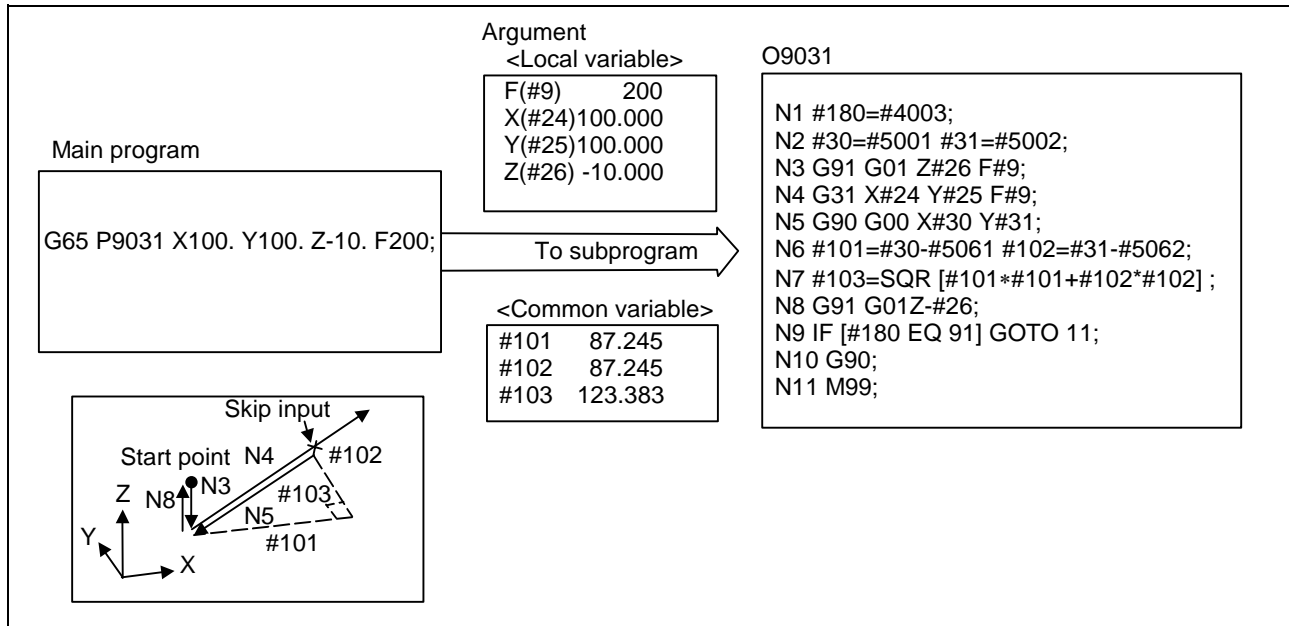

Skip input
Start point N4   #102
N3
#103
N5
#101

#101	X axis measurement amount	N1	G90/G91 modal recording
#102	X axis measurement amount	N2	X, Y start point recording
#103	Measurement linear segment amount	N3	Z axis entry amount
		N4	X, Y measurement (Stop at skip input)
#5001	X axis measurement start point	N5	Return to X, Y start point
#5002	Y axis measurement start point	N6	X, Y measurement incremental value calculation
		N7	Measurement linear segment calculation
#5061	X axis skip input point	N8	Z axis escape
#5062	Y axis skip input point	N9, N10	G90/G91 modal return
		N11	Return to main program from subprogram

(Example 2)   Reading of skip input coordinates


```
N1 G91 G28 X0 Y0;
N2 G90 G00 X0 Y0;
N3 X0Y-100.;
N4 G31 X-150. Y-50. F80;
N5 #111=#5061#112=#5062;
N6 G00 Y0;
N7 G31 X0;
N8 #121=#5061#122=#5062;
N9 M02;
```
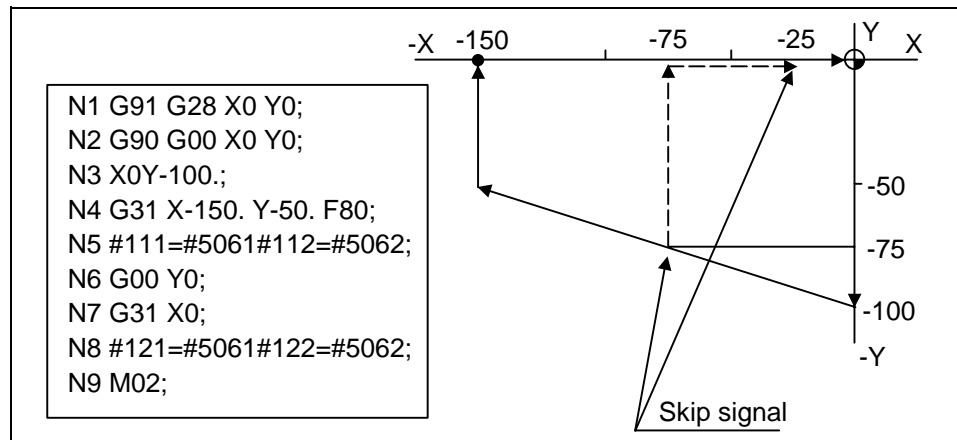
#111 = -75. + ε #112 = -75. + ε
#121 = -25. + ε #122 = -75. + ε
ε is the error caused by response delay.
(Refer to the section on the skip function for details.)
#122 is the N4 skip signal input coordinates as there is no Y command at N7.

Variable name setting and quotation

Any name (variable name) can be given to common variables #500 to #519. It must be composed of not more than 7 alphanumerics and it must begin with a letter. Do not use "#" in variable names. It causes an alarm when the program is executed.

Format

```
SETVN n [ NAME1, NAME2, ………. ]  :
n                          : Head number of variable to be named
NAME1                      : #n name (variable name)
NAME2                      : #n + 1 name (variable name)
```

Variable names are separated by a comma (,).

Detailed description
(1) Once variable names have been set, they will not be cleared even when the power is switched off.
(2) Variables in programs can be quoted by their variable names. In cases like this, the variables should be enclosed in square parentheses.
      (Example 1) G01X [#POINT1] ;

                                  [#NUMBER] = 25 ;

(3) The variable numbers, data and variable names appear on the screen of the setting and display unit.

(Example 2)

Program ...... SETVN500 [A234567, DIST, TOOL25] ;

```
[Common variables]
    #500      -12345.678    A234567
    #501       5670.000     DIST
    #502       -156.500     TOOL25
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    #518        10.000      NUMBER
Common variable     #(502) Data (-156.5) Name (TOOL25)
```

(Note) At the head of the variable name, do not use the characters determined by the NC for use in arithmetic commands, etc. (e.g. SIN, COS).

Number of workpiece machining times

The number of workpiece machining times can be read using variables #3901 and #3902.
By substituting a value in these variables, the number of workpiece machining times can be changed.

Type	Variable No.	Data setting range
Number of workpiece machining times	#3901	0 to 999999
Maximum workpiece value	#3902	

(Note) Always substitute a positive value for the number of workpiece machining times.

Tool life management

(1)   Definition of variable numbers

(a) Designation of group No.
 #60000

The tool life management data group No. to be read with #60001 to #64700 is designated by substituting a value in this variable. If a group No. is not designated, the data of the group registered first is read. This is valid until reset.

(b) Tool life management system variable No. (Read)
 #60001 to #64700

```
# ? ? ? ? ?
```

+ Variable No. or data type
Data class
6: Tool life management
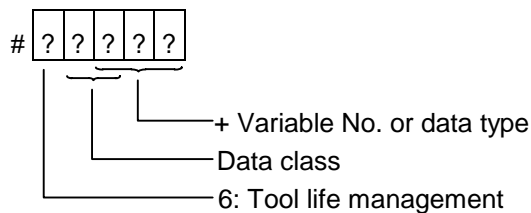
(c) Details of data classification

Data class	M System	L System	Remarks
00	For control	For control	Refer to following types
05	Group No.	Group No.	Refer to registration No.
10	Tool No.	Tool No.	Refer to registration No.
15	Tool data flag	Method	Refer to registration No.
20	Tool status	Status	Refer to registration No.
25	Life data	Life time/No. of times	Refer to registration No.
30	Usage data	Usage time/No. of times	Refer to registration No.
35	Tool length compensation data	-	Refer to registration No.
40	Tool radius compensation data	-	Refer to registration No.
45	Auxiliary data	-	Refer to registration No.

The group No., L System method, and life data are common for the group.

256

(d) Registration No.

M system	1 to 200
L system	1 to 16

(e) Data type

Type	M System	L System	Remarks
1	Number of registered tools	Number of registered tools	
2	Life current value	Life current value	
3	Tool selected No.	Tool selected No.	
4	Number of remaining registered tools	Number of remaining registered tools	
5	Signal being executed	Signal being executed	
6	Cutting time cumulative value (minute)	Cutting time cumulative value (minute)	
7	Life end signal	Life end signal	
8	Life prediction signal	Life prediction signal	

Variable No.	Item	Type	Details	Data range
60001	Number of registered tools	Common to system	Total number of tools registered in each group.	0 to 200
60002	Life current value	For each group (Designate group No. #60000)	Usage time/No. of uses of tool being used. Spindle tool usage data or usage data for tool in use (#60003).	0 to 4000 minutes 0 to 9999 times
60003	Tool selected No.		Registration No. of tool being used. Spindle tool registration No. (If spindle tool is not data of the designated group, ST:1 first tool, or if ST:1 is not used, the first tool of ST:0. When all tools have reached their lives, the last tool.)	0 to 200
60004	Number of remaining registered tools		No. of first registered tool that has not reached its life.	0 to 200
60005	Signal being executed		"1" when this group is used in program being executed. "1" when spindle tool data group No. and designated group No. match.	0/1
60006	Cutting time cumulative value (minute)		Indicates the time that this group is used in the program being executed.	
60007	Life end signal		"1" when lives of all tools in this group have been reached. "1" when all tools registered in designated group reach lives.	0/1
60008	Life prediction signal		"1" when new tool is selected with next command in this group. "1" when there is a tool for which ST is "0: Not used" in the designated group, and there are no tools for which ST is "1: Tools in use".	0/1

Variable No.	Item	Type	Details	Data range
60500 +***	Group No.	Each group/ registration No.	This group's No.	1 to 99999999
61000 +***	Tool No.	(Designate the group No. #60000 and registration No. *** .)	Tool No.	1 to 99999999
61500 +***	Tool data flag	Note the group No., method and life are common for the groups.	Usage data count method, length compensation method, radius compensation method, etc., parameters. bit 0, 1 : Tool length compensation data format bit 2, 3 : Tool radius compensation data format 　　　0: Compensation No. method 　　　1: Incremental value compensation amount method 　　　2: Absolute value compensation amount method bit 4, 5 : Tool life management method 　　　0: Usage time 　　　1: No. of mounts 　　　2: No. of usages	0 to FF (H)
62000 +***	Tool status		Tool usage state 0: Not used tool 1: Tool being used 2: Normal life tool 3: Tool error 1 4: Tool error 2	0 to 4
62500 +***	Life data		Life time or No. of lives for each tool	0 to 4000 minutes 0 to 9999 times
63000 +***	Usage data		Usage time or No. of uses for each tool	0 to 4000 minutes 0 to 9999 times
63500 +***	Tool length compensation data		Length compensation data set as compensation No., absolute value compensation amount or increment value compensation amount method.	Compensation No.: 0 to No. of tool compensation sets Absolute value compensation amount ±99999.999 Increment value compensation amount ±99999.999

258

Variable No.	Item	Type	Details	Data range
64000 +***	Tool radius compensation data		Radius compensation data set as compensation No., absolute value compensation amount or increment value compensation amount method.	Compensation No.: 0 to No. of tool compensation sets Absolute value compensation amount ±99999.999 Increment value compensation amount ±99999.999
64500 +***	Auxiliary data		Spare data	0 to 65535

Example of program for tool life management

(1) Normal commands

```
#101 = #60001 ; ........... Reads the number of registered tools.
#102 = #60002 ; ........... Reads the life current value.
#103 = #60003 ; ........... Reads the tool selection No.
#60000 = 10 ; ............... Designates the group No. of the life data
                              to be read.
#104 = #60004 ; ........... Reads the remaining number of registered
                              tools in group 10.
#105 = #60005 ; ........... Reads the signal being executed in group 10.
#111 = #61001 ; ........... Reads the group 10, #1 tool No.
#112 = #62001 ; ........... Reads the group 10, #1 status.
#113 = #61002 ; ........... Reads the group 10, #2 status.
%
```

Designated program No. is valid until reset.

(2) When group No. is not designated.

```
#104 = #60004 ; ........... Reads the remaining number of registered tools in the group
                              registered first.
#111 = #61001 ; ........... Reads the #1 tool No. in the group registered first.
%
```

(3) When non-registered group No. is designated. (Group 9999 does not exist.)

```
#60000 = 9999 ; ...........        Designates the group No.
#104 = #60004 ; ...........        #104 = -1.
```

(4) When registration No. not used is designated. (Group 10 has 15 tools)

```
#60000 = 10 ; ...............       Designates the group No.
#111 = #61016 ; ...........        #101 = -1.
```

(5) When registration No. out of the specifications is designated.

```
#60000 = 10 ;
#111 = #61017 ; ...........        Program error (P241)
```

(6) When tool life management data is registered with G10 command after group No. is designated.

#60000 = 10 ;	Designates the group No.
G10 L3 ;	Starts the life management data registration.
P10 LLn NNn ;	10 is the group No., Ln is the life per tool, Nn is the method.
TTn ;	Tn is the tool No.
:	
G11 ;	Registers the group 10 data with the G10 command.

The group 10 life data is registered.

#111 = #61001 ;	Reads the group 10, #1 tool No.
G10 L3 ;	Starts the life management data registration.
P1 LLn NNn ;	1 is the group No., Ln is the life per tool, Nn is the method.
TTn ;	Tn is the tool No.
:	
G11 ;	Registers the life data with the G10 command. (The registered data is deleted.)
#111 = 61001 ;	Group 10 does not exist. #201 = -1.

The life data other than group 10 is registered.

Precautions for tool life management

(1) If the tool life management system variable is commanded without designating a group No., the data of the group registered at the head of the registered data will be read.
(2) If a non-registered group No. is designated and the tool life management system variable is commanded, "-1" will be read as the data.
(3) If an unused registration No. tool life management system variable is commanded, "-1" will be read as the data.
(4) Once commanded, the group No. is valid until NC reset.

Reading the parameters

System data can be read in with the system variables.

(Note) These can be used only with some models.

Variable No.	Application
#100000	Parameter # designation
#100001	System reserved
#100002	Axis No./spindle No. designation
#100010	Parameter value read

The parameter values are read in with the following four blocks using these four system variables.

#100000 = 1001 ; ....Designates the parameter #.
#100001 = 0 ;
#100002 = 1 ; ....Designates the axis No./spindle No.
#100 = #100010 ; ....Reads the parameter value.

(1) Parameter # designation (#100000)

The parameter to be read in is designated by substituting the parameter # in this system variable.
If the parameters are read without designating this number, the parameters will be read in the same manner as if the minimum parameter # (#1) is designated. Once designated, the setting is held until the parameter # is designated again or until it is reset.
A program error (P39) will occur if a parameter # that does not exist is set.

(2) System reserved (#100001)

Always set "0".
When using the PLC axis, set "10".

(3) Axis/spindle No. designation (#100002)

(a) System variable for axis/spindle No. designation
The axis No./spindle No. of the parameter to be read in is designated by substituting an index value in this system variable. This designation will be ignored when reading in parameters that are not for a specific axis or spindle.
If the parameters are read without designating this number, the parameters will be read in the same manner as if the index value 1 (1st axis/1st spindle) is designated. Once designated, the setting is held until the index value is designated again or until it is reset.
A program error (P39) will occur if an axis/spindle No. that does not exist is set.

(b) Index values

Index values	Axis parameter	Spindle parameter
1	1st axis	1st spindle
2	2nd axis	2nd spindle
3	3rd axis	-
4	4th axis	-

(4)   Parameter read (#100010)

The designated parameter data is read with this system variable.
The following data is read according to the parameter type.

Type	Read data
Value	The values displayed on the Parameter screen are output.
Text	ASCII codes are converted into decimal values.

Example of programs for reading parameters

(1)   To read the parameter [#1002 axisno (Number of axes)]

    #100000 = 1002 ; ....Designates [#1002].
    #100001 = 0 ;
    #101 = #100010 ; ....Reads the number of axes.
    #100001 = 10 ; ....Designates [PLC axis].
    #110 = #100010 ; ....Reads the number of PLC axes.

(2)   To read the parameter [#2037 G53ofs (#1 reference position)]

[Conditions]

	<1st axis>	<2nd axis>
#2037 G53ofs	1000.000	200.000

    #100002 = 1 ; ....Designates [1st axis].
    #100000 = 2037 ; ....Designates [#2037].
    #101 = #100010 ; ....Reads the [#1 reference position] for the 1st axis.
                                         (#101 = 100.000.)
    #100002 = 2 ; ....Designates [2nd axis].
    #102 = #100010 ; ....Reads the [#1 reference position] for the 2nd axis.
                                         (#102 = 200.000.)

(3)   To read each parameter for each axis and spindle

    #100002 = 1 ; ....Designates [1st spindle].
    #100000 = 3001 ; ....Designates [#3001].
    #101 = #100010 ; ....Reads the [#3001 slimt1 (Limit rotation speed gears 00)]
                                         for 1st spindle.
    #100000 = 3002 ; ....Designates [#3002].
    #102 = #100010 ; ....Reads the [#3002 slimt2 (Limit rotation speed gears 01)]
                                         for 1st spindle.
    #100002 = 2 ; ....Designates [2nd spindle].
    #100000 = 3001 ; ....Designates [#3001].
      #201 = #100010 ; ....Reads the [#3001 slimt1 (Limit rotation speed gears 00)]
                                         for 2nd spindle.
    #100000 = 3002 ; ....Designates [#3002].
    #202 = #100010 ; ....Reads the [#3002 slimt2 (Limit rotation speed gears 01)]
                                         for 2nd spindle

Example of parameter read macro program

<Macro specifications>

Q341 A_. Q_ . ;

    A_....... Storage common variable   Designates the common variable No. for storing the data read in.

    Q_....... Parameter # designation   For an axis/spindle parameter, designates the axis/spindle No. with one digit after the decimal point.

<Macro contents>

#100000 = FIX [#17] ;	....	Designates parameter #.
#100002 = FIX [#17∗10] MOD 10 ;	....	Designates axis/spindle No.
#[#1] = #100010 ;	....	Reads parameter data.
M99 ;		

Precautions for reading parameters

(1) The number of axes and spindles is the maximum number in the specifications determined by the model.

(2) The inch/metric changeover function for the setting and display is valid even for the data read in.

Reading PLC data

PLC data can be read in with the system variables.

(Note 1) These can be used only with some models.

(Note 2) The read devices are limited.

Variable No.	Application
#100100	Device type designation
#100101	Device No. designation
#100102	Number of read bytes designation
#100103	Read bit designation
#100110	PLC data read

The PLC data is read in with the following five blocks using these five system variables.

#100100 = 1 ; ....Designates the device type.

#100101 = 0 ; ....Designates the device No.

#100102 = 1 ; ....Designates the number of bytes.

#100103 = 2 ; ....Designates the bit. (Valid only when reading word device bits.)

#100 = #100110 ; ......Reads in the PLC data.

(1)   Device designation (#100100)

(a) System variable for device designation
The type of device to be read in can be designated by substituting the device designation value in this system variable.
If the data is read without designating this number, the data will be read in the same manner as if the minimum device designation value (0: M device) is designated. Once designated, the setting is held until the device is designated again or until it is reset.
A program error (P39) will occur if a device that does not exist is set.

(b) Device designation values

Device designation value	Device	Unit	Device No.	Device designation value	Device	Unit	Device No.
0	M	Bit	M0 to M8191	10	F	Bit	F0 to F127
1	D	Word	D0 to D1023	13	L	Bit	L0 to L255
2	C	Bit	C0 to C127				
4	X *	Bit	X0 to X4BF				
5	Y *	Bit	Y0 to Y53F				
6	R	Word	R0 to R8191				
7	T	Bit	T0 to T255				
9	SM	Bit	SM0 to SM127				

The unit is the amount of data per device No. "Word" has 16 bits, and "Bit" has one bit.
* is a device that expresses the device No. as a hexadecimal.

(Note) The device which is used in 2-part system cannot be used because E60/E68 is 1-part system specifications.

(2) Device No. designation (#100101)

The device to be read in is designated by substituting the device No. in this system variable.
Convert a device expressed as a hexadecimal into a decimal when designating.
If the data is read without designating this number, the data will be read in the same manner as if the minimum device No. (0) is designated. Once designated, the setting is held until the device No. is designated again or until it is reset.
A program error (P39) will occur if a device No. that does not exist is set.

(3) Number of bytes designation (#100102)

(a) System variable for number of bytes designation
The reading size is designated by substituting the number of bytes designation value in this system variable.
If the data is read without designating this number, the data will be read in the same manner as if the minimum number of bytes designation value (0: bit designation) is designated. Once designated, the setting is held until the number of bytes is designated again or until it is reset.
A program error (P39) will occur if a number of bytes that does not exist in the specifications is set.

(b) Number of bytes designation value

Number of bytes designation value	Read in data			Operation	
	Size	Sign	Range	Word device	Bit device
0	1 bit	-	0 to 1	The number of bits designated is read in.	The bits for the designated device No. are read in.
1	1 byte	No	0 to 255	The low-order byte is read in.	8 bits are read in from the designated device No.
101		Yes	-128 to 127		
2	2 bytes	No	0 to 65535	Two bytes are read in.	16 bits are read in from the designated device No.
102		Yes	-32768 to 32767		
4	4 bytes	No	0 to 4294967295	The designated device (L) and next device (H) are read in.	32 bits are read in from the designated device No.
104		Yes	-2147483648 to 2147483647		

0 to 4 are designated without a sign, and 101 to 104 are designated with a sign.

(4)   Bit designation (#100103)

(a)   System variable for bit designation
The bit to be read in is designated by substituting the bit designation value in this system variable.
This designation is valid only when reading the bits for a 16-bit device, and is invalid in all other cases.
If the data is read without designating this number, the data will be read in the same manner as if the minimum bit designation value (0: bit 0) is designated. Once designated, the setting is held until the bit is designated again or until it is reset.
A program error (P39) will occur if a bit that does not exist is set.

(b)   Bit designation values

Bit designation value	Read in bit
0	Bit 0
1	Bit 1
:	:
15	Bit 15

(5)   PLC data read (#100110)

The data for the designated device is read in with this system variable.
Refer to the table for number of bytes designation for details on the range of data read in.

Examples of programs for reading PLC data

(1) To read a bit device

```
#100100 = 0 ; ........... Designates [M device].
#100101 = 0 ; ........... Designates [Device No. 0].
#100102 = 0 ; ........... Designates [Bit].
#100 = #100110 ; ...... Reads M0 (one bit).
#100102 = 1 ; ........... Designates [1 byte].
#101 = #100110 ; ...... Reads M0 to M7 (eight bits).
                        (If M7 to M0 is 0001 0010, this will be #102 = 18 (0x12).)
#100102 = 102 ; ........ Designates [Signed two bytes].
#102 = #100110 ;  ..... Reads M0 to M15 (16 bits).
                        (If M15 to M0 is 1111 1110 1101 1100, this will be
                        #102 = -292 (0xFEDC).)
#100102 = 4 ; ........... Designates [4 bytes].
#104 = #100110 ;  ..... Reads M0 to M31 (32 bits).
                        (If M31 to M0 is 0001 0010 0011 0100 0101 0110 0111 1000,
                        this will be #104 = 305419896 (0x12345678).)
```

(2) To read a word device

```
#100100 = 1 ; ........... Designates [D device].
#100101 = 0 ; ........... Designates [Device No. 0].
#100102 = 0 ; ........... Designates [Bit].
#100103 = 1 ; ........... Designates [Bit 1].
#100 = #100110 ; ...... Reads the D0 bit 1.
                        (If D0 = 0x0102, this will be #101 =1.)
#100102 = 1 ; ........... Designates [1 byte].
#101 = #100110 ;  ..... Reads the low-order byte of D0.
                        (If D0 = 0x0102, this will be #101 =2.)
#100102 = 2 ; ........... Designates [2 bytes].
#102 = #100110 ;  ..... Reads D0. (If D0 = 0x0102, this will be #102 =258.)
#100102 = 104 ; ........ Designates [Signed four bytes].
#104 = #100110 ;  ..... Reads D0 and D1.
                        (If D0 = 0xFFFE and D1 = 0xFFFF, this will be #104 =-2.)
```

Examples of using macro program for reading PLC data

<Macro specifications>

G340 F_. A_. Q_. H_. ;

F_. ....Number of bytes designation	F0 ..... Designates bit.	
	F1 ..... Designates one byte.	
	F2 ..... Designates two bytes.	
A_. ....Device designation	A0 ..... Designates M.	
	A1 ..... Designates D.	
	A2 ..... Designates C.	
	A4 ..... Designates X.	
	A5 ..... Designates Y.	
	A6 ..... Designates R.	
	A7 ..... Designates T.	
Q_.............Device No. designation	For a bit, designates the bit with two digits after the decimal point.	
	For a byte, a decimal value is not designated.	
H_............ Storage common variable	Designates the common variable No. for storing the read data.	

<Macro contents>

```
#100100 = #1 ;                            ......   Designates device.
#100101 = FIX [#17] ;                     ......   Designates device No.
#100102 = #9 ;                            ......   Designates number of bytes.
#100103 = FIX [#17∗100] MOD 100 ;         ......   Designates bit.
#[#11] = #100110 ;                        ......   Reads PLC data.
M99 ;
```

Precautions for reading PLC data

(1) As the PLC data is read asynchronously from the ladder execution, the data is not necessarily from the running program. Take care when reading devices that change.

(2) If reading of a device that does not exist is attempted by designating the device No. and number of bytes, the 0 value will be read in only for the sections that do not exist.

Time reading variables

The following operations can be carried out using the system variable extension for the user macro time.

(1) By adding time information system variable #3011 and #3012, the current date (#3011) and current time (#3012) can be read and written.

(2) By adding parameter #1273/bit1, the unit (millisecond unit/hour unit) of the cumulative time during system variable #3002 automatic start can be changed.

Variable No.	Details
#3001	The cumulative time during power ON can be read and the value can be substituted. The unit is a millisecond unit.
#3002	The cumulative time during automatic start can be read and the value can be substituted. The unit can be changed between millisecond and hour with parameter #1273/bit1.

#3011	The current date can be read and written.
	YYYY/MM/DD is read as a YYYYMMDD value.
	When a YYYYMMDD value is written in, it will be set as YY/MM/DD (the last two digits of the year are displayed).
	Command range for year/month/date setting
	    Year (YYYY) : 2000 to 2099
	    Month (MM) : 1 to 12
	    Date (DD) : 1 to maximum number of days in month

#3012	The current time can be read and written.
	HH/MM/SS is read as a HHMMSS value.
	When an HHMMSS value is written in, it will be set as HH/MM/DD.
	Command range for hour/minute/second setting
	    Hour (HH) : 0 to 23 (24-hour system)
	    Minute (MM) : 0 to 59
	    Second (SS) : 0 to 59

(3) The cumulative time returns to 0 at approx. 2.44×10^{11} milliseconds (approx. 7.7 years).

(4) If a negative value or a value exceeding 244335917226 milliseconds (67871.08811851 hours for #3002 time designation) is set for the cumulative time, a program error (P35) will occur.

(5) If a value exceeding the command range is set for the date or time, a program error (P35) will occur.

(6) Always set the month/date/hour/minute/second as a two-digit value when setting the date and time.
If the value only has one digit, always add 0. (February 14, 2001 $\Rightarrow$ #3001= 20010214 ;, etc.)

Examples of using time reading variable

(Example 1) To read the current date (February 14, 2001) in common variable #100
#100 = #3011 ; (20010214 is inserted in #100)

(Example 2) To write current time (18 hours, 13 minutes, 6 seconds) into system variable #3012
#3012 = 181306 ; (The command value cumulative time #2: time is set to 18:13:06.)

(Example 3) By setting the following type of program, the machining start/end time
(year/month/date/hour/minute/second) can be viewed.

```
#100=#3011 ;   ⇒ Machining start year/month/date
#101=#3012 ;   ⇒ Machining start hour/minute/second
G28 X0 Y0 Z0 ;
G92 ;
G0 X50. ;
    .
    .
    .
#102=#3011 ;   ⇒ Machining end year/month/date
#103=#3012 ;   ⇒ Machining end hour/minute/second
M30 ;
```

Restrictions and precautions for using time reading variable

(1) #3011 reads the date as an eight-digit value, so the difference of the two dates read in will not be the difference of days.

(2) #3012 reads the time as a six-digit value, so the difference of the two times read in will not be the difference of hours.

13.5.6　Arithmetic commands

A variety of arithmetic operations can be performed between variables.

Command format

#i = <formula>

<Formula> is a combination of constants, variables, functions and operators.
Constants can be used instead of #j and #k below.

(1)	Definition and substitution of variables	#i = #j	Definition, substitution
(2)	Addition arithmetic	#i = #j + #k	Addition
		#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	Multiplication
		#i = #j / #k	Division
		#i = #j MOD #k	Remainder
		#i = #j AND #k	Logical product (at every bit of 32 bits)
(4)	Functions	#i = SIN [#k]	Sine
		#i = COS [#k]	Cosine
		#i = TAN [#k]	Tangent (sin/cos used for tan)
		#i = ASIN [#k]	Arcsine
		#i = ATAN [#j]	Arctangent (ATAN or ATN may be used)
		#i = ACOS [#j]	Arc-cosine
		#i = SQRT [#k]	Square root (SQRT or SQR may be used)
		#i = ABS [#k]	Absolute value
		#i = BIN [#k]	Conversion from BCD to BIN
		#i = BCD [#k]	Conversion from BIN to BCD
		#i = ROUND [#k]	Rounding off (ROUND or RND may be used)
		#i = FIX [#k]	Discard fractions less than 1
		#i = FUP [#k]	Add for fractions less than 1
		#i = LN [#k]	Natural logarithm
		#i = EXP [#k]	Exponent with e (=2.718 .....) as bottom

(Note 1) A value without a decimal point is basically treated as a value with a decimal point at the end (1 = 1.000).

(Note 2) Offset amounts from #10001 and work coordinate system offset values from #5201 are handled as data with a decimal point. Consequently, data with a decimal point will be produced even when data without a decimal point have been defined in the variable numbers.

(Example)

```
#101    = 1000  ;        Common variables
                          after execution       #101   1000.000
#10001  = #101  ;                                #102   1000.000
#102    = #10001 ;
```
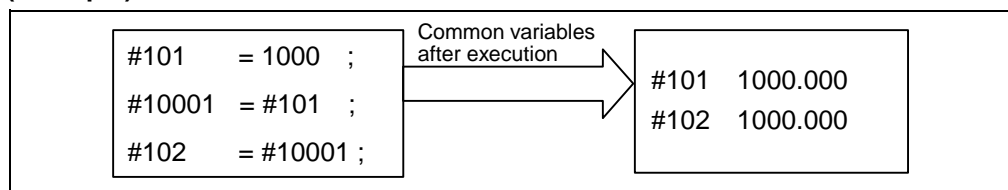
(Note 3) The <formula> after a function must be enclosed in the square parentheses.

Sequence of arithmetic operations

(1) The sequence of the arithmetic operations (1) through (3) is, respectively, the functions followed by the multiplication arithmetic followed in turn by the addition arithmetic.

#101 = #111 + #112∗SIN[#113]

    (1)   Function

    (2)   Multiplication arithmetic

    (3)   Addition arithmetic

(2) The part to be given priority in the operation sequence should be enclosed in square parentheses. Up to 5 pairs of such parentheses including those for the functions may be used.

#101 = SQRT [[[#111 = #112] ∗SIN[#113] + #114] ∗#15] ;

First pair of parentheses

Second pair of parentheses

Third pair of parentheses

Examples of operation commands

(1)	Main program and argument designation	G65 P100 A10 B20. ; #101=100.000 #102=200.000 ;	#1 10.000 #2 20.000 #101 100.000 #102 200.000
(2)	Definition and substitution (=)	#1=1000 #2=1000. #3=#101 #4=#102 #5=#10001 (#10001=-10.)	#1 1000.000 #2 1000.000 #3 100.000 ⎫ #4 200.000 ⎬ From common variables #5 -10.000 ... From offset amount
(3)	Addition and subtraction (+, -)	#11=#1+1000 #12=#2-50. #13=#101+#1 #14=#10001-3. (#10001 = -10.) #15=#10001+#102	#11 2000.000 #12 950.000 #13 1100.000 #14 -13.000 #15 190.000
(4)	Multiplication and division (∗, /)	#21=100∗100 #22=100.∗100 #23=100∗100 #24=100.∗100. #25=100/100 #26=100./100 #27=100/100. #28=100./100. #29=#10001∗#101 (#10001 = -10.) #30=#10001/#102	#21 10000.000 #22 10000.000 #23 10000.000 #24 10000.000 #25 1.000 #26 1.000 #27 1.000 #28 1.000 #29 -1000.000 #30 -0.050
(5)	Remainder (MOD)	#19=48 #20=9 #31=#19 MOD #20	#19/#20 = 48/9 = 5 with 3 over #31 = 3
(6)	Logical sum (OR)	#3=100 #4=#3 OR 14	#3 = 01100100 (binary) 14 = 00001110 (binary) #4 = 01101110 = 110
(7)	Exclusive OR (XOR)	#3=100 #4=#3 XOR 14	#3 = 01100100 (binary) 14 = 00001110 (binary) #4 = 01101010 = 106
(8)	Logical product (AND)	#9=100 #10=#9 AND 15	#9 = 01100100 (binary) 15 = 00001111 (binary) #10 = 00000100 = 4

(9)	Sin (SIN)	#501 = SIN [60] #502 = SIN [60.] #503 = 1000∗SIN [60] #504 = 1000∗SIN [60.] #505 = 1000.∗SIN [60] #506 = 1000.∗SIN [60.] **(Note)** SIN [60] is equivalent to SIN [60.]	#501 #502 #503 #504 #505 #506	0.860 0.860 866.025 866.025 866.025 866.025
(10)	Cosine (COS)	#541 = COS [45] #542 = COS [45.] #543 = 1000∗COS [45] #544 = 1000∗COS [45.] #545 = 1000.∗COS [45] #546 = 1000.∗COS [45.] **(Note)** COS [45] is equivalent to COS [45.]	#541 #542 #543 #544 #545 #546	0.707 0.707 707.107 707.107 707.107 707.107
(11)	Tangent (TAN)	#551 = TAN [60] #552 = TAN [60.] #553 = 1000∗TAN [60] #554 = 1000∗TAN [60.] #555 = 1000.∗TAN [60] #556 = 1000.∗TAN [60.] **(Note)** TAN [60] is equivalent to TAN [60.]	#551 #552 #553 #554 #555 #556	1.732 1.732 1732.051 1732.051 1732.051 1732.051
(12)	Arcsine ASIN	#531 = ASIN [100.500/201.] #532 = ASIN [100.500/201] #533 = ASIN [0.500] #534 = ASIN [-0.500]	#531 #432 #533 #534 **(Note)**	30.000 30.000 30.000 -30.000 When #1273/bit 0 is set to 1, #534 will be 330⁰.
(13)	Arctangent (ATAN or ATN)	#561 = ATAN [173205/100000] #562 = ATAN [173205/100000.] #563 = ATAN [173.205/100] #564 = ATAN [173.205/100.] #565 = ATAN [1.73205]	#561 #562 #563 #564 #565	60.000 60.000 60.000 60.000 60.000
(14)	Arccosine (ACOS)	#521 = ACOS [100./141.421] #522 = ACOS [100./141.421]	#521 #522	45.000 45.000
(15)	Square root (SQR or SQRT)	#571 = SQRT [1000] #572 = SQRT [1000.] #573 = SQRT [10. ∗10. +20. ∗20.] **(Note)** In order to increase the accuracy, proceed with the operation inside parentheses.	#571 #572 #573	31.623 31.623 22.360
(16)	Absolute value (ABS)	#576 = -1000 #577 = ABS [#576] #3 = 70. #4 = -50. #580 = ABS [#4 - #3]	#576 #577 #580	-1000.000 1000.000 120.000
(17)	BIN, BCD	#1 = 100 #11 = BIN [#1] #12 = BCD [#1]	 #11 #12	 64 256

(18)	Rounding off (ROUND or RND)	#21 = ROUND [14/3] #22 = ROUND [14./3] #23 = ROUND [14/3.] #24 = ROUND [14./3.] #25 = ROUND [-14/3] #26 = ROUND [-14./3] #27 = ROUND [-14/3.] #28 = ROUND [-14./3.]	#21 #22 #23 #24 #25 #26 #27 #28	5 5 5 5 -5 -5 -5 -5
(19)	Discarding fractions below decimal point (FIX)	#21 = FIX [14/3] #22 = FIX [14./3] #23 = FIX [14/3.] #24 = FIX [14./3.] #25 = FIX [-14/3] #26 = FIX [-14./3] #27 = FIX [-14/3.] #28 = FIX [-14./3.]	#21 #22 #23 #24 #25 #26 #27 #28	4.000 4.000 4.000 4.000 -4.000 -4.000 -4.000 -4.000
(20)	Adding fractions less than 1 (FUP)	#21 = FUP [14/3] #22 = FUP [14./3] #23 = FUP [14/3.] #24 = FUP [14./3.] #25 = FUP [-14/3] #26 = FUP [-14./3] #27 = FUP [-14/3.] #28 = FUP [-14./3.]	#21 #22 #23 #24 #25 #26 #27 #28	5.000 5.000 5.000 5.000 -5.000 -5.000 -5.000 -5.000
(21)	Natural logarithms (LN)	#101 = LN [5] #102 = LN [0.5] #103 = LN [-5]	#101 #102 Error	1.609 -0.693 "P282"
(22)	Exponents (EXP)	#104 = EXP [2] #105 = EXP [1] #106 = EXP [-2]	#104 #105 #106	7.389 2.718 0.135

Arithmetic accuracy

As shown in the following table, errors will be generated when performing arithmetic operations once and these errors will accumulate by repeating the operations.

Arithmetic format	Average error	Maximum error	Type of error
a = b + c a = b − c	2.33×10^{-10}	5.32×10^{-10}	Min. $|\varepsilon/b|$, $|\varepsilon/c|$
a = b∗c	1.55×10^{-10}	4.66×10^{-10}	Relative error $|\varepsilon/a|$
a = b/c	4.66×10^{-10}	1.86×10^{-9}	
a = $\sqrt{b}$	1.24×10^{-9}	3.73×10^{-9}	
a = SIN [b] a = COS [b]	5.0×10^{-9}	1.0×10^{-8}	Absolute error $|\varepsilon|°$
a = ATAN [b/c]	1.8×10^{-6}	3.6×10^{-6}	

(Note)   SIN/COS is calculated for the function TAN.

Notes on reduced accuracy

(1)   Addition and subtraction

It should be noted that when absolute values are used subtractively in addition or subtraction, the relative error cannot be kept below 10^{-8}.
For instance, it is assumed that the real values produced as the arithmetic calculation result of #10 and #20 are as follows (these values cannot be substituted directly) :

#10 = 2345678988888.888

#20 = 2345678901234.567

Performing #10 - #20 will not produced #10 - 320 = 87654.321. There are 8 significant digits in the variables and so the values of #10 and #20 will be as follows (strictly speaking, the internal values will differ somewhat from the values below because they are binary numbers) :

#10 = 2345679000000.000

#20 = 2345678900000.000

Consequently, #10 - #20 = 100000.000 will generate a large error.

(2)   Logical operations

EQ, NE, GT, LT, GE and LE are basically the same as addition and subtraction and so care should be taken with errors. For instance, to determine whether or not #10 and #20 are equal in the above example :

IF [#10EQ#20]

It is not always possible to provide proper evaluation because of the above mentioned error. Therefore, when the error is evaluated as in the following expression :

IF [ABS [#10 - #20] LT200000]

and the difference between #10 and #20 falls within the designated range error, both values should be considered equal.

(3)   Trigonometric functions

Absolute errors are guaranteed with trigonometric functions but since the relative error is not under 10^{-8}, care should be taken when dividing or multiplying after having used a trigonometric function.

13.5.7 Control commands

The flow of programs can be controlled by IF-GOTO- and WHILE-DO-.

Branching

Format

IF [conditional expression] GOTO n; (n = sequence number in the program)

When the condition is satisfied, control branches to "n" and when it is not satisfied, the next block is executed.
IF [conditional expression] can be omitted and, when it is, control passes to "n" unconditionally.
The following types of [conditional expressions] are available.

#i EQ #j	=	When #i and #j are equal
#i NE #j	≠	When #i and #j are not equal
#i GT #j	>	When #i is greater than #j
#i LT #j	<	When #i is less than #j
#i GE #j	≥	When #i is #j or more
#i LE #j	≤	When #i is #j or less

"n" of GOTO n must always be in the same program. Program error (P231) will result if it is not. A formula or variable can be used instead of #i, #j and "n".
In the block with sequence number "n" which will be executed after a GOTO n command, the sequence number must always be at the head of the block.
Otherwise, program error (P231) will result.
If "/" is at the head of the block and Nn follows, control can be branched to the sequence number.
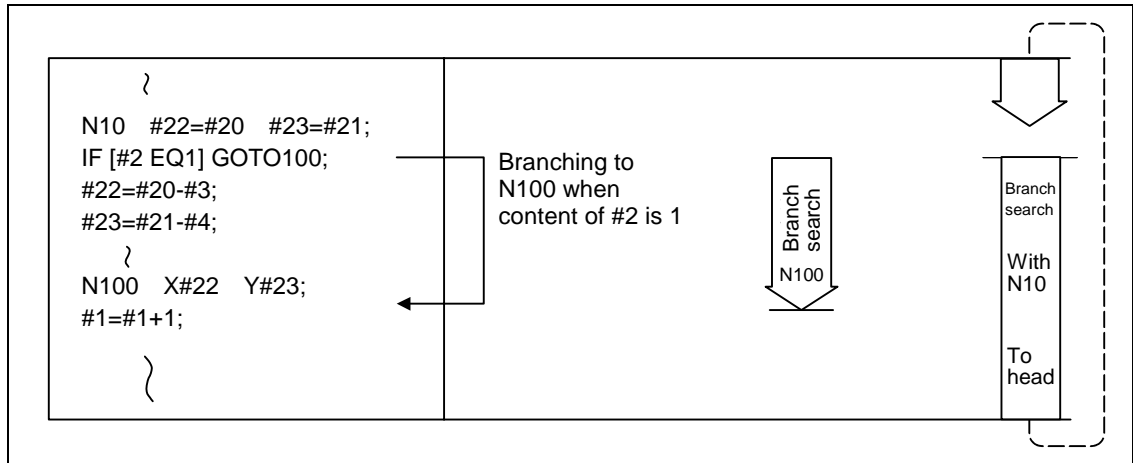
(Note 1) When the sequence number of the branch destination is searched, the search is conducted up to the end of the program (% code) from the block following IF; and if it is not found, it is then conducted from the top of the program to the block before IF;. Therefore, branch searches in the opposite direction to the program flow will take longer to execute compared with branch searches in the forward direction.

(Note 2) EQ and NE should be used only for integers. For comparison of numeric values with decimals, GE, GT, LE, and LT should be used.

Iteration

Format

> **WHILE [conditional expression] DOm ; (m = 1, 2, 3 ..... 127)**
>
>   ⟩
>
>   **END m ;**

While the conditional expression is established, the blocks from the following block to ENDm are repeatedly executed; when it is not established, execution moves to the block after ENDm. DOm may come before WHILE, WHILE [conditional expression] DOm and ENDm must be used as a pair. IF WHILE [conditional expression] is omitted, these blocks will be repeatedly ad infinitum. The repeating identification numbers range from 1 through 127 (DO1, DO2, DO3, ... DO127). Up to 27 nesting levels can be used.

(1) Same identifier number can be used any number of times.	(2) Any number may be used for the WHILE −DOm identifier number.

(1)

Possible

```
┌─ WHILE ~ DO1 ;  ─┐
│                  │
│  END1 ;          │
└──────────────────┘
```

Possible

```
┌─ WHILE ~ DO1 ;  ─┐
│                  │
│  END1 ;          │
└──────────────────┘
```

(2)

Possible

```
┌─ WHILE ~ DO1 ;  ─┐
│       ⟩          │
│  END1 ;          │
└──────────────────┘
        ⟩
┌┄ WHILE ~ DO3 ;  ┄┐
┆       ⟩          ┆
┆  END3 ;          ┆
└┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┘
        ⟩
┌┄ WHILE ~ DO2 ;  ┄┐
┆       ⟩          ┆
┆  END2 ;          ┆
└┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┘
        ⟩
┌┄ WHILE ~ DO1 ;  ┄┐
┆       ⟩          ┆
┆  END1 ;          ┆
└┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┄┘
```

(3) Up to 27 nesting levels for WHILE− DOm. "m" is any number from 1 to 127 for the nesting depth.	(4) The number of WHILE − DOm nesting levels cannot exceed 27.

(3) Possible

```
┌─WHILE ~ DO1 ; ─┐         DO1
│ ┌─WHILE ~ DO2 ; ─┐        DO2
│ │     :         │
│ │ ┌─WHILE~DO27; ─┐        DO27
│ │ │             │
│ │ │  END 27 ;   │
│ │ └─────────────┘
│ │     :
│ │  END 2 ;
│ └─────────────────┘
│  END 1 ;
└───────────────────┘
```
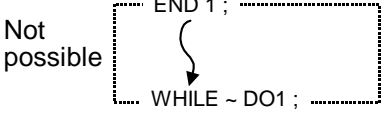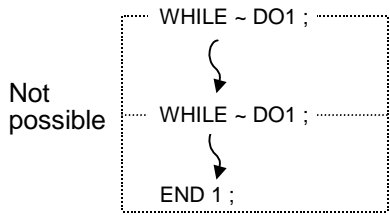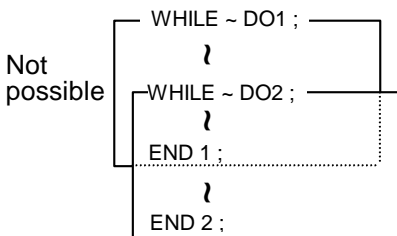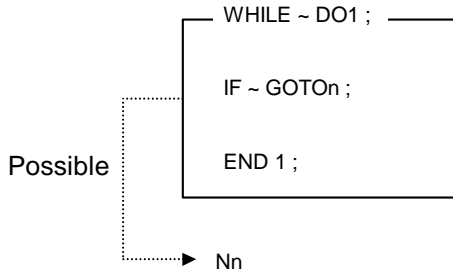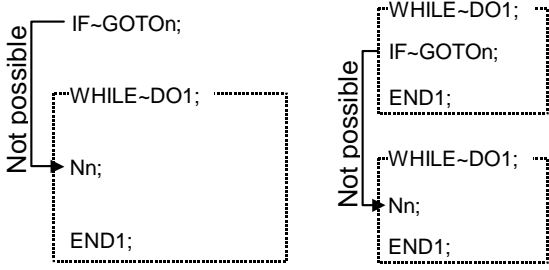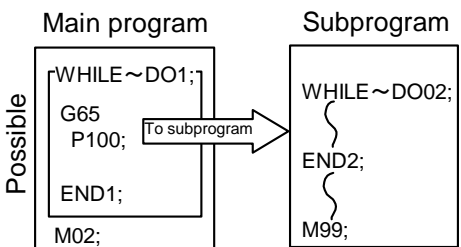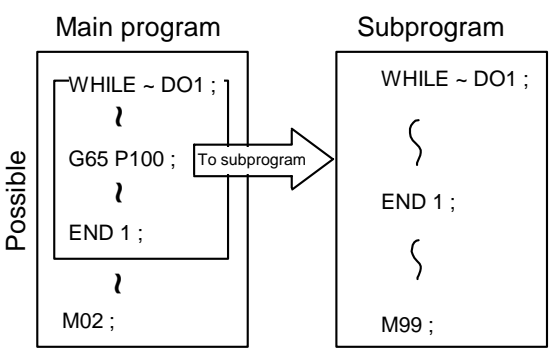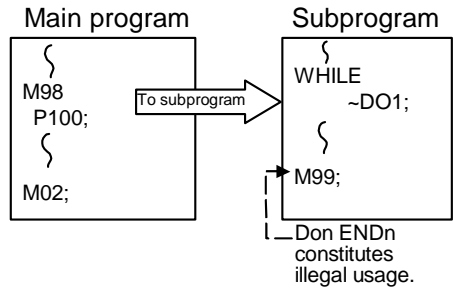
(Note) :With nesting, "m" which has been used once cannot be used.

(4) Not possible

```
┌── WHILE ~ DO1 ; ──┐
│ ┌─ WHILE ~ DO2 ; ─┐
│ │    :           │
│ │ ┌ WHILE ~ DO3 ; ┐
│ │ │    ⟩          │
│ │ │ ┌WHILE ~ DO28;┐
│ │ │ │    ⟩        │
│ │ │ │ END 28;     │
│ │ │ └─────────────┘
│ │ │    ⟩
│ │ │  END 3 ;
│ │ │    .
│ │  END 2 ;
│ │    ⟩
│  END 1 ;
└───────────────────┘
```

(5) WHILE - DOm must be designated first and ENDm last.

Not possible

```
┌─ END 1 ;
│   ⟨
└─ WHILE ~ DO1 ;
```

(6) WHILE - DOm and ENDm must correspond on a 1:1 (pairing) basis in the same program.

Not possible

```
┌┄ WHILE ~ DO1 ; ┄┐
│        ↓        │
│   WHILE ~ DO1 ; │
│        ↓        │
└┄ END 1 ; ┄┄┄┄┄┄┘
```

(7) Two WHILE - DOm's must not overlap.

Not possible

```
┌─ WHILE ~ DO1 ; ─┐
│       ⟨         │
│┌─WHILE ~ DO2 ;  │
││      ⟨         │
│└─ END 1 ;       │
│       ⟨         │
└─ END 2 ;
```

(8) Branching externally is possible from the WHILE − DOm range.

Possible

```
┌─ WHILE ~ DO1 ; ─┐
│                 │
│   IF ~ GOTOn ;  │
│                 │
│   END 1 ;       │
│                 │
└─────────────────┘
        ┄► Nn
```

(9) No branching is possible inside WHILE − DOm.

Not possible

```
┌─ IF~GOTOn;
│
│  WHILE~DO1; ┄┄┐
│              │
└─► Nn;        │
              │
   END1; ┄┄┄┄┄┘
```

Not possible

```
┌┄ WHILE~DO1; ┄┐
│  IF~GOTOn;   │
│  END1; ┄┄┄┄┄┘
│
│┄ WHILE~DO1; ┄┐
└─► Nn;        │
   END1; ┄┄┄┄┄┘
```

(10) Subprograms can be called by M98, G65 or G66 between WHILE - DOm's.

Possible

Main program

```
┌─WHILE~DO1;
│ G65
│  P100;        To subprogram ⇒
│ END1;
└ M02;
```

Subprogram

```
WHILE~DO02;
 ⟨
END2;
 ⟨
M99;
```

(11) Calls can be initiated by G65 or G66 between WHILE - DOm's and commands can be issued again from 1. Up to 27 nesting levels are possible for the main program and subprograms.

Possible

Main program

```
┌─WHILE ~ DO1 ;
│      ⟨
│ G65 P100 ;    To subprogram ⇒
│      ⟨
│ END 1 ;
│      ⟨
└ M02 ;
```

Subprogram

```
WHILE ~ DO1 ;
     ⟨
END 1 ;
     ⟨
M99 ;
```

(12) A program error will occur at M99 if WHILE and END are not paired in the subprogram (including macro subprogram).

Main program

```
│   ⟨
│ M98
│  P100;         To subprogram ⇒
│   ⟨
│ M02;
```

Subprogram

```
 ⟨
WHILE
     ~DO1;
 ⟨
M99;
```

└─Don ENDn constitutes illegal usage.

(Note)   As the fixed cycles G73 and G83 and the special fixed cycle G34 use WHILE, these will be added multiple times.

13.5.8 External output commands
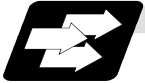
Function and purpose

Besides the standard user macro commands, the following macro instructions are also available as external output commands. They are designed to output the variable values or characters via the RS-232C interface.

Command format

```
POPEN   For preparing the processing of data outputs
PCLOS   For terminating the processing of data outputs
BPRNT   For character output and variable value binary output
DPRNT   For character output and digit-by-digit variable numerical output
```

Command sequence

POPEN	·············· Open command
↓	
DPRNT	·············· Data output command
↓	
PCLOS	·············· Closed command
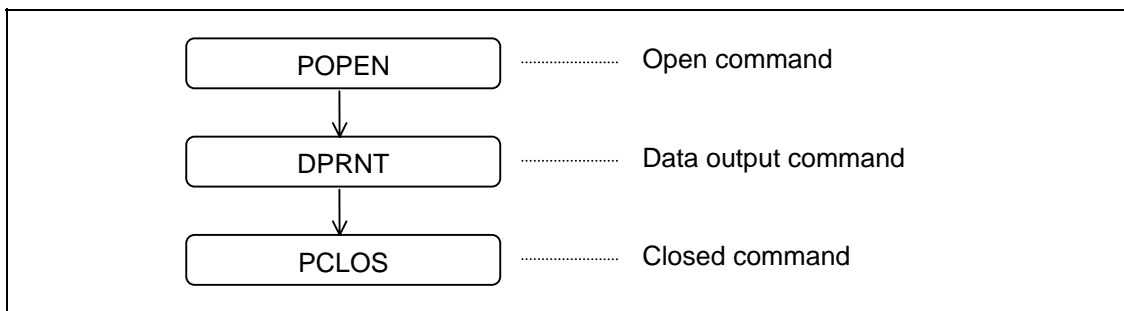
Open command : POPEN

(1) The command is issued before the series of data output commands.
(2) The DC2 control code and % code are output from the NC system to the external output device.
(3) Once POPEN; has been issued, it will remain valid until PCLOS; is issued.

Close command : PCLOS

(1) This command is issued when all the data outputs are completed.
(2) The DC4 control code and % code are output from the NC unit to the external output device.
(3) This command is used together with the open command and it should not be issued unless the open mode has been established.
(4) Issue the close command at the end of the program even when operation has been suspended by resetting or some other operation during data output.

Data output command :   DPRNT

DPRNT [/1 # v1 [d1 c1] / 2 # v2 [d2 c2] ……….]

l1   : Character string
v1   : Variable number
d1   : Significant digits above decimal point
c1   : Significant digits below decimal point   } $c + d \leq 8$

---

(1) The character output and decimal output of the variable values are done with ISO codes.
(2) The commanded character string is output as is by the ISO code.
     Alphanumerics (A to Z, 0 to 9) and special characters (+, −, ∗, /) can be used.
(3) The required significant digits above and below the decimal point in the variable values are commanded within square parentheses. As a result, the variable values equivalent to the commanded number of digits including the decimal point are output in ISO code in decimal notation from the high-order digits. Trailing zeroes are not omitted.
(4) Leading zeroes are suppressed.
     The leading zeroes can also be replaced by blank if so specified with a parameter. This can justify printed data on the last column.

13.5.9 Precautions

Precautions

When the user macro commands are employed, it is possible to use the M, S, T and other NC control commands together with the arithmetic, decision, branching and other macro commands for preparing the machining programs. When the former commands are made into executable statements and the latter commands into macro statements, the macro statement processing should be accomplished as quickly as possible in order to minimize the machining time, because such processing is not directly related to machine control.

As a result, the parameter #8101 "macro single" can be set and the macro statements can be processed in parallel with the execution of the executable statement.

(The parameter can be set OFF during normal machining to process all the macro statements together or set ON during a program check to execute the macro statements block by block. This enables the setting to be made in accordance with the intended objective in mind.)

Example of program

G91G28X0Y0Z0 ;	·····(1)
G92X0Y0Z0 ;	·····(2)
G00X-100.Y-100. ;	·····(3)
#101=100.∗ COS [210.] ;	·····(4) — Macro statement
#102=100.∗ SIN [210.] ;	·····(5)
G01X#101Y#102F800 ;	·····(6)

Macro statements are:
(1) Arithmetic commands (block including =)
(2) Control commands (block including GOTO, DO-END, etc.)
(3) Macro call commands (including macro calls based on G codes and cancel commands (G65, G66, G66.1, G67))
Executable statements indicate statements other than macro statements.
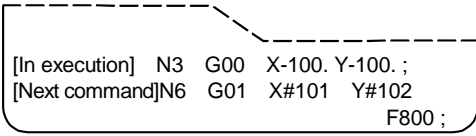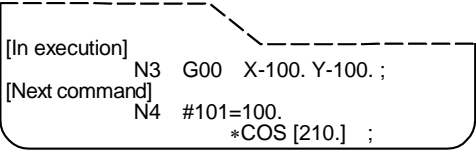
Flow of processing

282
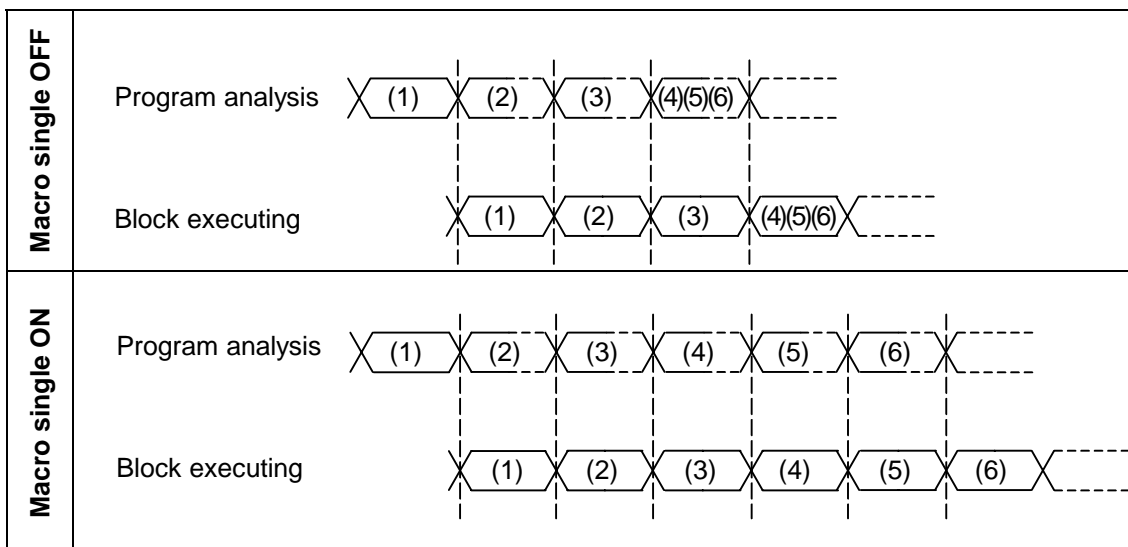
Machining program display

Macro single ON	[In execution] N3 G00 X-100. Y-100. ; [Next command]N6 G01 X#101 Y#102 F800 ;	N4, N5 and N6 are processed in parallel with the control of the executable statement of N3, N6 is an executable statement and so it is displayed as the next command. If the N4, N5 and N6 analysis is in time during N3 control, the machine movement will be continuously controlled.

Macro single OFF	[In execution]
	          N3 G00 X-100. Y-100. ;
	[Next command]
	          N4 #101=100.
	                  ∗COS [210.] ;
		N4 is processed in parallel with the control of the NC executable statement of N3, and it is displayed as the next command. N5 and N6 is executed after N3 has finished, and so the machine control is held on standby during the N5 and N6 analysis time.

13.5.10 Actual examples of using user macros

The following three examples will be described.
(Example 1) SIN curve
(Example 2) Bolt hole circle
(Example 3) Grid

(Example 1) SIN curve

G65 Pp1 Aa1 Bb1 Cc1 Ff1 ;

a1；Initial value 0°

b1；Final value 360°

c1；R of %∗SINθ

f1 ；Feedrate

(SINθ) Y

100.

0   90.   180.   270.   360.   X

-100.

Main program

⋮
G65P9910A0B360.C100.F100;
⋮

Local variable set by argument

#1=0
#2=360.000
#3=100.000
#9=100.000

To subprogram

O9910 (Subprogram)

WHILE [#1LE#2] DO1;
#101=#3∗SIN [#1] ;
G90G01X#1Y#10F#9;   } (Note 1)
#1=#1+10.;
END1;
M99;

(Note 1) Commanding with one block is possible when
G90G01X#1Y [#3∗SIN [#1]] F#9 ; is issued.
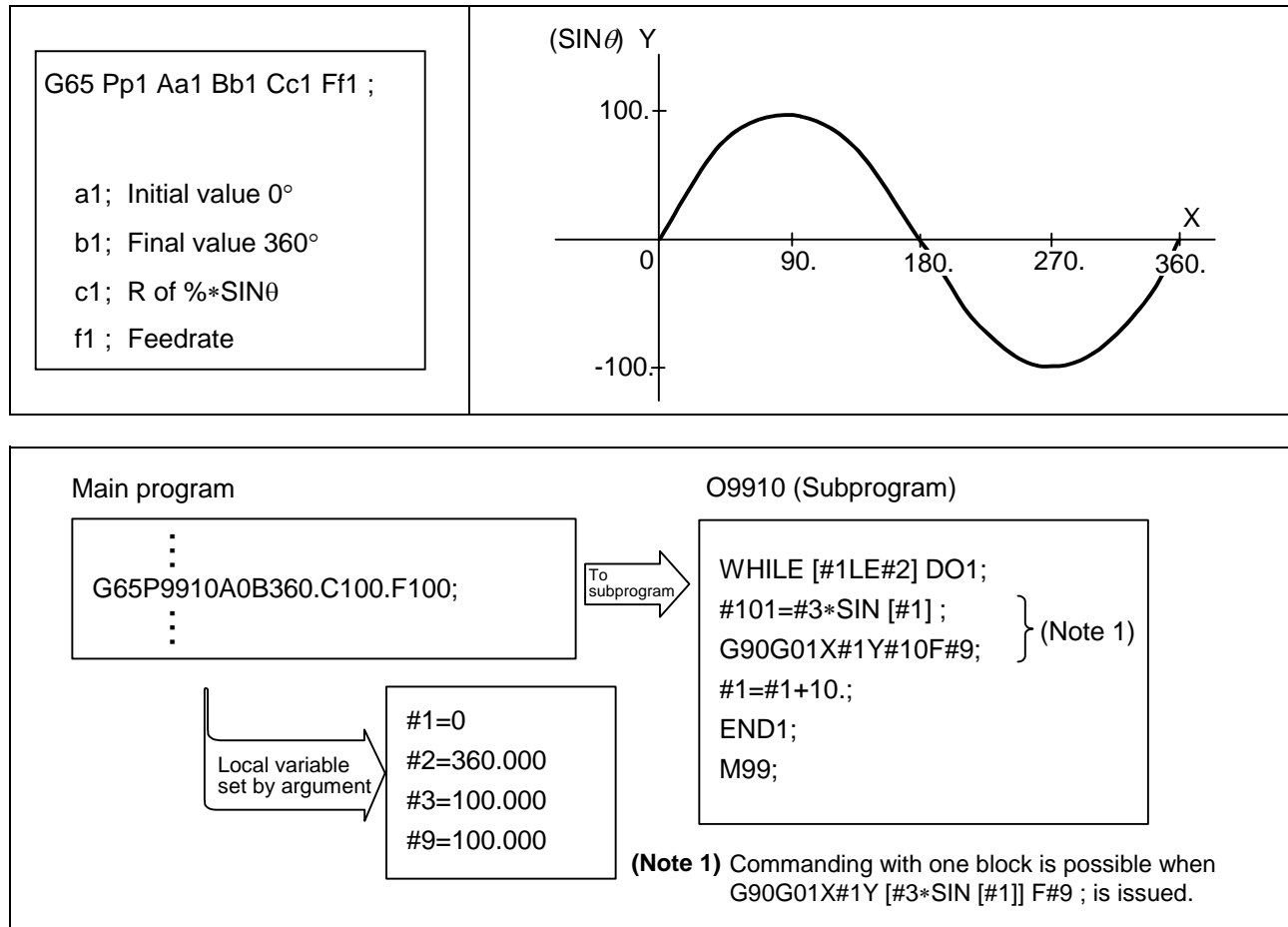
(Example 2) Bolt hole circle

After defining the hole data with fixed cycle (G72 to G89), the macro command is issued as the hole position command.

Main program

```
      〉
G81Z−100.R50.F300L0
G65P9920Aa1Bb1Rr1Xx1Yy1 ;
      〉
```

To subprogram

a1 ; Start angle
b1 ; No. of holes
r1 ; Radius
x1 ; X axis center position
y1 ; Y axis center position
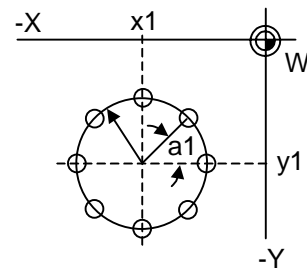
O9920 (Subprogram)

```
#101=0 ;
#102=#4003 ;
#103=#5001 ;            (Note 1)
#104=#5002 ;
#111=#1 ;

WHILE [#101LT#2] DO1 ;

#120=#24+#18∗COS [#111] ;
#121=#25+#18∗SIN [#111] ;   (Note 1)

#122=#120 #123=#121 ;
IF [#102EQ90] GOTO100 ;

#122=#120 − #103 ;
#123=#121 − #104 ;    (Note 1)

#103=#120
#104=#121       (Note 1)

N100   X#122Y#123 ;
#101=#101+1 ;           (Note 1)
#111=#1+360.∗#101/#2 ;

 END1 ;
 M99 ;
```

(Note 1) The processing time can be shortened by programming in one block.

O9920

```
0 → #101
G90, G91 mode
Read in → #102
Read previous coordinates
X → #103
Y → #104
Start angle → 111
```
#101 = No. of hole count

#102 = G90 or G91

#103 = X axis current position
#104 = Y axis current position
#111 = Start angle

#101 ≤ No. of holes — N → END

Y

```
Radius∗COS [#111]
+ Center coordinates X→#120
Radius∗SIN [#111]
+ Center coordinates Y→#121
#120 → #122
#121 → #123
```
#120 = Hole position X coordinates
#121 = Hole position Y coordinates
#122 = X axis absolute value
#123 = Y axis absolute value

#102=90 — Y

Judgment of G90, G91 mode

N

```
#120-#103 → #122
#121-#104 → #123
#120 → #103
#121 → #104
```
#122 = X axis incremental value
123 = Y axis incremental value
X axis current position update
Y axis current position update

N100X#122Y#123 — Drilling command

```
#101+1 → #101
360 deg.∗#101/
No. of holes+#1 → #111
```
No.of holes counter up

#111 = Hole position angle

```
G28 X0 Y0 Z0;
T1 M06;
G90 G43 Z100.H01;
G54 G00 X0 Y0;
G81 Z-100.R3.F100 L0 M03;
G65 P9920 X-500. Y-500. A0 B8 R100.;    To subprogram
G65 P9920 X-500. Y-500. A0 B8 R200.;    To subprogram
G65 P9920 X-500. Y-500. A0 B8 R300.;    To subprogram
      ⋮
```
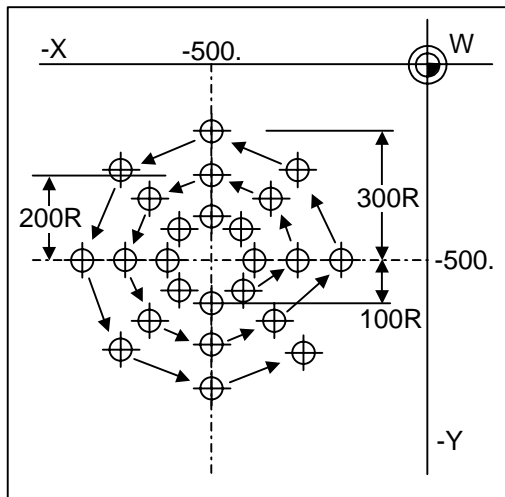

(Example 3) Grid

After defining the hole data with the fixed cycle (G72 to G89), macro call is commanded as a hole position command.

```
G81 Zz1 Rr1 Ff1;
G65Pp1 Xx1 Yy1 Ii1 Jj1 Aa1 Bb1;

    x1 ; X axis hole position
    y1 ; Y axis hole position
    i1 ; X axis interval
    j1 ; Y axis interval
    a1 ; No. of holes in X direction
    b1 ; No. of holes in Y direction
```

Subprogram is on next page
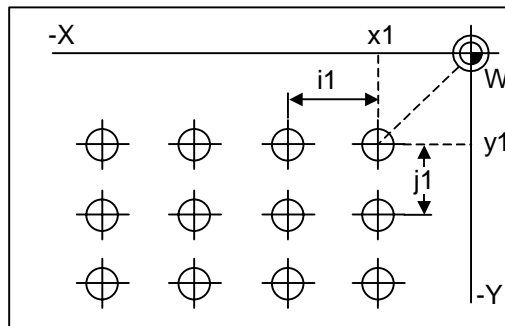

```
G28 X0 Y0 Z0;
T1 M06;
G90 G43 Z100.H01;
G54 G00 X0 Y0;
G81 Z-100. R3.F100 L0 M03;
G65 P9930 X0 Y0 I-100. J-75. A5B3;    To subprogram
      ⟅
G84 Z-90. R3. F250 M03;
G65 P9930 X0 I-100. J-75. A5B3;    To subprogram
      ⟅
```


286
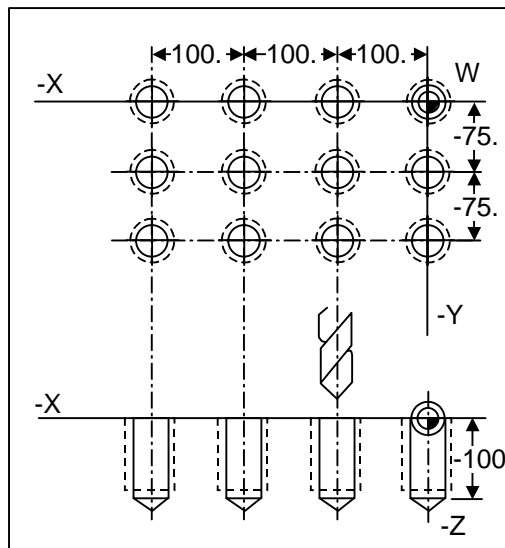
O9930 (Subprogram)　　　　　　　　O9930

```
#101=#24 ;
#102=#25 ;


#103=#4 ;
#104=#5 ;


#106=#2 ;
```
(Note 1)

```
WHILE [#106GT0] DO1 ;

#105=#1 ;

WHILE [#105GT0] DO2 ;



G90 X#101 Y#102 ;



#101=#101+#103 ;
#105=#105−1 ;
```
(Note 1)
```
END2 ;


#101=#101-#103;
#102=#102+#104;
```
(Note 1)
```
#103=−#103 ;
#106=#106−1 ;



END1 ;



M99 ;
```

Start point X coordinates	: $x_1 \rightarrow$ #101
Start point Y coordinates	: $y_1 \rightarrow$ #102
X axis interval	: $i_1 \rightarrow$ #103
Y axis interval	: $j_1 \rightarrow$ #104
No. of holes in Y direction	: $b_1 \rightarrow$ #106

#101 = X axis start point
#102 = Y direction interval
#103 = X direction interval
#106 = No. of holes in
　　　　Y direction

#106 > 0 — N → END　　No. of holes in Y direction

Y

Y direction drilling completion check　　No. of holes in X direction set

#105 > 0　　No. of holes in Y direction check

X#101　Y#102　　Positioning, drilling

#101 + #103 → #101　　X coordinates update

#105 − 1 → #105　　No. of holes in X direction −1

#101 − #103 → #101
#102 + #104 → #102　　X coordinates revision / Y coordinates update

−#103 → #103　　X axis drilling direction reversal

#106 − 1 → #106　　No. of holes in Y direction −1

(Note 1)　The processing time can be shortened by programming in one block.

287
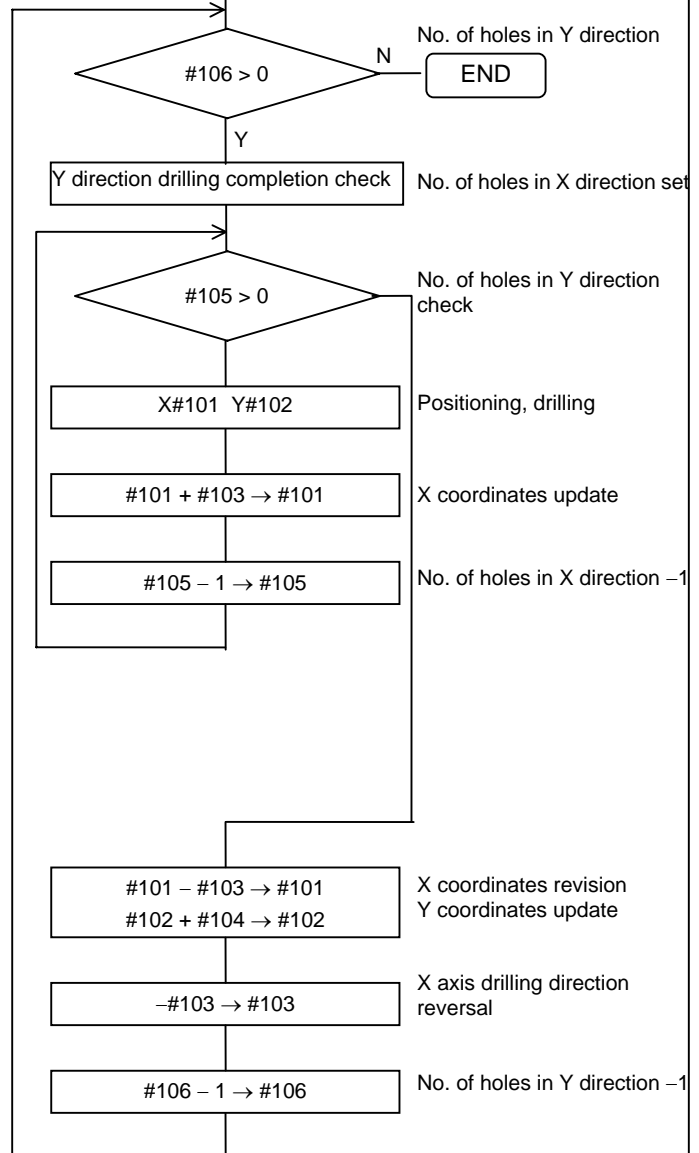
13.6 G command mirror image; G50.1, G51.1
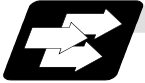
Function and purpose

When cutting a shape that is symmetrical on the left and right, programming time can be shortened by machining the one side and then using the same program to machine the other side. The mirror image function is effective for this.

For example, when using a program as shown below to machine the shape on the left side, a symmetrical shape can be machined on the right side by applying mirror image and executing the program.
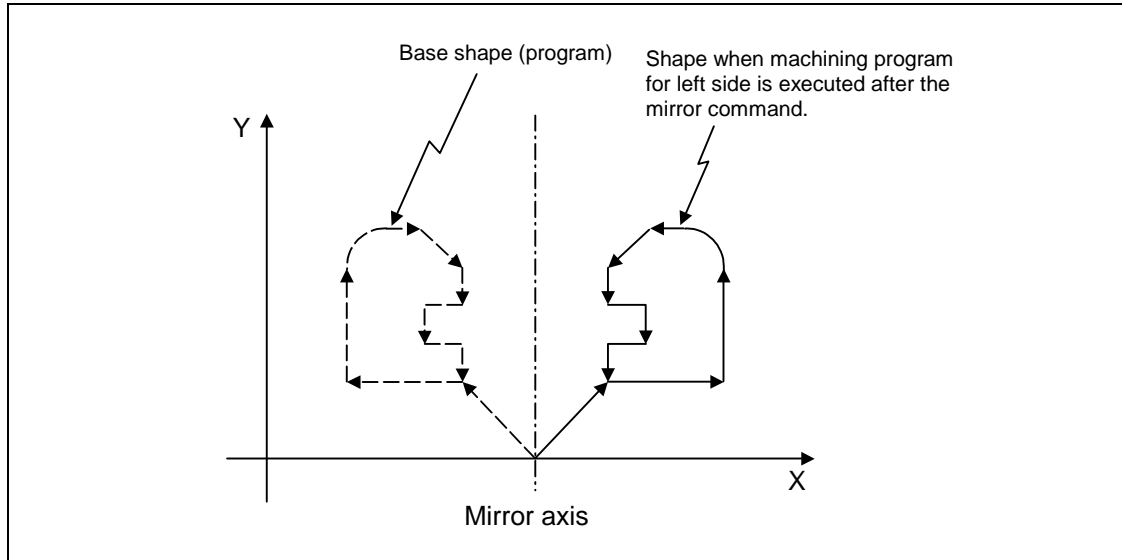
Command format

G51.1	**Xx₁**	**Yy₁**	**Zz₁ ; (Mirror image ON)**
G50.1	**Xx2**	**Yy2**	**Zz2 ; (Mirror image OFF)**
Xx/Yy/Zz		: Mirror image command axis	

Detailed description

(1) The coordinate word for G51.1 is commanded with the mirror image command axis, and the coordinate value commands the mirror image center coordinate with an absolute value or incremental value.

(2) The coordinate word in G50.1 expresses the axis for which mirror image is to be turned OFF, and the coordinate value is ignored.

(3) If mirror image is applied on only one axis in the designated plane, the rotation direction and compensation direction will be reversed for the arc or tool diameter compensation and coordinate rotation, etc.

(4) This function is processed on the local coordinate system, so the center of the mirror image will change when the counter is preset or when the workpiece coordinates are changed.

(5) Reference point return during mirror image
   If the reference point return command (G28, G30) is executed during the mirror image, the mirror image will be valid during the movement to the intermediate point, but will not be applied on the movement to the reference point after the intermediate point.

Intermediate point when mirror is applied

Intermediate point

Path on which mirror is applied

Programmed path

Mirror center
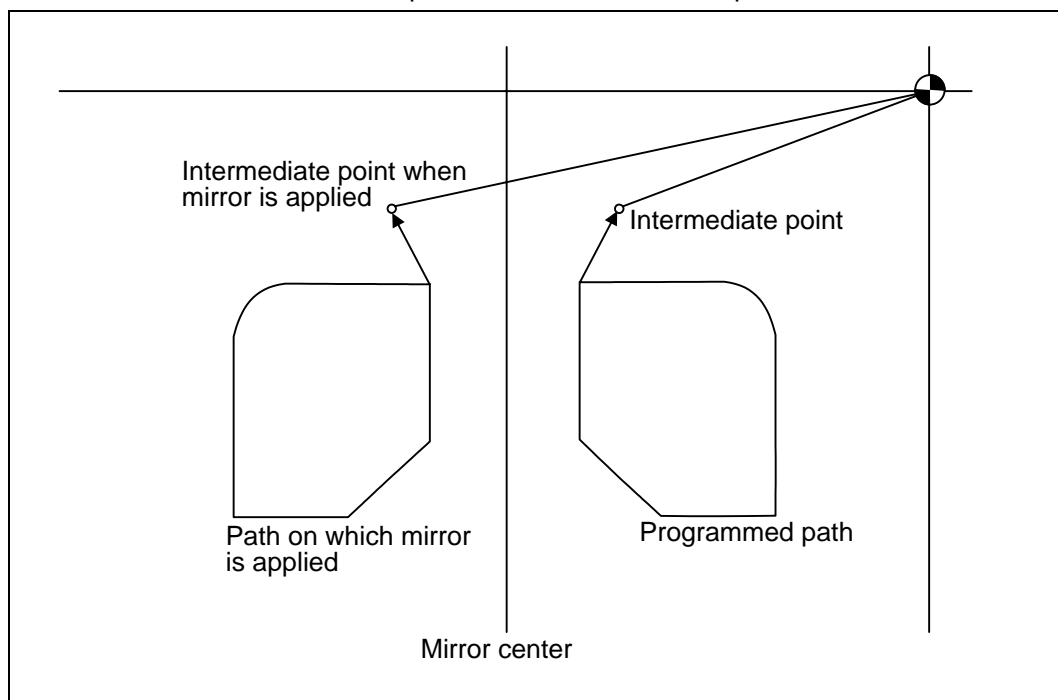
(6) Return from reference point during mirror image
   If the return command (G29) from the reference point is commanded during the mirror image, the mirror will be applied on the intermediate point.

(7) The mirror image will not be applied on the G53 command.

289

Precautions

⚠ CAUTION

⚠ **Turn the mirror image ON and OFF at the mirror image center.**

If mirror image is canceled at a point other than the mirror center, the absolute value and machine position will deviate as shown below. (In this state, execute the absolute value command (positioning with G90 mode), or execute reference point return with G28 or G30 to continue the operation.) The mirror center is set with an absolute value, so if the mirror center is commanded again in this state, the center may be set to an unpredictable position.

Cancel the mirror at the mirror center or position with the absolute value command after canceling.

Absolute value (position commanded in program)

Machine position

When moved with the incremental command after mirror cancel

Issue mirror cancel command here

Issue mirror axis command here

Mirror center
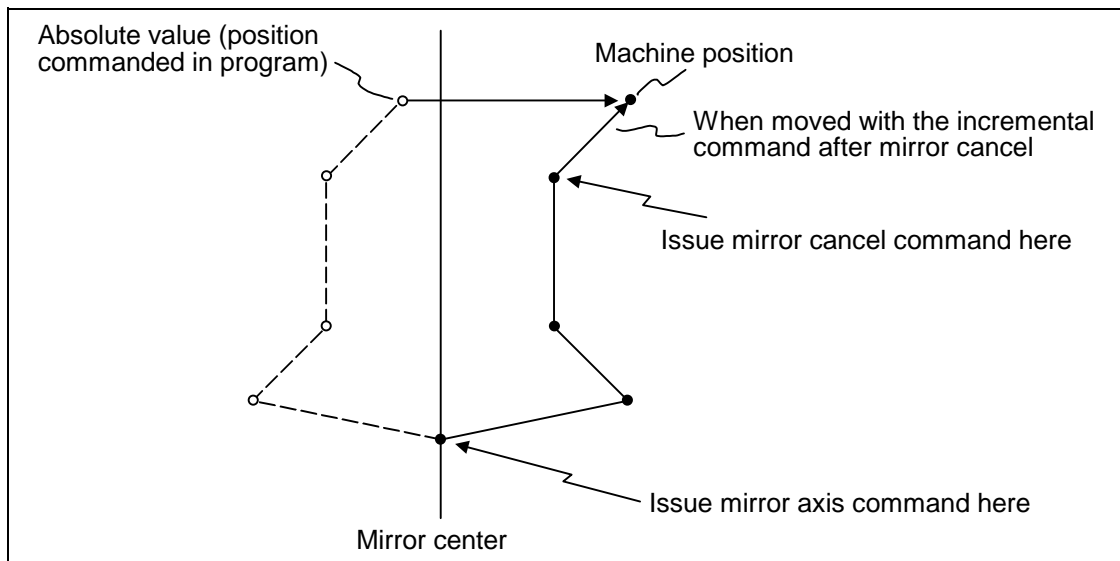
Combination with other functions

(1) Combination with diameter compensation

The mirror image (G51.1) will be processed after the diameter compensation (G41, G42) is applied, so the following type of cutting will take place.

Programmed path

When only diameter compensation is applied

When only mirror image is applied

When both mirror image and diameter compensation are applied

Mirror center
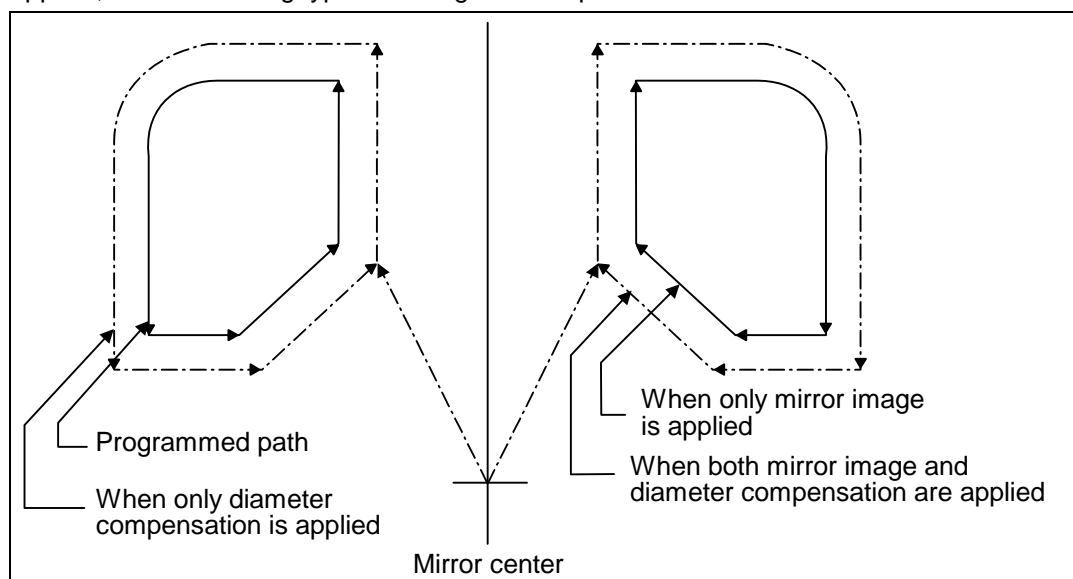
13.7 Corner chamfering, corner rounding

Chamfering at any angle or corner rounding is performed automatically by adding ,C_ or ,R_ to the end of the block to be commanded first among those command blocks which shape the corner with lines only.

13.7.1 Corner chamfering " ,C_ "
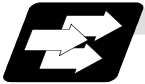
Function and purpose

The corner is chamfered in such a way that the positions produced by subtracting the lengths commanded by ",C" from the imaginary starting and final corners which would apply if no chamfering were to be performed, are connected.

Command format

N100 G01 X_ Y_ ,C_ ;
N200 G01 X_ Y_ ;
,C_   : Length up to chamfering starting point or end point from imaginary corner

Chamfering is performed at the point where N100 and N200 intersect.

Example of program

(1) G91 G01 X100., C10. ;
(2) X100. Y100. ;
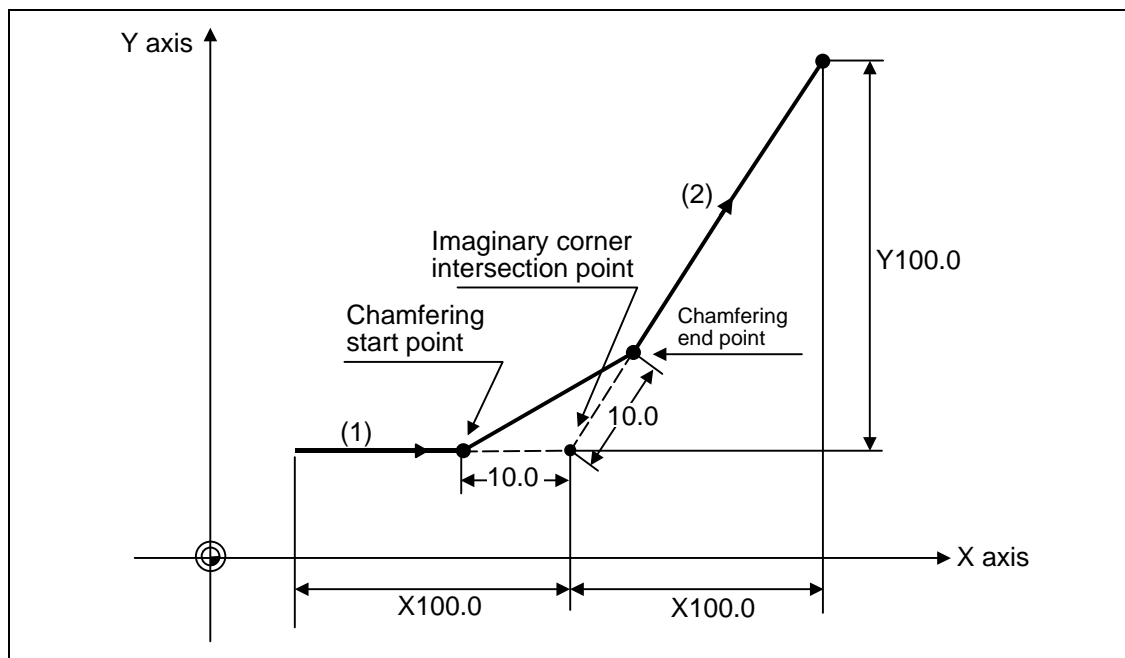
291

Detailed description

(1) The start point of the block following the corner chamfering serves as the imaginary corner intersection point.

(2) When the comma in ",C" is not present, it is handled as a C command.

(3) When both the corner chamfer and corner rounding commands exist in the same block, the latter command is valid.

(4) Tool offset is calculated for the shape which has already been subjected to corner chamfering.

(5) When scaling is commanded, scaling will also be applied to the commanded corner chamfer amount.

(6) Program error "P381" results when there is an arc command in the block following the corner chamfering block.

(7) Program error "P382" results when the block following the corner chamfering block does not have a linear command.

(8) Program error "P383" results when the movement amount in the corner chamfering block is less than the chamfering amount.

(9) Program error "P384" results when the movement amount in the block following the corner chamfering block is less than the chamfering amount.

13.7.2 Corner rounding " ,R_ "
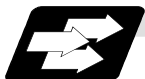
Function and purpose

The imaginary corner, which would exist if the corner were not to be rounded, is rounded with the arc having the radius which is commanded by ",R_" only when configured of linear lines.

Command format

N100 G01 X_ Y_ , R_ ;
N200 G02 X_ Y_ ;
,R_   : Arc radius of corner rounding

Corner rounding is performed at the point where N100 and N200 intersect.

Example of program

(1) G91 G01 X100., R10. ;
(2) X100. Y100. ;
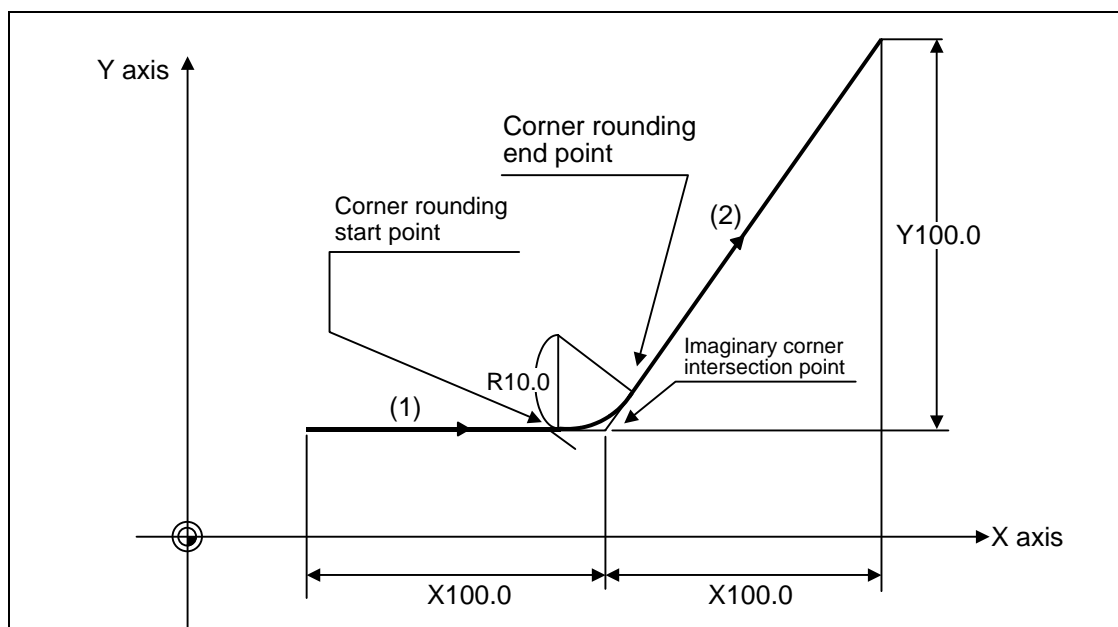
Detailed description

(1) The start point of the block following the corner R serves as the imaginary corner intersection point.
(2) When the comma in ",R" is not present, it is handled as an R command.
(3) When both the corner chamfer and corner rounding commands exist in the same block, the latter command is valid.
(4) Tool offset is calculated for the shape which has already been subjected to corner rounding.
(5) Program error "P381" results when there is an arc command in the block following the corner rounding block.
(6) Program error "P382" results when the block following the corner rounding block does not have a linear command.
(7) Program error "P383" results when the movement amount in the corner rounding block is less than the R value.
(8) Program error "P384" results when the movement amount in the block following the corner rounding block is less than the R value.

13.8 Linear angle command

Function and purpose

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

Command format

> **N1 G01 Xx$_1$ (Yy$_1$) Aa$_1$;**
> **N1 G01 Xx$_2$ (Yy$_2$) A$-$a$_2$;   (A-a$_2$ can also be set as As$_3$.)**
> **or**
> **N1 G01 Xx$_1$ (Yy$_1$) ,Aa$_1$;**
> **N1 G01 Xx$_2$ (Yy$_2$) ,A$-$a$_2$;**

This designates the angle and the X or Y axis coordinates.

Select the command plane with G17 to G19.
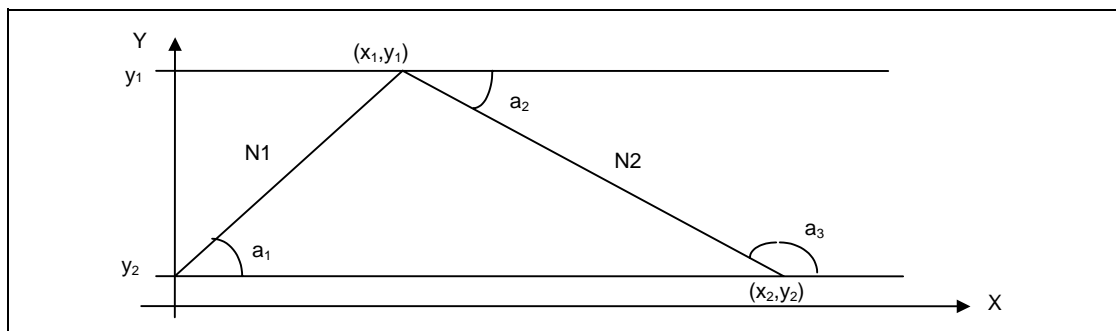
Detailed description

(1) The angle is from the + direction of the horizontal axis on the selected plane. The counter-clockwise (CCW) direction is considered to be + and the clockwise direction (CW) −.

(2) Either of the axes on the selected plane is commanded for the end point.

(3) The angle is ignored when the angle and the coordinates of both axes are commanded.

(4) When only the angle has been commanded, this is treated as a geometric command.

(5) The angle of either the start point (a_1) or end point (a_2) may be used.

(6) If address A is used for the axis name or 2nd miscellaneous function, use ",A" as the angle.

(7) This function is valid only for the G01 command; it is not valid for other interpolation or positioning commands.

(8) The range of slope "a" is $-360.000 \leq a \leq 360.000$.
When a value outside this range is commanded, it will be divided by 360 (degrees) and the remainder will be commanded.
(Example) If 400 is commanded, $40°$ (remainder of 400/360) will become the command angle.

(9) If "A" and ",A" are commanded in the same block, ",A" will be interpreted as the angle.

(Note) A program error (P33) will occur if this function is commanded during the high-speed machining mode III or high-speed high-accuracy II mode.

13.9 Geometric command

Function and purpose

When it is difficult to find the intersection point of two straight lines with a continuous linear interpolation command, this point can be calculated automatically by programming the command for the angle of the straight lines.
Example


```
N1   G01   Aa1   Ff1   ;
N2   Xx1   Zz1   Aa2   ;
```
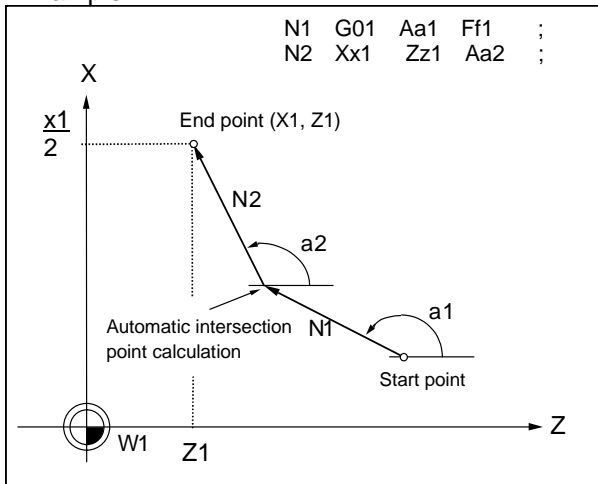
a: Angle (°) formed between straight line and horizontal axis on plane.
The plane is the selected plane at this time.

(Note 1) This function cannot be used when using the A axis or 2nd miscellaneous function A.
(Note 2) When this function is executed while executing high-speed machining mode III, the program error (P33) will occur.

Command format

N1 G01 Xx_1 (Yy_1) Aa$_1$;
N1 G01 Xx_2 (Y$_2$) A-a$_2$;   (A-a$_2$ can also be set as As$_3$.)

Detailed description

(1) Automatic calculation of two-arc contact

When two continuous circular arcs contact with each other and it is difficult to find the contact, the contact is automatically calculated by specifying the center coordinates position or radius of the first circular arc and the end point (absolute position) and center position or radius of the second circular arc.

Example

```
    G18     G02     Ii1     Kk1     Ff1         ;
            G03     Xxc     Zzc     Ii2     Kk2     Ff2     ;
    OR
    G18     G02     Ii1     Kk1     Ff1         ;
            G03     Xxc     Zzc     Rr2     Ff2     ;
    OR
    G18     G02     Rr1     Ff1     ;
            G03     Xxc     Zzc     Ii2     Kk2     Ff2     ;
```
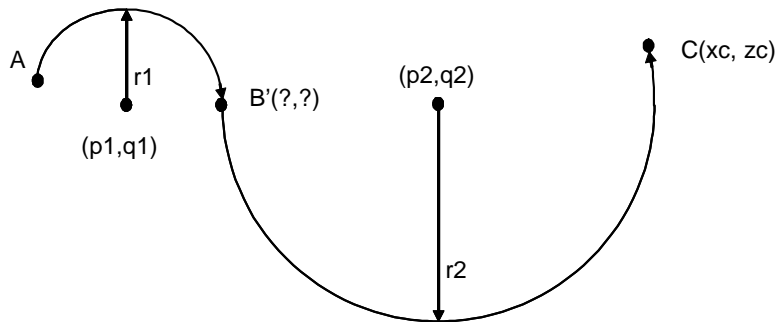

I and K : Incremental position from arc end point
P and Q : Arc center position (absolute position)

I and K are the arc center position (incremental position); distances from the start point in the first block or distances from the end point in the second block. P and Q (X, Z arc center position (absolute position)) can be commanded instead of I and K commands.

(2) Automatic calculation of linear-arc intersection

When it is difficult to find the intersections of a given line and circular arc, the intersections are automatically calculated by programming the following blocks.

Example

```
G18   G01   Aa1   Ff1   ;
      G02   Xxc   Zzc   Ii2   Kk2   Hh2   Ff2   ;
```
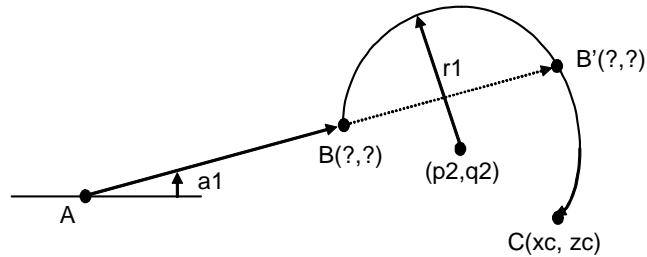

I and K : Incremental position from arc end point
P and Q : Arc center position (absolute position)
H = 0   : Intersection with shorter line (B point)
H = 1   : Intersection with longer line (B' point)

(3) Automatic calculation of arc-linear intersection

When it is difficult to find the intersections of a given circular arc and line, the intersections are automatically calculated by programming the following blocks.

Example

```
G18   G03   Ii1   Kk1   Hh1   Ff1   ;
      G01   Xxc   Zzc   Aa1   Ff1   ;
```
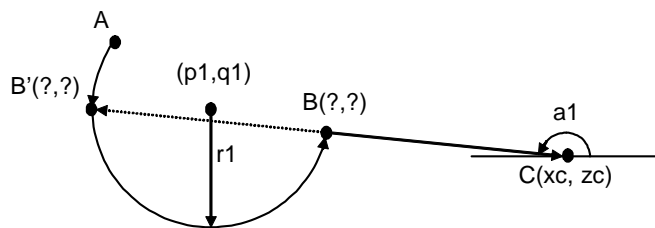

I and K : Incremental position from arc end point
P and Q : Arc center position (absolute position) (L3 only)
H = 0   : Intersection with shorter line (B point)
H = 1   : Intersection with longer line (B' point)

(4) Automatic calculation of linear-arc contact

When it is difficult to find the contact of a given line and circular arc, the contact is automatically calculated by programming the following blocks.

Example

```
G01   Aa1   Ff1   ;
G03   Xxc   Zzc   Rr1   Ff1   ;
```

C(xc, zc)

r1

B (?,?)

a1

A
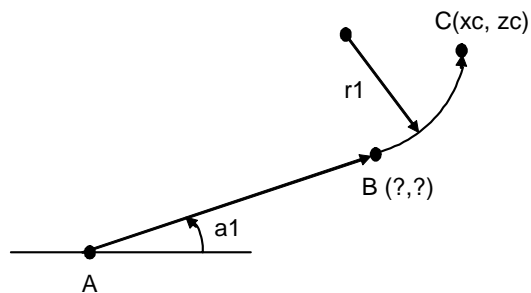
(5) Automatic calculation of arc-linear contact

When it is difficult to find the contact of a given circular arc and line, the contact is automatically calculated by programming the following blocks.

Example

```
G02   Rr1   Ff1   ;
G01   Xxc   Zzc   Aa1   Ff1   ;
```

A

B (?,?)

r1

a1

C(xc, zc)
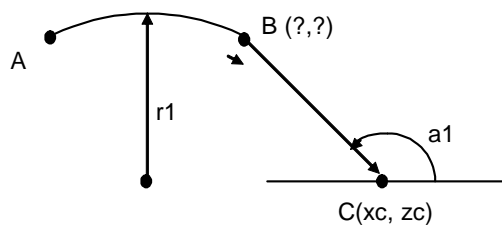
13.10　Polar coordinate command; G16　　　　　　　　　　　　　　　[E68]

Function and purpose

With this function, the end point coordinate value is commanded with the polar coordinate of the radius and angle.

Command format

G16 ;　　　　Polar coordinate command mode ON

G15 ;　　　　Polar coordinate command mode OFF

(1) The polar coordinate command is applied in the interval from turning ON to OFF of the polar coordinate command mode.

　G1x ;　　　　　　　　　　Plane selection for polar coordinate command (G17/G18/G19)
　G16 ;　　　　　　　　　　Polar coordinate command mode ON
　G9x G01 Xx1 Yy1 F2000 ;　Polar coordinate command
　:　　　　　　　　　　　　G9x : Center selection for polar coordinate command (G90/G91)
　　　　　　　　　　　　　　　G90 ⋯The workpiece coordinate system zero point is the polar coordinate center.
　　　　　　　　　　　　　　　G91 ⋯The present position is the polar coordinate center.
　　　　　　　　　　　　　x1 : 1st axis for the plane ⋯The radius commanded
　　　　　　　　　　　　　y1 : 2nd axis for the plane ⋯The angle commanded

For G90/G17(X-Y plane)
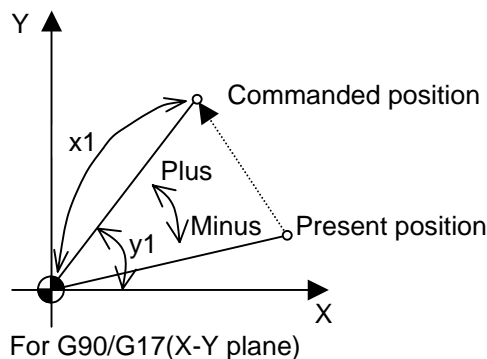
　G15 ;　　　　　　　　　　　Polar coordinate command mode OFF

(2) The plane selection during the polar coordinate command mode is carried out with G17, G18 and G19.

(3) The polar coordinate command is a modal. The polar coordinate command mode when the power is turned ON is OFF (G15). Whether to initialize the modal at reset or not can be selected with the parameter (#1210 RstGmd/bit 11) setting.

(4) During polar coordinate command mode, command the radius with the 1st axis for the selected plane, and the angle with the 2nd axis. For example, when the X-Y plane is selected, command the radius with the address "X", and the angle with the address "Y".

(5) For the angle, the counterclockwise direction of the selected plane is positive and the clockwise direction is negative.

(6) The radius and angle can be commanded with both the absolute value and incremental value (G90, G91).

(7) When the radius is commanded with the absolute value, command the distance from the zero point in the workpiece coordinate system (note that the local coordinate system is applied when the local coordinate system is set).

(8) When the radius is commanded with the incremental value command, considering the end point of the previous block as the polar coordinate center, command the incremental value from that end point. The angle is commanded with the incremental value of the angle from the previous block.

(9) When the radius is commanded with the negative value, the same operation as the command that the radius command value is changed to the absolute value and 180° is added to the angle command value.

Command position

(1) When the zero point in the workipiece coordinate system is applied to the polar coordinate center

The zero point in the workipiece coordinate system is applied to the polar coordinate center by commanding the radius value with the absolute value. Note that the zero point in the local coordinate system is applied to the polar coordinate center if the local coordinate system (G52) is used.

Command position when the zero point in the workipiece coordinate system is applied to the polar coordinate center
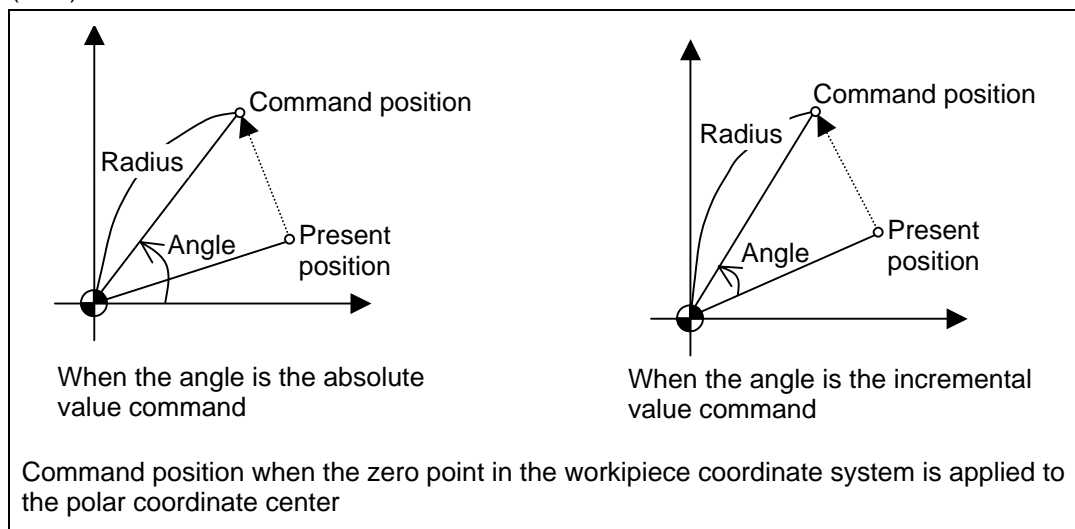
(2) When the present position is applied to the polar coordinate center

The present position is applied to the polar coordinate center by commanding the radius value with the incremental value.

Command position when the present position is applied to the polar coordinate center
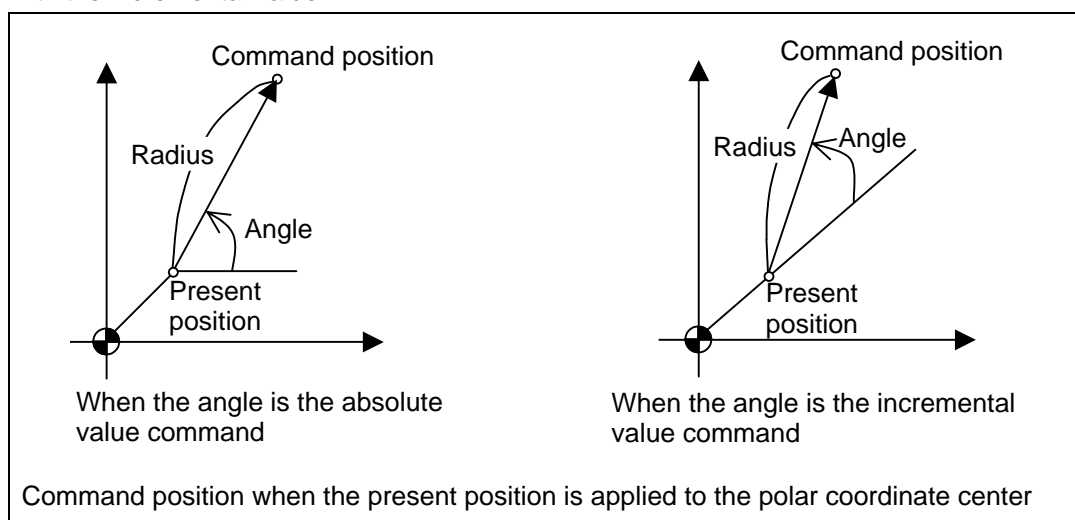
(3) When the radius value command is omitted

When the radius value command is omitted, the zero point in the workpiece coordinate system is applied to the polar coordinate center. Note that the zero point in the local coordinate system is applied to the polar coordinate center if the local coordinate system (G52) is used.

When the angle is the absolute value command

When the angle is the incremental value command

Command position when the radius value command is omitted
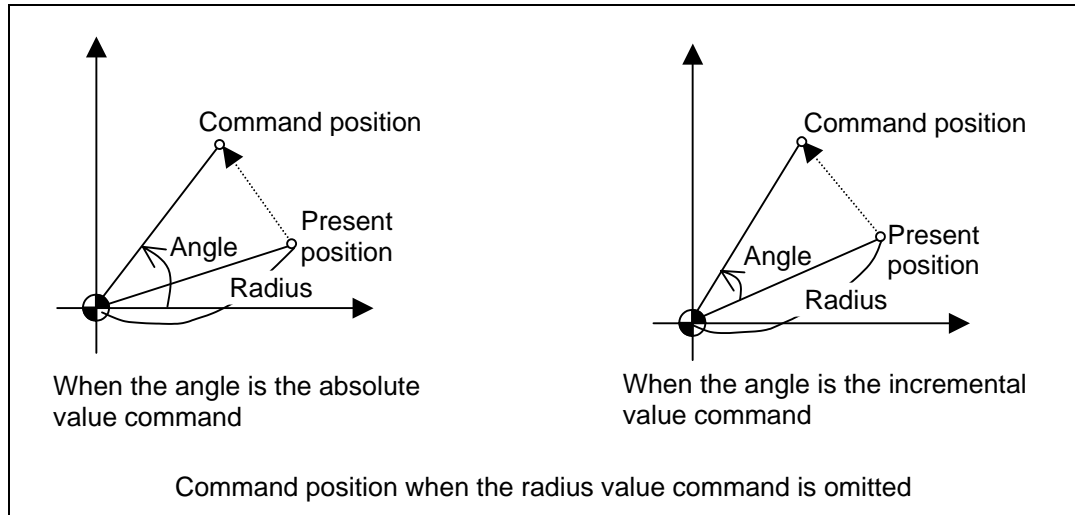
(4) When the angle command is omitted

When the angle command is omitted, the angle of the present position in the workpiece coordinate system is applied to the angle command.
The zero point in the workpiece coordinate system is applied to the polar coordinate center by commanding the radius value with the absolute value. Note that the zero point in the local coordinate system is applied to the polar coordinate center if the local coordinate system (G52) is used.
If the radius value is commanded with the incremental value, the present position is applied to the polar coordinate center.

When the radius is the absolute value command

When the radius is the incremental value command

Command position when the angle command is omitted
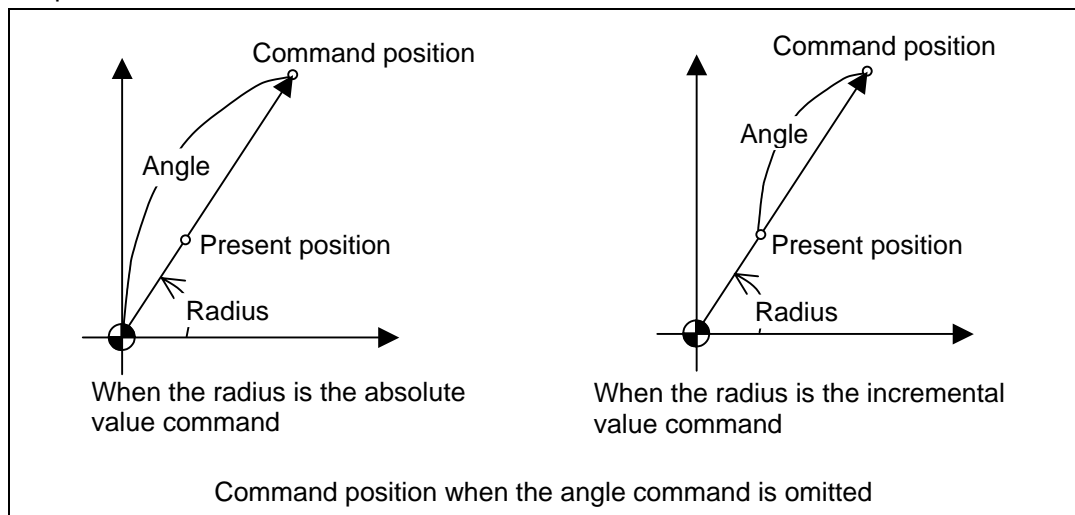
Axis command not interpreted as the polar coordinate command

The axis command with the following command is not interpreted as the polar coordinate command during the polar coordinate command mode. The movement command that has no axes commands for the 1st axis and 2nd axis in the selected plane mode is also not interpreted as polar coordinate command during the polar coordinate command mode.

Function	G code
Dwell	G04
Program parameter input/compensation input	G10
Local coordinate system setting	G52
Machine coordinate system setting	G92
Machine coordinate system selection	G53
Coordinate rotation by program	G68
Scaling	G51
G command mirror image	G51.1
Reference point check	G27
Reference point return	G28
Start point return	G29
2nd to 4th reference point return	G30
Tool change position return 1	G30.1
Tool change position return 2	G30.2
Tool change position return 3	G30.3
Tool change position return 4	G30.4
Tool change position return 5	G30.5
Tool change position return 6	G30.6
Automatic tool length measurement	G37
Skip	G31
Multi-skip function 1	G31.1
Multi-skip function 2	G31.2
Multi-skip function 3	G31.2
Linear angle command	G01 Aa1

Example of program

When the zero point in the workpiece coordinate system is the polar coordinate zero point

- The polar coordinate zero point is the zero point in the workpiece coordinate system.
- The plane is the X-Y plane.
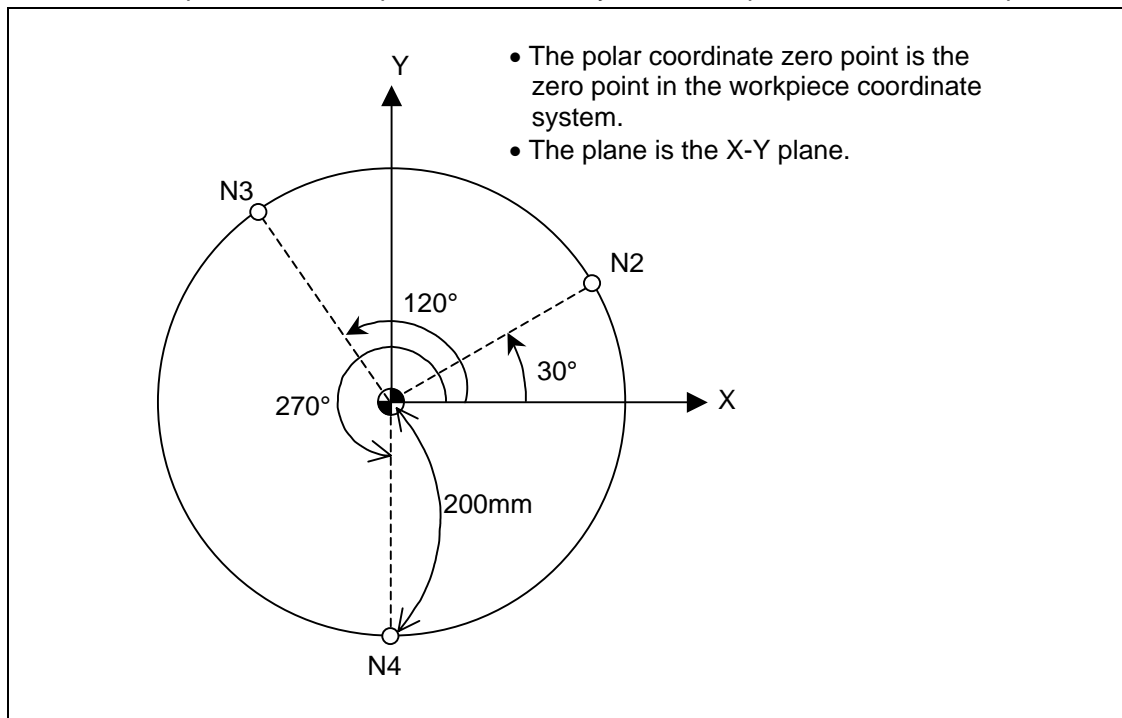
(1) When the radius value and angle are the absolute value command

N1 G17 G90 G16 ;	Polar coordinate command, X-Y plane selection
	The polar coordinate zero point is the zero point in the workpiece coordinate system.
N2 G85 X200. Y30. Z-20. F200. ;	Radius 200mm, angle 30°
N3 Y120. ;	Radius 200mm, angle 120°
N4 Y270. ;	Radius 200mm, angle 270°
N5 G15 G80 ;	Polar coordinate command cancel

(2) When the radius value is the absolute value command and the angle is the incremental value command

N1 G17 G90 G16 ;	Polar coordinate command, X-Y plane selection
	The polar coordinate zero point is the zero point in the workpiece coordinate system.
N2 G85 X200. Y30. Z-20. F200. ;	Radius 200mm, angle 30°
N3 G91 Y90. ;	Radius 200mm, angle + 90°
N4 Y150. ;	Radius 200mm, angle +150°
N5 G15 G80 ;	Polar coordinate command cancel

Precautions

(1) If the following commands are carried out during the polar coordinate command mode, or if the polar coordinate command is carried out during the following command mode, an program error (P34) will occur.

Function	G code
High-speed machining mode 3	G05 P3

(2) When the mirror image (G code/parameter/external signal) is canceled anywhere except at the mirror image center during the polar coordinate command mode, the absolute value and machine position will deviate. The mirror center is set with an absolute value and so if another mirror center is assigned in this state, the center may be set at an unforeseen position. Cancel the mirror image above the mirror center or, after cancellation, assign a positioning command using absolute value command that the radius and angle of the polar coordinate command are designated.

13.11 Circle cutting; G12, G13

Function and purpose

Circle cutting starts the tool from the center of the circle, and cuts the inner circumference of the circle. The tool continues cutting while drawing a circle and returns to the center position.

Command format

G12 (G13) Ii_1 Dd_1 Ff_1 ;	
G12	: Clockwise (CW)
G13	: Counterclockwise (CCW)
I	: Radius of circle (incremental value), the symbol is ignored
D	: Offset No. (The offset No. and offset data are not displayed on the setting and display unit.)
F	: Feedrate

Detailed description

(1) The symbol + for the offset amount indicates reduction, and − indicates enlargement.
(2) The circle cutting is executed on the plane G17, G18 or G19 currently selected.

Circle radius

i_1

— — — — — Offset amount symbol +

—·—·—·— Offset amount symbol −

d1 offset amount +

d1 offset amount −

For G12 (tool center path)
0→1→2→3→4→5→6→7→0

For G13 (tool center path)
0→7→6→5→4→3→2→1→0
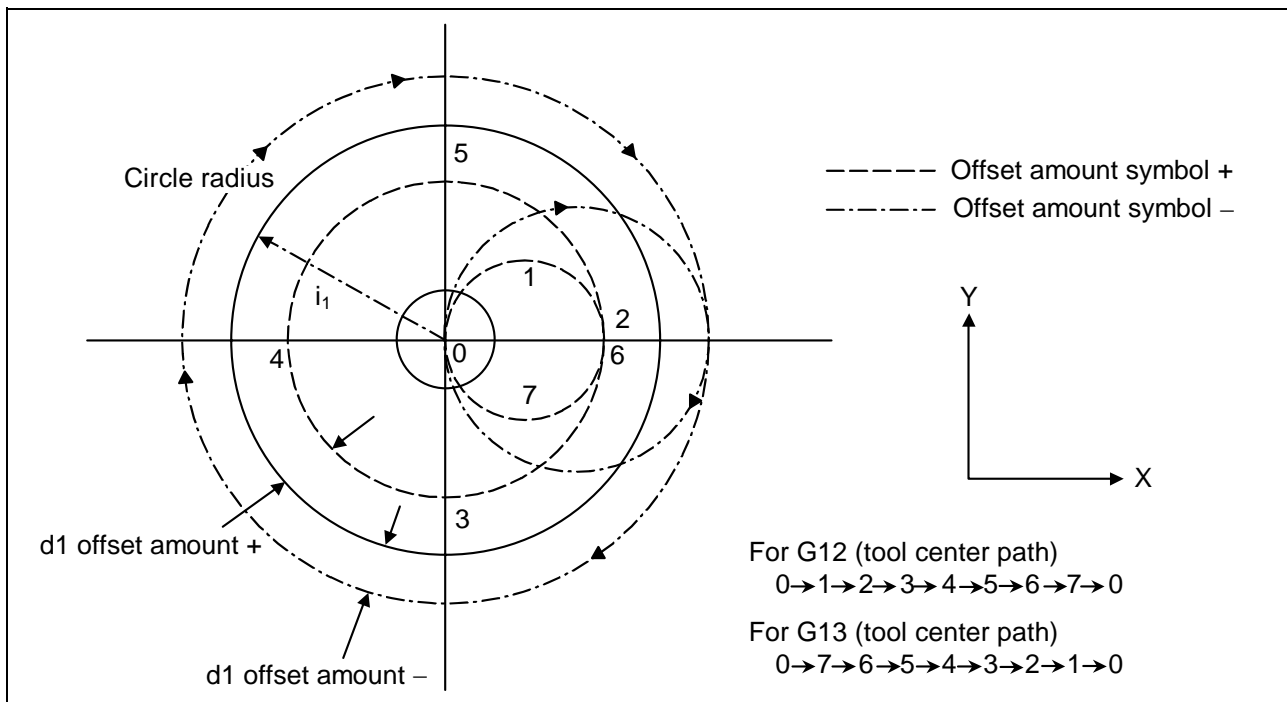
Example of program

(Example 1) G12 I5000 D01 F100 ; (Input setting unit 0.01)
When offset amount is +10.00mm

10.000m   Offset

50.000m   Radius
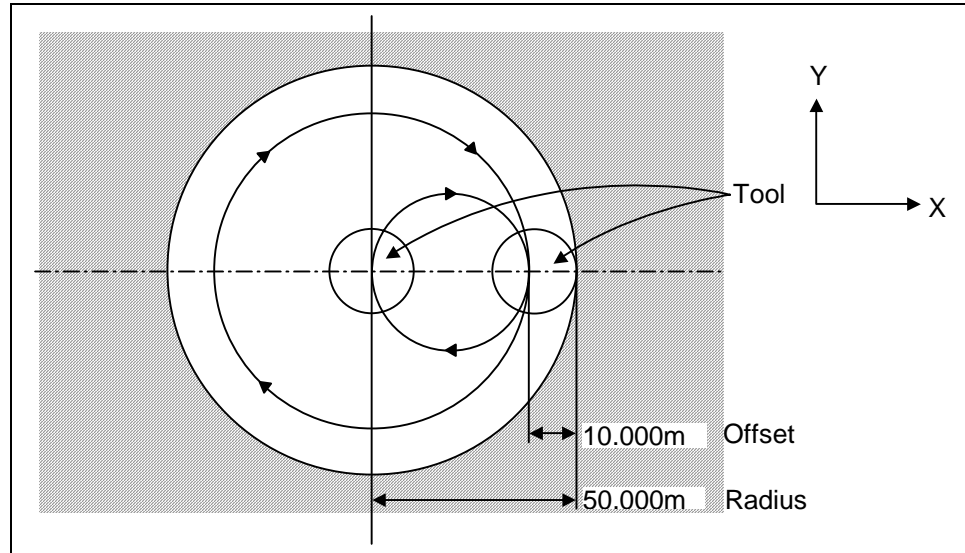
Precautions

(1) If the offset No. "D" is not issued or if the offset No. is illegal, the program error (P170) will occur.
(2) If [Radius (I) = offset amount] is 0 or negative, the program error (P233) will occur.
(3) If G12 or G13 is commanded during diameter compensation (G41, G42), the diameter compensation will be validated on the path after compensating with the D commanded with G12 or G13.
(4) If an address, not included in the format, is commanded in the same block as G12 and G13, a program error (P32) will occur.

13.12 Program parameter input; G10, G11

Function and purpose

The parameters set from the setting and display unit can be changed in the machining programs.
The data format used for the data setting is as follows.

Command format

> **G10 L50 ; Data setting command**
>
> **P major classification number N data number H□ bit type data ;**
> **P major classification number A axis number N data number D byte type data ;**
> **P major classification number A axis number N data number S word type data ;**
> **P major classification number A axis number N data number L 2 word type data ;**
>
> **G11 ; Data setting mode cancel (data setting completed)**

There are 8 types of data formats according to the type of parameter (axis-common and axis-independent) and data type, as listed below.

With axis-common data	
Axis-common bit-type parameter	P___ N___ H□____ ;
Axis-common byte-type parameter	P___ N___ D_____ ;
Axis-common word-type parameter	P___ N___ S_____ ;
Axis-common 2-word-type parameter	P___ N___ L_____ ;
With axis-independent data	
Axis-independent bit-type parameter	P__ A__ N___ H□___ ;
Axis-independent byte-type parameter	P__ A__ N___ D____ ;
Axis-independent word-type parameter	P__ A__ N___ S____ ;
Axis-independent 2-word-type parameter	P__ A__ N__ L____ ;

(Note 1)   The sequence of addresses in a block must be as shown above.

(Note 2)   Refer to Appendix Table 1 for the P, N number correspondence table.

(Note 3)   For a bit type parameter, the data type will be H□ (□ is a value between 0 and 7).

(Note 4)   The axis number is set in the following manner: 1st axis is 1, 2nd axis is 2, and so forth.

(Note 5)   Command G10L50, G11 in independent blocks. A program error (P33, P421) will occur if not commanded in independent blocks.

Example of program

(Example) To turn ON bit 2 of bit selection #6401

```
G10  L50 ;

P8  N1  H21 ;

G11 ;
```

13.13 Macro interrupt

Function and purpose

A user macro interrupt signal (UIT) is input from the machine to interrupt the program being currently executed and instead call another program and execute it. This is called the user macro interrupt function.

Use of this function allows the program to operate flexibly enough to meet varying conditions.

For setting the parameters of the function, refer to the Operation manual and the machine parameters in Appendix 1.

Command format

M96 P__ ;	**User macro interrupt enable**
M97 ;	**User macro interrupt disable**
P	:Interrupt program number

The user macro interrupt function is enabled and disabled by the M96 and M97 commands programmed to make the user macro interrupt signal (UIT) valid or invalid. That is, if an interrupt signal (UIT) is input from the machine side in a user macro interrupt enable period from when M96 is issued to when M97 is issued or the NC is reset, a user macro interrupt is caused to execute the program specified by P__ instead of the one being executed currently.

Another interrupt signal (UIT) is ignored while one user macro interrupt is being in service. It is also ignored in a user macro interrupt disable state such as after an M97 command is issued or the system is reset.

M96 and M97 are processed internally as user macro interrupt control M codes.

Interrupt enable conditions

A user macro interrupt is enabled only during execution of a program. The requirements for the user macro interrupt are as follows :

(1) An automatic operation mode, memory or tape, or MDI has been selected.

(2) The system is running in automatic mode.

(3) No other macro interrupt is being processed.

(Note 1) A macro interrupt is disabled in manual operation mode (JOG, STEP, HANDLE, etc.)

Outline of operation

(1) When a user macro interrupt signal (UIT) is input after an M96Pp1 ; command is issued by the current program, interrupt program Op1 is executed. When an M99; command is issued by the interrupt program, control returns to the main program.

(2) If M99Pp$_2$; is specified, the blocks from the one next to the interrupted block to the last one are searched for the block with sequence number Np2 ;. Control thus returns to the block with sequence number Np2 that is found first in the above search.
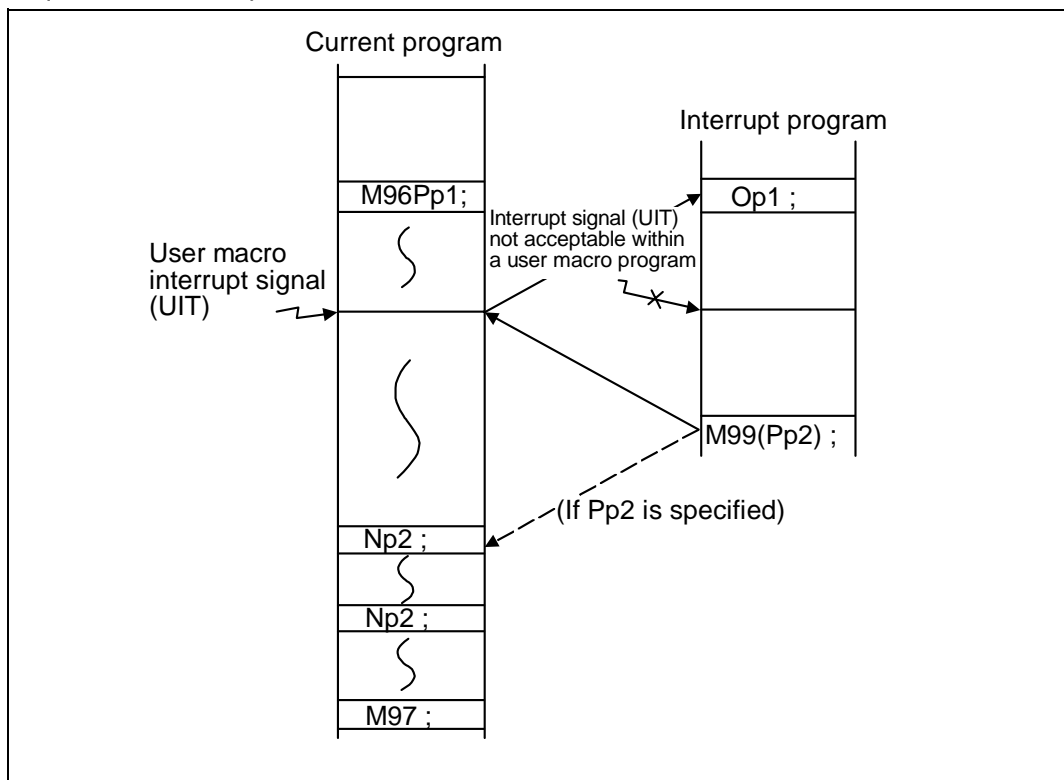
Interrupt type

Interrupt types 1 and 2 can be selected by the parameter "#1113 INT_2".

[Type 1]
- When an interrupt signal (UIT) is input, the system immediately stops moving the tool and interrupts dwell, then permits the interrupt program to run.
- If the interrupt program contains a move or miscellaneous function (MSTB) command, the commands in the interrupted block are lost. After the interrupt program completes, the main program resumes operation from the block next to the interrupted one.
- If the interrupted program contains no move and miscellaneous (MSTB) commands, it resumes operation, after completion of the interrupt program, from the point in the block where the interrupt was caused.

If an interrupt signal (UIT) is input during execution of a miscellaneous function (MSTB) command, the NC system waits for a completion signal (FIN). The system thus executes a move or miscellaneous function command (MSTB) in the interrupt program only after input of FIN.

[Type 2]
- When an interrupt signal (UIT) is input, the program completes the commands in the current block, then transfers control to the interrupt program.
- If the interrupt program contains no move and miscellaneous function (MSTB) commands, the interrupt program is executed without interrupting execution of the current block.

However, if the interrupt program has not ended even after the execution of the original block is completed, the system may stop machining temporarily.

[Type 1]

Main program

block(1)   block(2)   block(3)

If the interrupt program contains a move or miscellaneous function command, the reset block (2) is lost.

block(1)   block(2)   block(3)

Interrupt program

User macro interrupt

If the interrupted program contains no move and miscellaneous commands, it resumes operation from where it left in block (2), that is, all the reset commands.

block(1)   block(2)   block(2)   block(3)

Interrupt program

User macro interrupt

Executing

[Type 2]

Main program

block(1)   block(2)   block(3)

block(1)   block(2)   block(3)

Interrupt program

User macro interrupt signal

If the interrupted program contains no move and miscellaneous commands, the interrupted program is kept executed in parallel to execution of the interrupt program block (3).

block(1)   block(2)   block(3)

Interrupt program

User macro interrupt

The move or miscellaneous command in the interrupt program is executed after completion of the current block.

Calling method

User macro interrupt is classified into the following two types depending on the way an interrupt program is called. These two types of interrupt are selected by parameter "#1229 set01/bit0".
Both types of interrupt are included in calculation of the nest level. The subprograms and user macros called in the interrupt program are also included in calculation of the nest level.
a. Subprogram type interrupt
b. Macro type interrupt

Subprogram type interrupt	The user macro interrupt program is called as a subprogram. As with calling by M98, the local variable level remains unchanged before and after an interrupt.
Macro type interrupt	The user macro interrupt program is called as a user macro. As with calling by G65, the local variable level changes before and after an interrupt. No arguments in the main program can be passed to the interrupt program.

Acceptance of user macro interrupt signal (UIT)

A user macro interrupt signal (UIT) is accepted in the following two modes: These two modes are selected by a parameter "#1112 S_TRG".
a. Status trigger mode
b. Edge trigger mode

Status trigger mode	The user macro interrupt signal (UIT) is accepted as valid when it is on. If the interrupt signal (UIT) is ON when the user macro interrupt function is enabled by M96, the interrupt program is activated. By keeping the interrupt signal (UIT) ON, the interrupt program can be executed repeatedly.
Edge trigger mode	The user macro interrupt signal (UIT) is accepted as valid at its rising edge, that is, at the instance it turns on. This mode is useful to execute an interrupt program once.

User macro interrupt signal (UIT)

ON

OFF

(Status trigger mode)

User macro interrupt

(Edge trigger mode)

Accepting user macro interrupt signal (UIT)
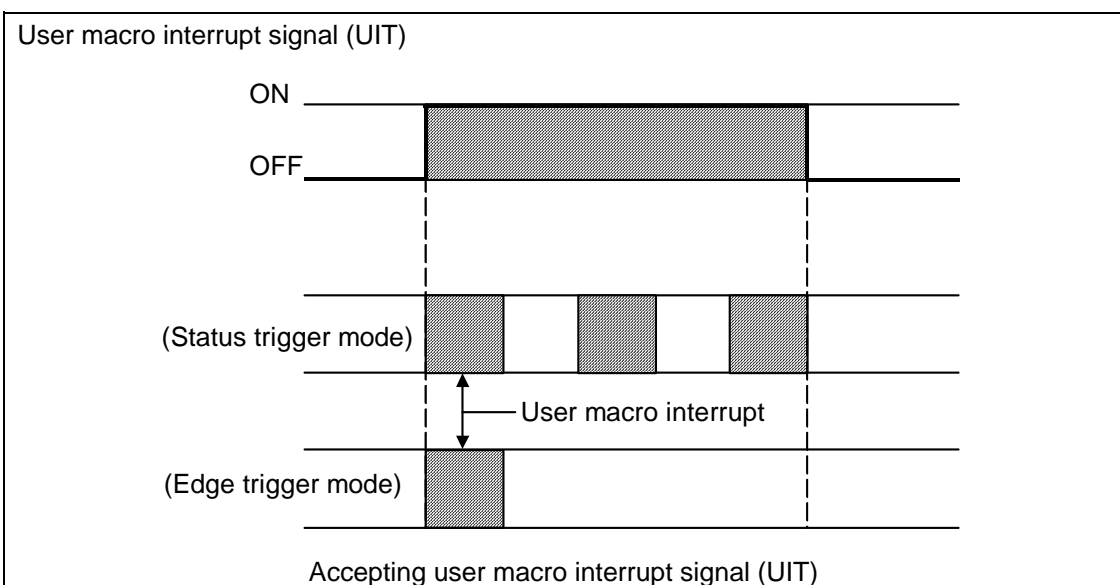
Returning from user macro interrupt

M99 (P__) ;
An M99 command is issued in the interrupt program to return to the main program. Address P is used to specify the sequence number of the return destination in the main program. The blocks from the one next to the interrupted block to the last one in the main program are first searched for the block with sequence number Np2;. If it is not found, all the blocks before the interrupted one are then searched. Control thus returns to the block with sequence number that is found first in the above search.
(This is equivalent to M99P__ used after M98 calling.)

Modal information affected by user macro interrupt

If modal information is changed by the interrupt program, it is handled as follows after control returns from the interrupt program to the main program.

Returning with M99;	The change of modal information by the interrupt program is invalidated and the original modal information is not restored. With interrupt type 1, however, if the interrupt program contains a move or miscellaneous function (MSTB) command, the original modal information is not restored.
Returning with M99P__;	The original modal information is updated by the change in the interrupt program even after returning to the main program. This is the same as in returning with M99P__; from a program called by M98.

Main program being executed

M96Pp1 ;

User macro interrupt signal (UIT)

Interrupt program

Op1 ;

(Modal change)

{ Modal before interrupt is restored. }

M99(p2) ;

(With Pp2 specified)

Np2 ;

{ Modal modified by interrupt program remains effective. }

Modal information affected by user macro interrupt
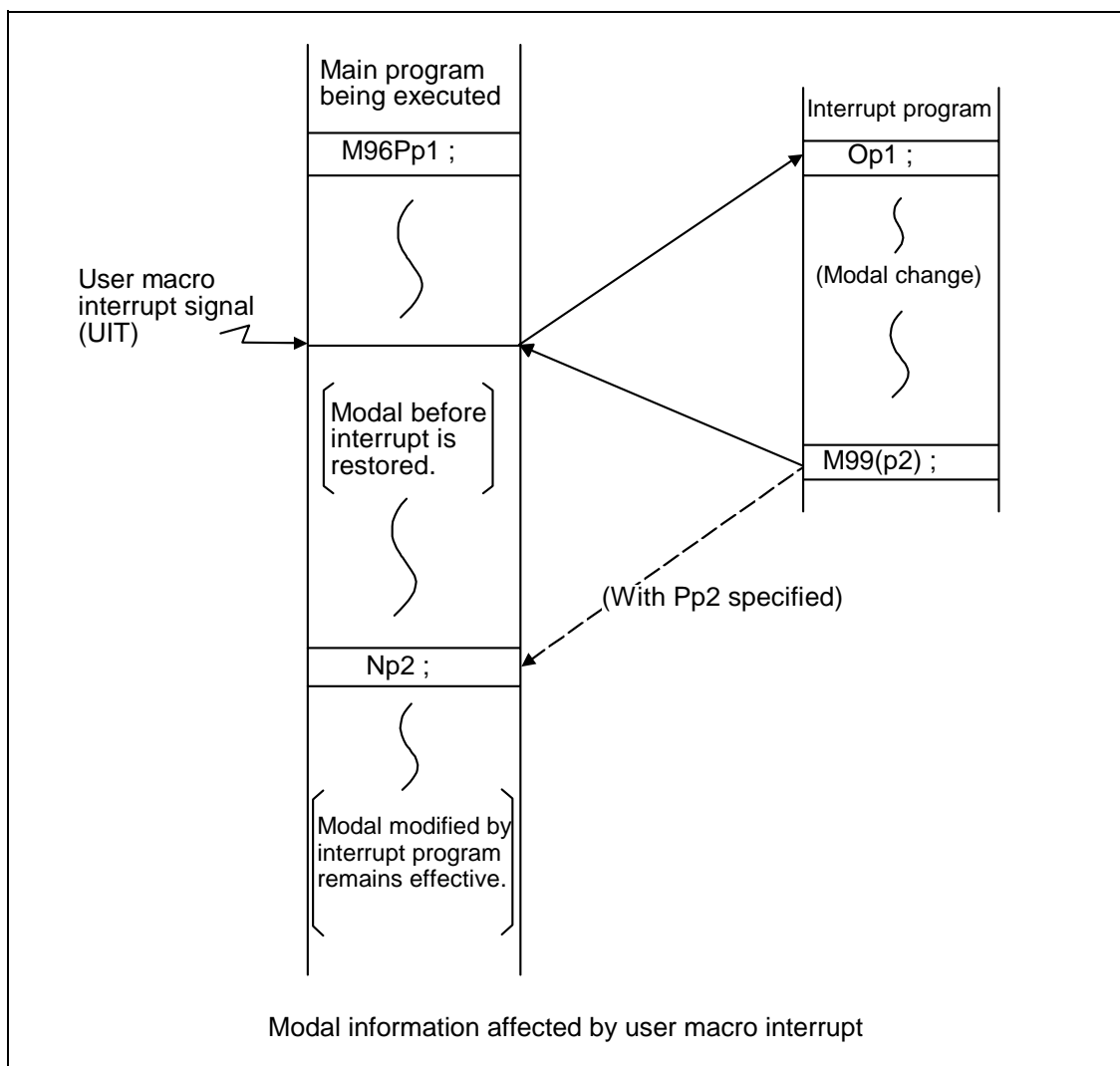
Modal information variables (#4401 to #4520)

Modal information when control passes to the user macro interrupt program can be known by reading system variables #4401 to #4520.
The unit specified with a command applies.

System variable	Modal information	
#4401 to #4421	G code (group 01 to group 21)	Some groups are not used.
#4507	D code	
#4509	F code	
#4511	H code	
#4513	M code	
#4514	Sequence number	
#4515	Program number	
#4519	S code	
#4520	T code	

The above system variables are available only in the user macro interrupt program. If they are used in other programs, program error (P241) results.

M code for control of user macro interrupt

The user macro interrupt is controlled by M96 and M97. However, these commands may have been used for other operation. To be prepared for such case, these command functions can be assigned to other M codes.
(This invalidates program compatibility.)
User macro interrupt control with alternate M codes is possible by setting the alternate M code in parameters "#1110 M96_M" and "#1111 M97_M" and by validating the setting by selecting parameter "#1109 subs_M".
(M codes 03 to 97 except 30 are available for this purpose.)
If the parameter "#1109 subs_M" used to enable the alternate M codes is not selected, the M96 and M97 codes remain effective for user macro interrupt control.
In either case, the M codes for user macro interrupt control are processed internally and not output to the outside.

Parameters

Refer to the Operation Manual for details on the setting methods.
(1) Subprogram call validity "#1229 set 01/bit 0"
   1 : Subprogram type user macro interrupt
   0 : Macro type user macro interrupt
(2) Status trigger mode validity "#1112 S_TRG"
   1 : Status trigger mode
   0 : Edge trigger mode
(3) Interrupt type 2 validity "#1113 INT_2"
   1 : The executable statements in the interrupt program are executed after completion of execution of the current block. (Type 2)
   0 : The executable statements in the interrupt program are executed before completion of execution of the current block. (Type 1)
(4) Validity of alternate M code for user macro interrupt control "#1109 subs_M"
   1 : Valid
   0 : Invalid
(5) Alternate M codes for user macro interrupt
   Interrupt enable M code (equivalent to M96) "#1110 M96_M"
   Interrupt disable M code (equivalent to M97) "#1111 M97_M"
   M codes 03 to 97 except 30 are available.

Restrictions

(1) If the user macro interrupt program uses system variables #5001 and after (position information) to read coordinates, the coordinates pre-read in the buffer are used.
(2) If an interrupt is caused during execution of the tool diameter compensation, a sequence number (M99P___;) must be specified with a command to return from the user macro interrupt program. If no sequence number is specified, control cannot return to the main program normally.

13.14 Tool change position return
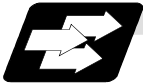
Function and purpose

By specifying the tool changing position in a parameter ("#8206 TOOL CHG. P") and also specifying a tool changing position return command in a machining program, the tool can be changed at the most appropriate position.
The axes that are going to return to the tool changing position and the order in which the axes begin to return can be changed by commands.

Command format

(1) The format of tool changing position return commands is as follows.

> **G30. n;**
> n = 1 to 6 :   Specify the axes that return to the tool changing position and the order in which they return.

For the commands and return order, see next table.

Command	Return order	
G30.1	Z axis → X axis • Y axis	(→ added axis)
G30.2	Z axis → X axis → Y axis	(→ added axis)
G30.3	Z axis → Y axis → X axis	(→ added axis)
G30.4	X axis → Y axis • Z axis	(→ added axis)
G30.5	Y axis → X axis • Z axis	(→ added axis)
G30.6	X axis • Y axis • Z axis	(→ added axis)

(Note 1) An arrow (→) indicates the order of axes that begin to return. An period (•) indicates that the axes begin to return simultaneously. (Example : "Z axis → X axis, Y axis" indicate that the Z axis returns to the tool changing position, then the X and Y axes does.)

(2) The tool changing position return on/off for the added axis can be set with parameter "#1092 Tchg_A" for the added axis. Note, however, that the added axis always return to the tool changing position only after the standard axes complete returning (see the above table).
The added axis alone cannot return to the tool changing position.

(3) When the axis address is commanded in the same block as the tool change position return command, the program error (P33) will occur.

Example of operates

(1) The figure below shows an example of how the tool operates during the tool change position return command. (Only operations of X and Y axes in G30.1 to G30.3 are figured.)
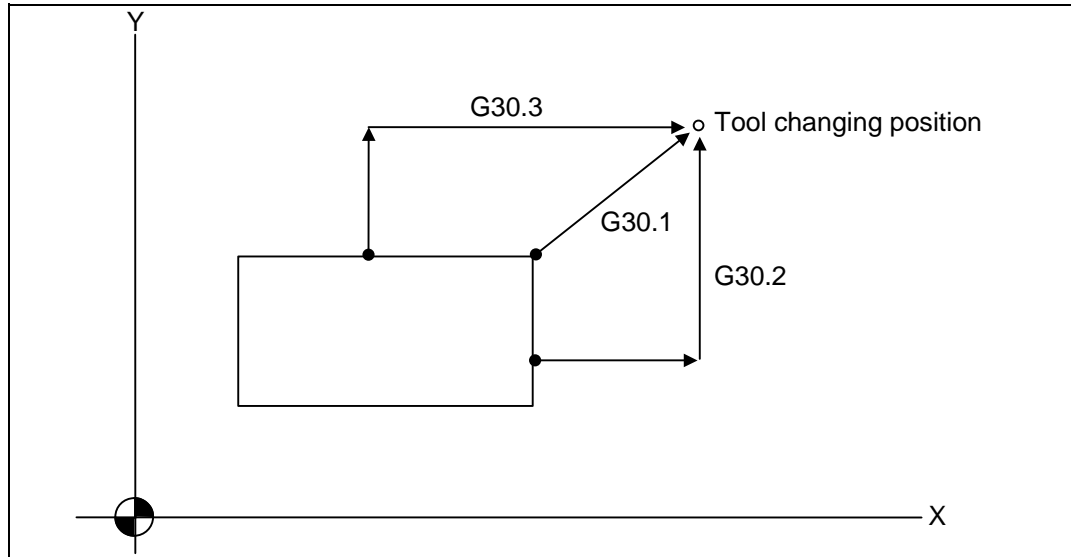
(a) G30.1 command: The Z axis returns to the tool changing position, then the X and Y axes simultaneously do the same thing.

(b) G30.2 command: The Z axis returns to the tool changing position, then the X axis does the same thing. After that, the Y axis returns to the tool changing position.

(c) G30.3 command: The Z axis returns to the tool changing position, then the X axis does the same thing. After that, the X axis returns to the tool changing position.

(d) G30.4 command: The X axis returns to the tool changing position, then the Y axis and Z axis simultaneously do the same thing.

(e) G30.5 command: The Y axis returns to the tool changing position, then the X and Z axes return to the tool changing position simultaneously.

(f) G30.6 command: The X, Y and Z axes return to the tool changing position simultaneously.

(Note) If tool changing position return is valid for an added axis, the added axis also returns to the tool changing position after the X, Y and Z axes reach the tool changing position.

(2) After all necessary tool changing position return is completed by a G30.n command, tool changing position return complete signal TCP (X22B) is turned on. When an axis out of those having returned to the tool changing position by a G30.n command leaves the tool changing position, the TCP signal is turned off.

With a G30.1 command, for example, the TCP signal is turned on when the Z axis has reached the tool changing position after the X and Y axes did. The TCP signal is then turned off when the X or Y axis leaves the position. If tool changing position return for added axes is on with parameter "#1092 Tchg_A", the TCP signal is turned on when the added axis or axes have reached the tool changing position after the standard axes did. It is then turned off when one of the X, Y, Z, and added axes leaves the position.

```
[TCP signal output timing chart]        (G30.1 command)


Work program                          G30.1;              T02;    G00X-100.…

Arrival of Z axis to tool
changing position

Arrival of X, Y axes to tool
changing position

Tool changing position return
complete signal (TCP)
```
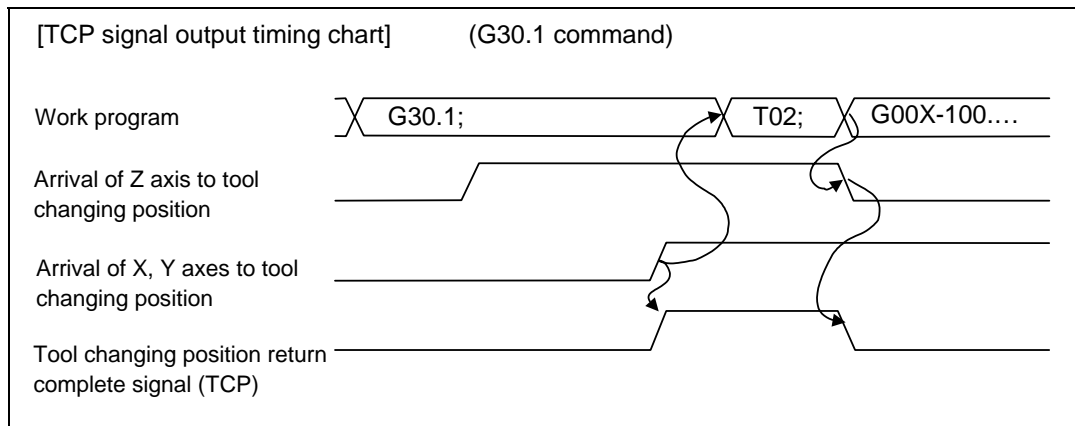
(3) When a tool changing position return command is issued, tool offset data such as for tool length offset and tool diameter compensation for the axis that moved is canceled.

(4) This command is executed by dividing blocks for every axis. If this command is issued during single-block operation, therefore, a block stop occurs each time one axis returns to the tool changing position. To make the next axis return to the tool changing position, therefore, a cycle start needs to be specified.

13.15 High-accuracy control; G61.1 [E68]
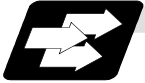
Function and purpose

This function aims to improve the error caused by the accuracy of the control system during machine machining. The parameter method and G code command method, which turn initial high-accuracy ON, are used to enter the high-accuracy control mode.

In G code command method, there are G61.1 (conventional format) and G08. They can be switched with the parameters. The G61.1 command is explained as following.

Trouble such as the following occurred when using normal control:
(1) Corner rounding occurred at linear and linear-connected corners because the following command movement started before the previous command finished. (Refer to Fig. 1)
(2) When cutting circle commands, an error occurred further inside the commanded path, and the resulting cutting path was smaller than the commanded path. (Refer to Fig. 2)

Fig. 1 Rounding at linear corners

Fig. 2 Radius reduction error in circle commands

This function uses the following fix functions to minimize the increase in machining time while reducing the shape error.

(1) Pre-interpolation acceleration/deceleration (linear acceleration/deceleration)

(2) Optimum speed control

(3) Vector accuracy interpolation

(4) Feed forward

(5) Arc entrance/exit speed control

(6) S-pattern filter control

Command format

> **G61.1 Ff1 ;**
> G61.1:   High-accuracy control mode
> Ff1   Feedrate command

The high-accuracy control mode is validated from the block containing the G61.1 command.

```
 G64                 G61.1
```

The high-accuracy control mode is canceled with one of the following G commands.
- G61 (exact stop check)
- G62 (automatic corner override)
- G63 (tapping mode)
- G64 (cutting mode)

(1)   The feedrate command F is clamped by the rapid traverse rate or maximum cutting feedrate set with the parameters.

(2)   The modal holding state of the high-accuracy control mode differs according to the conditions of the base specification parameter "#1151 rstint" (reset initial) and "#1148 I_G611" (initial high-accuracy).

Parameter		Default state	Reset			Emergency stop	Emergency stop cancel	Block inter-ruption	Block stop	NC alarm	OT
Reset initial (#1151)	Initial high accuracy (#1148)	Power ON	Reset 1	Reset 2	Reset & rewind	Emergency stop switch / External emergency stop	Emergency stop switch / External emergency stop	Mode changeover (automatic/manualFeed hold)	Single block	Servo alarm	H/W OT
OFF	OFF	G64	H	G64		H	H		H		
ON			G64				G64				
OFF	ON	G61.1	H	G61.1		H	H				
ON			G61.1				G61.1				

H (hold): Modal hold

Pre-interpolation acceleration/deceleration

Acceleration/deceleration control is carried out for the movement commands to suppress the impact when the machine starts or stops moving. However, with conventional post-interpolation acceleration/deceleration, the corners at the block seams are rounded, and path errors occur regarding the command shape.

In the high-accuracy control function mode, acceleration/deceleration is carried out before interpolation to solve the above problems. This pre-interpolation acceleration/deceleration enables machining on a machining path that more closely follows the command.

The acceleration/deceleration time can be reduced because constant inclination acceleration/deceleration is carried out.

(1) Basic patterns of acceleration/deceleration control in linear interpolation commands

	Acceleration/deceleration waveform pattern	
Normal mode		(a) Because of the constant time constant acceleration/deceleration, the rising edge/falling edge of the waveform becomes more gentle as the command speed becomes slower. (b) The acceleration/deceleration time constant can be independently set for each axis. Linear type, exponential function type, or both can be selected. Note that if the time constant of each axis is not set to the same value, an error will occur in the path course. #2002 clamp : G01 clamp speed #2007 G1tL : Linear type acceleration/ deceleration time constant #2008 G1t1 : Exponential type acceleration/ deceleration time constant
High-accuracy control mode		(a) Because of the constant inclination type linear acceleration/deceleration, the acceleration/deceleration time is reduced as the command speed becomes slower. (b) The acceleration/deceleration time constant becomes one value (common for each axis) in the system. #2002 clamp : G01 clamp speed #1206 G1bF : Target speed #1207 G1btL : Acceleration/deceleration time to target speed **(Note)** G1bF and G1btL are values for specifying the inclination of the acceleration/ deceleration time; the actual cutting feed maximum speed is clamped by the "#2002 clamp" value.

(2) Path control in circular interpolation commands
When commanding circular interpolation with the conventional post-interpolation acceleration/ deceleration control method, the path itself that is output from the CNC to the servo runs further inside the commanded path, and the circle radius becomes smaller than that of the commanded circle. This is due to the influence of the smoothing course droop amount for CNC internal acceleration/deceleration.

With the pre-interpolation acceleration/deceleration control method, the path error is eliminated and a circular path faithful to the command results, because interpolation is carried out after the accelera-tion/deceleration control. Note that the tracking lag due to the position loop control in the servo system is not the target here.

The following shows a comparison of the circle radius reduction error amounts for the conventional post-interpolation acceleration/deceleration control and pre-interpolation acceleration/deceleration control in the high-accuracy control mode.

R : Commanded radius (mm)
ΔR: Radius error   (mm)
F : Cutting feedrate   (mm/min)

The compensation amount of the circle radius reduction error (ΔR) is theoretically calculated as shown in the following table.

Post-interpolation acceleration/deceleration control (normal mode)	Pre-interpolation acceleration/deceleration control (high-accuracy control mode)
Linear acceleration/deceleration $$\Delta R = \frac{1}{2R}\left(\frac{1}{12}Ts^2 + Tp^2\right)\left(\frac{F}{60}\right)^2$$ Exponential function acceleration/deceleration $$\Delta R = \frac{1}{2R}\left(Ts^2 + Tp^2\right)\left(\frac{F}{60}\right)^2$$	Linear acceleration/deceleration $$\Delta R = \frac{1}{2R}\left\{Tp^2\left(1 - Kf^2\right)\right\}\left(\frac{F}{60}\right)^2$$ (a) Because the item Ts can be ignored by using the pre-interpolation acceleration/deceleration control method, the radius reduction error amount can be reduced. (b) Item Tp can be negated by making Kf = 1.

Ts : Acceleration/deceleration time constant in the CNC (s)
Tp : Servo system position loop time constant (s)
Kf : Feed forward coefficient

Optimum speed control

(1) Optimum corner deceleration

By calculating the angle of the seam between blocks, and carrying out acceleration/ deceleration control in which the corner is passed at the optimum speed, highly accurate edge machining can be realized.

When the corner is entered, that corners optimum speed (optimum corner speed) is calculated from the angle with the next block. The machine decelerates to that speed in advance, and then accelerates back to the command speed after the corner is passed.

Corner deceleration is not carried out when blocks are smoothly connected. In this case, the criteria for whether the connection is smooth or not can be designated by the machining parameter "#8020 DCC ANGLE".

When the corner angle is larger than the parameter "DCC ANGLE" for a linear−linear connection, or for a circle, etc., the acceleration ΔV occurs due to the change in the direction of progress after the corner is passed at a speed V.

| V | Speed before entering the corner |
| ΔV | Speed change at the corner |

Speed after the corner is passed

The corner angle V is controlled so that this ΔV value becomes less than the pre-interpolation acceleration/ deceleration tolerable value set in the parameters ("#1206 G1bF", "#1207 G1btL").

In this case the speed pattern is as follows.

N01 G01X100.Y1.F500 ; N02 G01X100.Y-1.F500 ;

The optimum corner speed is represented by V0.

V0 is obtained from the pre-interpolation acceleration/deceleration tolerable value ($\Delta V'$) and the corner angle (outside angle) θ.

$$\Delta V' = \frac{G1bF}{G1btL}$$

Combined speed pattern

$$V0 = \sqrt{V0x^2 + V0y^2}$$

To further reduce the corner speed V0 (to further improve the edge accuracy), the V0 value can be reduced in the machining parameter "#8019 R COMPEN".

X axis speed pattern

$$V0' = \frac{V0 \times (100 - Ks)}{100}$$

Ks: R COMPEN

Y axis speed pattern

(Note 1) In this case, the cycle time may increase due to the increase in the time required for acceleration/ deceleration.

(Note 2) V0 can be increased by setting a negative value for the accuracy coefficient.
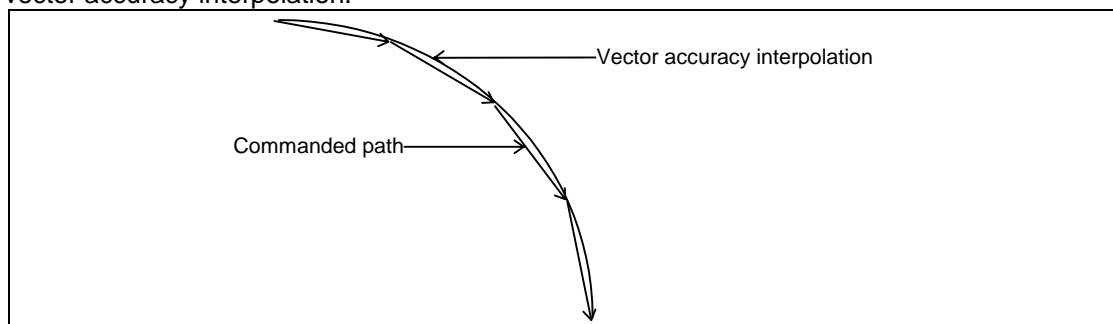
The accuracy coefficient differs according to parameter "#8201 COMP CHANGE".

#8201 COMP CHANGE	Accuracy coefficient used
0	#8019 R COMPEN
1	#8022 CORNER COMP

The corner speed V0 can be maintained at a set speed or more so that the corner speed does not drop too far.
Set "#2096 crncsp" for each axis, and make a composite speed so that the moving axis does not exceed this setting.

Note that the speed is controlled with the optimum corner deceleration speed in the following cases.

• When the composite corner deceleration speed is less than the optimum corner deceleration speed.
• When the corner deceleration minimum speed parameter setting for the moving axes is set to "0" for even one axis.

(2) Arc speed clamp
During circular interpolation, even when moving at a constant speed, acceleration is generated as the advance direction constantly changes. When the arc radius is large compared to the commanded speed, control is carried out at the commanded speed. However, when the arc radius is relatively small, the speed is clamped so that the generated acceleration does not exceed the tolerable acceleration/deceleration speed before interpolation, calculated with the parameters.
This allows arc cutting to be carried out at an optimum speed for the arc radius.

F   : Commanded speed (mm/min)
R   : Commanded arc radius (mm)
$\Delta\theta$: Angle change per interpolation unit
ΔV : Speed change per interpolation unit

The tool is fed with the arc clamp speed F' so that ΔV does not exceed the tolerable acceleration/deceleration speed before interpolation $\Delta V'$.

$$F' \leq \sqrt{R \times \Delta V' \times 60 \times 1000} \ (mm/min)$$

$$\Delta V' = \frac{G1bF\ (mm/min)}{G1btL\ (ms)}$$

When the above F' expression is substituted in the expression expressing the maximum logical arc radius reduction error amount ΔR explained in the section "Pre-interpolation acceleration/deceleration", the commanded radius R is eliminated, and ΔR does not rely on R.

$$\Delta R \leq \frac{1}{2R} \{Tp^2 (1 - Kf^2)\} \left(\frac{F}{60}\right)^2$$

$$\leq \frac{1}{2R} \{Tp^2 (1 - Kf^2)\} \left(\frac{\Delta V' \times 1000}{60}\right)$$

ΔR: Arc radius reduction error amount
Tp : Position loop gain time constant of servo system
Kf : Feed forward coefficient
F : Cutting feedrate

In other words, with the arc command in the high-accuracy control mode, in logical terms regardless of the commanded speed F or commanded radius R, machining can be carried out with a radius reduction error amount within a constant value.

To further lower the arc clamp speed (to further improve the roundness), the arc clamp speed can be lowered with the machining parameter "#8019 R COMPEN". In this case, speed control is carried out to improve the maximum arc radius reduction error amount $\Delta R'$ by the set percentage.

$$\Delta R' \leq \frac{\Delta R \times (100 - Ks)}{100} \ (mm)$$

$\Delta R'$: Maximum arc radius reduction error amount
Ks : R COMPEN (%)

After setting the "R COMPEN", the above $\Delta R'$ will appear on the parameter screen.
#8019 R COMPEN (0.078) 50

Accuracy coefficient
setting value

$\Delta R'$

(Note 1) The maximum arc radius reduction error amount ΔR can be increased by setting a negative value for the "accuracy coefficient".
(Note 2) When the "R COMPEN" is set with positive value, the arc clamp speed will drop, so in a machining program with many arc commands, the machining time will take longer.
(Note 3) If the "accuracy coefficient" is not set (0), arc speed clamping will not be applied.

(Note 4) The "accuracy coefficient" differs according to parameter "#8021 COMP CHANGE".

#8201 COMP CHANGE	Accuracy coefficient used
0	#8019 R COMPEN
1	#8023 CURVE COMP

Vector accuracy interpolation

When a fine segment is commanded and the angle between the blocks is extremely small (when not using optimum corner deceleration), interpolation can be carried out more smoothly using the vector accuracy interpolation.

Feed forward control

With this function, the constant speed error caused by the position loop control of the servo system can be greatly reduced. However, as machine vibration is induced by the feed forward control, there are cases when the coefficient cannot be increased.
In this case, use this function together with the smooth high gain (SHG) control function and stably compensate the delay by the servo system's position loop to realize a high accuracy. As the response is smoother during acceleration/deceleration, the position loop gain can be increased.

(1) Feed forward control

(2) Reduction of arc radius reduction error amount using feed forward control
With the high-accuracy control, the arc radius reduction error amount can be greatly reduced by combining the pre-interpolation acceleration/deceleration control method above-mentioned and the feed forward control/SHG control.
The logical radius reduction error amount ∆R in the high-accuracy control mode is obtained with the following expression.

Feed forward control	SHG control + feed forward control
$$\Delta R \leq \frac{1}{2R} \{ Tp^2 (1 - Kf^2) \} \left(\frac{F}{60} \right)^2$$ R : Arc radius (mm) F : Cutting feedrate (mm/min) Tp : Position loop time constant (s) Kf : Feed forward coefficient	
By setting Kf to the following value, the delay elements caused by the position loop in the servo system can be eliminated, and the logical ∆R can be set to 0.	
Kf = 1 (Feed forward gain 100%)	The equivalent feed forward gain to set Kf to 1 can be obtained with the following expression. $$100 \sqrt{1 - \left\{ 1 - \left(\frac{fwd\_g}{50} \right)^2 \right\} \left(\frac{\text{PGN1 for conventional control}}{2 \times \text{PGN1 for SHG control}} \right)^2}$$

* The feed forward gain can be set independently for G00 and G01.

Path for pre-interpolation acceleration/deceleration control method (Kf = 1)

Path for pre-interpolation acceleration/deceleration control method (Kf = 0)

Path for post-interpolation acceleration/deceleration control method

(Note) If the machine vibrates when Kf is set to 1, Kf must be lowered or the servo system must be adjusted.

Arc entrance/exit speed control

There are cases when the speed fluctuates and the machine vibrates at the joint from the straight line to arc or from the arc to straight line.

This function decelerates to the deceleration speed before entering the arc and after exiting the arc to reduce the machine vibration. If this is overlapped with corner deceleration, the function with the slower deceleration speed is valid.

The validity of this control can be changed with the base specification parameter "#1149 cireft".
The deceleration speed is designated with the base specification parameter "#1209 cirdcc".

(Example 1) When not using corner deceleration

<Program>	<Operation>
G61.1 ; • • N1 G01 X-10. F3000 ; N2 G02 X-5. Y-5. J-2.5 ; N3 G01 Y-10. ; • •	

<Deceleration pattern>

(Example 2) When using corner deceleration

<Program>	<Operation>

G61.1 ;
   •
   •
N1 G01 X-10. F3000 ;
N2 G02 X5. Y-5. I2.5 ;
N3 G01 X10. ;
   •
   •	

<Deceleration pattern>

S-pattern filter control

This control interpolates while smoothing the changes in the segments distributed to each axis element with vector accuracy interpolation. With this, the fluctuation amplified by feed forward control is reduced and the effect onto the machine is reduced.

S-pattern filter can be separately set for G00 and G01 by setting base specification parameter "#1150 Fldc0/bit7" to "1".

In this case, G00 S-pattern filter is set by the base specification parameter "#1150 Fldc0". G01 S-pattern filter is set by the base specification parameter "#1131 Fldcc".

When the base specification parameter "#1150 Fldc0/bit7" is "0", both G00 and G01 S-pattern filter are set with the base specification parameter "#1131 Fldcc".

The S-pattern filter can be set to 7.1, 14.2, 28.4 or 56.8 [ms].

<When filter is invalid>   <When filter is valid>

Command to drive unit

Relation with other functions

(1) The modals must be set as shown below when commanding G61.1.

Function	G code
High-speed high-accuracy control II, High-speed machining cancel	G05 P0
Cylindrical interpolation cancel	G07.1
High-accuracy control cancel	G64 etc.
Tool radius compensation mode cancel	G40
Tool length compensation cancel	G49
Programmable mirror image OFF	G50.1
Mirror image with settings	Cancel
Mirror image with signals	Cancel
No macro modal call	G67
Feed per revolution cancel	G94
Constant surface speed control mode cancel	G97
Interruption type macro mode	M97

(2) An alarm will occur if high-accuracy control is commanded in the following modes.

- During cylindrical interpolation   Program error (P481)

(3) A program error (P29) will occur if the following commands are issued during the high-accuracy control mode.

- Cylindrical interpolation

Precautions

(1) Command G61.1 in an independent block.
(2) The G61.1 command can be used when the basic parameter "#1267 ext03/bit0" is set to "0". If G61.1 is commanded when the parameter is set to "1", a program error (P34) will occur.
(3) Refer to the detailed explanations on the speed clamp during the arc command.

13.16 High-accuracy control; G08 [E68]
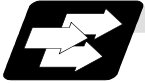
Function and purpose

This function aims to improve the error caused by the accuracy of the control system during machine machining. The parameter method and G code command method, which turn initial high-accuracy ON, are used to enter the high-accuracy control mode.

In G code command method, there are G61.1 (conventional format) and G08. They can be switched with the parameters. The G08 command is explained as following.

Trouble such as the following occurred when using normal control:
(1) Corner rounding occurred at linear and linear-connected corners because the following command movement started before the previous command finished. (Refer to Fig. 1)
(2) When cutting circle commands, an error occurred further inside the commanded path, and the resulting cutting path was smaller than the commanded path. (Refer to Fig. 2)

Fig. 1 Rounding at linear corners

Fig. 2 Radius reduction error in circle commands

This function uses the following fix functions to minimize the increase in machining time while reducing the shape error.

(1) Pre-interpolation acceleration/deceleration (linear acceleration/deceleration)

(2) Optimum speed control

(3) Vector accuracy interpolation

(4) Feed forward

(7) Arc entrance/exit speed control

(8) S-pattern filter control

Command format

```
G08  P1(P0) ;

G08                 : High-accuracy control mode
P1                  : High-accuracy control mode start
P0                  : High-accuracy control mode cancel
```

The "G08 P1" high-accuracy control mode is canceled with P0.
Command G08P_ in an independent block.
The decimal places below the decimal point are ignored for P address.

(Note) G code group for in G08 is "0"; the priority is given to the function of the G code group 0 over the function of the G code group 13. After "G08 P1" is commanded, G code group 13 is changed automatically to G64 (cutting) mode. Other command of "13" results in error. Even if high-accuracy control mode is canceled by "G08 P0" command, G64 (cutting) mode will not be changed. If you want to return to the function of G code group "13" when "G08 P1" has been commanded, command again after high-accuracy control mode is canceled.

(1) The feedrate command F is clamped by the rapid traverse rate or maximum cutting feedrate set with the parameters.

(2) The modal holding state of the high-accuracy control mode differs according to the conditions of the base specification parameter "#1151 rstint" (reset initial) and "#1148 I_G611" (initial high-accuracy).

Parameter		Default state	Reset			Emergency stop	Emergency stop cancel	Block interruption	Block stop	NC alarm	OT
Reset initial (#1151)	Initial high accuracy (#1148)	Power ON	Reset 1	Reset 2	Reset & rewind	Emergency stop switch / External emergency stop	Emergency stop switch / External emergency stop	Mode changeover (automatic/manual) / Feed hold	Single block	Servo alarm	H/W OT
OFF	OFF	G08P0	H	G08P0	G08P0	H	H	H	H	H	H
ON	OFF	G08P0	G08P0	G08P0	G08P0	H	G08P0	H	H	H	H
OFF	ON	G08P1	H	G08P1	G08P1	H	H	H	H	H	H
ON	ON	G08P1	G08P1	G08P1	G08P1	H	G08P1	H	H	H	H

H (hold): Modal hold

Pre-interpolation acceleration/deceleration

Acceleration/deceleration control is carried out for the movement commands to suppress the impact when the machine starts or stops moving. However, with conventional post-interpolation acceleration/deceleration, the corners at the block seams are rounded, and path errors occur regarding the command shape.

In the high-accuracy control function mode, acceleration/deceleration is carried out before interpolation to solve the above problems. This pre-interpolation acceleration/deceleration enables machining on a machining path that more closely follows the command.

The acceleration/deceleration time can be reduced because constant inclination acceleration/deceleration is carried out.

(2)   Basic patterns of acceleration/deceleration control in linear interpolation commands

	Acceleration/deceleration waveform pattern	
Normal mode		(a) Because of the constant time constant acceleration/deceleration, the rising edge/falling edge of the waveform becomes more gentle as the command speed becomes slower. (b) The acceleration/deceleration time constant can be independently set for each axis. Linear type, exponential function type, or both can be selected. Note that if the time constant of each axis is not set to the same value, an error will occur in the path course. #2002 clamp : G01 clamp speed #2007 G1tL : Linear type acceleration/ deceleration time constant #2008 G1t1 : Exponential type acceleration/ deceleration time constant
High-accuracy control mode		(a) Because of the constant inclination type linear acceleration/deceleration, the acceleration/deceleration time is reduced as the command speed becomes slower. (b) The acceleration/deceleration time constant becomes one value (common for each axis) in the system. #2002 clamp : G01 clamp speed #1206 G1bF : Target speed #1207 G1btL : Acceleration/deceleration time to target speed **(Note)** G1bF and G1btL are values for specifying the inclination of the acceleration/ deceleration time; the actual cutting feed maximum speed is clamped by the "#2002 clamp" value.

(2) Path control in circular interpolation commands

When commanding circular interpolation with the conventional post-interpolation acceleration/ deceleration control method, the path itself that is output from the CNC to the servo runs further inside the commanded path, and the circle radius becomes smaller than that of the commanded circle. This is due to the influence of the smoothing course droop amount for CNC internal acceleration/deceleration.

With the pre-interpolation acceleration/deceleration control method, the path error is eliminated and a circular path faithful to the command results, because interpolation is carried out after the acceleration/deceleration control. Note that the tracking lag due to the position loop control in the servo system is not the target here.

The following shows a comparison of the circle radius reduction error amounts for the conventional post-interpolation acceleration/deceleration control and pre-interpolation acceleration/deceleration control in the high-accuracy control mode.

R : Commanded radius (mm)
ΔR: Radius error   (mm)
F : Cutting feedrate   (mm/min)

The compensation amount of the circle radius reduction error (ΔR) is theoretically calculated as shown in the following table.

Post-interpolation acceleration/deceleration control (normal mode)	Pre-interpolation acceleration/deceleration control (high-accuracy control mode)
Linear acceleration/deceleration $$\Delta R = \frac{1}{2R}\left(\frac{1}{12}Ts^2 + Tp^2\right)\left(\frac{F}{60}\right)^2$$ Exponential function acceleration/deceleration $$\Delta R = \frac{1}{2R}\left(Ts^2 + Tp^2\right)\left(\frac{F}{60}\right)^2$$	Linear acceleration/deceleration $$\Delta R = \frac{1}{2R}\left\{Tp^2\left(1 - Kf^2\right)\right\}\left(\frac{F}{60}\right)^2$$ (a) Because the item Ts can be ignored by using the pre-interpolation acceleration/deceleration control method, the radius reduction error amount can be reduced. (b) Item Tp can be negated by making Kf = 1.

Ts : Acceleration/deceleration time constant in the CNC (s)
Tp : Servo system position loop time constant (s)
Kf : Feed forward coefficient

Optimum speed control

(1) Optimum corner deceleration

By calculating the angle of the seam between blocks, and carrying out acceleration/ deceleration control in which the corner is passed at the optimum speed, highly accurate edge machining can be realized.

When the corner is entered, that corners optimum speed (optimum corner speed) is calculated from the angle with the next block. The machine decelerates to that speed in advance, and then accelerates back to the command speed after the corner is passed.

Corner deceleration is not carried out when blocks are smoothly connected. In this case, the criteria for whether the connection is smooth or not can be designated by the machining parameter "#8020 DCC ANGLE".

When the corner angle is larger than the parameter "DCC ANGLE" for a linear−linear connection, or for a circle, etc., the acceleration ΔV occurs due to the change in the direction of progress after the corner is passed at a speed V.

V　Speed before entering the corner
ΔV　Speed change at the corner

Speed after the corner is passed

The corner angle V is controlled so that this ΔV value becomes less than the pre-interpolation acceleration/ deceleration tolerable value set in the parameters ("#1206 G1bF", "#1207 G1btL").

In this case the speed pattern is as follows.

N01 G01X100.Y1.F500 ;　N02 G01X100.Y-1.F500 ;

Combined speed pattern

X axis speed pattern

Y axis speed pattern

The optimum corner speed is represented by V0.
V0 is obtained from the pre-interpolation acceleration/deceleration tolerable value ($\Delta V'$) and the corner angle (outside angle) θ.

$$\Delta V' = \frac{G1bF}{G1btL}$$

To further reduce the corner speed V0 (to further improve the edge accuracy), the V0 value can be reduced in the machining parameter "#8019 R COMPEN".

$$V0' = \frac{V0 \times (100 - Ks)}{100}$$

Ks: R COMPEN

(Note 1) In this case, the cycle time may increase due to the increase in the time required for acceleration/ deceleration.

(Note 2) V0 can be increased by setting a negative value for the accuracy coefficient.

$$V0 = \sqrt{V0x^2 + V0y^2}$$

The accuracy coefficient differs according to parameter "#8201 COMP CHANGE".

#8201 COMP CHANGE	Accuracy coefficient used
0	#8019 R COMPEN
1	#8022 CORNER COMP

The corner speed V0 can be maintained at a set speed or more so that the corner speed does not drop too far.
Set "#2096 crncsp" for each axis, and make a composite speed so that the moving axis does not exceed this setting.

Note that the speed is controlled with the optimum corner deceleration speed in the following cases.

• When the composite corner deceleration speed is less than the optimum corner deceleration speed.
• When the corner deceleration minimum speed parameter setting for the moving axes is set to "0" for even one axis.

(2) Arc speed clamp
   During circular interpolation, even when moving at a constant speed, acceleration is generated as the advance direction constantly changes. When the arc radius is large compared to the commanded speed, control is carried out at the commanded speed. However, when the arc radius is relatively small, the speed is clamped so that the generated acceleration does not exceed the tolerable acceleration/deceleration speed before interpolation, calculated with the parameters.
   This allows arc cutting to be carried out at an optimum speed for the arc radius.

F : Commanded speed (mm/min)
R : Commanded arc radius (mm)
$\Delta\theta$: Angle change per interpolation unit
ΔV : Speed change per interpolation unit

The tool is fed with the arc clamp speed F' so that ΔV does not exceed the tolerable acceleration/deceleration speed before interpolation $\Delta V'$.

$$F' \leq \sqrt{R \times \Delta V' \times 60 \times 1000} \text{ (mm/min)}$$

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

When the above F' expression is substituted in the expression expressing the maximum logical arc radius reduction error amount ΔR explained in the section "Pre-interpolation acceleration/deceleration", the commanded radius R is eliminated, and ΔR does not rely on R.

$$\Delta R \leq \frac{1}{2R} \{ Tp^2 (1 - Kf^2) \} (\frac{F}{60})^2$$

$$\leq \frac{1}{2R} \{ Tp^2 (1 - Kf^2) \} (\frac{\Delta V' \times 1000}{60})$$

ΔR: Arc radius reduction error amount
Tp : Position loop gain time constant of servo system
Kf : Feed forward coefficient
F : Cutting feedrate

In other words, with the arc command in the high-accuracy control mode, in logical terms regardless of the commanded speed F or commanded radius R, machining can be carried out with a radius reduction error amount within a constant value.

To further lower the arc clamp speed (to further improve the roundness), the arc clamp speed can be lowered with the machining parameter "#8019 R COMPEN". In this case, speed control is carried out to improve the maximum arc radius reduction error amount $\Delta R'$ by the set percentage.

$$\Delta R' \leq \frac{\Delta R \times (100 - Ks)}{100} \text{ (mm)}$$

$\Delta R'$: Maximum arc radius reduction error amount
Ks : R COMPEN (%)

After setting the "R COMPEN", the above $\Delta R'$ will appear on the parameter screen.
#8019 R COMPEN (0.078) 50

Accuracy coefficient
setting value
$\Delta R'$

(Note 1) The maximum arc radius reduction error amount ΔR can be increased by setting a negative value for the "accuracy coefficient".
(Note 2) When the "R COMPEN" is set with positive value, the arc clamp speed will drop, so in a machining program with many arc commands, the machining time will take longer.
(Note 3) If the "accuracy coefficient" is not set (0), arc speed clamping will not be applied.

(Note 4) The "accuracy coefficient" differs according to parameter "#8021 COMP CHANGE".

#8201 COMP CHANGE	Accuracy coefficient used
0	#8019 R COMPEN
1	#8023 CURVE COMP

Vector accuracy interpolation

When a fine segment is commanded and the angle between the blocks is extremely small (when not using optimum corner deceleration), interpolation can be carried out more smoothly using the vector accuracy interpolation.

Feed forward control

With this function, the constant speed error caused by the position loop control of the servo system can be greatly reduced. However, as machine vibration is induced by the feed forward control, there are cases when the coefficient cannot be increased.
In this case, use this function together with the smooth high gain (SHG) control function and stably compensate the delay by the servo system's position loop to realize a high accuracy. As the response is smoother during acceleration/deceleration, the position loop gain can be increased.

(1)   Feed forward control

(2) Reduction of arc radius reduction error amount using feed forward control
With the high-accuracy control, the arc radius reduction error amount can be greatly reduced
by combining the pre-interpolation acceleration/deceleration control method above-mentioned
and the feed forward control/SHG control.
The logical radius reduction error amount ΔR in the high-accuracy control mode is obtained
with the following expression.

Feed forward control	SHG control + feed forward control
$$\Delta R \ \le \ \frac{1}{2R} \ \{ Tp^2 (1 - Kf^2) \} \ \left(\frac{F}{60} \right)^2$$ R : Arc radius (mm) F : Cutting feedrate (mm/min) Tp : Position loop time constant (s) Kf : Feed forward coefficient	
By setting Kf to the following value, the delay elements caused by the position loop in the servo system can be eliminated, and the logical ΔR can be set to 0.	
Kf = 1 (Feed forward gain 100%)	The equivalent feed forward gain to set Kf to 1 can be obtained with the following expression. $$100 \sqrt{1 - \left\{ 1 - \left(\frac{fwd\_g}{50} \right)^2 \right\} \left(\frac{\text{PGN1 for conventional control}}{2 \times \text{PGN1 for SHG control}} \right)^2}$$

* The feed forward gain can be set independently for G00 and G01.

Path for pre-interpolation acceleration/deceleration control method (Kf = 1)

Path for pre-interpolation acceleration/deceleration control method (Kf = 0)

Path for post-interpolation acceleration/deceleration control method

(Note) If the machine vibrates when Kf is set to 1, Kf must be lowered or the servo system
must be adjusted.

Arc entrance/exit speed control

There are cases when the speed fluctuates and the machine vibrates at the joint from the straight line to arc or from the arc to straight line.

This function decelerates to the deceleration speed before entering the arc and after exiting the arc to reduce the machine vibration. If this is overlapped with corner deceleration, the function with the slower deceleration speed is valid.

The validity of this control can be changed with the base specification parameter "#1149 cireft".

The deceleration speed is designated with the base specification parameter "#1209 cirdcc".

(Example 1) When not using corner deceleration

(Example 2) When using corner deceleration

<Program>	<Operation>
G08 P1 ; • • N1 G01 X-10. F3000 ; N2 G02 X5. Y-5. I2.5 ; N3 G01 X10. ; • •	

<Deceleration pattern>

S-pattern filter control

This control interpolates while smoothing the changes in the segments distributed to each axis element with vector accuracy interpolation. With this, the fluctuation amplified by feed forward control is reduced and the effect onto the machine is reduced.

S-pattern filter can be separately set for G00 and G01 by setting base specification parameter "#1150 Fldc0/bit7" to "1".

In this case, G00 S-pattern filter is set by the base specification parameter "#1150 Fldc0". G01 S-pattern filter is set by the base specification parameter "#1131 Fldcc".

When the base specification parameter "#1150 Fldc0/bit7" is "0", both G00 and G01 S-pattern filter are set with the base specification parameter "#1131 Fldcc".

The S-pattern filter can be set to 7.1, 14.2, 28.4 or 56.8 [ms].

Relation with other functions

(1) The modals must be set as shown below when commanding G08P1.

Function	G code
High-speed high-accuracy control II, High-speed machining cancel	G05 P0
Cylindrical interpolation cancel	G07.1
High-accuracy control cancel	G08P0
Tool radius compensation mode cancel	G40
Tool length compensation cancel	G49
Programmable mirror image OFF	G50.1
Mirror image with settings	Cancel
Mirror image with signals	Cancel
No macro modal call	G67
Feed per revolution cancel	G94
Constant surface speed control mode cancel	G97
Interruption type macro mode	M97

(2) An alarm will occur if high-accuracy control is commanded in the following modes.

- During cylindrical interpolation   Program error (P481)

(4) A program error (P29) will occur if the following commands are issued during the high-accuracy control mode.

- Cylindrical interpolation

Precautions

(1) Command G08P1 or G08P0 in an independent block.
(2) The G08 command can be used when the basic parameter "#1267 ext03/bit0" is set to "0". If G08 is commanded when the parameter is set to "1", a program error (P34) will occur.
(3) Refer to the detailed explanations on the speed clamp during the arc command.

13.17 High-speed machining mode

13.17.1 High-speed machining mode III [E68]

Function and purpose

This function runs a machining program that approximates a freely curved surface with fine segments at high speed. This is effective in increasing the speed of machining dies of a freely curved surface.

High-accuracy machining can be realized by combined use of this function with the high-accuracy control function.

This function requires the "High-speed machining mode III" specifications.

Command format

G05 P3 ;...........High-speed machining mode III ON
G05 P0 ;...........High-speed machining mode III OFF

Example of program

```
G28  X0.  Y0.  Z0. ;
G91  G00  X-100.  Y-100. ;
G01  F3000 ;                        ⎫
G05  P3 ;                           ⎬  High-speed machining mode III ON

X0.1 ;                              ⎫
X0.1  Y0.001 ;                      ⎬  INC modal, linear cutting
X0.1  Y0.002 ;                      ⎭
G90  G00  X0  Y0  Z0 ;              Rapid traverse
G02  X_Y_R_F_ ;                     ABS modal, R designation arc
G03  X_Y_I_J_F_ ;                   Arc
 •
 •
 •
G05  P0 ;                           High-speed machining mode III OFF
M02;
```

Detailed description

(1) The following G codes can be used in high-speed machining mode III.
G00, G01, G02, G03, G90, G91, G17, G18, G19

(2) It is possible to command only axis names, movement amounts (variables and calculating not possible), and F commands I/J/K/R/P in high-speed machining mode III. (Comments can be used.)

(3) Tape, MDI, IC card and memory operation are all possible with the high-speed machining mode III.

(4) Override, maximum cutting speed clamp, single block operation, dry run, handle interrupt, graphic trace, and high-accuracy control functions are valid even in the high-speed machining mode III modal.

(5) Depending on the No. of characters in one block, the machining speed may decrease.

(6) When high-speed machining mode III is validated, absolute/incremental, G00 to G03, and F command modal information are carried over.

(7) High-speed machining mode III commands are possible in a sub-program.

Program example

Main program	Sub-program	
01	O1000	
•	G05 P3 ;	High-speed machining mode III ON
•		
G61.1 ;	•	
M98 P1000 L100 ;	•	
•	•	
•	•	
•	•	
•	G05 P0 ;	High-speed machining mode III OFF
	M99 ;	

M99 commands cannot be used in the high-speed machining mode III modal.
(A program error (P33) will occur.)
Always turn OFF the high-speed machining mode III before the M99 command.

Restrictions

(1) This function requires the "High-speed machining mode III" specifications.
If G05 is commanded when there are no specifications, program error (P65) will occur.

(2) When machining with the high-speed machining mode III, the program path and actual path may differ.
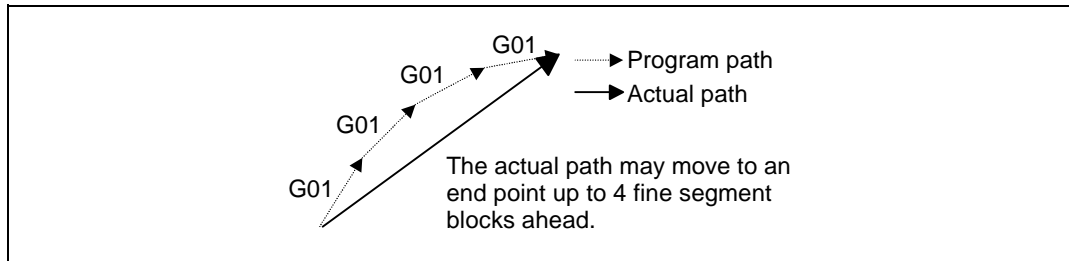
(3) Cancel tool radius compensation, tool length offset, G command mirror image, and program coordinate rotation before commanding G05 P3. If these functions are not canceled, program error (P34) will occur.
The modal information of the Modal Inform. screen will also change to the modal information during the G05 P3 command.
The modal updated during the high-speed machining mode III modal is carried over even after the high-speed machining mode III is canceled.

(4) The program error (P33) will occur if any command other than the following is carried out during high-speed machining mode III modal: Axis names, absolute/incremental movement amounts, F commands, G05 P0, G90/G91, G17/G18/G19, G00/G01/G02/G03, I/J/K/R/P commands, or comments.
Change to the standard mode (G05 P0) before issuing commands other than those above.

(5) The optional block skip (/) cannot be used during high-speed machining mode III modal.
The program error (P33) will occur.

(6) The sequence No. (N) cannot be used during the high-speed machining mode III modal.
The program error (P33) will occur.

(7) The automatic operation process is prioritized during high-speed machining mode III modal, so the monitor display, etc., may be delayed.

(8) The machine decelerates temporarily in the G05 P3 and G05 P0 command blocks, so turn the machine ON/OFF at the places where the tool separates from the workpiece.
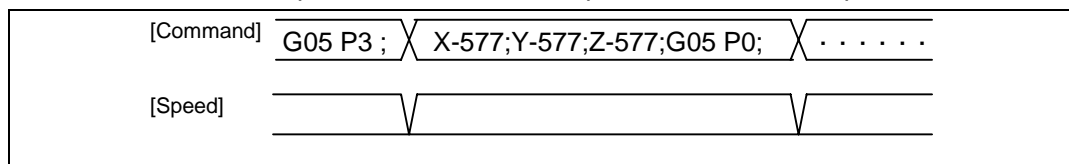
(9) During high-speed machining mode operation by tape operation, the machining speed may be suppressed due to the No. of characters in one block.

(10) If there is no P command in the G05 command block, a program error (P33) will occur.

(11) Decimal points are invalid in P commands.

(12) Program error (P35) will occur for commands other than P0/P3.
Note that program error (P39) will occur for P1/P2 commands.

(13) Variable commands and user macro cannot be used during high-speed machining mode III modal.

(14) If an address other than G, P or N is commanded in the G5P3, G5P0 command block, a program error (P33) will occur.

13.18 Scaling ; G50/G51

Function and purpose

By multiplying the moving axis command values within the range specified under this command by the factor, the shape commanded by the program can be enlarged or reduced to the desired size.

Command format

(1) Scaling ON

G51 Xx Yy Zz Pp ; Scaling ON
Xx, Yy, Zz   : Scaling center coordinates
Pp   : Scaling factor

(2) Scaling cancel

G50 ; Scaling cancel

sc   : Scaling center
p1, p2, p3 : Program shape
s1, s2, s3 : Shape after scaling
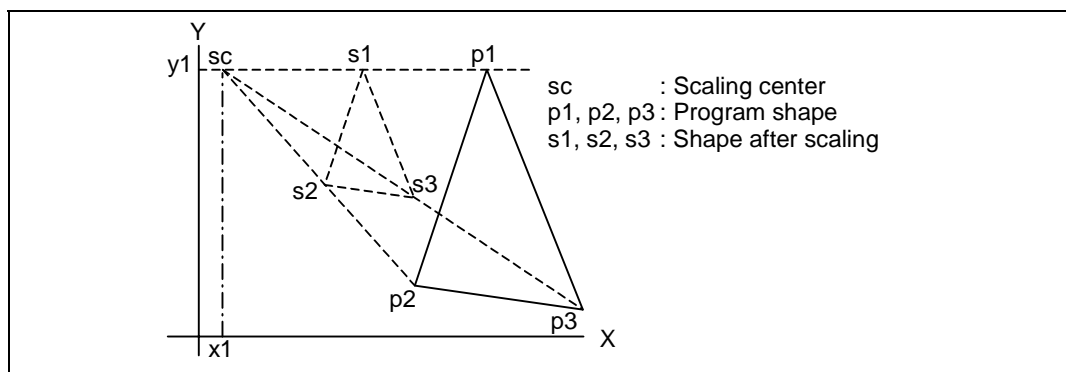
Detailed description

(1) Specifying the scaling axis, scaling center and its factor

Commanding G51 selects the scaling mode. The G51 command only specifies the scaling axis, its center and factor, and does not move the axis.

Though the scaling mode is selected by the G51 command, the axis actually valid for scaling is the axis where the scaling center has been specified.

    (a) Scaling center
- Specify the scaling center in accordance with the then absolute/incremental mode (G90/G91).
- The scaling center must be specified also when the current position is defined as a center.
- As described above, the axis valid for scaling is only the axis whose center has been specified.

    (b) Scaling factor
- Use the address P to specify the scaling factor.
- Minimum command unit : 0.000001
- Command range: Either 1 to 99999999 (0.0000001 to 99.999999 times) or 0.0000001 to 99.999999 is valid, but the decimal point command is valid only after the G51 command.
- When the factor is not specified in the same block as G51, the factor set with the parameter (#8072 SCALING P) is used.
- If changed during the scaling mode, the value of this parameter is not made valid. Scaling is performed with the setting value that was used when G51 was commanded.
- When the factor is not specified in both the program and parameter, it is calculated as 1.

    (c) A program error will occur in either of the following cases.
- Scaling was commanded though there was no scaling specification. (Program error (P350))
- The upper limit of the factor specifying range was exceeded in the same block as G51. (Program error (P 35))
  (If the machining parameter scaling factor is used, a factor less than 0.0000001 or more than 99.999999 is calculated as 1.)

(2) Scaling cancel

When G50 is commanded, scaling is canceled.

Precautions

(1) Scaling is not applied to the offset amounts of tool radius compensation, tool position offset, tool length offset and like. (Compensation and offset are calculated for the shape after scaling.)

(2) Scaling is valid for only the movement command in automatic operation. It is invalid for manual movement.

(3) For X, Y and Z, scaling is valid for only the specified axes and is not applied to the axes not specified.

(4) When an arc is commanded and scaling is valid for one of the two axes configuring the arc plane, a program error (P70) will occur.

(5) Commanding M02, M30 or M00 (only when M0 is reset) during the scaling mode selects the cancel mode.

(6) When the coordinate system is shifted (G92, G52 command) during scaling, the scaling center is also shifted by the difference amount.

(7) If manual interrupt is made during scaling, manual ABS selection is ignored and operation performed is the same as in manual ABS OFF.

Example of program

(Example 1)

Scaling center

N11

N03

N09

N08

N06

N07

D01=25.000

—— Tool path after 1/2 scaling

—— Program path after 1/2 scaling

—— Tool path when scaling is not applied

—— Program path when scaling is not applied

<Program>
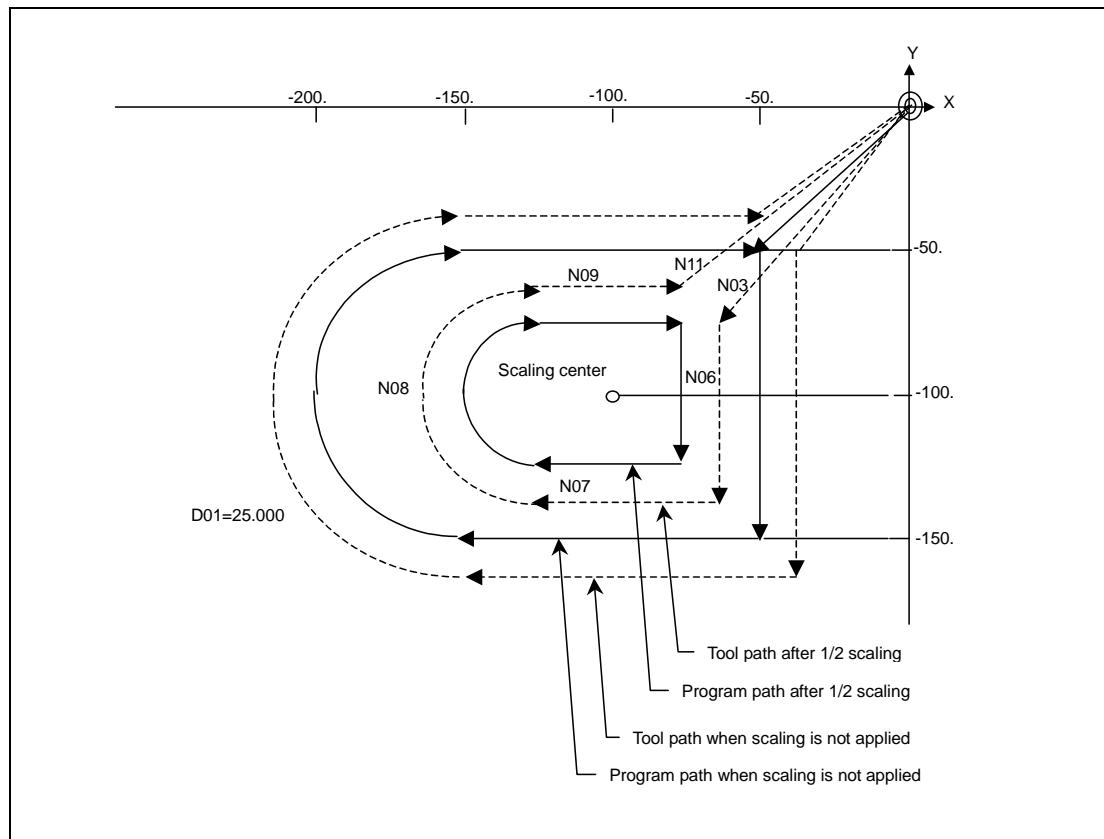
```
N01 G92       X0 Y0 Z0;
N02 G90 G51 X-100. Y-100. P0.5;
N03 G00 G43 Z-200. H02;
N04 G41       X-50.  Y-50.  D01;
N05 G01       Z-250. F1000;
N06           Y-150. F200;
N07           X-150.;
N08 G02       Y-50.  J50.;
N09 G01       X-50.;
N10 G00 G49 Z0;
N11 G40 G50 X0 Y0;
N12 M02;
```

Relation with other functions

(1) G27 reference point verify command

When G27 is commanded during scaling, scaling is canceled at completion of the command.

(2) Reference point return command (G28, G29, G30)

When the G28 or G30 reference point return command is issued during scaling, scaling is canceled at the midpoint and the axis returns to the reference point. When the midpoint is to be ignored, the axis returns to the reference point directly. When G29 is commanded during scaling, scaling is applied to the movement after the midpoint.

(3) G60 (unidirectional positioning) command

If the G60 (unidirectional positioning) command is given during scaling, scaling is applied to the final positioning point and is not applied to the creep amount. Namely, the creep amount is uniform regardless of scaling.

(4) Workpiece coordinate system switching

When the workpiece coordinate system is switched during scaling, the scaling center is shifted by the difference between the offset amounts of the new and old workpiece coordinate systems.

(5) During coordinate rotation

When scaling is commanded during coordinate rotation, the scaling center rotates. Scaling is executed at that rotated scaling center.

(6) G51 command

When the G51 command is issued during the scaling mode, the axis whose center was newly specified is also made valid for scaling. Also, the factor under the latest G51 command is made valid.

13.19 Program coordinate rotation; G68/G69

Function and purpose

When machining a complicated shape at a position rotated in respect to the coordinate system, the shape before rotation can be programmed on the local coordinate system, rotation angle designated with the program coordinate rotation command, and the rotated shaped machined.

Command format

(1) Coordinate rotation ON

G68 Xx Yy Rr; Coordinate rotation ON
G68   : Coordinate rotation command

Xx, Yy   : Rotation center coordinates.
             Two axes (X,Y or Z) corresponding to the selected plane are designated
             with absolute positions.

Rr   : Rotation angle   The counterclockwise direction is +.

Select the command plane with G17 to G19.

(2) Coordinate rotation cancel

G69 ; Coordinate rotation cancel
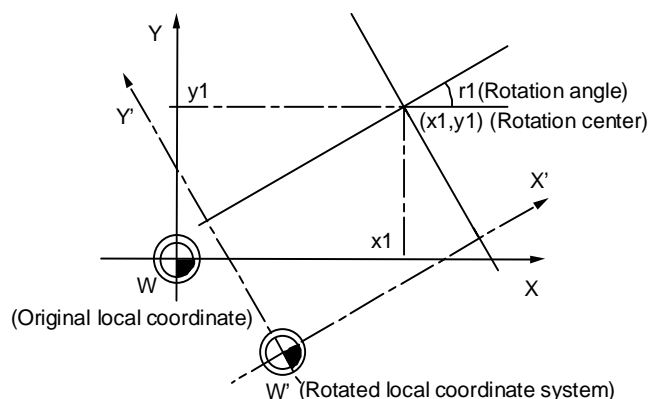
Detailed description

(1) Always command the rotation center coordinate (x1, y1) with an absolute value. Even if commanded with an incremental address, it will not be handled as an incremental value.

(2) If the rotation center coordinates (x1, y1) are omitted, the position where the G68 command was executed will be the rotation center.

(3) The rotation takes place in the counterclockwise direction by the angle designated in rotation angle r1.

(4) The rotation angle r1 setting range is -360.000 to 360.000. If a command exceeding 360 degrees is issued, the remainder divided by 360 degrees will be the command.

(5) The program coordinate rotation is a function used on the local coordinate system. The relation of the rotated coordinate system, workpiece coordinate system and basic machine coordinate system is shown below.
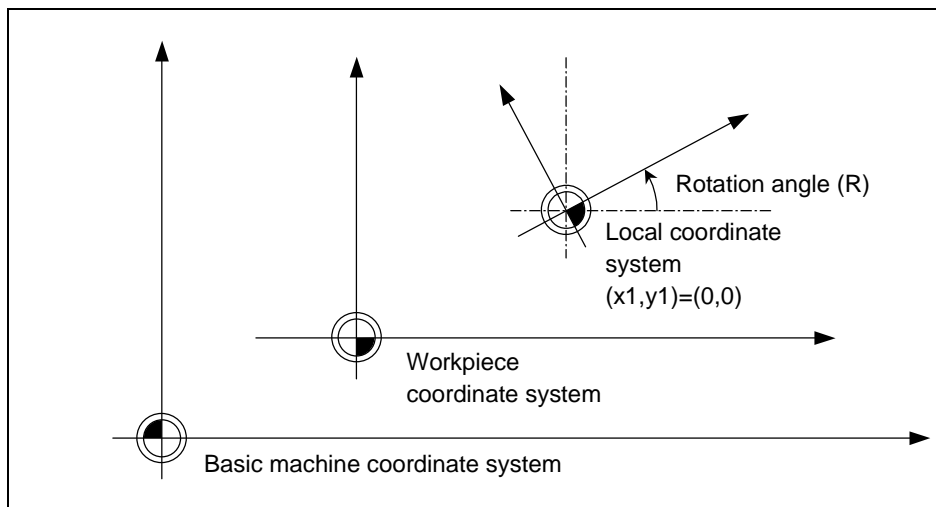
(6) The coordinate rotation command during coordinate rotation is processed as center coordinates and rotation angle changes.

(7) If M02 or M30 is commanded or the reset signal is input during the coordinate rotation mode, the coordinate rotation mode will be canceled.

(8) G68 is displayed on the modal information screen during the coordinate rotation mode. When the mode is canceled, the display changes to G69. (The modal value is not displayed for the rotation angle command R.)

(9) The program coordinate rotation function is valid only in the automatic operation mode.

Example of program

```
N01 G28 X0. Y0.;
N02 G54 G52 X200. Y100. ;        Local coordinate designation
N03 T10 ;
N04 G68 X-100. Y0. R60. ;        Coordinate rotation ON
N05 M98 H101 ;                   Subprogram execution
N06 G69 ;                        Coordinate rotation cancel
N07 G54 G52 X0 Y0 ;              Local coordinate system cancel
N08 M02 ;                        End

Subprogram
(Shape programmed with original coordinate system)
N101 G00 X-100. Y-40.;
N102 G83 X-150. R-20. F100 ;
N103 G00 Y40. ;
N104 G83 X-150. R-20. F100 ;
N105 M99
```
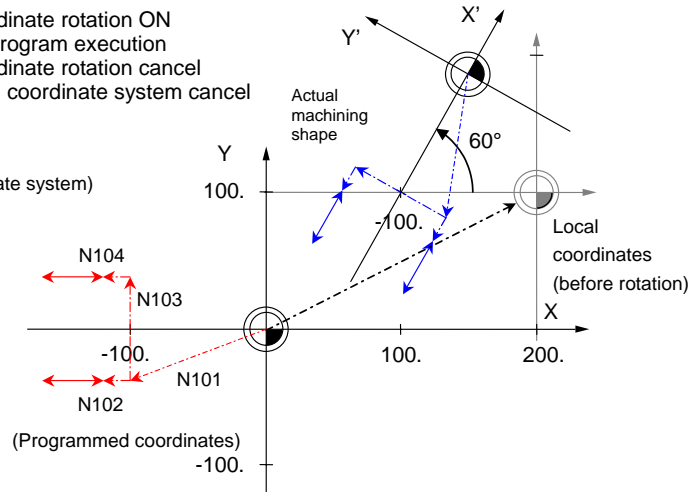
Precautions

(1) Always command an absolute value for the movement command immediately after G68 and G69.

(2) If manual absolute is ON and manual interrupt is issued for the coordinate rotation axis, do not use automatic operation for the following absolute value command.

(3) The intermediate point during reference point return is the position after the coordinates are rotated.

(4) If the workpiece coordinate system offset amount is changed during the coordinate rotation mode, the rotation center for the program coordinate rotation will be shifted. (The center will follow the coordinate system.)

(5) If workpiece coordinate system setting (G92) is executed during the coordinate rotation mode, the rotation center for program coordinate rotation will not shift. (The center position will be the same looking from the basic machine coordinate system.)

(6) If the workpiece coordinates are changed during the coordinate rotation mode (ex., from G54 to G55), the rotation center of the program coordinate rotation will be the position in the commanded coordinate system. (The center position will be the same looking from the basic machine coordinate system.)

(7) If coordinate rotation is executed to the G00 command for only one axis, two axes will move. If G00 non-interpolation (parameter "#1086 G0Intp" = 1) is set, each axis will move independently at the respective rapid traverse rates. If the axis must be moved linearly (interpolated) from the start point to the end point (such as during the hole machining cycle), always turn G00 non-interpolation OFF (parameter "#1086 G0Intp" = 0). The feedrate in this case is the composite speed of each axis' rapid traverse rate, so the movement speed will be faster than when moving only one axis (before coordinate rotation).

(8) If the coordinate rotation specifications are not provided, a program error (P260) will occur when coordinate rotation is commanded.

(9) The offset operation during the coordinate rotation mode compensates the local coordinate system after coordinate rotation. The compensation direction is the coordinate system before rotation.

(10) Mirror image during the coordinate rotation mode is applied on the local coordinate system after coordinate rotation.

(11) All position displays show the positions after coordinate rotation with the local coordinate system before rotation.

(12) When the coordinate value variables are read, the positions are all on the coordinate system before rotation.

(13) The coordinates can also be rotated for the a parallel axis. Select the plane that contains the parallel axis before issuing the G68 command. The plane cannot be selected in the same block as the G68 command.

(14) The coordinates can be rotated for the rotation axis. The angle will be interpreted as a length when rotating.

Relation with other functions

(1) Program error (P111) will occur if the plane selection code is commanded during the coordinate rotation mode.

(2) Program error (P485) will occur if pole coordinate interpolation is commanded during the coordinate rotation mode.

(3) Program error (P481) will occur if coordinate rotation is commanded during the pole coordinate interpolation mode.

(4) Program error (P485) will occur if cylindrical interpolation is commanded during the coordinate rotation mode.

(5) Program error (P481) will occur if coordinate rotation is commanded during the cylindrical interpolation mode.

(6) Program error (P34) will occur if the workpiece coordinate system preset (G92.1) is commanded during the coordinate rotation mode.

(7) Program error (P34) will occur if high-accuracy control mode, high-speed machining mode 3 is commanded during the coordinate rotation mode.

14. Coordinates System Setting Functions

14.1 Coordinate words and control axes

Function and purpose

There are three controlled axis for the basic specifications. When using E68, up to 4 axes can be controlled.

Pre-determined alphabetic coordinate words that correspond to the axes are used to designate each machining direction.

For XY table

X-Y and revolving table [E68]
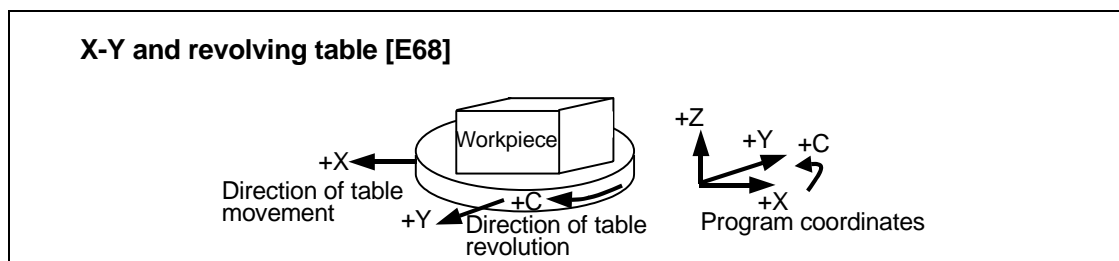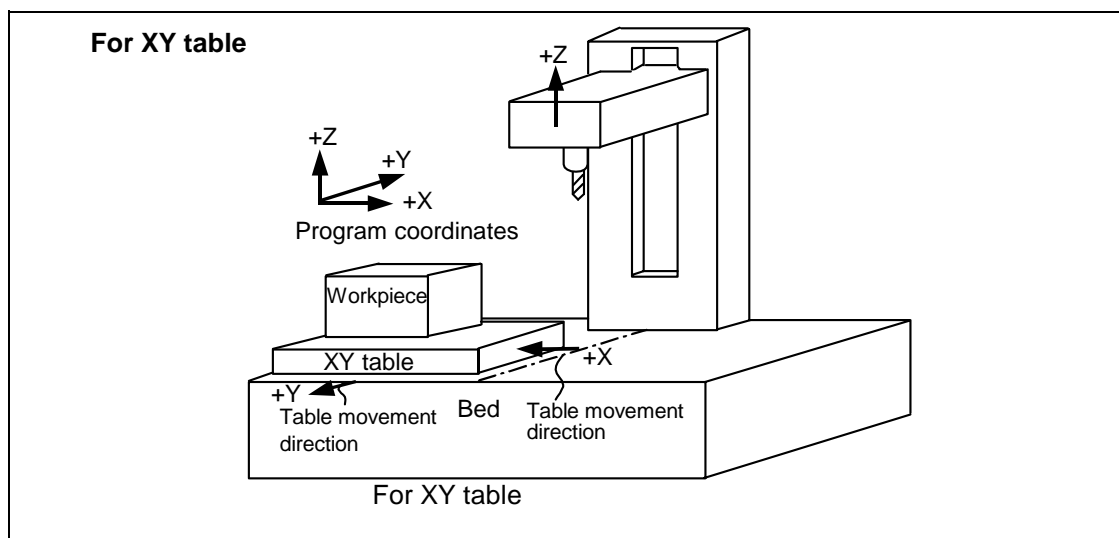
14.2 Basic machine, work and local coordinate systems

Function and purpose

The basic machine coordinate system is fixed in the machine and it denotes that position which is determined inherently by the machine.

The work coordinate systems are used for programming and in these systems the reference point on the workpiece is set as the coordinate zero point.

the local coordinate systems are created on the work coordinate systems and they are designed to facilitate the programs for parts machining.
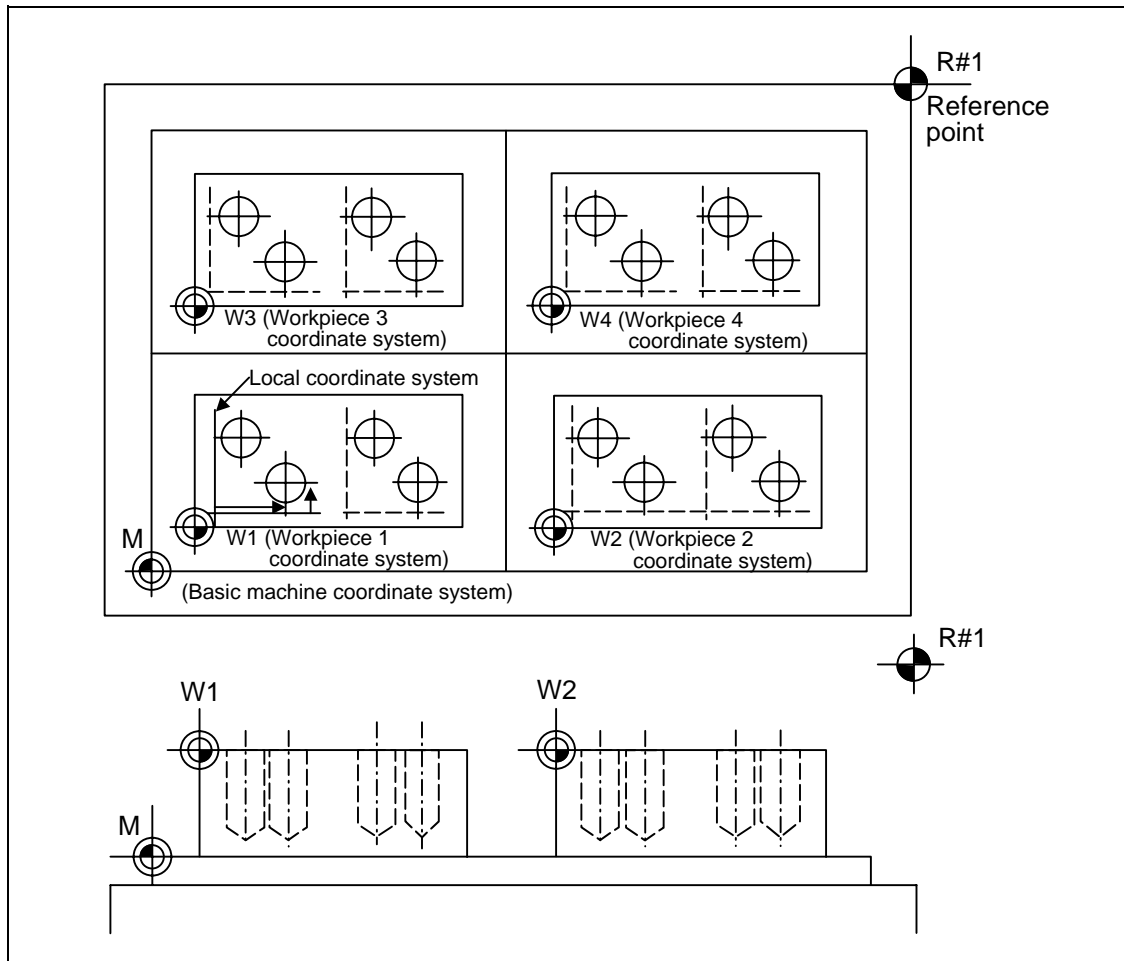
14.3 Machine zero point and 2nd, 3rd, 4th reference points (Zero point)

Function and purpose

The machine zero point serves as the reference for the basic machine coordinate system. It is inherent to the machine and is determined by the reference (zero) point return.
2nd, 3rd and 4th reference (zero points) points (zero points) relate to the position of the coordinates which have been set beforehand by parameter from the zero point of the basic machine coordinate system.
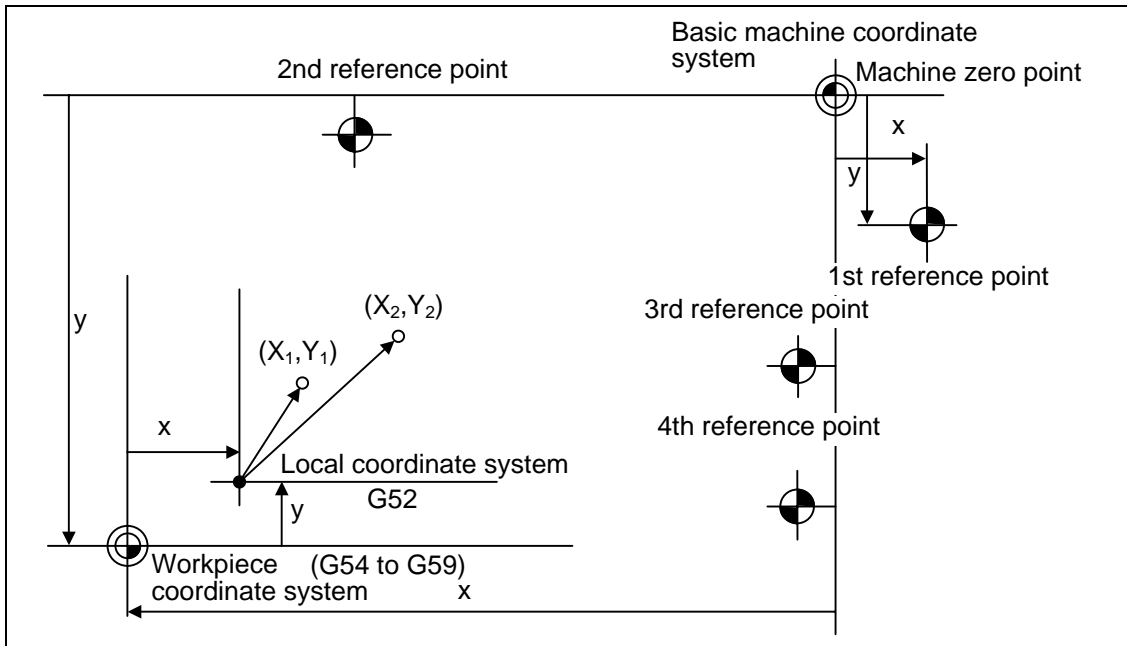
14.4 Basic machine coordinate system

Function and purpose

The basic machine coordinate system is the coordinate system that expresses the position (tool change position, stroke end position, etc.) that is characteristic to the machine.
The tool is moved to the position commanded on the basic machine coordinate system with the G53 command and the coordinate command that follows.

Command format

Basic machine coordinate system selection

(G90) G53 Xx Yy Zz $\alpha\alpha$;
$\alpha\alpha$:Additional axis

Detailed description

(1) When the power is switched on, the basic machine coordinate system is automatically set as referenced to the reference (zero) point return position, which is determined by the automatic or manual reference (zero) point return.

(2) The basic machine coordinate system is not changed by the G92 command.

(3) The G53 command is valid only in the block in which it has been designated.

(4) In the incremental value command mode (G91), the G53 command provides movement with the incremental value in the coordinate system being selected.

(5) Even if G53 is commanded, the tool diameter offset amount for the commanded axis will not be canceled.

(6) The 1st reference point coordinate value indicates the distance from the basic machine coordinate system 0 point to the reference point (zero point) return position.

(7) The G53 commands will all move with rapid traverse.

(8) If the G53 command and G28 command (reference point return) are issued in the same block, the command issued last will be valid.
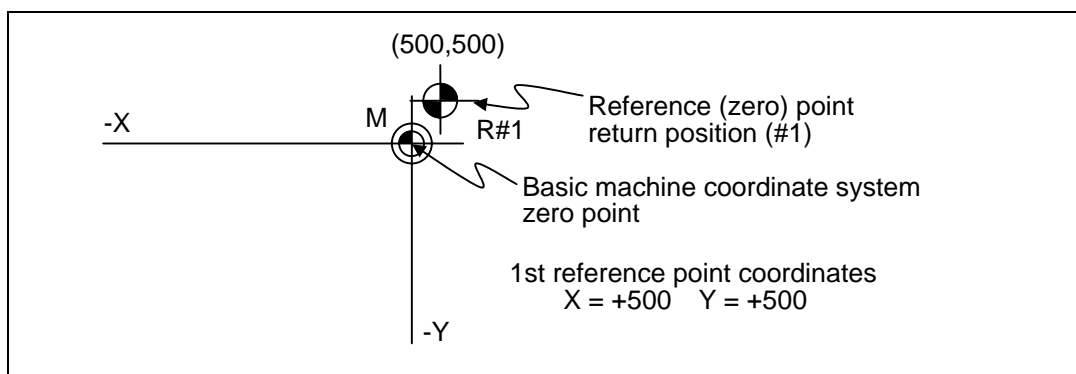
14.5 Coordinate system setting ;G92

Function and purpose

By commanding G92, the absolute value (workpiece) coordinate system and current position display value can be preset in the command value without moving the machine.

Command format

$$\text{G92 } Xx_1 \text{ } Yy_1 \text{ } Zz_1 \text{ } \alpha\alpha_1 \text{ ;}$$

$\alpha\alpha$:Additional axis

Detailed description

(1) After the power is turned on, the first reference point return will be done with dog-type, and when completed, the coordinate system will be set automatically. (Automatic coordinate system setting)
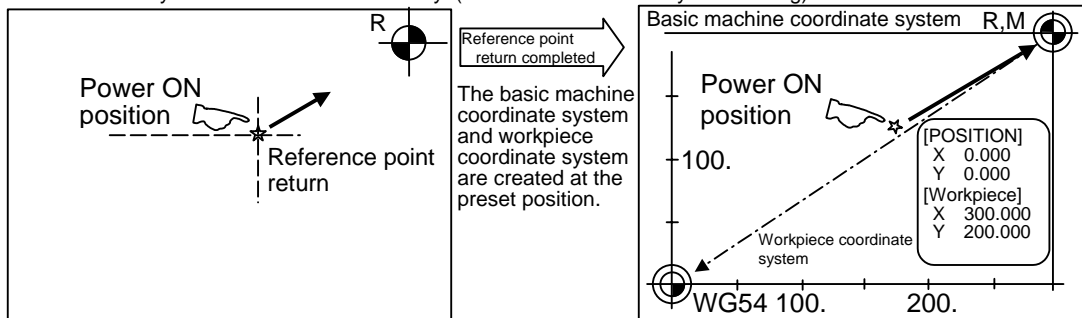
(2) By commanding G92, the absolute value (workpiece) coordinate system and POSITION display value can be preset in the command value without moving the machine.
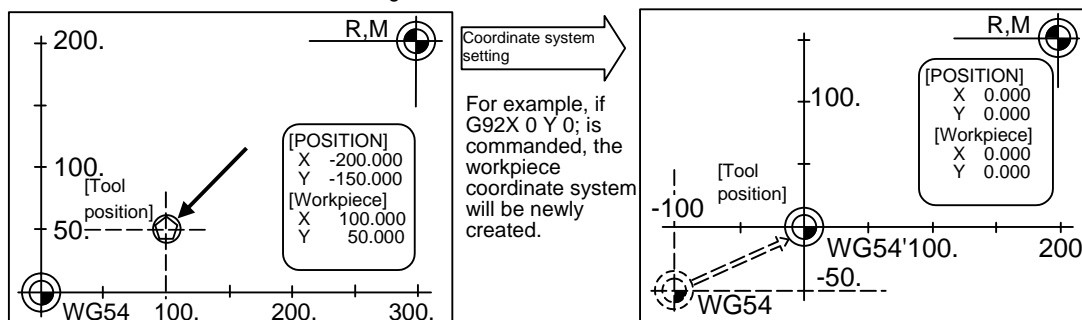
(Note) If the workpiece coordinate system deviated because the axis is moved manually when the manual absolute position switch is OFF, etc., the workpiece coordinate system can be corrected with the following steps.
(1) Execute reference point return while the coordinate system is deviated.
(2) After that, command G92G53X0Y0Z0;. With this command, the workpiece coordinate value and relative position will be displayed, and the workpiece coordinate system will be preset to the offset value.

14.6 Automatic coordinate system setting

Function and purpose

This function creates each coordinate system according to the parameter values input beforehand from the setting and display unit when the reference point is reached with the first manual reference point return or dog-type reference point return when the NC power is turned ON.
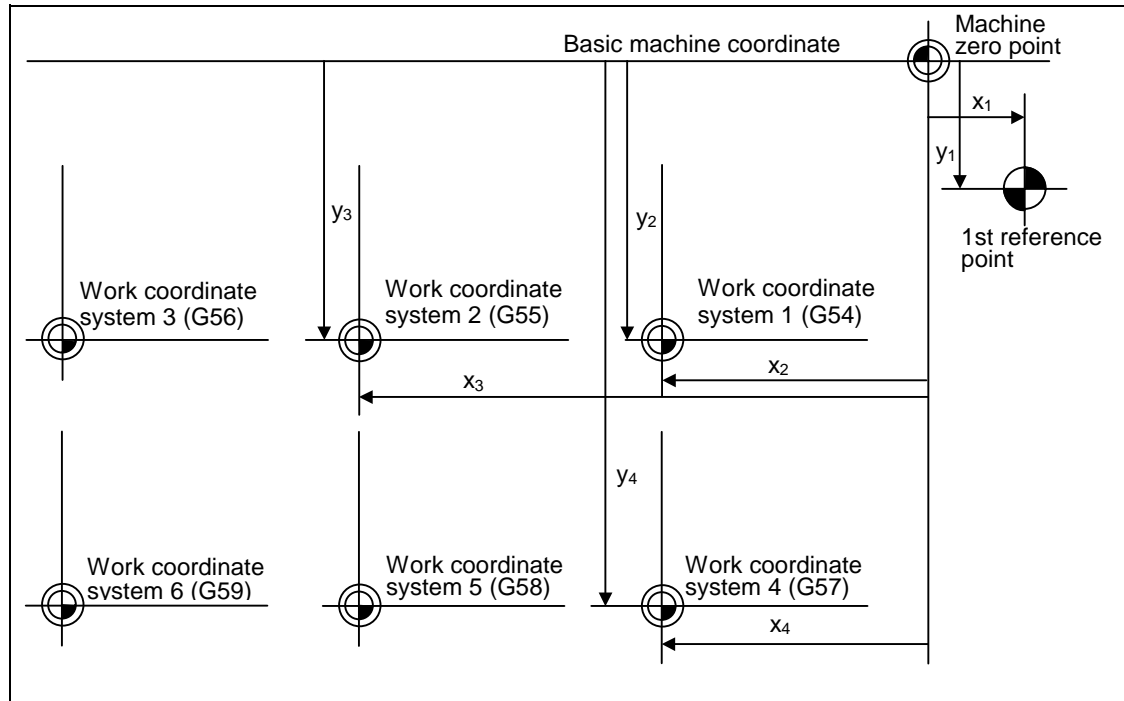
Detailed description

(1) The coordinate systems created by this function are as follow:
   (a) Basic machine coordinate system
   (b) Work coordinate systems (G54 to G59)

(2) The parameters related to the coordinate system all provide the distance from the zero point of the basic machine coordinate system. Therefore, it is decided at which position in the basic machine coordinate system the first reference point should be set and then the zero point positions of the work coordinate systems are set.

(3) When the automatic coordinate system setting function is executed, shifting of the workpiece coordinate system with G92, setting of the local coordinate system with G52, shifting of the workpiece coordinate system with origin set, and shifting of the workpiece coordinate system with manual interrupt will be canceled.

(4) The dog-type reference point return will be executed in the following conditions.
   • The first manual reference point return after power ON.
   • The first automatic reference point return after power ON.
   • The second and following manual reference point return when the dog-type is selected with a parameter.
   • The second and following automatic reference point return when the dog-type is selected with a parameter.

14.7 Reference (zero) point return; G28, G29

Function and purpose

(1) After the commanded axes have been positioned by G0, they are returned respectively at rapid traverse to the first reference (zero) point when G28 is commanded.

(2) By commanding G29, the axes are first positioned independently at high speed to the G28 or G30 intermediate point and then positioned by G0 at the commanded position.
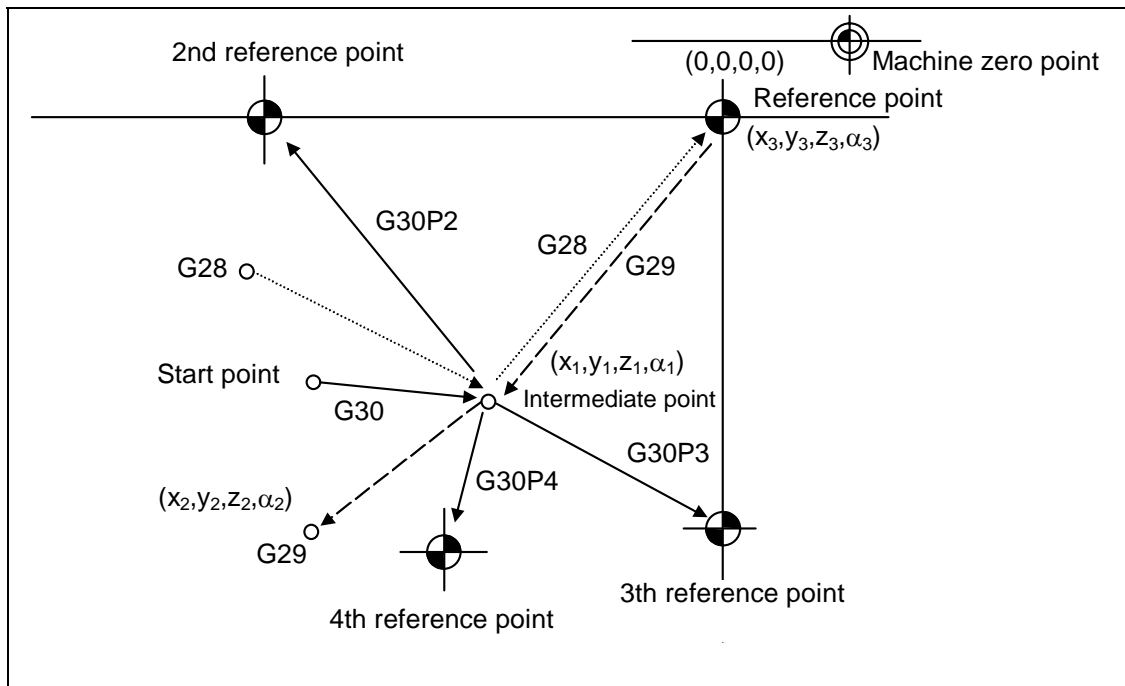
Command format

G28 Xx$_1$ Yy$_1$ Zz$_1$ αα$_1$; Automatic reference point return
G29 Xx$_2$ Yy$_2$ Zz$_2$ αα$_2$; Start position return
αα$_1$/αα$_2$: additional axis

Detailed description

(1) The G28 command is equivalent to the following:

G00 Xx_1 Yy_1 Zz_1 ;
G00 Xx_3 Yy_3 Zz_3 ;

In this case, x_3, y_3 and z_3 are the reference point coordinates and they are set by a parameter "#2037 G53ofs" as the distance from the zero point of the basic machine coordinate system.

(2) After the power has been switched on, the axes which have not been subject to manual reference (zero) point are returned by the dog type of return just as with the manual type. In this case, the return direction is regarded as the command sign direction. If the return type is straight-type return, the return direction will not be checked. For the second and subsequence returns, the return is made at high speed to the reference (zero) point which was stored at the first time and the direction is not checked.

(3) When reference (zero) point return is completed, the zero point arrival output signal is output and also #1 appears at the axis name line on the setting and display unit screen.

(4) The G29 command is equivalent to the following:

G00 Xx_1 Yy_1 Zz_1 ;　　Rapid traverse (non-interpolation type) applies
G00 Xx_2 Yy_2 Zz_2 ;　　independently for each axis for the positioning from the reference point to the intermediate point.

In this case, x_1, y_1 and z_1 are the coordinates of the G28 or G30 intermediate point.

(5) Program error (P430) results when G29 is executed if automatic reference (zero) point return (G28) is not performed after the power has been switched on.

(6) When the Z axis is canceled, the movement of the Z axis to the intermediate point will be ignored, and only the position display for the following positioning will be executed. (The machine itself will not move.)

(7) The intermediate point coordinates (x_1, y_1, z_1) of the positioning point are assigned by the position command modal. (G90, G91).

(8) G29 is valid for either G28 or G30 but the commanded axes are positioned after a return has been made to the latest intermediate point.

(9) The tool offset will be canceled during reference point return unless it is already canceled, and the offset amount will be cleared.

Example of program

(Example1) G28 Xx_1 Zz_1 ;

R

Reference (zero) point
position (#1)

1st operation after power
has been switched on

$G0Xx_3Zz_3$;

2nd and subsequent
operations

(x_1, z_1)
Intermediate point

$G0Xx_1 Zz_1$;

Return start position

1st operation after power
has been switched on

2nd and subsequent
operations

Rapid traverse rate

Near-point dog

Reference (zero) point
position (#1)

R

(Example2) G29 $Xx_2 Zz_2$;

R

Present position

$(G0)Xx_1 Zz_1$;

G28, G30 intermediate point (x_1, z_1)
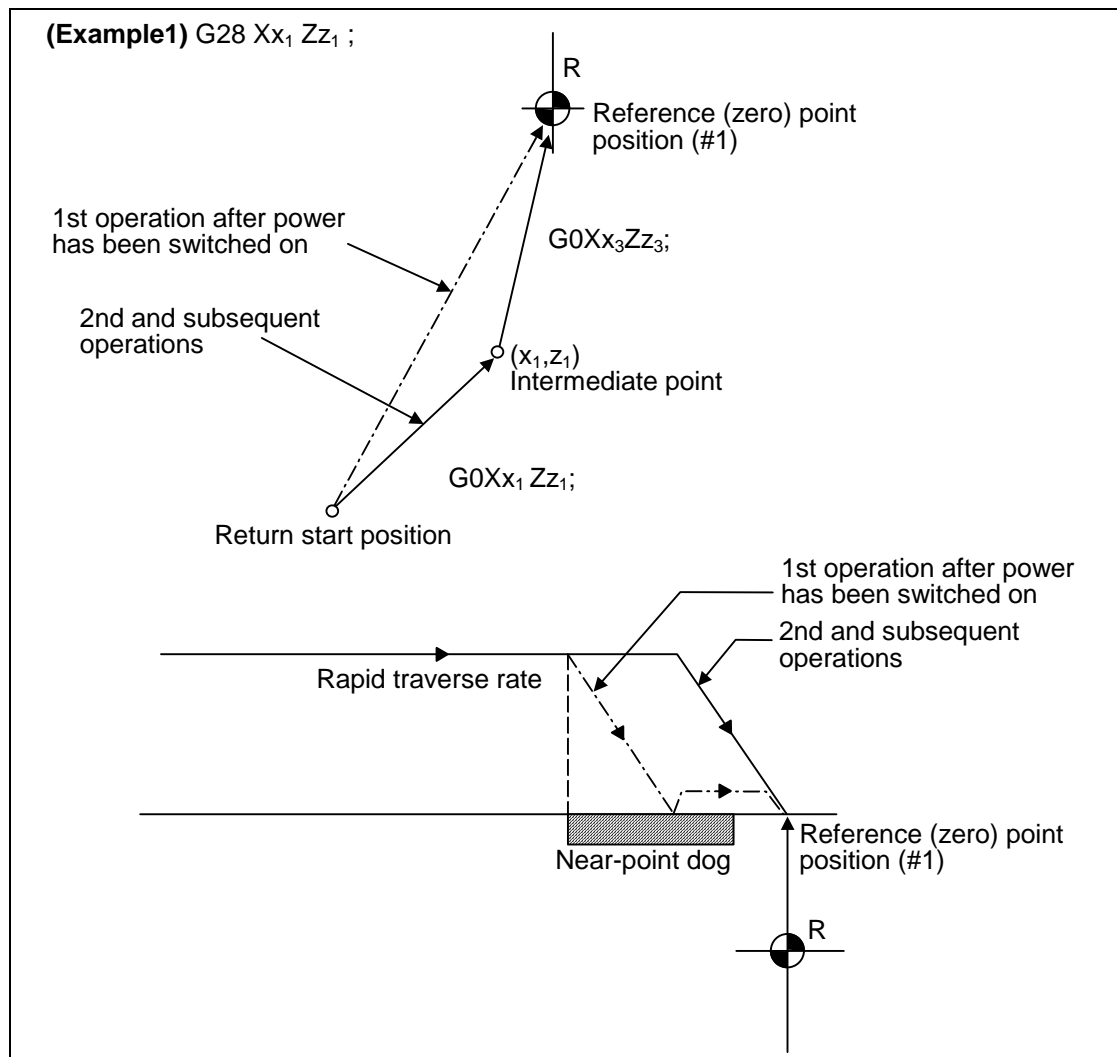
G0 $Xx_2 Zz_2$;

(x_2, z_2)
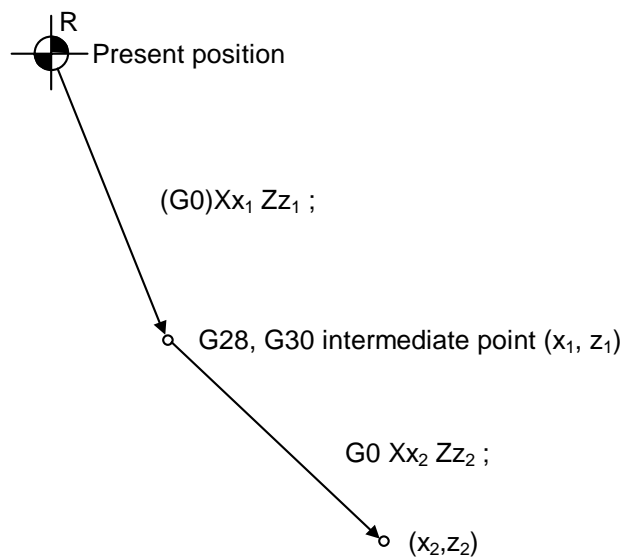
(Example 3)

G28 $Xx_1 Zz_1$; (From point A to reference (zero) point)
G30 $Xx_2 Zz_2$; (From point B to 2nd reference (zero) point)
G29 $Xx_3 Zz_3$; (From point C to point D)
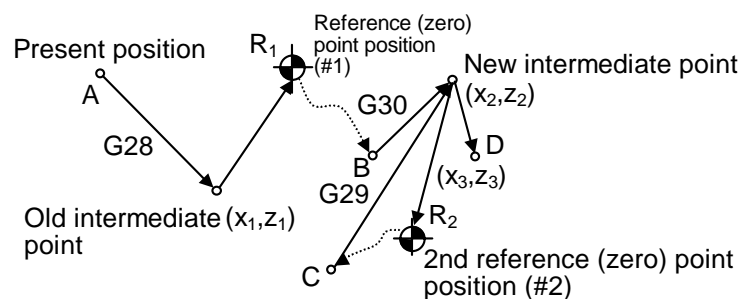
Present position

R_1

Reference (zero)
point position
(#1)

New intermediate point
(x_2, z_2)

A

G30

G28

D

B

(x_3, z_3)

Old intermediate (x_1, z_1)
point

G29

R_2

C

2nd reference (zero) point
position (#2)

366

14.8 2nd, 3rd and 4th reference (zero) point return; G30

Function and purpose

The tool can return to the second, third, or fourth reference (zero) point by specifying G30 P2 (P3 or P4).
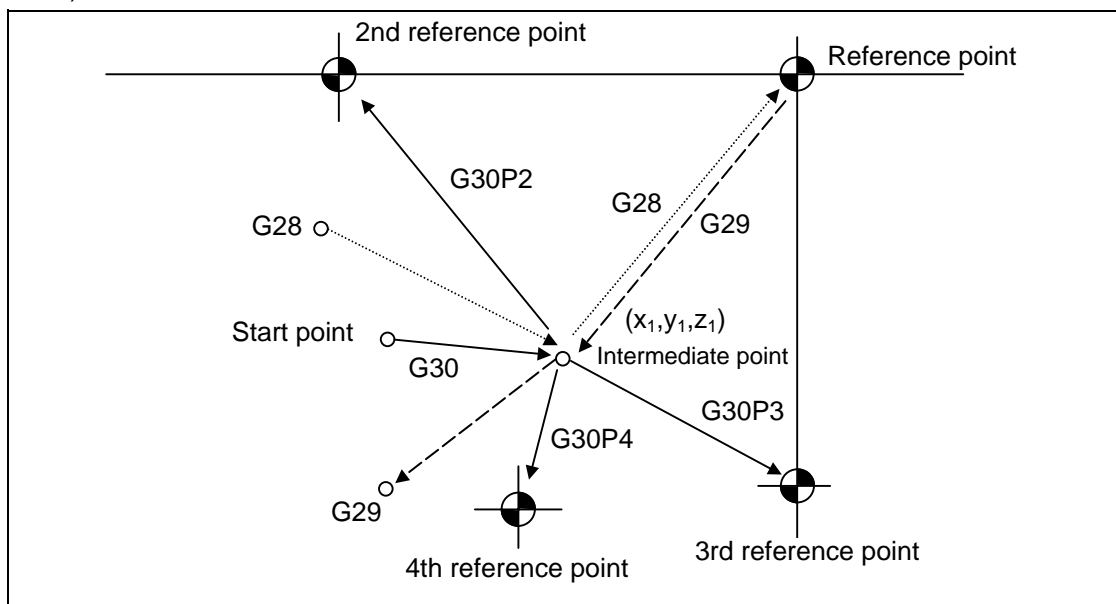
Command format

G30 P2 (P3, P4) Xx_1 Yy_1 Zz_1 aa_1;
$\alpha\alpha_1$:Additional axis

Detailed description

(1) The second, third, or fourth reference (zero) point return is specified by P2, P3, or P4. A command without P or with P0, P1, P5 or a greater P number is ignored, returning the tool to the second reference (zero) point.

(2) In the second, third, or fourth reference (zero) point return mode, as in the first reference (zero) point return mode, the tool returns to the second, third, or fourth reference (zero) pint via the intermediate point specified by G30.

(3) The second, third, and fourth reference (zero) point coordinates refer to the positions specific to the machine, and these can be checked with the setting and display unit.

(4) If G29 is specified after completion of returning to the second, third, and fourth reference (zero) points, the intermediate position used last is used as the intermediate position for returning by G29.
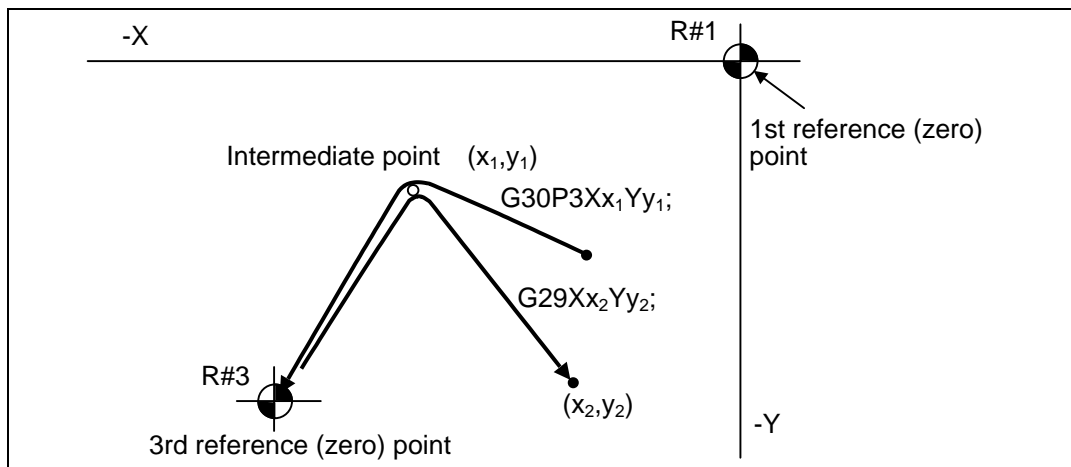
(5) With reference (zero) point return on a plane during compensation, the tool moves without tool diameter compensation (zero compensation) from the intermediate point. with a subsequent G29 command, the tool moves with tool diameter compensation until the G29 command from the intermediate point.
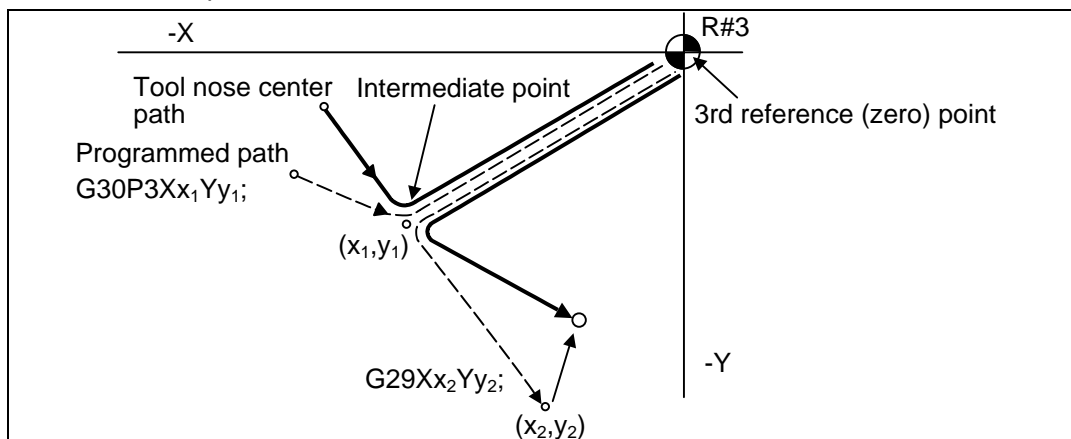
(6) The tool length offset amount for the axis involved is canceled after the second, third and fourth reference (zero) point returns.

(7) With second, third and fourth reference (zero) point returns in the machine lock status, control from the intermediate point to the reference (zero) point will be ignored. When the designated axis reaches as far as the intermediate point, the next block will be executed.

(8) With second, third and fourth reference (zero) point returns in the mirror image mode, mirror image will be valid from the start point to the intermediate point and the tool will move in the opposite direction to that of the command. However, mirror image is ignored from the intermediate point to the reference (zero) point and the tool moves to the reference (zero) point.

-X

R#3

3rd reference (zero) point

X-axis mirror image

-Y

G30P3Xx_1Yy_1;

No mirror image
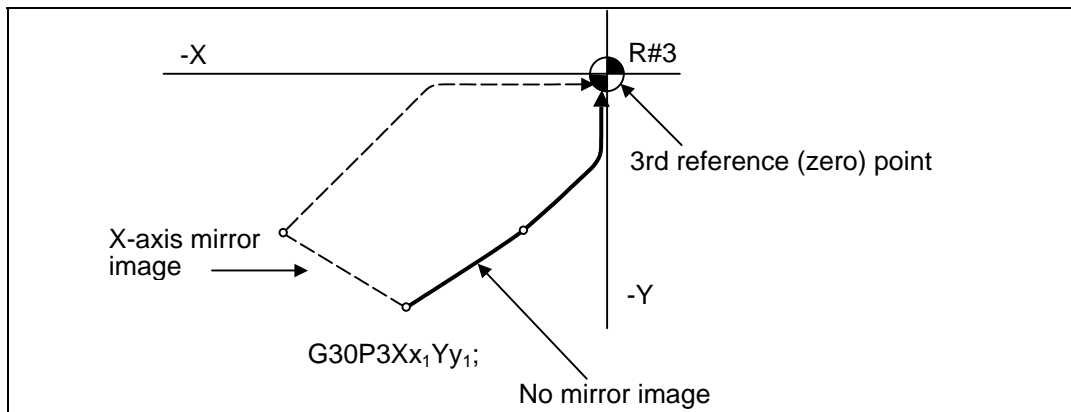
14.9 Simple zero point return

Function and purpose

By setting the parameters (#1222 aux06/bit7) , the G28, G29 and G30 positioning operations can be simplified and the time shortened.
The time is shortened by approx. several ten ms.
When this function is validated, the positioning accuracy may drop.

Methods of shortening time

(1) G90 without intermediate point

Fig. 1   The graph shows the commanded speed during movement.
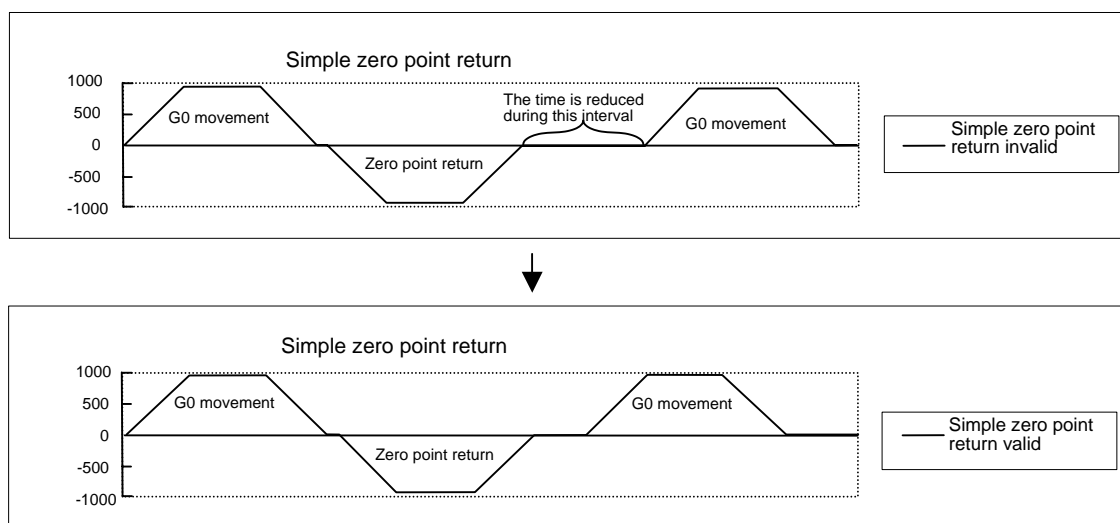
(2) G90 with intermediate point

Fig. 2   The graph shows the commanded speed during movement.
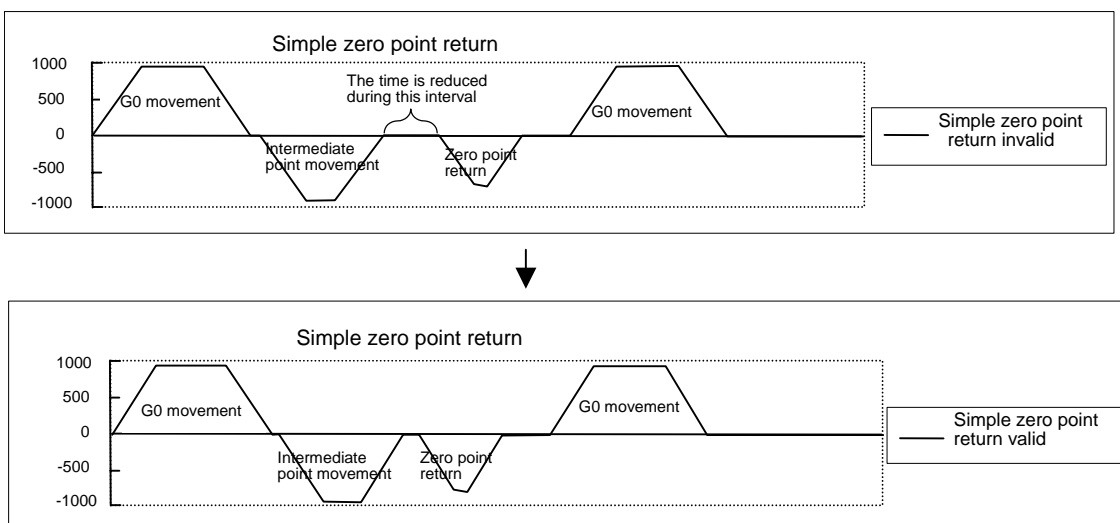
(3) G91 without intermediate point
Fig. 3   The graph shows the commanded speed during movement.

Simple zero point return
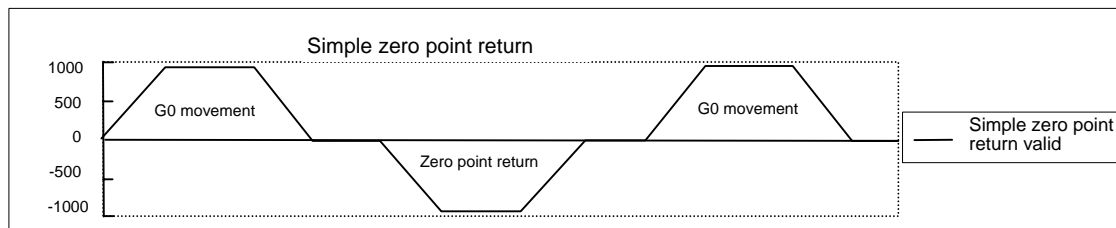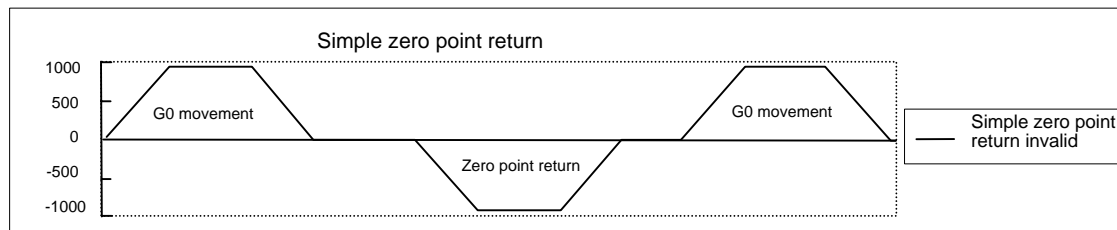
(4) G91 with intermediate point

Fig. 4   The graph shows the commanded speed during movement.
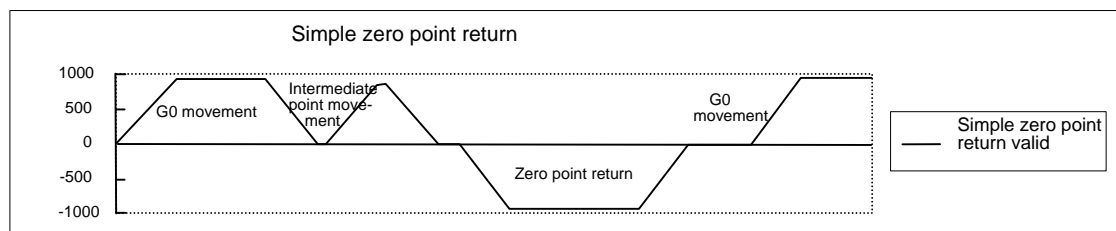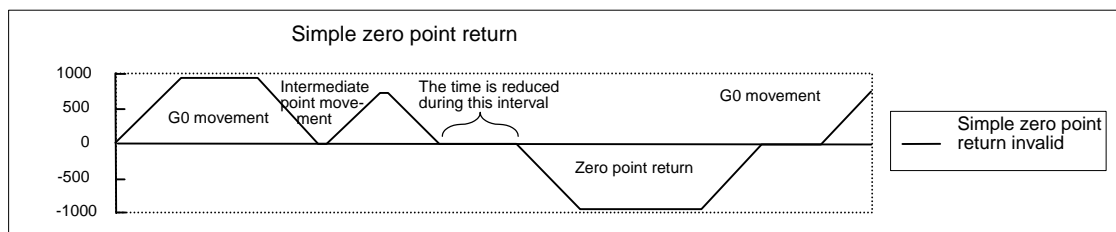
Simple zero point return

Accuracy

(1) G90 without intermediate point

```
(Program example)
G90;
G00 X30.0 Y30.0 Z30.0;
G28 XYZ;
G00 X-30.0 Y30.0 Z30.0;
M30;
```

The accuracy of the movement before G28 is maintained.
The G28 zero point return accuracy will drop. (The accuracy is affected by the positioning speed and movement distance.)

[Simple zero point return invalid]   [Simple zero point return valid]
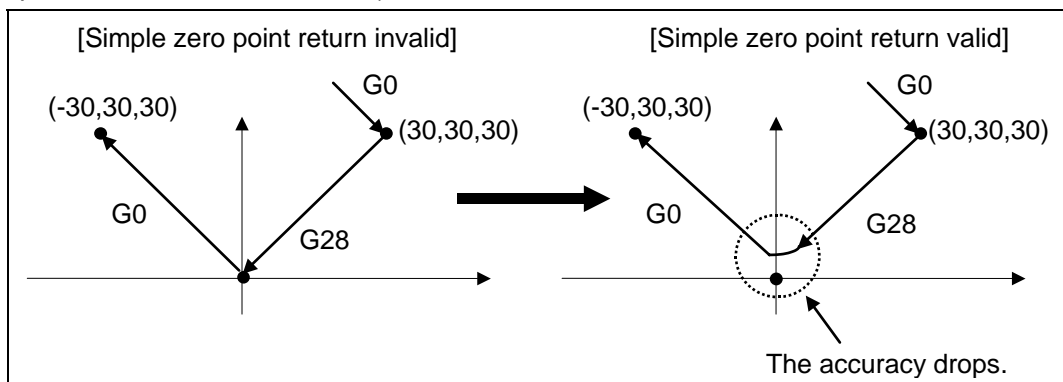
The accuracy drops.

(2) G90 with intermediate point

```
(Program example)
G90;
G00 X30.0 Y30.0 Z30.0;
G28 X30.0 Y15.0 Z10.0;
G00 X-30.0 Y30.0 Z30.0;
M30;
```

The accuracy of the movement before G28 is maintained.
The accuracy of the movement to the intermediate point is maintained.
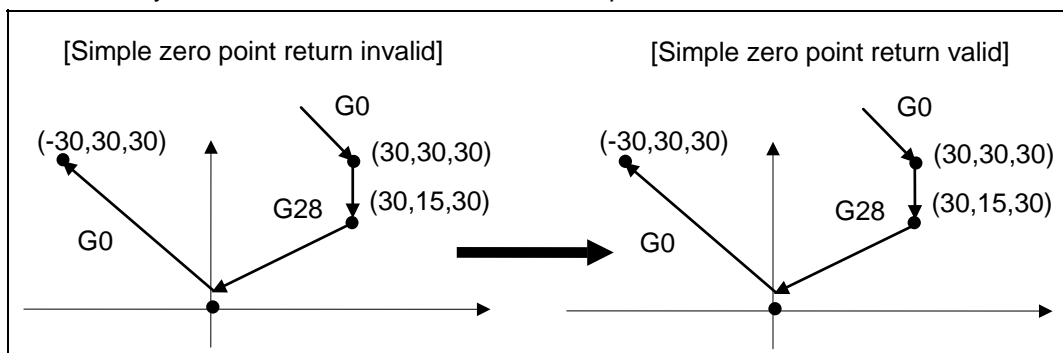
[Simple zero point return invalid]   [Simple zero point return valid]

(3) G91 without intermediate point

```
(Program example)
G91;
G00 X30.0 Y30.0 Z30.0;
G28 XYZ;
G00 X-30.0 Y30.0 Z30.0;
M30;
```

The accuracy of the movement before G28 will drop. (The accuracy is affected by the positioning speed and movement distance.)
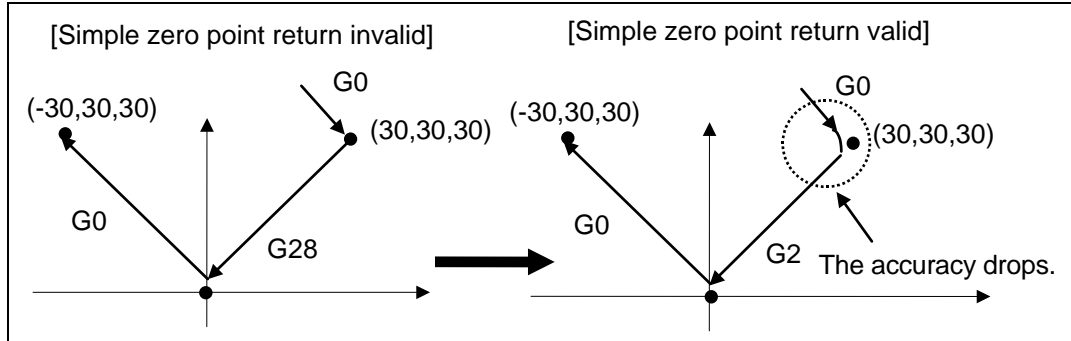
(4) G91 with intermediate point

```
(Program example)
G91;
G00 X30.0 Y30.0 Z30.0;
G28 X0.0 Y-15.0 Z0.0;
G00 X-30.0 Y30.0 Z30.0;
M30;
```

The accuracy of the movement before G28 is maintained.
The accuracy of the movement to the intermediate point will drop. (The accuracy is affected by the positioning speed and movement distance.)
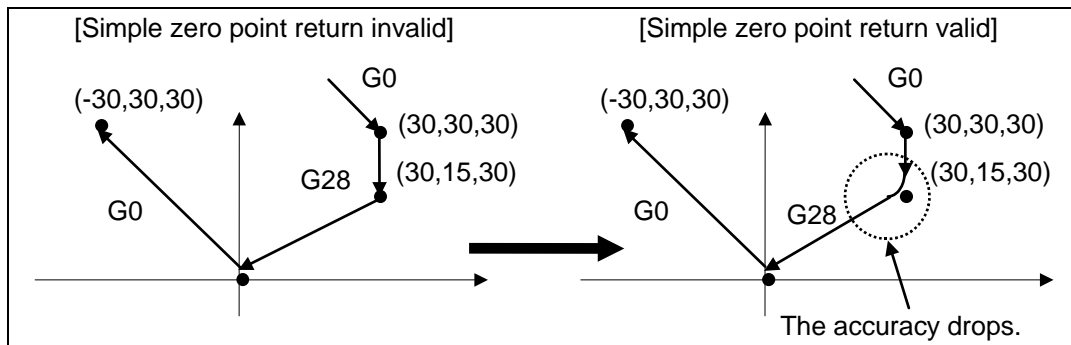
Precautions

(1) When the simple zero point return function is valid, the accuracy of the rapid feed command, cutting feed command, and movement command to the intermediate point before G28 may drop.
If the rapid feed command and cutting feed command before G28 must be accurate, use one of the following methods.
• Invalidate this function.
• Carry out an in-position check in the block before G28/G29/G30, and confirm the accuracy. (G09: Exact stop check, etc.)
• Input the required dwell command (time) in the block before G28/G29/G30.

14.10 Reference point check; G27

Function and purpose

This command first positions the tool at the position assigned by the program and then, if that positioning point is the first reference point, it outputs the reference point arrival signal to the machine in the same way as with the G28 command. Therefore, when a machining program is prepared so that the tool will depart from the first reference point and return to the first reference point, it is possible to check whether the tool has returned to the reference point after the program has been run.

Command format

G27 Xx$_1$ Yy$_1$ Zz$_1$ Pp$_1$;	
G27	: Check command
Xx$_1$ Yy$_1$ Zz$_1$	: Return control axis
Pp$_1$	: Check number
	P1 : 1st reference point check
	P2 : 2nd reference point check
	P3 : 3rd reference point check
	P4 : 4th reference point check

Detailed description

(1) If the P command has been omitted, the first reference point will be checked.
(2) The number of axes whose reference points can be checked simultaneously depends on the number of axes which can be controlled simultaneously.
   Note that the display shows one axis at a time from the final axis.
(3) An alarm will occur if the reference point is not reached after the command is completed.

14.11 Work coordinate system setting and offset ; G54 to G59 (G54.1)
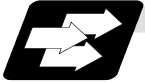
Function and purpose

(1) The work coordinate systems are for facilitating the programming of workpiece machining in which the reference point of the workpiece to be machined is to serve as the zero point.

(2) These commands enable the tool to move to the positions in the workpiece coordinate system. There are 6 workpiece coordinate systems which are used by the programmer for programming. (G54 ~ G59)
In addition to the six sets of workpiece coordinate systems between G54 and G59, there are 48 additional workpiece coordinate system sets. (The 48 sets are options.)

(3) Among the workpiece coordinate systems currently selected by these commands, any workpiece coordinate system with coordinates which have been commanded by the present position of the tool is reset. (The "present position of the tool" includes the offset amounts for tool radius, tool length and tool position offset.)

(4) An hypothetical machine coordinate system with coordinates which have been commanded by the present position of the tool is set by this command. (The "present position of the tool" includes the offset amounts for nose radius, tool length and tool position offset.) (G54 ~ G92)

Command format

(1) Work coordinate system selection (G54 to G59)

(G90) G54 Xx_1 Yy_1 Zz_1 $\alpha\alpha_1$;
$\alpha\alpha_1$:Additional axis

(2) Work coordinate system setting (G54 to G59)

(G54) G92 Xx_1 Yy_1 Zz_1 $\alpha\alpha_1$;
$\alpha\alpha_1$:Additional axis

(3) Extended work coordinate system selection (P1 to P48)

G54.1 Pn ;

(4) Extended work coordinate system setting (P1 to P48)

G54.1 Pn ;
G92 Xx Yy Zz $\alpha\alpha$; (where, α is the additional axis)

(5) Extended work coordinate system offset amount setting (P1 to P48)

G10 L20   Pn Xx Yy Zz $\alpha\alpha$; (where, α is the additional axis)

![hand icon] **Detailed description**

(1) With any of the G54 through G59 commands, the tool diameter offset amounts for the commanded axes will not be canceled even if work coordinate system selection is commanded.
(2) The G54 work coordinate system is selected when the power is switched on.
(3) Commands G54 through G59 are modal commands (group 12).
(4) The coordinate system will move with G92 in a work coordinate system.
(5) The offset setting in a work coordinate system denotes the distance from the zero point of the basic machine coordinate system.
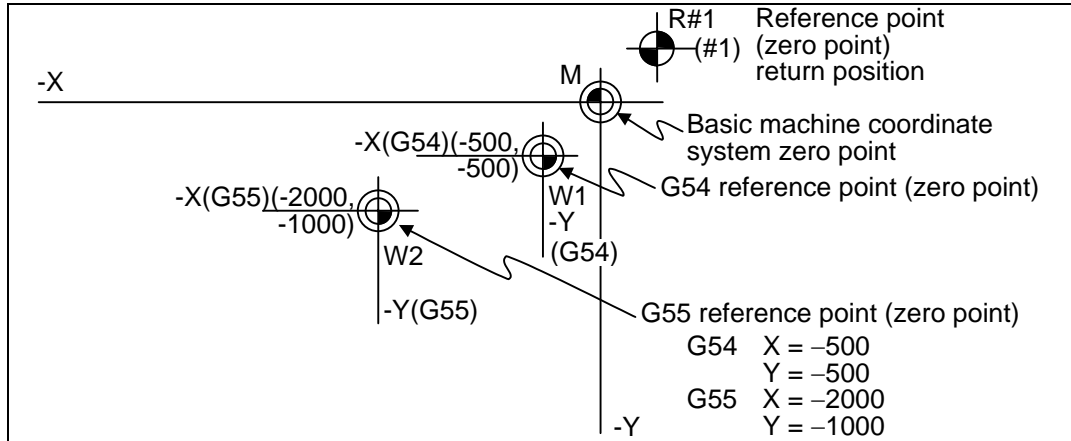
(6) The offset settings of work coordinate systems can be changed any number of times. (They can also be changed by G10 L2 Pp1 Xx1 Zz1.)
Handling when L or P is omitted

G10 L2 Pn Xx Yy Zz ;	;n=0 : Set the offset amount in the external workpiece coordinate system. n=1 to 6 : Set the offset amount in the designated workpiece coordinate system. Others : The program error (P35) will occur.
G10 L2 Xx Yy Zz ;	Set the offset amount in the currently selected workpiece coordinate system. When in G54.1 modal, the program error (P33) will occur.
G10 L20 Pn Xx Zz ;	n=1 to 48: Set the offset amount in the designated workpiece coordinate system. Others: The program error (P35) will occur.
G10 L20 Xx Zz ;	Set the offset amount in the currently selected workpiece coordinate system. When in G54 to G59 modal, the program error (P33) will occur.
G10 Pn Xx Yy Zz ; G10 Xx Yy Zz ;	L2 (workpiece offset) will be judged if there is no L value.

(7) A new work coordinate system 1 is set by issuing the G92 command in the G54 (work coordinate system 1) mode. At the same time, the other work coordinate systems 2 through 6 (G55 to G59) will move in parallel and new work coordinate systems 2 through 6 will be set.

(8) An imaginary machine coordinate system is formed at the position which deviates from the new workpiece reference (zero) point by an amount equivalent to the work coordinate system offset amount.

After the power has been switched on, the imaginary machine coordinate system is matched with the basic machine coordinate system by the first automatic (G28) or manual reference (zero) point return.
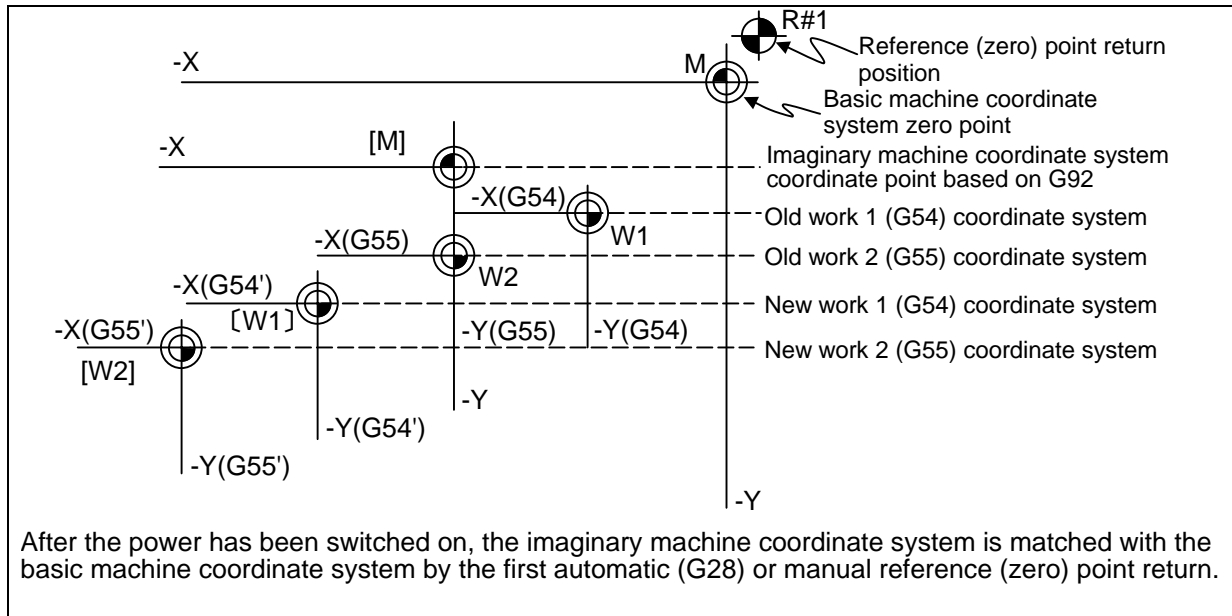
(9) By setting the imaginary basic machine coordinate system, the new work coordinate system will be set at a position which deviates from that imaginary basic machine coordinate system by an amount equivalent to the work coordinate system offset amount.

(10) When the first automatic (G28) or manual reference (zero) point return is completed after the power has been switched on, the basic machine coordinate system and work coordinate systems are set automatically in accordance with the parameter setting.

(11) If G54X-Y-; is commanded after the reference return (both automatic or manual) executed after the power is turned ON, the program error (P62) will occur. (A speed command is required as the movement will be controlled with the G01 speed.)

(12) Do not command a G code for which a P code is used in the same block as G54.1. The P code will be used in the prioritized G command.

(13) When number of workpiece offset sets additional specifications is not added, the program error (P172) will occur when the G10 L20 command is executed.

(14) The local coordinate system cannot be used during G54.1 modal. The program error (P438) will occur when the G52 command is executed during G54.1 modal.

(15) A new workpiece coordinate system P1 can be set by commanding G92 in the G54.1 P1 mode.
   However, the workpiece coordinate system of the other workpiece coordinate systems G54 to G59, G54.1 and P2 to P48 will move in parallel with it, and a new workpiece coordinate system will be set.

(16) The offset amount of the extended workpiece coordinate system is assigned to the variable number as shown in Table 1.

Table 1 Variable numbers of the extended workpiece coordinate offset system

	1st axis to nth axis		1st axis to nth axis
P 1	#7001 to #700n	P25	#7481 to #748n
P 2	#7021 to #702n	P26	#7501 to #750n
P 3	#7041 to #704n	P27	#7521 to #752n
P 4	#7061 to #706n	P28	#7541 to #754n
P 5	#7081 to #708n	P29	#7561 to #756n
P 6	#7101 to #710n	P30	#7581 to #758n
P 7	#7121 to #712n	P31	#7601 to #760n
P 8	#7141 to #714n	P32	#7621 to #762n
P 9	#7161 to #716n	P33	#7641 to #764n
P10	#7181 to #718n	P34	#7661 to #766n
P11	#7201 to #720n	P35	#7681 to #768n
P12	#7221 to #722n	P36	#7701 to #770n
P13	#7241 to #724n	P37	#7721 to #772n
P14	#7261 to #726n	P38	#7741 to #774n
P15	#7281 to #728n	P39	#7761 to #776n
P16	#7301 to #730n	P40	#7781 to #778n
P17	#7321 to #732n	P41	#7801 to #780n
P18	#7341 to #734n	P42	#7821 to #782n
P19	#7361 to #736n	P43	#7841 to #784n
P20	#7381 to #738n	P44	#7861 to #786n
P21	#7401 to #740n	P45	#7881 to #788n
P22	#7421 to #742n	P46	#7901 to #790n
P23	#7441 to #744n	P47	#7921 to #792n
P24	#7461 to #746n	P48	#7941 to #794n

⚠ CAUTION

⚠ If the workpiece coordinate system offset amount is changed during single block stop, the new setting will be valid from the next block.

Example of program

(Example 1)

```
(1) G28   X0Y0 ;
(2) G53 X−1000   Y−500 ;
(3) G53   X0Y0 ;
```
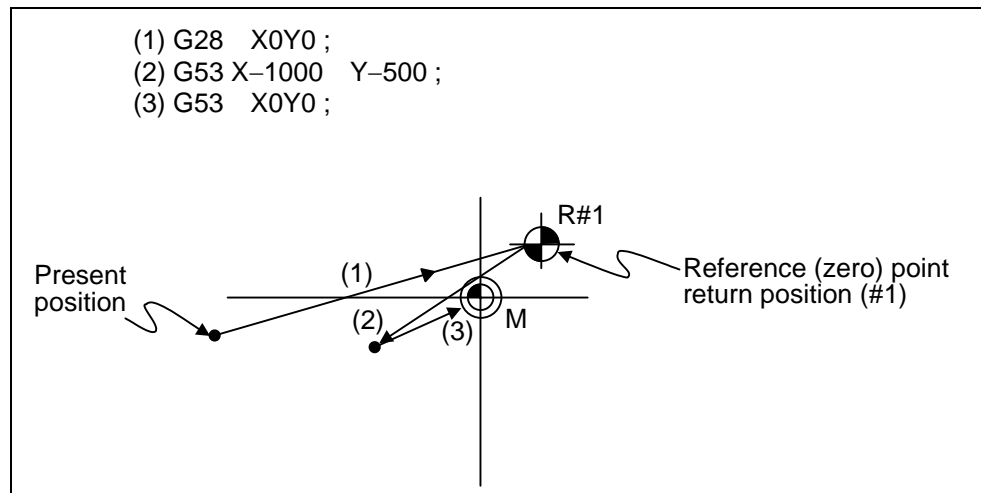

When the first reference point coordinate is zero, the basic machine coordinate system zero point and reference (zero) point return position (#1) will coincide.

(Example 2)
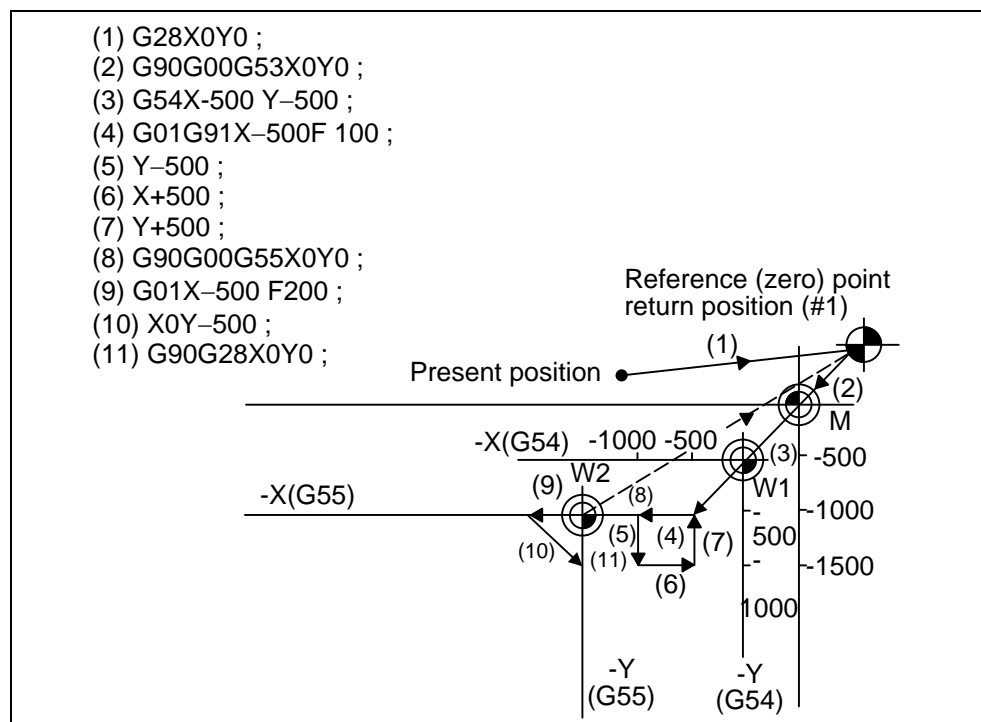
```
(1) G28X0Y0 ;
(2) G90G00G53X0Y0 ;
(3) G54X-500 Y−500 ;
(4) G01G91X−500F 100 ;
(5) Y−500 ;
(6) X+500 ;
(7) Y+500 ;
(8) G90G00G55X0Y0 ;
(9) G01X−500 F200 ;
(10) X0Y−500 ;
(11) G90G28X0Y0 ;
```

(Example 3) When work coordinate system G54 has shifted (−500, −500) in example 2 (It is assumed that 3 through 10 in example 2 have been entered in subprogram 01111.)

(1) G28 X0 Y0 ;	
(2) G90 G53 X0 Y0 ;	(This is not required when there is no G53 offset.)
(3) G54 X −500Y−500 ;	Amount by which work coordinate system deviates
(4) G92 X0 Y0 ;	New work coordinate system is set.
(5) M98 P1111 ;	
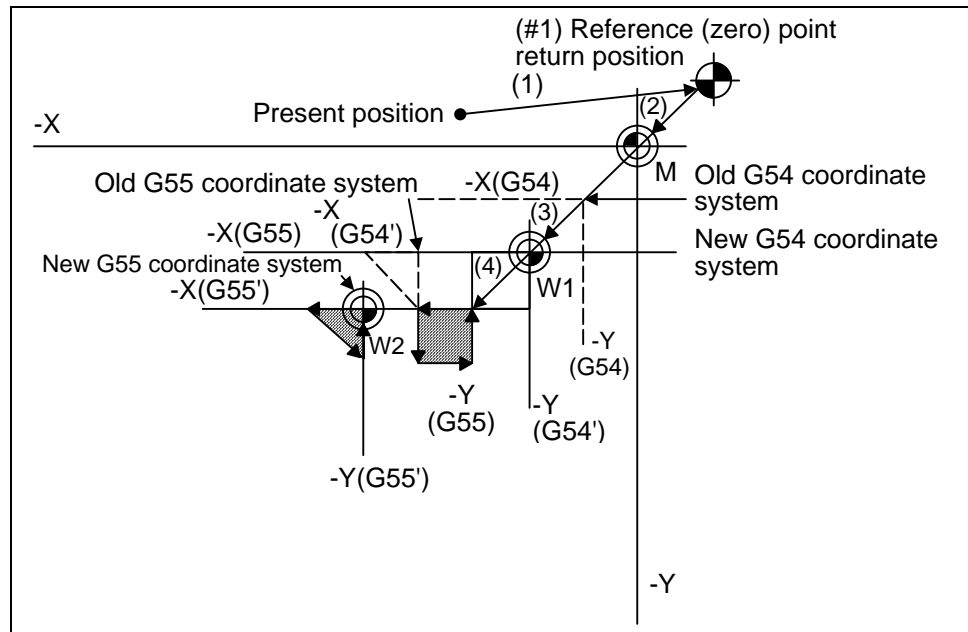
(Note)   The work coordinate system will shift each time steps 3 through 5 are repeated. The reference point return (G28) command should therefore be issued upon completion of the program.

(Example 4) When six workpieces are placed on the same coordinate system of G54 to G59, and each is to be machined with the same machining.

(1) Setting of workpiece offset data

```
Workpiece1  X = −100.000  Y = −100.000 ....................................... G54
        2  X = −100.000  Y = −500.000 ....................................... G55
        3  X = −500.000  Y = −100.000 ....................................... G56
        4  X = −500.000  Y = −500.000 ....................................... G57
        5  X = −900.000  Y = −100.000 ....................................... G58
        6  X = −900.000  Y = −500.000 ....................................... G59
```

(2) Machining program (subprogram)

```
O100;
N1 G90 G0 G43X-50. Y-50. Z-100. H10;       Positioning
N2 G02 X-200. F50;
        Y-200.  ;
        X- 50.  ;                          Edge cutting
        Y- 50.  ;
N3 G28 X0 Y0 Z0 ;

N4 G98 G81 X-125. Y-75. Z-150. R-100. F40;              1
        X-175. Y-125. ;                     Drilling     2
        X-125. Y-175. ;                                  3
        X- 75. Y-125. ;                                  4
    G80;
N5 G28 X0 Y0 Z0 ;

N6 G98 G84 X-125. Y-75. Z-150. R-100. F40 ;             1
        X-175. Y-125. ;                     Tapping      2
        X-125. Y-175. ;                                  3
        X- 75. Y-125. ;                                  4
    G80;
    M99;
```

(3) Positioning program (main)

```
G28 X0 Y0 Z0 ;
                                       When power is turned ON
N1  G90 G54 M98 P100 ;
N2      G55 M98 P100 ;
N3      G57 M98 P100 ;
N4      G56 M98 P100 ;
N5      G58 M98 P100 ;
N6      G59 M98 P100 ;
N7  G28 X0 Y0 Z0 ;
N8  M02 ;
%
```
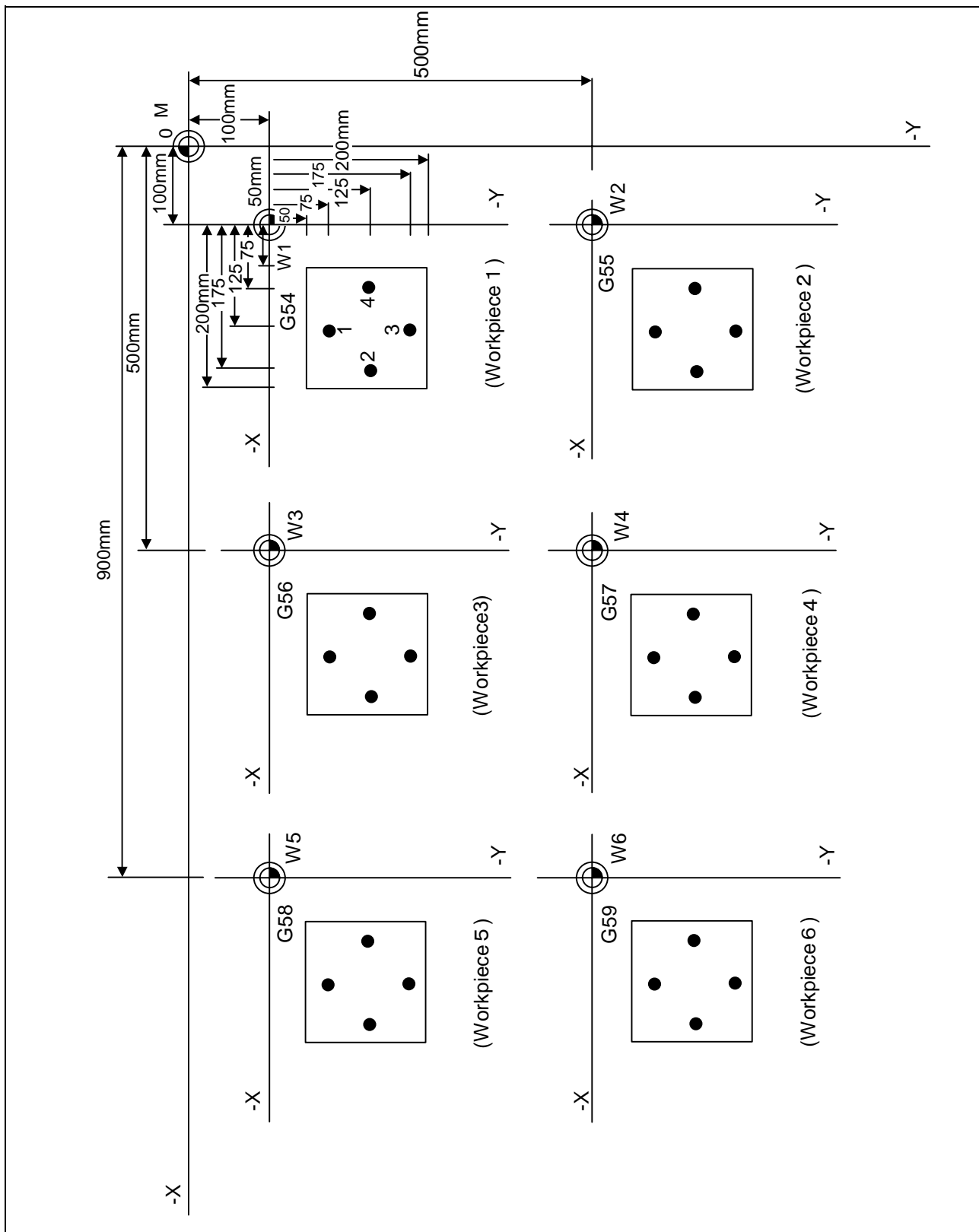
(Example 5) Program example when continuously using 48 sets of added workpiece coordinate system offsets.

In this example, the offsets for each workpiece are set beforehand in P1 to P48 when 48 workpieces are fixed on a table, as shown in the drawing below.
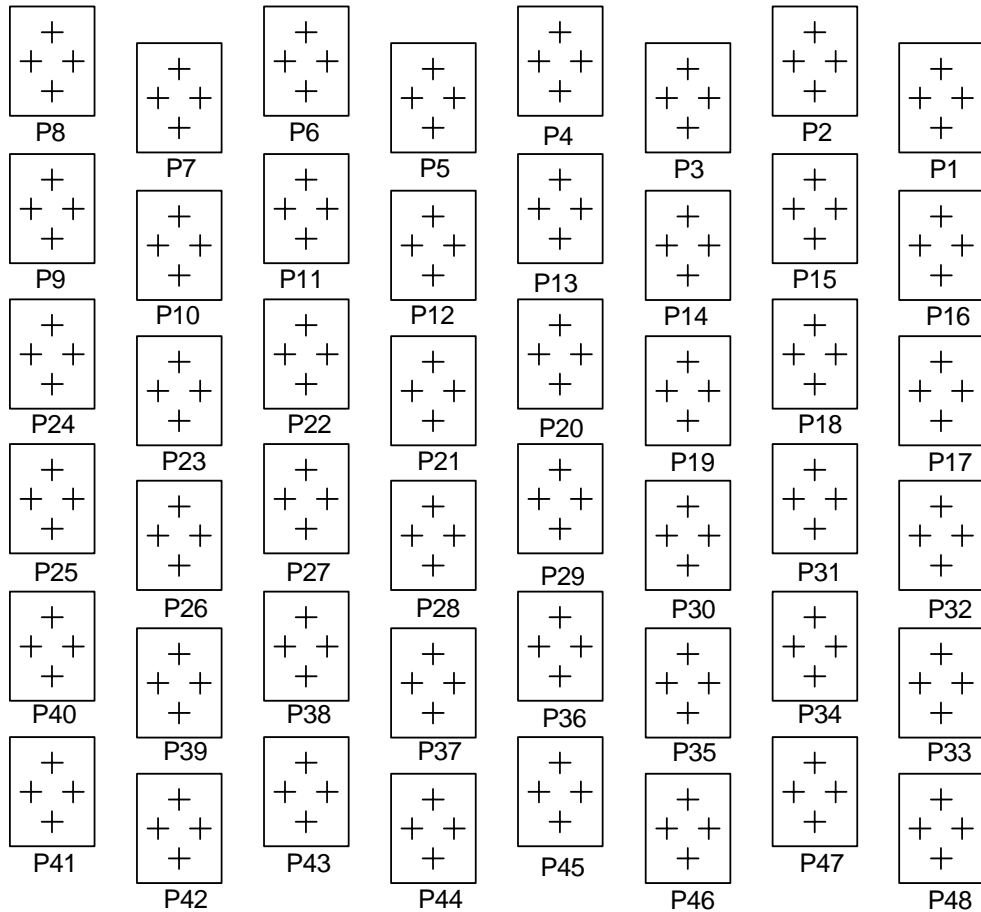
01000	Reference point return	01001
G28 XYZ ;	Initialize added workpiece coordinate system P No.	G43 X-10.Y-10.Z-100.H10.; Contour
#100=1 ;	Absolute value mode	G01 X-30.;
G90 ;	Repeat P No. to 48	Y-30.;
WHILE [#100LE48]D01 ;	Set workpiece coordinate system	X-10.;
G54.1 P#100 ;	Read sub-program	Y-10.;
M98 P1001 ;	P No. +1	G00 G40 Z10.;
#100=#100+1 ;		G98 G81·X-20.Y-15.Z-150.R5.F40; Drilling
END1 ;	Return to reference point	X-25.Y-20.;
G28 Z ;		X-20.Y-25.;
G28 XY ;		X-15.Y-20.;
M02 ;		G80 ;
		M99 ;

(Example 6) Program example when the added workpiece coordinate system offsets are transferred to the standard workpiece coordinate system offsets and used.

In this example, the workpiece coordinate system offsets for each workpiece are set beforehand in P1 to P24 when the workpiece is fixed on a rotating table, as shown in the drawing below.
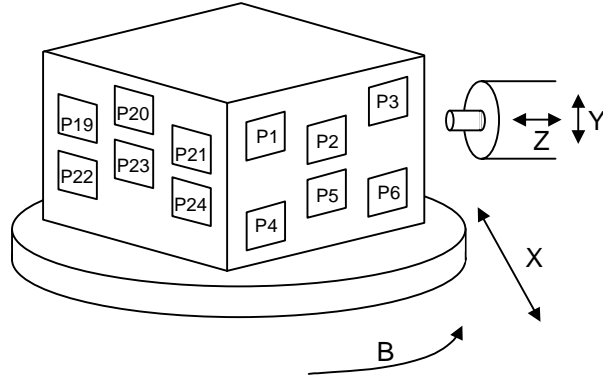

```
O20000   (Main)
G28 XYZB ;                       Zero point return
G90 ;                            Absolute value mode
G00 B0 ;                         Position table to face 1
G65 P2001 A1 ;                   Load workpiece offsets
M98 P2002 ;                      Drilling
G00 B90. ;                       Position table to face 2
G65 P2001 A7 ;
M98 P2002 ;
G00 B180. ;                      Position table to face 3
G65 P2001 A13 ;
M98 P2002 ;
G00 B270. ;                      Position table to face 4
G65 P2001 A19 ;
M98 P2002 ;
G28 XYB ;                        Return to zero point
M02 ;
%
O2001   Transmission of workpiece offsets
#2=5221 ;                        Leading No. of workpiece coordinate system
variables
#3=(#1-1)*20+7001 ;              Leading No. of added workpiece coordinate
                                 system variables
#5=0 ;                           No. of sets counter clear
WHILE [#5 LT 6] DO1 ;            Check No. of sets
#6=#6+1 ;                        Set transmission source 1st axis variable No.
#7=#7+1 ;                        Set transmission destination 1st axis variable No.
#4=#4+1 ;                        Clear No. of axes counter
WHILE [#4 LT 6] DO2 ;            Check No. of axes
#[#6]=#[#7] ;                    Transmit variable data
#6=#6+1 ;                        Set transmission source next axis
#7=#7+1 ;                        Set transmission destination next axis
#4=#4+1 ;                        No. of axes counter +1
END2 ;
#2=#2+20 ;                       Transmission source  Set lead of next No. of sets
#3=#3+20 ;                       Transmission destination  Set lead of next No. of
sets
#5=#5+1 ;                        No. of sets counter +1
END1 ;
M99 ;
%
```
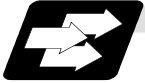
14. Coordinates System Setting Functions

```
O2002   (Drilling)
G54 M98 H100 ;              Drilling in G54 coordinate system
G55 M98 H100 ;              In G55
G56 M98 H100 ;              In G56
G57 M98 H100 ;              In G57
G58 M98 H100 ;              In G58
G59 M98 H100 ;              In G59
G28 Z0 ;
M99 ;
N100 G98 G81 X-20. Y-15. Z-150. R5. F40 ; Fixed cycle for Drilling call
X-25. Y-20. ;
X-20. Y-25. ;
X-15. Y-20. ;
G80 ;
G28 Z ;
M99 ;
%
```

14.12 Local coordinate system setting

Function and purpose

The local coordinate systems can be set independently on the G54 through G59 work coordinate systems using the G52 command so that the commanded position serves as the programmed zero point.

The G52 command can also be used instead of the G92 command to change the deviation between the zero point in the machining program and the machining workpiece zero point.

Command format

G54 (54 to G59) G52Xx_1 Yy_1 Zz_1 $\alpha\alpha_1$; (where, α is the additional axis)

Detailed description

(1) The G52 command is valid until a new G52 command is issued, and the tool does not move. This command comes in handy for employing another coordinate system without changing the zero point positions of the work coordinate systems (G54 to G59).

(2) The local coordinate system offset will be cleared by the dog-type manual reference (zero) point return or reference (zero) point return performed after the power has been switched on.

(3) The local coordinate system is canceled by (G54 to G59) G52X0 Y0 Z0 α0 ;.

(4) Coordinate commands in the absolute value mode (G90) cause the tool to move to the local coordinate system position.
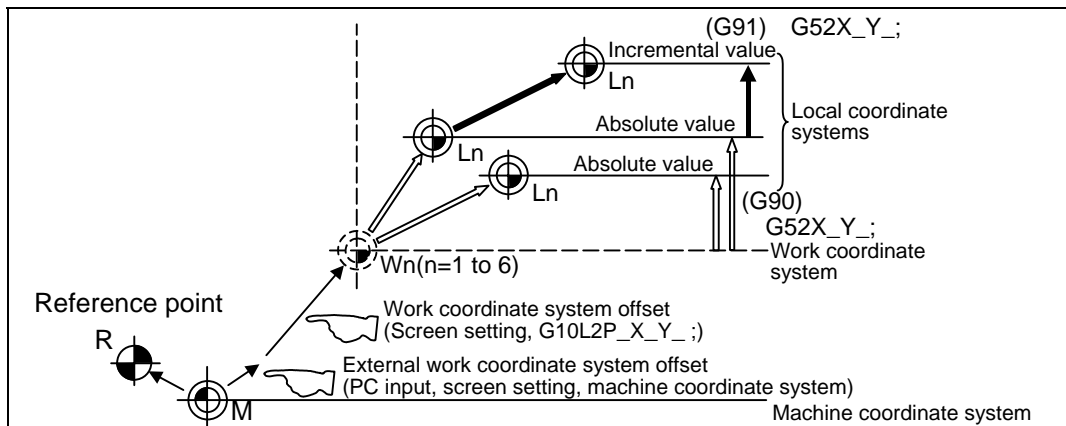
(Note) If the program is executed repeatedly, the workpiece coordinate system will deviate each time. Thus, when the program is completed, the reference point return operation must be commanded.

(Example 1) Local coordinates for absolute value mode (The local coordinate system offset is not cumulated)
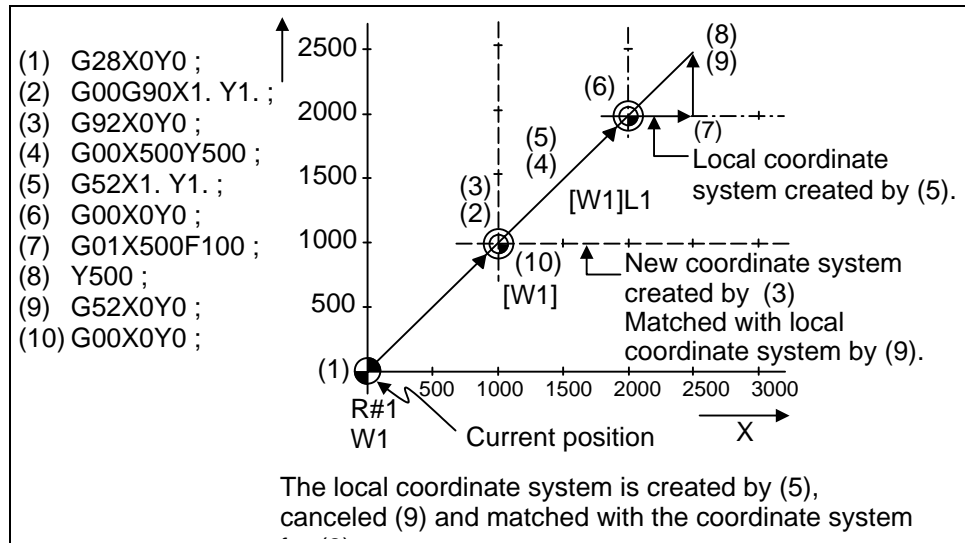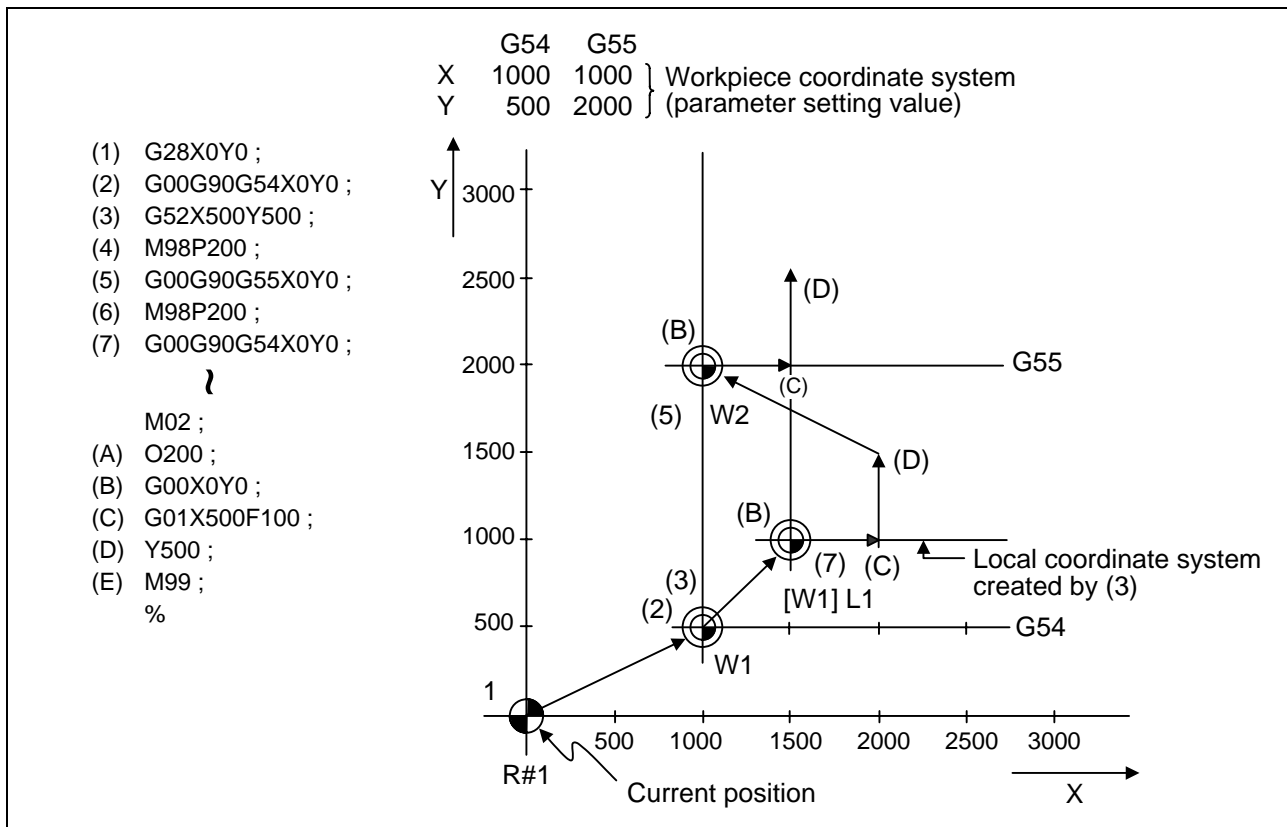
```
(1)  G28X0Y0 ;
(2)  G00G90X1. Y1. ;
(3)  G92X0Y0 ;
(4)  G00X500Y500 ;
(5)  G52X1. Y1. ;
(6)  G00X0Y0 ;
(7)  G01X500F100 ;
(8)  Y500 ;
(9)  G52X0Y0 ;
(10) G00X0Y0 ;
```


The local coordinate system is created by (5), canceled (9) and matched with the coordinate system

(Note)   If the program is executed repeatedly, the workpiece coordinate system will deviate each time. Thus, when the program is completed, the reference point return operation must be commanded.

(Example 2) Local coordinates for incremental value mode (The local coordinate system offset is cumulated.)
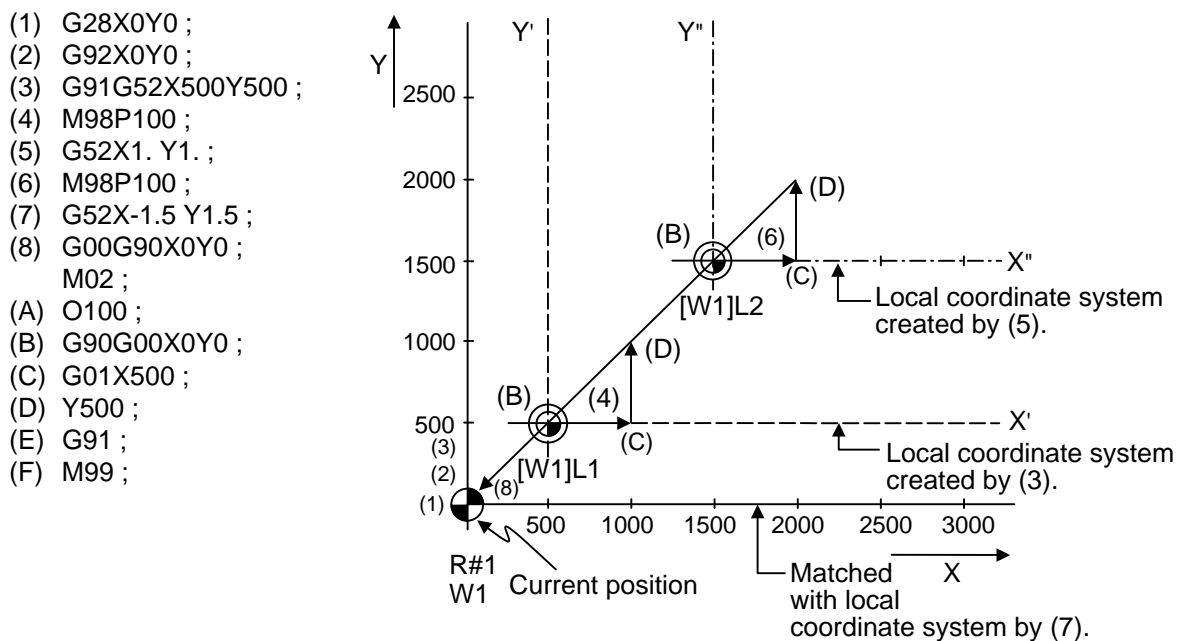
```
(1)  G28X0Y0 ;
(2)  G92X0Y0 ;
(3)  G91G52X500Y500 ;
(4)  M98P100 ;
(5)  G52X1. Y1. ;
(6)  M98P100 ;
(7)  G52X-1.5 Y1.5 ;
(8)  G00G90X0Y0 ;
     M02 ;
(A)  O100 ;
(B)  G90G00X0Y0 ;
(C)  G01X500 ;
(D)  Y500 ;
(E)  G91 ;
(F)  M99 ;
```


(Explanation)

The local coordinate system X'Y' is created at the XY coordinate system (500,500) position by (3).
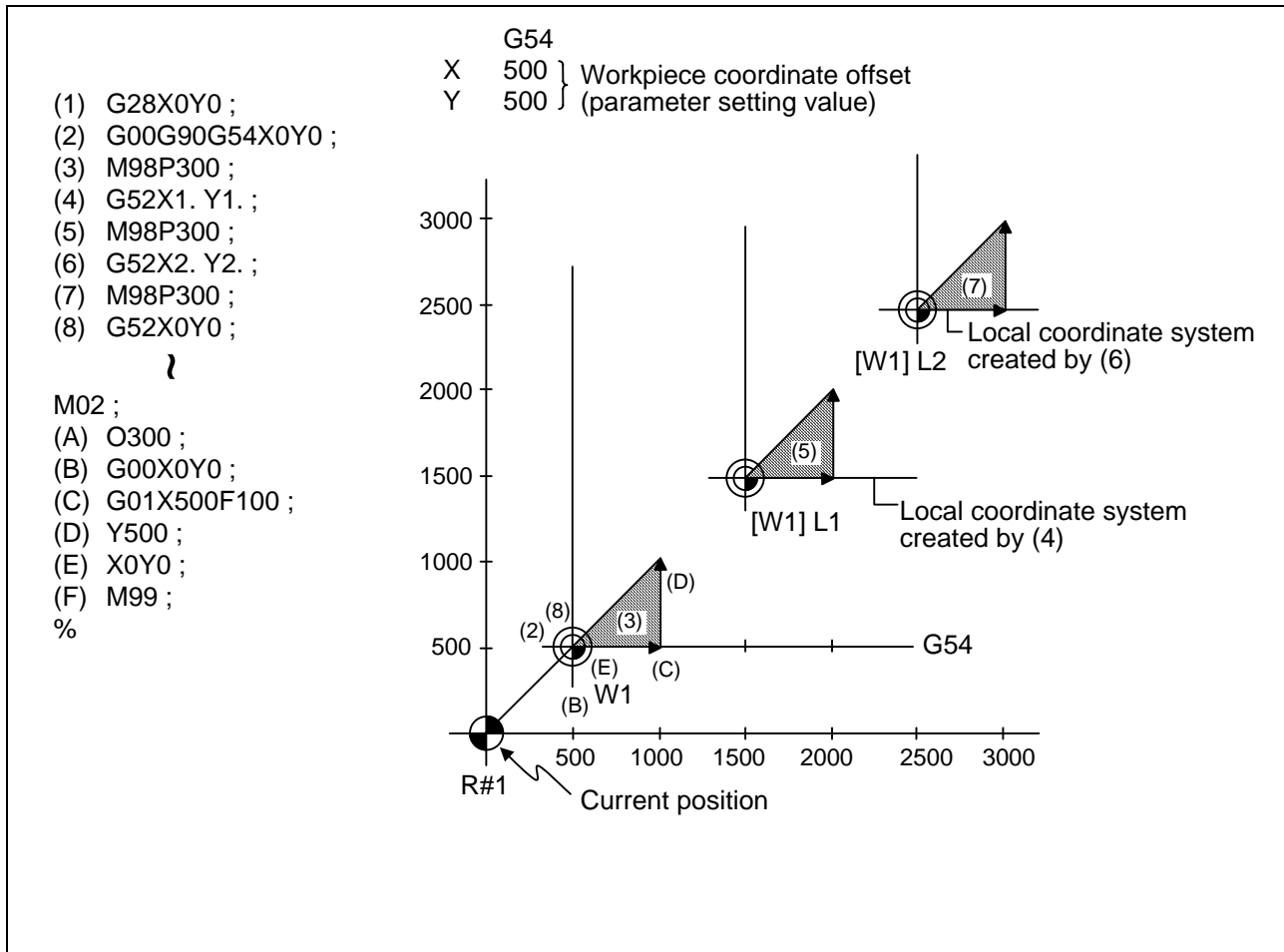The local coordinate system X"Y" is created at the X'Y' coordinate system (1000,1000) position by (5).
The local coordinate system is created at the X"Y" coordinate system (-1500, -1500) position by (7).
In other words, the same occurs as when the local coordinate system and XY coordinate system are matched and the local coordinate system is canceled.

(Example 3) When used together with workpiece coordinate system

```
                              G54    G55
                       X     1000   1000 ⎤ Workpiece coordinate system
                       Y      500   2000 ⎦ (parameter setting value)

(1)  G28X0Y0 ;
(2)  G00G90G54X0Y0 ;
(3)  G52X500Y500 ;
(4)  M98P200 ;
(5)  G00G90G55X0Y0 ;
(6)  M98P200 ;
(7)  G00G90G54X0Y0 ;
          ⎰
     M02 ;
(A)  O200 ;
(B)  G00X0Y0 ;
(C)  G01X500F100 ;
(D)  Y500 ;
(E)  M99 ;
     %
```

(Diagram showing coordinate axes with Y up to 3000, X across to 3000, points W1, W2, G54, G55 coordinate systems, local coordinate system created by (3), current position at R#1)

(Explanation)

The local coordinate system is created at the G54 coordinate system (500,500) position by (3), but the local coordinate system is not created for the G55 coordinate system.

During the movement for (7), the axis moves to the G54 local coordinate system's reference point (zero point).

The local coordinate system is canceled by G90G54G52X0Y0;.

(Example 4) Combination of workpiece coordinate system G54 and multiple local coordinate systems

```
(1)  G28X0Y0 ;
(2)  G00G90G54X0Y0 ;
(3)  M98P300 ;
(4)  G52X1. Y1. ;
(5)  M98P300 ;
(6)  G52X2. Y2. ;
(7)  M98P300 ;
(8)  G52X0Y0 ;
      {
M02 ;
(A)  O300 ;
(B)  G00X0Y0 ;
(C)  G01X500F100 ;
(D)  Y500 ;
(E)  X0Y0 ;
(F)  M99 ;
%
```


G54
X   500 } Workpiece coordinate offset
Y   500 } (parameter setting value)

Local coordinate system created by (6)

Local coordinate system created by (4)

[W1] L2

[W1] L1

G54

W1

R#1   Current position

(Explanation)

The local coordinate system is created at the G54 coordinate system (1000,1000) position by (4).
The local coordinate system is created at the G54 coordinate system (2000,2000) by (6).
The G54 coordinate system and local coordinate system are matched by (8).

14.13 Coordinate system for rotary axis

Function and purpose

The axis designated as the rotary axis with the parameters is controlled with the rotary axis' coordinate system.

The rotary axis includes the rotating type (short-cut valid/invalid) and linear type.

The range of each coordinate system is 0 to 359.999° for the rotating type, and 0 to ±99999.999° for the linear type.

The rotary axis is commanded with a degree (°) unit regardless of the inch or metric designation.

The rotary axis type is common for all rotary axes and can be set with the parameters "#1089 Cut_RT" and "#1090 Lin_RT".

	Rotary axis			
	Rotating type rotary axis		Linear type rotary axis	Linear axis
	Short-cut invalid	Short-cut valid		
#1089 setting value	0	1	--	--
#1090 setting value	0		1	--
Workpiece coordinate value	Displayed in the range of 0° to 359.999°.		Displayed in the range of 0° to ±99999.999°.	
Machine coordinate value/relative position	Displayed in the range of 0° to 359.999°.			Displayed in the range of 0° to ±99999.999°.
ABS command	The incremental amount from the end point to the current position is divided by 360, and the axis moves by the remainder amount according to the sign.	The incremental amount from the end point to the current position is divided by 360°, and the axis takes a shortcut to moves by the remainder amount.	In the same manner as the normal linear axis, moves according to the sign by the amount obtained by subtracting the current position from the end point (without rounding up to 360 degrees).	
INC command	Moves in the direction of the commanded sign by the commanded incremental amount starting at the current position.			
Reference position return	Until the intermediate point: Depends on the absolute command or the incremental command.			
	From the intermediate point to the reference position: Returns with movement within 360 degrees.			Moves and returns in the reference position direction by the difference from the intermediate point to the reference position.

![icon] **Example of operation**

Examples of differences in the operation and counter displays according to the type of rotation coordinate are given below.
(The workpiece offset is set as 0°.)

(1)   Rotary type (#1090 = 0) + short-cut invalid (#1089 = 0)
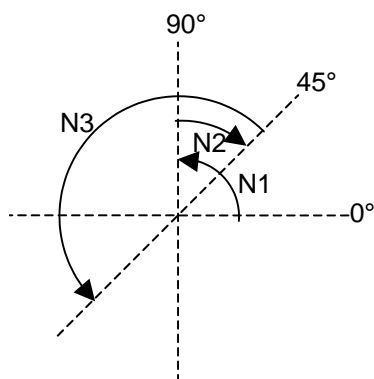
(a) The machine coordinate position, workpiece coordinate position and current position are displayed in the range of 0 to 359.999°.
(b) For the absolute command, the axis moves according to the sign.
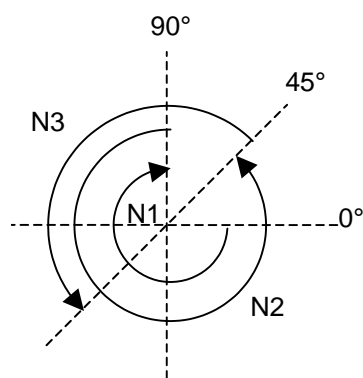
Program	Workpiece coordinate counter	Machine coordinate counter
G28 C0.		
N1 G90 C-270.	90.000	90.000
N2 C405.	45.000	45.000
N3 G91 C180	225.000	225.000

(2)   Rotation type (When #1090 = 0) + short-cut valid (#1089 = 1)

(a) The machine coordinate position, workpiece coordinate position and current position are displayed in the range of 0 to 359.999°.
(b) For the absolute command, the axis rotates to the direction having less amount of movement to the end point.

Program	Workpiece coordinate counter	Machine coordinate counter
G28 C0.		
N1 G90 C-270.	90.000	90.000
N2 C405.	45.000	45.000
N3 G91 C180	225.000	225.000

(3) Linear type (When #1090 = 1)

    (a) The machine coordinate position, workpiece coordinate position and current position are displayed in the range of 0 to 99999.999°.

    (b) For absolute value commands that exceed 360°, the axis will move in the same manner as a linear axis.

       (Example) If the command is 420°, the axis will pass the 360° position and will move to the 60° position.

    (c) During reference position return, the axis moves in the same manner as the linear axis until the intermediate point.

      The axis returns with a rotation within 360° from the intermediate point to the reference position.

    (d) During absolute position detection, the system starts up at the position where the power was turned OFF when the power is turned ON again.
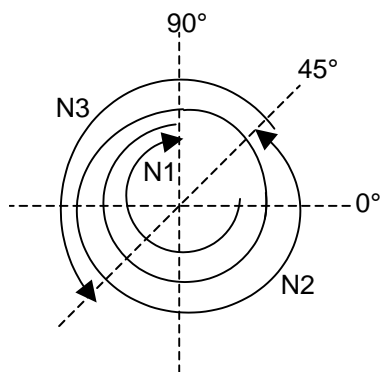
Program	Workpiece coordinate counter	Machine coordinate counter	POSITION counter
G28 C0.			
N1 G90 C-270.	-270.000	-270.000	-270.000
N2 C405.	405.000	405.000	405.000
N3 G91 C180	585.000	585.000	585.000

When power is turned ON again ↓

Workpiece	Machine
585.000	585.000

15. Measurement Support Functions

15.1 Automatic tool length measurement; G37
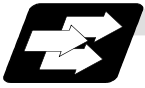
Function and purpose

These functions issue the command values from the measuring start position as far as the measurement position, move the tool in the direction of the measurement position, stop the machine once the tool has arrived at the sensor, cause the NC system to calculate automatically the difference between the coordinate values at that time and the coordinate values of the commanded measurement position and provide this difference as the tool offset amount.

When offset is already being applied to a tool, it moves the tool toward the measurement position with the offset still applied, and if a further offset amount is generated as a result of the measurement and calculation, it provides further compensation of the present offset amount.

If there is one type of offset amount at this time, and the offset amount is distinguished between tool length offset amount and wear offset amount, the wear amount will be automatically compensated.

Command format

G37Z__R__D__F__ ;

Z : Measuring axis address and coordinates of measurement position ..... X, Y, z, α (where, α is the additional axis)
R : This commands the distance between the measurement position and point where the movement is to start at the measuring speed.
D : This commands the range within which the tool is to stop.
F : This commands the measuring feedrate.

When R_, D_ of F_ is omitted, the value set in the parameter is used instead.

<Parameter>   ("TLM" on machining parameter screen)

- #8004 SPEED (measuring feedrate)   : 0 to 60000 (mm/min)
- #8005 ZONE r (deceleration range)   : 0 to 99999.999 (mm)
- #8006 ZONE d (measurement range)   : 0 to 99999.999 (mm)

Example of execution

For new measurement
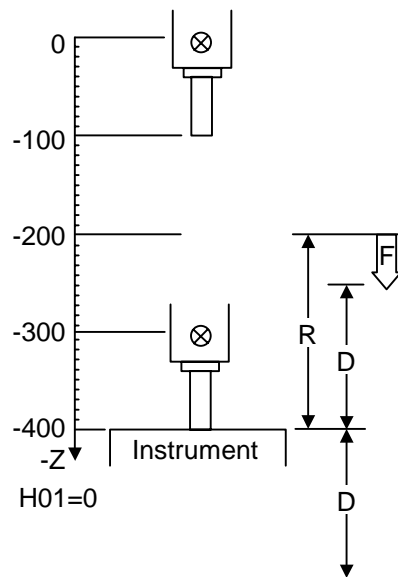

```
T01 ;
M06   T02 ;
G90   G00   G43   Z0   H01 ;
G37   Z-400   R200   D150   F1 ;
Coordinate value when measurement position is reached = -300
-300 - (-400) = 100
0+100 = 100    Where, H01 = 100
```
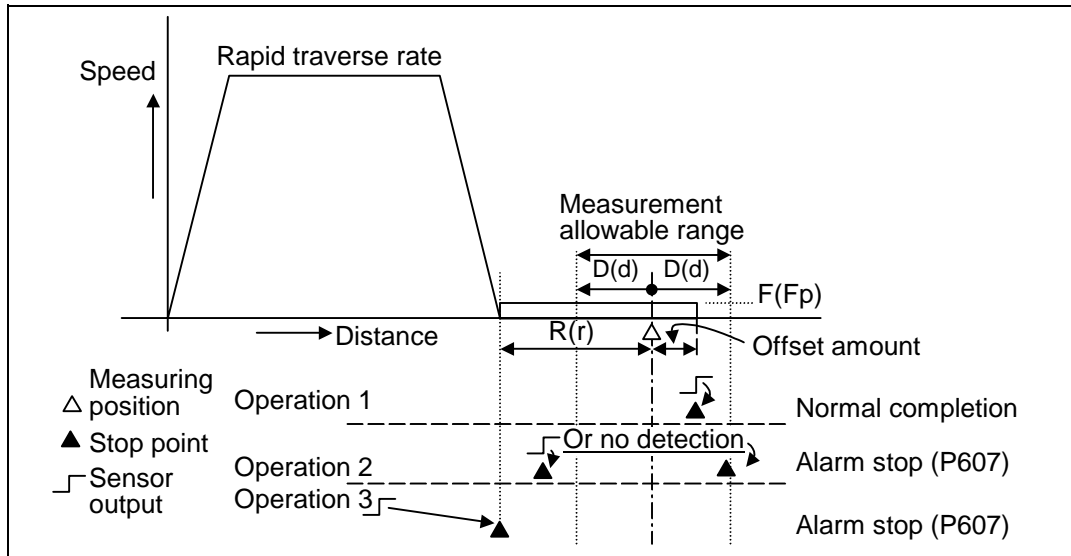
Detailed description

(1) Operation with G37 command

(2) The sensor signal (measuring position arrival signal) is used in common with the skip signal.

(3) The feedrate will be 1mm/min if the F command and parameter measurement speed are 0.

(4) An updated offset amount is valid unless it is assigned from the following Z axis (measurement axis) command of the G37 command.

(5) Excluding the corresponding values at the PLC side, the delay and fluctuations in the sensor signal processing range from 0 to 0.2ms.
As a result, the measuring error shown below is caused.

$$\text{Maximum measuring error (mm)} = \text{Measuring speed (mm/min)} \bullet \frac{1}{60} \bullet \frac{0.2 \text{ (ms)}}{1000}$$

(6) The machine position coordinates at that point in time are ready by sensor signal detection, and the machine will overtravel and stop at a position equivalent to the servo droop.
Maximum overtravel (mm)

$$= \text{Measuring speed (mm/min)} \bullet \frac{1}{60} \bullet \frac{1}{\text{Position loop gain (s}^{-1})}$$

The standard position loop gain is 33 (s^{-1}).

396

Precautions

(1) Program error (P600) results if G37 is commanded when the automatic tool length measurement function is not provided.
(2) Program error (P604) results when no axis has been commanded in the G37 block or when two or more axes have been commanded.
(3) Program error (P605) results when the H code is commanded in the G37 block.
(4) Program error (P606) results when G43_H is not commanded prior to the G37 block.
(5) Program error (P607) results when the sensor signal was input outside the allowable measuring range or when the sensor signal was not detected even upon arrival at the end point.
(6) When a manual interrupt is applied while the tool is moving at the measuring speed, a return must be made to the position prior to the interrupt and then operation must be resumed.
(7) The data commanded in G37 or the parameter setting data must meet the following conditions:

$$| \text{Measurement point} - \text{start point} | > \text{R command or parameter } r$$
$$> \text{D command or parameter } d$$

(8) When the D command and parameter d in (7) above are zero, operation will be completed normally only when the commanded measurement point and sensor signal detection point coincide. Otherwise, program error (P607) will results.
(9) When the R and D commands as well as parameters r and d in (7) above are all zero, program error (P607) will result regardless of whether the sensor signal is present or not after the tool has been positioned at the commanded measurement point.
(10) The automatic tool length measurement command (G37) must be commanded together with the G43H_ command that designates the offset No.

$$\begin{bmatrix} \text{G43H\_\_ ;} \\ \text{G37Z\_\_R\_\_D\_\_F\_\_ ;} \end{bmatrix}$$

15.2 Skip Function; G31

Function and purpose

When the skip signal is input externally during linear interpolation based on the G31 command, the machine feed is stopped immediately, the remaining distance is discarded and the command in the following block is executed.

Command format

> **G31 Xx Yy Zz $\alpha\alpha$ Ff ; (where, a is the additional axis)**
>
> x, y, z, α : Axis coordinates; they are commanded as absolute or incremental
>   values according to the G90/G91 modal when commanded.
> f   : Feedrate (mm/min)

Linear interpolation can be executed using this function. If the skip signal is input externally while this command is being executed, the machine will stop, the remaining commands will be canceled and operation will be executed from the next block.

Detailed description

(1) If Ff is assigned as the federate in the same block as the G31 dommand block, command feed f will apply; if it not assigned, the value set in the parameter "#1174 Skip_F" will serve as the feedrate. In either case, the F modal will not be updated.

(2) The machine will not automatically accelerate and decelerate with the G31 block. The G31 maximum speed will depend on the machine specifications.

(3) Override is invalid with the G31 command and it is fixed at 100%. Dry run is also invalid. The stop conditions (feed hold, interlock, override zero and stroke end) are valid. External deceleration is also valid.

(4) The G31 command is unmodal and so it needs to be commanded each time.

(5) If the skip command is input during G31 command start, the G31 command will be completed immediately.
When a skip signal has not been input until the G31 block completion, the G31 command will also be completed upon completion of the movement commands.

(6) When the G31 command is issued during tool diameter compensation, program error (P608) will result.

(7) When there is no F command in the G31 command and the parameter speed is also zero, program error (P603) will result.

(8) With machine lock or with a command for the Z-axis only with the Z-axis cancel switch ON, the skip signal will be ignored and execution will continue as far as the end of the block.

15. Measurement Support Functions

Execution of G31

```
G90  G00  X-100000  Y0 ;
     G31  X-500000  F100 ;
     G01  Y-100000 ;
     G31  X0  F100 ;
          Y-200000 ;
     G31  X-50000  F100 ;
          Y-300000 ;
          X0 ;
```
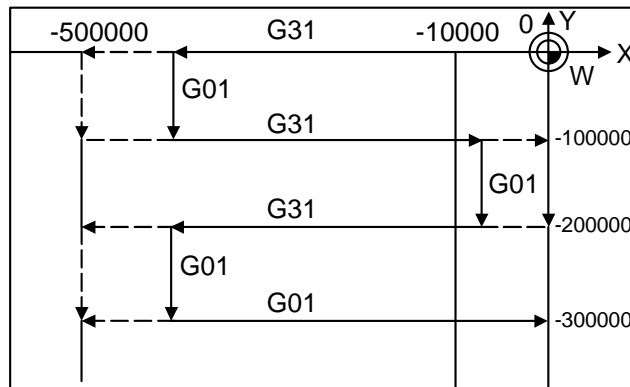

Detailed description (Readout of skip coordinates)

The coordinate positions for which the skip signal is input are stored in the system variables #5061 (1st axis) to #5063 (3rd axis), so these can be used in the user macros.

```
          ↳
G90  G00  X-100. ;
G31  X-200.  F60 ;          Skip command
#101 = #5061                Skip signal input coordinate values (workpiece
          ↳                 coordinate system) are readout to #101.
```

Detailed description (G31 coasting)

The amount of coasting from when the skip signal is input during the G31 command until the machine stops differs according to the parameter "#1174 skip_F" or F command in G31.
The time to start deceleration to a stop after responding to the skip signal is short, so the machine can be stopped precisely with a small coasting amount

$$\delta_0 = \frac{F}{60} \times Tp + \frac{F}{60} \times (t_1 \pm t_2) = \underbrace{\frac{F}{60} \times (Tp + t_1)}_{\delta_1} \pm \underbrace{\frac{F}{60} \times t_2}_{\delta_2}$$

δ_0 : Coasting amount (mm)
F : G31 skip speed (mm/min.)
Tp : Position loop time constant (s) = (position loop gain)$^{-1}$
t_1 : Response delay time (s) = (time taken from the detection to the arrival of the skip signal at the controller via PC)
t_2 : Response error time (0.001 s)

When G31 is used for calculation, the value calculated from the section indicated by $\delta 1$ in the above equation can be compensated, however, $\delta 2$ results in calculation error.

Stop pattern with skip signal input
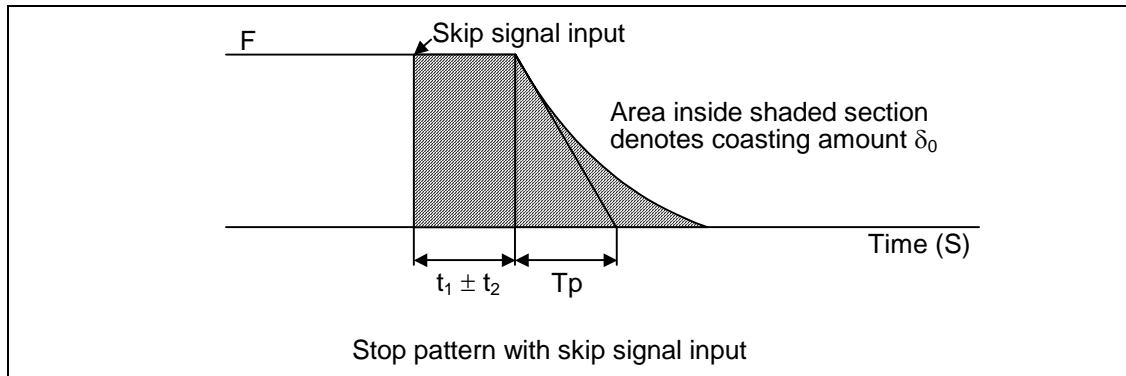
The relationship between the coasting amount and speed when Tp is 30ms and t_1 is 5ms is shown in the following figure.

Relationship between coasting amount and feedrate (example)

400
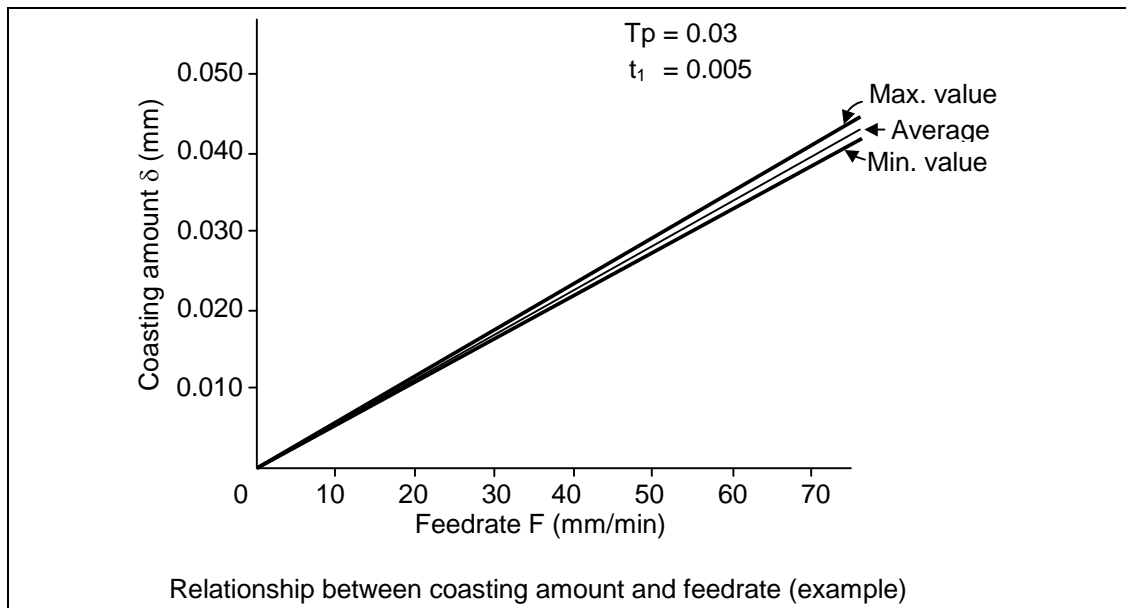
Detailed description (Skip coordinate readout error)

(1) Skip signal input coordinate readout

The coasting amount based on the position loop time constant Tp and cutting feed time constant Ts is not included in the skip signal input coordinate values.

Therefore, the work coordinate values applying when the skip signal is input can be read out across the error range in the following formula as the skip signal input coordinate values. However, coasting based on response delay time t1 results in a measurement error and so compensation must be provided.

$$\varepsilon = \pm \frac{F}{60} \times t_2$$

ε :   Readout error (mm)
F :   Feedrate (mm/min)
t_2 :   Response error time 0.001 (s)

Readout error of skip input coordinates

Readout error of skip input coordinates
Readout error with a 60mm/min feedrate is:

$$\varepsilon = \pm \frac{60}{60} \times 0.001 = \pm 0.001 \text{ (mm)}$$

Measurement value is within readout error range of $\pm 1\mu m$.
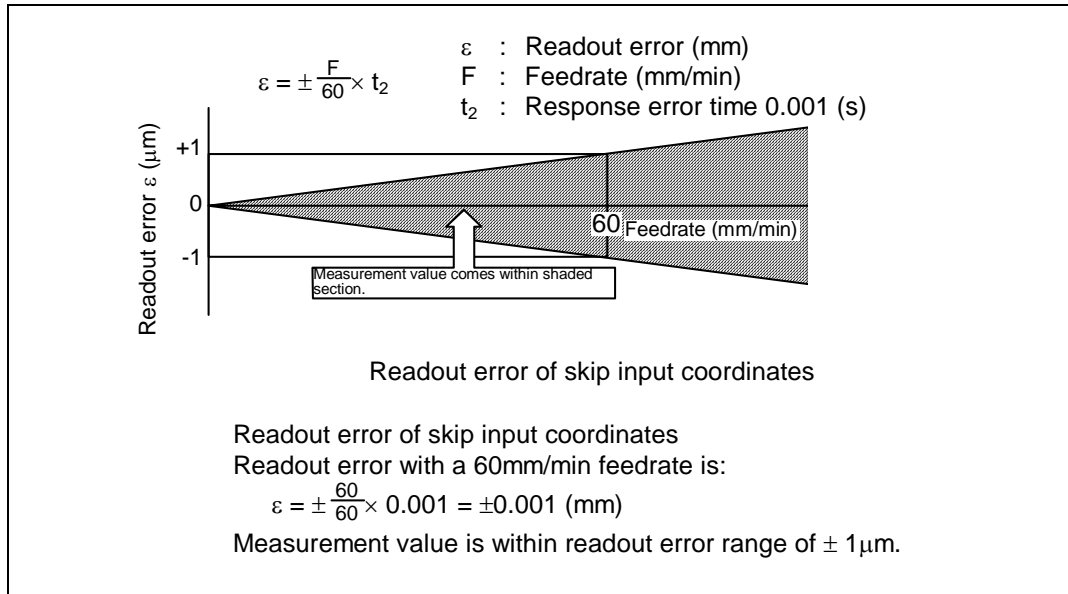
(2) Readout of other coordinates

The readout coordinate values include the coasting amount. Therefore, when coordinate values are required with skip signal input, reference should be made to the section on the G31 coasting amount and compensation provided. As in the case of (1), the coasting amount based on the delay error time t2 cannot be calculated, and this generates a measuring error.

Examples of compensating for coasting

(1) Compensating for skip signal input coordinates

```
#110 = Skip feedrate ;

#111 = Response delay time t₁ ;
```

$\wr$	
G31 X100. F100 ;	—— Skip command
G04 ;	—— Machine stop check
#101 = #5061 ;	—— Skip signal input coordinate readout
#102 = #110∗#111/60 ;	—— Coasting based on response delay time
#105 = #101−#102−#103 ;	—— Skip signal input coordinates
$\wr$	

(2) Compensating for work coordinates

```
#110 = Skip feedrate ;

#111 = Response delay time t₁ ;

#112 = Position loop time constant Tp ;
```

$\wr$	
G31 X100. F100 ;	—— Skip command
G04 ;	—— Machine stop check
#101 = #5061 ;	—— Skip signal input coordinate readout
#102 = #110∗#111/60 ;	—— Coasting based on response delay time
#103 = #110∗#112/60 ;	—— Coasting based on position loop time constant
#105 = #101−#102−#103 ;	—— Skip signal input coordinates
$\wr$	

15.3 Multi-stage skip function; G31.n, G04

Function and purpose

The setting of combinations of skip signals to be input enables skipping under various conditions. The actual skip operation is the same as with G31.

The G commands which can specify skipping are G31.1, G31.2, G31.3, and G04, and the correspondence between the G commands and skip signals can be set by parameters.

Command format

> **G31.1 Xx Yy Zz αα Ff ;**
>
> Xx Yy Zz αα   ; Command format axis coordinate word and target coordinates
> Ff   ; Feedrate (mm/min)

Same with G31.2 and G31.3 ; Ff is not required with G04

As with the G31 command, this command executes linear interpolation and when the preset skip signal conditions have been met, the machine is stopped, the remaining commands are canceled, and the next block is executed.

Detailed description

(1) Feedrate G31.1 set with the parameter corresponds to "#1176 skip1f", G31.2 corresponds to "#1178 skip2f", and G31.3 corresponds to "#1180 skip3f".
(2) A command is skipped if it meets the specified skip signal condition.
(3) The G31.n and G04 commands work the same as the G31 command for other than (1) and (2) above.
(4) The feedrates corresponding to the G31.1, G31.2, and G31.3 commands can be set by parameters.
(5) The skip conditions (logical sum of skip signals which have been set) corresponding to the G31.1, G31.2, G31.3 and G04 commands can be set by parameters.

Parameter setting	Valid skip signal		
	1	2	3
1	○		
2		○	
3	○	○	
4			○
5	○		○
6		○	○
7	○	○	○

(Skip when ○ signal is input.)

Example of operation

(1) The multi-stage skip function enables the following control, thereby improving measurement accuracy and shortening the time required for measurement.

```
Parameter settings :
Skip condition    Skip speed
G31.1    : 7      20.0mm/min (f1)
G31.2    : 3      5.0mm/min (f2)
G31.3    : 1      1.0mm/min (f3)
```

Program example :

```
N10G31.1 X200.0 ;
N20G31.2 X40.0 ;
N30G31.3 X1.0 ;
```
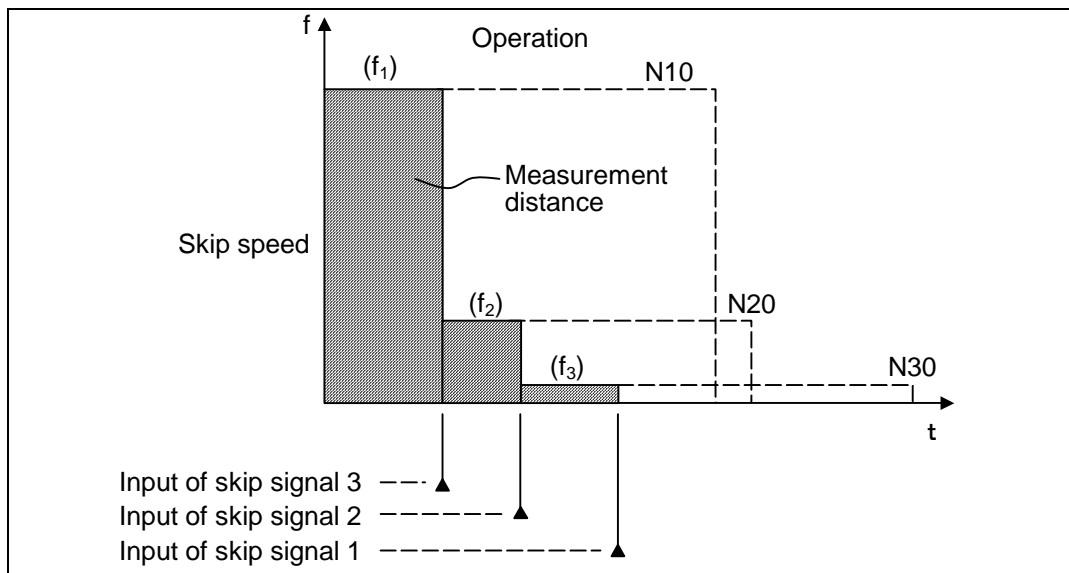

(Note 1) If skip signal 1 is input before skip signal 2 in the above operation, N20 is skipped at that point and N30 is also ignored.

(2) If a skip signal with the condition set during G04 (dwell) is input, the remaining dwell time is canceled and the following block is executed.

15.4 Multi-skip function 2

Function and purpose

During linear interpolation, command operation is skipped if skip signal parameter Pp specified with a skip command (G31), which indicates external skip signals 1 to 4, is met.

If the skip condition specified by the parameter "#1173 dwlskp" (indicating external skip signals 1 to 4) is met during execution of a dwell command (G04), the remaining dwell time is canceled and the following block is executed.

Command format

G31 Xx Yy Zz $\alpha\alpha$ Pp Ff ;
Xx Yy Zz $\alpha\alpha$: Command format axis coordinate word and target coordinates
Pp   : Skip signal parameter
Ff   : Feedrate (mm/min)

Detailed description

(1) The skip is specified by command speed f. Note that the F modal is not updated.

(2) The skip signal is specified by skip signal parameter p. p can range from 1 to 15. If p is specified outside the range, program error (P35) occurs.

Skip signal parameter P	Valid skip signal			
	4	3	2	1
1				○
2			○	
3			○	○
4		○		
5		○		○
6		○	○	
7		○	○	○
8	○			
~	~	~	~	~
13	○	○		○
14	○	○	○	
15	○	○	○	○

(Skip when ○ signal is input.)

(3)   The specified skip signal command is a logical sum of the skip signals.

(Example)
         G31 X100. P5 F100 ;
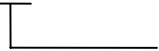
                                        Operation is skipped if skip signal 1 or 3 is input.

(4) If skip signal parameter Pp is not specified, the skip condition specified by the G31 parameter works. If speed parameter Ff is not specified, the skip speed specified by the G31 parameter works.

Relations between skip and multi-skip

Skip specifications	x		o	
	Condition	**Speed**	**condition**	**Speed**
G31 X100 ; Without P and F	Program error (P601)		Skip 1	Parameter
G31 X100 P5 ; Without F	Program error (P602)		Command value	Parameter
G31 X100 F100 ; Without P	Program error (P601)		Skip 1	Command value
G31 X100 P5 F100 ;	Program error (P602)		Command value	Command value

(Note) "Parameter" in the above table indicates that specified with a skip command (G31).

(5) If skip specification is effective and P is specified as an axis address, skip signal parameter P is given priority and axis address P is ignored.

(Example)
        G31 P500. F100 ;

                ⌐───────This is regarded as a skip signal parameter and program error (P35)
                        results.

(6) Those items other than (1) to (5) are the same with the ordinary skip function (G31 without P).

15.5 Programmable current limitation   [E68]

Function and purpose

This function allows the current limit value of the servo axis to be changed to a desired value in the program, and is used for the workpiece stopper, etc.
The commanded current limit value is designated with a ratio of the limit current to the rated current.

Command format

G10 L14 Xn ;
(n is the skip signal 1 to 8)
L14   : Current limit value setting (+ side/- side)
X   : Axis address
n   : Current limit value (%) Setting value: 1 to 300

Precautions and restrictions

(1) If the current limit value is reached when the current limit is valid, the current limit reached signal is output.

(2) The following two modes can be used as the operation after the current limit is reached. The external signal determines which mode applies.
  • Normal mode
    The movement command is executed in the current state.
    During automatic operation, the movement command is executed to the end, and then the next block is moved to with the droops still accumulated.
  • Interlock mode
    The internal interlock state has been entered while accumulating the droops, and the next movement is not executed.
    During automatic operation, the operation stops at the corresponding block, and the next block is not moved to.
    During manual operation, the following same direction commands are ignored.

(3) During the current limit, the position droop generated by the current limit can be canceled when the current limit changeover signal of external signals is canceled.
    (Note that the axis must not be moving.)

(4) The setting range of the current limit value is 1% to 300%. Commands that exceed this range will cause a program error (P35).

(5) If a decimal point is designated with the G10 command, only the integer will be valid.
    (Example) G10 L14 X10.123 ; The current limit value will be set to 10%.

(6) For the axis name "C", the current limit value cannot be set from the program (G10 command). To set from the program, set the axis name to one other than "C".

Appendix 1. Program Parameter Input N No. Correspondence Table

(Note 1) The units in the table indicate the minimum setting units for the parameter data.

(Note 2) The setting ranges given in the table are the setting ranges on the screen. Designate parameters related to the length by doubling the input setting unit.

(Example 1) If the input setting unit in the metric system is "B" (0.001mm), set 30mm in the parameter.

(Example 2) If the input setting unit in the inch system is "B" (0.0001inch), set 5inch in the parameter.

(Note 3) The binary type parameters must be converted into by-type data, and commanded with a decimal data after address D.

(Example 1) Binary data
$01010101_B = 55_H = 85_D$ · · · · · · · · · · · · Command 85

(Example 2) ASCII code
"M" = $01001101_B = 4D_H = 77_D$ · · · · · · Command 77
(B indicates Binary, H indicates Hexadecimal, and D indicates Decimal.)

[Machining parameters]   **Parameter 1.**

#	Item	P	A	N	Data type	Setting range	(Unit)	Remarks
8007	OVERRIDE	11	–	1010	D	0 to 100	(%)	
8008	MAX ANGLE	11	–	756	L	0 to 180	(°)	
8009	DSC. ZONE	11	–	760	L	0 to 99999.999	(mm)	
8013	G83 n	11	–	832	L	0 to 99999.999	(mm)	
8051	G71 THICK	11	–	784	L		(mm)	
8052	PULL UP	11	–	796	L		(mm)	
8053	G73 U	11	–	800	L		(mm)	
8054	W	11	–	804	L		(mm)	
8055	R	11	–	808	L		(time)	
8056	G74 RETRACT	11	–	820	L		(mm)	
8057	G76 LAST-D	11	–	824			(mm)	
8058	TIMES	11	–	997	D	0 to 99	(time)	
8059	ANGLE	11	–	998	D	0 to 99	(°)	

[Axis parameters]   **Parameter 1.**

#	Item	P	A	N	Data type	Setting range	(Unit)	Remarks
8202	OT-CHECK OFF	2	No.	897	H2	0 to 1		
8204	OT-CHECK-N	2	No.	916	L	±99999.999	(mm)	
8205	OT-CHECK-P	2	No.	912	L	±99999.999	(mm)	

Appendix 1.   Program Parameter Input N No. Correspondence Table

[PLC data]   **Setup parameter 6.**

#	Item	P	A	N	Data type	Setting range	(Unit)	Remarks
#6301 to #6348	PLC constant	5	–	1 to 48	L	±99999999		

[PLC timer]   **Setup parameter 6.**

#	Item	P	A	N	Data type	Setting range	(Unit)	Remarks
T0 to T15	10ms TIMER (T0 to T15)	6	–	0 to 15	S	0 to 32767	0.01s	
T16 to T95	100ms TIMER (T16 to T95)	6	–	16 to 95	S	0 to 32767	0.1s	
T96 to T103	100ms TIMER (T96 to T103) (INC.)	6	–	96 to 103	S	0 to 32767	0.1s	

[PLC counter]   **Setup parameter 6.**

#	Item	P	A	N	Data type	Setting range	(Unit)	Remarks
C0 to C23	PLC counter	7	–	0 to 23	L	0 to 32767		

[Bit selection]   **Setup parameter 6.**

#	Item	P	A	N	Data type	Setting range	(Unit)	Remarks
6401 to 6496	Bit selection	8	–	1 to 96	D or H0 to H7	0x00 to 0xFF 0 to 256 **(Note)** 0 to 1	(Hex) (Dec)	

(Note) Convert the 0x00 to 0xFF data into a decimal before using.

Appendix 2. Program Error

(The bold characters are the message displayed in the screen.)
These alarms occur during automatic operation, and the causes of these alarms are mainly program errors which occur, for instance, when mistakes have been made in the preparation of the machining programs or when programs which conform to the specification have not been prepared.

Error No.	Details	Remedy
P 10	**EXCS. AXIS No.** The number of axis addresses commanded in the same block exceeds the specifications.	• Divide the alarm block command into two. • Check the specifications.
P 11	**AXIS ADR. ERROR** The axis address commanded by the program and the axis address set by the parameter do not match.	• Revise the axis names in the program.
P 20	**DIVISION ERROR** An axis command which cannot be divided by the command unit has been issued.	• Check the program.
P 29	**NOT ACCEPT CMND** The normal line control command (G40.1, G41.1, G42.1) has been issued during the modal in which the normal line control is not acceptable.	• Check the program.
P 30	**PARITY H** The number of holes per character on the paper tape is even for EIA code and odd for ISO code.	• Check the paper tape. • Check the tape puncher and tape reader.
P 31	**PARITY V** The number of characters per block on the paper tape is odd.	• Make the number of characters per block on the paper tape even. • Set the parameter parity V selection OFF.
P 32	**ADDRESS ERROR** An address not listed in the specifications has been used.	• Check and revise the program address. • Check and correct the parameters values. • Check the specifications.
P 33	**FORMAT ERROR** The command format in the program is not correct.	• Check the program.
P 34	**G-CODE ERROR** A G code not listed in the specifications has been used. An illegal G code was commanded during the coordinate rotation command (G68).	• Check and correct the G code address in the program.

Appendix 2.   Program Error

Error No.	Details	Remedy
P 35	**CMD-VALUE OVER** The setting range for the addresses has been exceeded.	• Check the program.
P 36	**PROGRAM END ERR** "EOR" has been read during tape and memory operation.	• Enter the M02 and M30 command at the end of the program. • Enter the M99 command at the end of the subprogram.
P 37	**PROG. NO. ZERO** A zero has been specified for program and sequence numbers.	• The program numbers are designated across a range from 1 to 99999999. • The sequence numbers are designated across a range from 1 to 99999.
P 39	**NO SPEC ERR** • A non-specified G code was specified. • The high-speed program server operation specifications are not provided.	• Check the specifications.
P 40	**PREREAD BL. ERR** When tool radius compensation is executed, there is an error in the pre-read block and so the interference check is disabled.	• Reconsider the program.
P 60	**OVER CMP. LENG.** The commanded movement distance is excessive. (Over 2^{31})	• Reconsider the axis address command.
P 62	**F-CMD. NOTHING** • No feed rate command has been issued. • There is no F command in the cylindrical interpolation or pole coordinate interpolation immediately after the G95 mode is commanded.	• The default movement modal command at power ON is G01. This causes the machine to move without a G01 command if a movement command is issued in the program, and an alarm results. Use an F command to specify the feedrate. • Specify F with a thread lead command.
P 65	**NO G05P3 SPEC**	• Check the high-speed mode III specifications.
P 70	**ARC ERROR** • There is an error in the arc start and end points as well as in the arc center. • The difference of the involute curve through the start point and the end point is large. • When arc was commanded, one of the two axes configuring the arc plane was a scaling valid axis.	• Check the numerical values of the addresses that specify the start and end points, arc center as well as the radius in the program. • Check the "+" and "–" directions of the address numerical values.

Appendix 2.   Program Error

Error No.	Details	Remedy
P 71	**ARC CENTER** • The arc center is not sought during R-specified circular interpolation. • The curvature center of the involute curve cannot be obtained.	• Check the numerical values of the addresses in the program. • Check whether the start point or end point is on the inner side of the base circle for involute interpolation. When carrying out tool radius compensation, check that the start point and end point after compensation are not on the inner side of the base circle for involute interpolation. • Check whether the start point and end point are at an even distance from the center of the base circle for involute interpolation.
P 72	**NO HELICAL SPEC** A helical command has been issued though it is not included in the specifications.	• Check the helical specifications. • An Axis 3 command was issued by the circular interpolation command. If there is no helical specification, the linear axis is moved to the next block.
P 90	**NO THREAD SPEC** A thread cutting command has been issued though it is not included in the specifications.	• Check the specifications.
P 93	**SCREW PITCH ERR** The screw pitch has not been set correctly when the thread cutting command is issued.	• Issue the thread cutting command and then set the screw pitch command properly.
P100	**NO CYLIND SPEC** Cylindrical interpolation was commanded when the cylindrical interpolation specifications were not provided.	• Check the specifications.
P111	**PLANE CHG (CR)** Plane selection commands (G17, G18, and G19) were issued when a coordinate rotation command (G68) was issued.	• Before issuing the plane selection commands, issue G68 and then G69 (coordinate rotation cancel).
P112	**PLANE CHG (CC)** • A plane selection command (G17, G18, G19) has been issued when the tool radius compensation command (G41, G42) or nose R compensation command (G41, G42, G46) is issued. • The plane selection command was issued when nose R compensation is completed, there is no axial movement command after the G40 command, and the compensation has not been canceled.	• Issue the plane selection command after the tool radius compensation command or nose R compensation command has been canceled (issue axial movement command after the G40 cancel command).

Appendix 2.   Program Error

Error No.	Details	Remedy
P113	**ILLEGAL PLANE** The arc command axis is not on the selected plane.	• Issue arc command on the correctly selected plane.
P122	**NO AUTO C-OVR** An automatic corner override command (G62) has been issued though it is not included in the specifications.	• Check the specifications. • Delete the G62 command from the program.
P126	**ILL. CMD(H.A.)** An illegal command was issued during the high-accuracy control mode. • A G code group 13 command was issued during the high-accuracy control mode. • Milling, cylindrical interpolation or pole coordinate interpolation was commanded during the high-accuracy control mode.	• Reconsider the program.
P130	**2nd AUX. ADDR** The 2nd miscellaneous function address specified in the program does not match that set by the parameter.	• Check and correct the 2nd miscellaneous function address in the program.
P131	**NO G96 SPEC** (No constant peripheral speed) The constant peripheral speed command (G96) was issued despite the fact that such a command does not exist in the specifications.	• Check the specifications. • Change from the constant peripheral speed command (G96) to the rotation speed command (G97).
P132	**SPINDLE S = 0** No spindle speed command has been specified.	• Reconsider the program.
P133	**G96 P-No. ERR** An invalid constant peripheral speed control axis has been specified.	• Reconsider the parameter specified for the constant peripheral speed control axis.
P140	**NO T-POS OFST** The position compensation command (G45 to G48) specifications are not available.	• Check the specifications.
P141	**PAT-ROT ERROR** Position compensation was commanded during the figure rotation or coordinate rotation command.	• Reconsider the program.
P142	**T-OFFS G2 ERR** A position compensation invalid arc command was commanded.	• Reconsider the program.

414

Appendix 2. Program Error

Error No.	Details	Remedy
P150	**NO C-CMP SPEC** • Even though there were no tool radius compensation specifications, tool radius compensation commands (G41 and G42) were issued. • Even though there were no nose R compensation specifications, nose R compensation commands (G41, G42, and G46) were issued.	• Check the specifications.
P151	**G2, 3 CMP. ERR** A compensation command (G40, G41, G42, G43, G44, G46) has been issued in the arc mode (G02, G03).	• Issue the linear command (G01) or rapid traverse command (G00) in the compensation command block or cancel block. (Set the modal to linear interpolation.)
P152	**I.S.P NOTHING** In interference block processing during execution of a tool radius compensation (G41 or G42) or nose R compensation (G41, G42, or G46) command, the intersection point after one block is skipped cannot be determined.	• Reconsider the program.
P153	**I.F ERROR** An interference error has arisen while the tool radius compensation command (G41, G42) or nose R compensation command (G41, G42, G46) was being executed.	• Reconsider the program.
P155	**F-CYC ERR (CC)** A fixed cycle command has been issued in the radius compensation mode.	• The radius compensation mode is established when a fixed cycle command is executed and so the radius compensation cancel command (G40) should be issued.
P156	**BOUND DIRECT** At the start of G46 nose R compensation, the compensation direction is undefined if this shift vector is used.	• Change the vector to that with which the compensation direction is defined. • Exchange with a tool having a different tip point number.
P157	**SIDE REVERSED** During G46 nose R compensation, the compensation direction is inverted.	• Change the G command to that which allows inversion of the compensation direction (G00, G28, G30, G33, or G53). • Exchange with a tool having a different tip point number. • Turn ON the #8106 G46 inversion error avoidance parameter.
P158	**ILLEGAL TIP P.** During G46 nose R compensation, the tip point is illegal (other than 1 to 8).	• Change the tip point number to a legal one.

Appendix 2.   Program Error

Error No.	Details	Remedy
P170	**NO CORR. No.** The compensation number (DOO, TOO, HOO) command was not given when the radius compensation (G41, G42, G43, G46) command was issued. Alternatively, the compensation number is larger than the number of sets in the specifications.	• Add the compensation number command to the compensation command block. • Check the number of compensation number sets a correct it to a compensation number command within the permitted number of compensation sets.
P172	**G10 L-No. ERR** (G10 L-number error) The L address command is not correct when the G10 command is issued.	• Check the address L-Number of the G10 command and correct the number.
P173	**G10 P-No. ERR** (G10 compensation error) When the G10 command is issued, a compensation number outside the permitted number of sets in the specifications has been commanded for the compensation number command.	• First check the number of compensation sets and then set the address P designation to within the permitted number of sets.
P177	**LIFE COUNT ACT** Registration of tool life management data with G10 was attempted when the used data count valid signal was ON.	• The tool life management data cannot be registered when counting the used data. Turn the used data count valid signal OFF.
P178	**LIFE DATA OVER** The No. of registration groups, total No. of registered tools or the No. of registrations per group exceeded the specifications range.	• Review the No. of registrations. The maximum No. of registrations is shown below. <table><tr><td>No. of groups</td><td>80</td></tr><tr><td>No. of tools</td><td>80</td></tr><tr><td>Per group</td><td>16</td></tr></table>
P179	**GROUP No. ILL.** • When registering the tool life management data with G10, the group No. was commanded in duplicate. • A group No. that was not registered was designated during the T□□□□99 command. • An M code command must be issued as a single command but coexists in the same block as that of another M code command. • The M code commands set in the same group exist in the same block.	• The group No. cannot be commanded in duplicate. When registering the group data, register it in group units. • Correct to the correct group No.

416

Error No.	Details	Remedy
P180	**NO BORING CYC.** A fixed cycle command was issued though there are not fixed cycle (G72 - G89) specifications.	• Check the specifications. • Correct the program.
P181	**NO S-CMD (TAP)** The spindle rotation speed command has not been issued when the hole drilling fixed cycle command is given.	• Issue the spindle rotation speed command (S) when the hole drilling fixed cycle command G84, G74 (G84, G88) is given.
P182	**SYN TAP ERROR** Connection to the main spindle unit was not established.	• Check connection to the main spindle. • Check that the main spindle encoder exists.
P183	**PTC/THD, No.** The pitch or thread number command has not been issued in the tap cycle of a hole drilling fixed cycle command.	• Specify the pitch data and the number of threads by F or E command.
P184	**NO PTC/THD CMD** • The pitch or the number of threads per inch is illegal in the tap cycle of the hole drilling fixed cycle command. • The pitch is too small for the spindle rotation speed. • The thread number is too large for the spindle rotation speed.	• Check the pitch or the number of threads per inch.
P190	**NO CUTTING CYC** A lathe cutting cycle command was input although the lathe cutting cycle was undefined in the specification.	• Check the specification. • Delete the lathe cutting cycle command.
P191	**TAPER LENG ERR** In the lathe cutting cycle, the specified length of taper section is illegal.	• The radius set value in the lathe cycle command must be smaller than the axis shift amount.
P192	**CHAMFERING ERR** Chamfering in the thread cutting cycle is illegal.	• Set a chamfering amount not exceeding the cycle.
P200	**NO MRC CYC SPC** The fixed cycle for compound lathe I (G70 to G73) was commanded when the fixed cycle for compound lathe I specifications were not provided.	• Check the specification.

Appendix 2. Program Error

Error No.	Details	Remedy

P201	**PROG. ERR (MRC)**
	• When called with a fixed cycle for compound lathe I command, the subprogram contained at least one of the following commands:
	  • Reference point return command (G27, G28, G29, G30)
	  • Thread cutting (G33, G34)
	  • Fixed cycle skip-function (G31)
	• The first move block of the finish shape program in fixed cycle for compound lathe I contains an arc command.
		• Delete the following G codes from this subprogram that is called with the fixed cycle for compound lathe I commands (G70 to G73): G27, G28, G29, G30, G31, G33, G34, fixed cycle G-code.
		• Remove G2 and G3 from the first move block of the finish shape program in fixed cycle for compound lathe I.

P202	**BLOCK OVR (MRC)** The number of blocks in the shape program of the fixed cycle for compound lathe I is over 50 or 200 (this differs according to the model).	• Specify 50 or a less value. The number of blocks in the shape program called by the fixed cycle for compound lathe I commands (G70 to G73) must be decreased below 50 or 200 (this differs according to the model).
P203	**CONF. ERR (MRC)** The fixed cycle for compound lathe I (G70 to G73) shape program could not cut the work normally because it defined an abnormal shape.	• Check the fixed cycle for compound lathe I (G70 to G73) shape program.
P204	**VALUE ERR (MRC)** A command value of the fixed cycle for compound lathe (G70 to G76) is illegal.	• Check the fixed cycle for compound lathe (G70 to G76) command value.
P210	**NO PAT CYC SPC** A fixed cycle for compound lathe II (G74 to G76) command was input although it was undefined in the specification.	• Check the specification.
P220	**NO SPECIAL CYC** No special fixed cycle specifications are available.	• Check the specifications.
P221	**NO HOLE (S_CYC)** A 0 has been specified for the number of holes in special fixed cycle mode.	• Reconsider the program.
P222	**G36 ANGLE ERR** A G36 command specifies 0 for angle intervals.	• Reconsider the program.
P223	**G12 G13 R ERR** The radius value specified with a G12 or G13 command is below the compensation amount.	• Reconsider the program.
P224	**NO G12, G13 SPC** There are no circular cutting specifications.	• Check the specifications.

Error No.	Details	Remedy
P230	**NESTING OVER** • A subprogram has been called 8 or more times in succession from the subprogram. • The program in the IC card contains the M198 command. • The program in the IC card has been called more than once (the program in the IC card can be called only once at a time).	• Check the number of subprogram calls and correct the program so that it does not exceed 8 times. • When using the IC card, the IC card and the number of IC card program calls.
P231	**NO N-NUMBER** At subprogram call time, the sequence number set at return from the subprogram or specified by GOTO, was not set.	• Specify the sequence numbers in the call block of the subprogram. • When using an IC card, check the program and its No. in the IC card.
P232	**NO PROGRAM No.** • The specified subprogram is not registered in the memory or IC card. • The program file name registered in the IC card and O No. do not match.	• Enter the subprogram. • Check the program number in the IC card.
P241	**NO VARI NUMBER** The variable number commanded is higher than the numbers in the specifications.	• Check the specifications. • Check the program variable number.
P242	**EQL. SYM. MSSG.** The "=" sign has not been commanded when a variable is defined.	• Designate the "=" sign in the variable definition of the program.
P243	**VARIABLE ERR.** An invalid variable has been specified in the left or right side of an operation expression.	• Correct the program.
P252	**PAT.&COOD-ROT** A coordinate rotation related command (G68, G69) was issued during figure rotation.	• Reconsider the program.
P260	**NO COOD-RT SPC** Even though there were no coordinate rotation specifications, a coordinate rotation command was issued.	• Check the specifications.
P270	**NO MACRO SPEC** A macro specification was commanded though there are no such command specifications.	• Check the specifications.

Error No.	Details	Remedy
P271	**NO MACRO INT.** A macro interrupt command has been issued though it is not included in the specifications.	• Check the specifications.
P272	**MACRO ILL.** A statement and a macro statement exist together in the same block.	• Reconsider the program and place the executable statement and macro statement in separate blocks.
P273	**MACRO OVERCALL** The number of macro call nests exceeded the specifications.	• Reconsider the program and correct it so that the macro calls do not exceed the limit imposed by the specification.
P275	**MACRO ARG. EX.** The number of macro call argument type II sets has exceeded the limit.	• Reconsider the program.
P276	**CALL CANCEL** A G67 command was issued though it was not during the G66 command modal.	• Reconsider the program. • The G67 command is the call cancel command and so the G66 command must be designated first before it is issued.
P277	**MACRO ALM MESG** An alarm command has been issued in #3000.	• Refer to the operator messages on the DIAG screen. • Refer to the instruction manual issued by the machine manufacturer.
P280	**EXC. [,** The number of parentheses "[" or "]" which can be commanded in a single block has exceeded five.	• Reconsider the program and correct it so the number of "[" or "]" does not exceed five.
P281	**[,] ILLEGAL** The number of "[" and "]" parentheses commanded in a single block does not match.	• Reconsider the program and correct it so that "[" and "]" parentheses are paired up properly.
P282	**CALC. IMPOSS.** The arithmetic formula is incorrect.	• Reconsider the program and correct the formula.
P283	**DIVIDE BY ZERO** The denominator of the division is zero.	• Reconsider the program and correct it so that the denominator for division in the formula is not zero.
P290	**IF SNT. ERROR** There is an error in the IF conditional GOTO☐ statement.	• Reconsider the program.
P291	**WHILE SNT. ERR** There is an error in the WHILE conditional DO☐-END☐ statement.	• Reconsider the program.
P292	**SETVN SNT. ERR** There is an error in the SETVN☐ statement when the variable name setting was made.	• Reconsider the program. • The number of characters in the variable name of the SETVN statement must be 7 or less.

Error No.	Details	Remedy
P293	**DO-END EXCESS** The number of □'s for DO-END□ in the WHILE conditional DO□ – END□ statement has exceed 27.	• Reconsider the program and correct it so that the number of 's in the DO – END statement does not exceed 27.
P294	**DO-END MMC.** The DO's and END's are not paired off properly.	• Reconsider the program and correct it so that the DO's and END's are paired off properly.
P295	**WHILE/GOTO TPE** There is a WHILE or GOTO statement on the tape during tape operation.	• During tape operation, a program which includes a WHILE or GOTO statement cannot be executed and so the memory operation mode is established instead.
P296	**NO ADR (MACRO)** A required address has not been specified in the user macro.	• Review the program.
P297	**ADR-A ERR.** The user macro does not use address A as a variable.	• Review the program.
P298	**PTR OP (MACRO)** User macro G200, G201, or G202 was specified during tape or MDI operation.	• Review the program.
P300	**VER. NAME ERROR** The variable names have not been commanded properly.	• Reconsider the variable names in the program and correct them.
P301	**VAR NAME DUPLI** The name of the variable has been duplicated.	• Correct the program so that the name is not duplicated.
P360	**NO PROG.MIRR.** A mirror image (G50.1 or G51.1) command has been issued though the programmable mirror image specifications are not provided.	• Check the specifications.
P370	**NO OPOS MR SPC** The facing turret mirror image specifications are not provided.	• Check the specifications.
P371	**MIRR ILLEGAL** Facing turret mirror image was commanded to an axis for which external mirror image or parameter mirror image is valid. Facing turret mirror image validating mirror image for a rotary axis was commanded.	• Check the program. • Check the parameters.

Error No.	Details	Remedy
P380	**NO CORNER R/C** A command was issued for corner chamfering/corner rounding though there are no such specifications.	• Check the specifications. • Remove the corner chamfering/corner rounding command from the program.
P381	**NO ARC R/C SPC** Corner chamfering/corner rounding was specified in the arc interpolation block although corner chamfering/corner rounding II is unsupported.	• Check the specifications.
P382	**CORNER NO MOVE** The block next to corner chamfering/corner rounding is not a movement command.	• Replace the block succeeding the corner chamfering/corner rounding command by G01 command.
P383	**CORNER SHORT** In the corner chamfering/corner rounding command, the movement distance was shorter than the value in the corner chamfering/corner rounding command.	• Make the corner chamfering/corner rounding less than the movement distance since this distance is shorter than the corner chamfering/corner rounding.
P384	**CORNER SHORT** When the corner chamfering/corner rounding command was input, the movement distance in the following block was shorter than the length of the corner chamfering/corner rounding.	• Make the corner chamfering/corner rounding less than the movement distance since this distance in the following block is shorter than the corner chamfering/corner rounding.
P385	**G0 G33 IN CONR** A block with corner chamfering/corner rounding was given during G00 or G33 modal.	• Recheck the program.
P390	**NO GEOMETRIC** A geometric command was issued though there are no geometric specifications.	• Check the specifications.
P391	**NO GEOMETRIC 2** There are no geometric IB specifications.	• Check the specifications.
P392	**LES AGL (GEOMT)** The angular difference between the geometric line and line is 1° or less.	• Correct the geometric angle.
P393	**INC ERR (GEOMT)** The second geometric block was specified by an incremental value.	• Specify this block by an absolute value.
P394	**NO G01 (GEOMT)** The second geometric block contains no linear command.	• Specify the G01 command.
P395	**NO ADRS (GEOMT)** The geometric format is invalid.	• Recheck the program.

Error No.	Details	Remedy
P396	**PL CHG. (GEOMT)** A plane switching command was executed during geometric command processing.	• Execute the plane switching command before geometric command processing.
P397	**ARC ERR (GEOMT)** In geometric IB, the circular arc end point does not contact or cross the next block start point.	• Recheck the geometric circular arc command and the preceding and following commands.
P398	**NO GEOMETRIC 1B** Although the geometric IB specifications are not included, a geometric command is given.	• Check the specifications.
P421	**PRAM. IN ERROR** • The specified parameter number or set data is illegal. • An illegal G command address was input in parameter input mode. • A parameter input command was input during fixed cycle modal or nose R compensation.	• Check the program.
P430	**AXIS NOT RET.** • A command was issued to move an axis, which has not returned to the reference point, away from that reference point. • A command was issued to an axis removal axis.	• Execute reference point return manually. • The command was issued to an axis for which axis removal is validated so invalidate axis removal.
P431	**NO 2nd REF. SPC** A command for second, third or fourth reference point return was issued though there are no such command specifications.	• Check the specifications.
P434	**COLLATION ERR** One of the axes did not return to the start position when the origin point collate command (G27) was executed.	• Check the program.
P435	**G27/M ERROR** An M command was issued simultaneously in the G27 command block.	• An M code command cannot be issued in a G27 command block and so the G27 command and M code command must be placed in separate blocks.
P436	**G29/M ERROR** An M command was issued simultaneously in the G29 command block.	• An M code command cannot be issued in a G29 command block and so the G29 command and M code command must be placed in separate blocks.

Error No.	Details	Remedy
P438	**NOT USE (G52)** A local coordinate system command was issued during execution of the G54.1 command.	• Review the program.
P450	**NO CHUCK BARR.** The chuck barrier on command (G22) was specified although the chuck barrier was undefined in the specification.	• Check the specification.
P460	**TAPE I/O ERROR** • An error has arisen in the tape reader or, alternatively, in the printer during macro printing. • Restart search was attempted in the tape mode (tape/IC card operation) without a search operation after a reset upon a restart search.	• Check the power and cable of the connected devices. • Check the I/O device parameters. • Insert the IC card. • After reset, always perform a search operation to have the head of program indexed prior to a restart search.
P461	**FILE I/O ERROR** A file of the machining program cannot be read.	• In memory mode, the programs stored in memory may have been destroyed. Output all of the programs and tool data once and format them. • Ensure that the external device that contains the file is mounted.
P462	**DNC ERROR** A communication error occurred during the BTR operation.	• L01 DNC ERROR is displayed simultaneously, so remedy the problem according to the error No.
P480	**NO MILL SPEC** • Milling was commanded when the milling specifications were not provided. • Pole coordinate interpolation was commanded when the pole coordinate interpolation specifications were not provided.	• Check the specification.
P481	**MILL ILL. G** • An illegal G code was used during the milling mode. • An illegal G code was used during cylindrical interpolation or pole coordinate interpolation. • The G07.1 command was issued during the tool radius compensation.	• Check the program.

Appendix 2.   Program Error

Error No.	Details	Remedy
P482	**MILL ILL. AXIS** • A rotary axis was commanded during the milling mode. • Milling was executed even though an illegal value was set for the milling axis No. • Cylindrical interpolation or pole coordinate interpolation was commanded during mirror image. • Cylindrical interpolation or pole coordinate interpolation was commanded before the tool compensation was completed after the T command. • G07.1 was commanded when cylindrical interpolation was not possible (there is no rotary axis, or external mirror image is ON). • G12.1 was commanded when polar coordinate interpolation was not possible. • An axis other than a cylindrical coordinate system axis was commanded during cylindrical interpolation.	• Check the machining program, parameters and PLC I/F signal.
P484	**MILL AXIS RET.** • Movement was commanded to an axis that had not completed reference point return during the milling mode. • Movement was commanded to an axis that had not completed reference point return during cylindrical interpolation or pole coordinate interpolation.	• Carry out manual reference point return.

Error No.	Details	Remedy
P485	**MILL ILL. MODAL** • The milling mode was turned ON during nose R compensation or constant surface speed control. • A T command was issued during the milling mode. • The mode was switched from milling to cutting during tool compensation. • Cylindrical interpolation or pole coordinate interpolation was commanded during the constant surface speed control mode (G96). • The command unacceptable in the cylindrical interpolation was issued. • A T command was issued during the cylindrical interpolation or pole coordinate interpolation mode. • A movement command was issued when the plane was not selected just before or after the G07.1 command. • A plane selection command was issued during the pole coordinate interpolation mode. • Cylindrical interpolation or pole coordinate interpolation was commanded during tool radius compensation. • The G16 plane in which the radius value of a cylinder is 0 was specified. • A cylindrical interpolation or pole coordinate interpolation command was issued during program coordinate rotation (G68).	• Check the program. • Before issuing G12.1, issue G40 or G97. • Before issuing G12.1, issue a T command. • Before issuing G13.1, issue G40. • Specify the radius value of a cylinder other than 0, or specify the X axis's current value other than 0 before issuing G12.1/G16.
P486	**MILLING ERROR** • The milling command was issued during the mirror image (when parameter or external input is turned ON). • Pole coordinate interpolation, cylindrical interpolation or milling interpolation was commanded during facing turret mirror image. • The start command of the cylindrical interpolation or polar coordinate interpolation was issued during the normal line control.	• Check the program.

Appendix 2.   Program Error

Error No.	Details	Remedy
P511	**SYNC CODE ERR** • Two or more synchronization M codes were commanded in the same block. • The synchronization M code and "!" code were commanded in the same block.	• Check the program.
P600	**NO AUTO TLM.** An automatic tool length measurement command (G37) was execute though there are no such command specifications.	• Check the specifications.
P601	**NO SKIP SPEC.** A skip command (G31) was issued though there are no such command specifications.	• Check the specifications.
P602	**NO MULTI SKIP** A multiple skipping command (G31.1, G31.2 or G31.3) was issued though there are no such command specifications.	• Check the specifications.
P603	**SKIP SPEED 0** The skip speed is 0.	• Specify the skip speed.
P604	**TLM ILL. AXIS** No axis or more than one axis was specified in the automatic tool length measurement block.	• Specify only one axis.
P605	**T-CMD IN BLOCK** The T code is in the same block as the automatic tool length measurement block.	• Specify this T code before the block.
P606	**NO T-CMD BEFOR** The T code was not yet specified in automatic tool length measurement.	• Specify this T code before the block.
P607	**TLM ILL. SIGNL** Before the area specified by the D command or decelerating area parameter d, the measurement position arrival signal went ON. The signal remains OFF to the end.	• Check the program.
P608	**SKIP ERROR (CC)** A skip command was specified during radius compensation processing.	• Specify a radius compensation cancel (G40) command' or remove the skip command.

427

Appendix 2.   Program Error

Error No.	Details	Remedy
P610	**ILLEGAL PARA.** • The parameter setting is not correct.	• Check the program. • Check the parameter.
P612	**EXP. ERROR** A movement command for exponential function interpolation was issued during facing turret mirror image.	• Check the program.
P700	**CMD-VALUE ILL.** Spindle synchronization was commanded to a spindle that is not connected serially.	• Check the program. • Check the parameter.
P900	**NO TANZ. SPEC** A normal line control command (G40.1, G41.1, G42.1) was issued when the normal line control specifications were not provided.	• Check the specifications.
P901	**TAN. AXIS G92** A coordinate system preset command (G92) was issued to a normal line control axis during normal line control.	• Check the program.
P902	**TAN. AXIS LINE** • The normal line control axis was set to a linear axis. • The normal line control axis was set to the linear type rotary axis II axis. • The normal line control axis has not been set. • The normal line control axis was the same as the plane selection axis.	• Correct the normal line control axis.
P903	**PLANE CHG (TAN)** The plane selection command (G17, G18, G19) was issued during normal line control.	• Delete the plane selection command (G17, G18, G19) from the program for normal line control.
P990	**PREPRO S/W ERR** Combining commands that required pre-reading (nose R offset, corner chamfering/corner rounding, geometric I, geometric IB, and fixed cycle for compound lathe) resulted in eight or more pre-read blocks.	• Reduce the number of commands that require pre-reading or delete such commands.

Appendix 3. Order of G Function Command Priority
(Command in a separate block when possible)

(Note) Upper level : When commanded in the same block indicates that both commands are executed simultaneously

G Group / Commanded G code	01 G00 to G03	02 G17 to G19	03 G90, G91		05 G94, G95	06 G20, G21	07 G40 to G42	08 G43, G44, G49
G00 to G03.1 Positioning/ interpolation	G command commanded last is valid. Group 1 modal is updated / ○	○ / ○ Also possible during arc modal	○ / ○		○ / ○	○ / ○	Arc and G41, G42 cause error P151 / ○ Diameter is compensated, and then moves	Arc and G43 to G49 cause error P70 / ○ The G49 movement in the arc modal moves with G01
G04 Dwell	○ Group 1 modal is updated G04 is executed / ○	○ / ○	○ / ○		○ / ○	○ / ○	G04 is executed G40 to G42 are ignored / ○	G04 is executed G43 to G49 are ignored / ○
G09 Exact stop check	○ / ○	○ / ○	○ / ○		○ / ○	○ / ○	○ / ○	○ / ○
G10, G11 Program data setting	○ G10 is priority for axis No movement I, J, K rotation input / ○	○ G10 is used for axis, so the selected plan axis will be the basic axis. / ○	○ / ○		○ / ○	○ / ○	G10 to G11 are executed G40 to G42 are ignored / ○	G10 to G11 are executed G43 to G49 are ignored / ○
G17 to G19 Plane selection	○ / ○	○ G command commanded last is valid. / ○	○ / ○		○ / ○	○ / ○	○ / Plane axis changeover during diameter compensation causes error P112	○ / ○

Appendix 3. Order of G Function Command Priority

G Group / Commanded G code	01 G00 to G03	02 G17 to G19	03 G90, G91		05 G94, G95	06 G20, G21	07 G40 to G42	08 G43, G44, G49
G20, G21 Inch/metric changeover	◯ ◯	◯ ◯	◯ ◯		◯ ◯	Possible in same block ◯	◯ ◯	◯ ◯
G27 to G30 Reference point compare/ return	◯ G00 to G03.1 modals are updated G27 to G30 are executed ◯	◯ ◯	◯ ◯		◯ ◯	◯ ◯	G27 to G30 are executed G40 to G42 are ignored ◯	G27 to G30 are executed G43 to G49 are ignored ◯
G31 to G31.3 Skip	◯ ◯	◯ ◯	◯ ◯		◯ ◯	◯ ◯	Error:P608 Error:P608	◯ ◯
G33 Thread cutting	G command commanded last is valid. ◯	◯ ◯	◯ ◯		◯ ◯	◯ ◯	◯ ◯	◯ ◯
G37 Automatic tool length measurement	G37 is executed G00 to G33 are ignored ◯	◯ ◯	◯ ◯		◯ ◯	◯ ◯	G37 is executed G40 to G42 are ignored	G37 is executed G43 to G49 are ignored ◯
G40 to G42 Tool diameter compensation	Arc and G41, G42 cause error P151 G41 and G42 in arc modal cause error P151	◯ Plane axis changeover during diameter compensation causes error P112	◯ ◯		◯ ◯	◯ ◯	G command commanded last is valid. ◯	◯ ◯

430

Appendix 3. Order of G Function Command Priority

Commanded G code	09 G73 to G89	10 G98, G99	12 G54 to G59	13 G61 to G64	14 G66 to G67	17 G96, G97		19 G50.1 G51.1
G00 to G03.1 Positioning/ interpolation	Group 1 command is executed Group 9 is canceled ○	○ ○	○ ○	○ ○	G66 to G67 are executed G00 to G03.1 modals are updated ○	○ ○		○ During the arc command, all axis names become mirror center data Movement with mirror shape ○
G04 Dwell	G04 is executed G73 to G89 are ignored ○	○ ○	G04 is executed Group 12 is changed ○	○ ○	○	○ ○		G04 is executed G50.1 and G51.1 are ignored ○
G09 Exact stop check	○ ○	○ ○	○ ○	○ ○	○ ○	○ ○		○ ○
G10, G11 Program data setting	G10 to G11 are executed G73 to G89 are ignored ○	○ ○	G10 is executed G54 to G59 modals are updated ○	○ ○	G66 to G67 are executed G10 is ignored ○	○ ○		G10 to G11 are executed G50.1 and G51.1 are ignored ○
G17 to G19 Plane selection	○ ○	○ ○	○ ○	○ ○	○ ○	○ ○		○ ○

Appendix 3.   Order of G Function Command Priority

G Group Commanded G code	09 G73 to G89	10 G98, G99	12 G54 to G59	13 G61 to G64	14 G66 to G67	17 G96, G97		19 G50.1 G51.1
G20, G21 Inch/metric changeover	○ ○	○ ○	○ ○	○ ○	○ ○	○ ○		○ ○
G27 to G30 Reference point compare/ return	○ ○	○ ○	○ ○	○ ○	G66 to G67 are executed G27 to G30 are ignored ○	○ ○		G27 to G30 are executed G50.1 and G51.1 are ignored ○
G31 to G31.3 Skip	○ ○	○ ○	○ ○	○ ○	○ ○	○ ○		○ ○
G33 Thread cutting	Group 1 command is executed Group 9 is canceled ○	○ ○	○ ○	○ ○	G66 to G67 are executed G33 modals is updated ○	○ ○		○ ○
G37 Automatic tool length measurement	 ○	○ ○	 ○	 ○	G66 to G67 are executed G37 modals is ignored ○	○ ○		G37 is executed G50.1 and G51.1 are ignored ○
G40 to G42 Tool diameter compensation	Error:P155 Error:P155	○ ○	○ ○	○ ○	○ ○	○ ○		○ ○

Appendix 3.   Order of G Function Command Priority

G Group / Commanded G code	01 G00 to G03.1 G33	02 G17 to G19	03 G90, G91		05 G94, G95	06 G20, G21	07 G40 to G42	08 G43, G44, G49
G43, G44, G49 Length compensation	Arc and G43, G44 cause error P70 Arc and G43, G44 cause error P70	○ ○	○ ○		○ ○	○ ○	○ ○	○ G command commanded last is valid. ○
G50.1 G51.1 Program mirror image	○ ○	○ ○	○ ○		○ ○	○ ○	○ ○	○ ○
G52 Local coordinate system	○ ○	○ ○	○ ○		○ ○	○ ○	G52 is executed G40 to G42 are ignored ○	G52 is executed G43 to G49 are ignored ○
G53 Machine coordinate system	○ ○	 ○	○ ○		 ○	○ ○	G53 is executed G40 to G42 are ignored ○	G53 is executed G40 to G42 are ignored ○
G54 to G59 Workpiece coordinate system	○ ○	○ ○	○ ○		○ ○	○ ○	○ ○	○ ○
G61 to G64 Mode selection	○ ○	○ ○	○ ○		○ ○	○ ○	○ ○	○ ○
G65 Macro call	G65 is executed G00 to G03.1 modals are updated ○	○ ○	○ ○		○ ○	○ ○	○ ○	G65 is executed G43 to G49 modals are updated ○

Appendix 3.   Order of G Function Command Priority

Commanded G code	01 G00 to G03.1 G33	02 G17, G19	03 G90, G92		05 G94, G95	06 G20, G21	07 G40 to G42	08 G43, G44 G49
G66 to G67 Macro call	G66 to G67 are executed G00 to G03.1 modals are updated ○	○ ○	○ ○		○ ○	○ ○	○ ○	○ G66 to G67 are executed G43 to G49 modals are updated ○
G73 to G89 Fixed cycle	G73 to G89 are canceled G01 to G33 modals are updated ○	○ ○	○ ○		○ ○	○ ○	Error:P155 Fixed cycle during compensation Error:P155	○ ○
G90, G91 Absolute value/ incremental value	○ ○	○ ○	Use in same block ○		○ ○	○ ○	○ ○	○ ○
G92 Coordinate system setting	○ ○	○ ○	○ ○		○ ○	○ ○	○ ○	○
G94, G95 Synchronous/ asynchronous	○ ○	○ ○	○ ○		G command commanded last is valid. ○	○ ○	○ ○	○ ○
G96, G97 Constant surface speed control	○ ○	○ ○	○ ○		○ ○	○ ○	○ ○	○ ○
G98, G99 Initial point/ R point return	○ ○	○ ○	○ ○		○ ○	○ ○	○ ○	○ ○

Appendix 3.   Order of G Function Command Priority

G Group Commanded G code	09 G73 to G89	10 G98 to G99	12 G54 to G59	13 G61 to G65	14 G66 to G67	17 G96, G97		19 G50.1 G51.1
G43, G44, G49 Length compensation	○ ○	○ ○	○ ○	○ ○	G66 to G67 are executed G43 to G49 modals are updated ○	○ ○		○ ○
G50.1 G51.1 Program mirror image	○ ○	○ ○	○ ○	○ ○	G66 to G67 are executed G50.1 G51.1 is ignored ○	○ ○		G command commanded last is valid. ○
G52 Local coordinate system	G52 is executed G73 to G89 are ignored ○	○ ○	○ ○	○ ○	○	○ ○		G52 is executed G50.1 G51.1 is ignored ○
G53 Machine coordinate system	○	○ ○	○ ○	○ ○	○	○ ○		G53 is executed G50.1 G51.1 is invalid ○
G54 to G59 Workpiece coordinate system	○ ○	○ ○	G command commanded last is valid. ○	○ ○	G66 to G67 are executed G54 to G59 modals are updated ○	○ ○		○ ○
G61 to G64 Mode selection	○ ○	○ ○	○ ○	G command commanded last is valid. ○	○ ○	○ ○		○ ○
G65 Macro call	G65 is executed G73 to G89 are ignored ○	○ ○	○ ○	○ ○	Error ○	○ ○		G65 is executed G50.1 G51.1 is ignored ○

G Group / Commanded G code	09 G73 to G89	10 G98, G99	12 G54 to G59	13 G61 to G67	14 G66 to G67	17 G96, G97		19 G50.1 G51.1
G66 to G67 Macro call	G66 to G67 are executed G73 to G89 are ignored	○	G66 to G67 are executed G54 to G59 modals are updated	○	G command commanded last is valid.	○		G66 to G67 are executed G50.1 G51.1 is ignored
	○	○	○	○	G command commanded last is valid.	○		○
G73 to G89 Fixed cycle	G command commanded last is valid.	○	○	○	G66 to G67 are executed G73 to G89 are ignored	○		○ All axes become mirror center
	○	○	○	○	○	○		○
G90, G91 Absolute value/ incremental value	○	○	○	○	○	○		○
	○	○	○	○	○	○		○
G92 Coordinate system setting	G92 is executed G73 to G89 are ignored	○		○		○		○ Note that G92 is priority for axis
	○	○	○	○	○	○		○
G94, G95 Synchronous/ asynchronous	○	○	○	○	○	○		○
	○	○	○	○	○	○		○
G96, G97 Constant surface speed control	○	○	○	○	○	G command commanded last is valid.		○
	○	○	○	○	○	○		○
G98, G99 Initial point/R point return	○	G command commanded last is valid.	○	○	○	○		○
	○	○	○	○	○	○		○

Revision History

Date of revision	Manual No.	Revision details
Mar. 2006	IB(NA)1500174-A	First edition created.

Global service network

NORTH AMERICA FA Center

EUROPEAN FA Center

CHINA FA Center

KOREAN FA Center

ASEAN FA Center

HONG KONG FA Center

TAIWAN FA Center
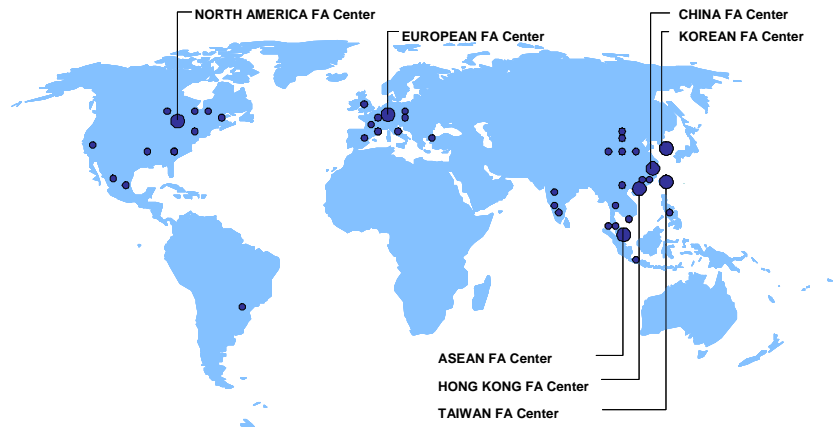
North America FA Center (MITSUBISHI ELECTRIC AUTOMATION INC.)

Illinois CNC Service Center
500 CORPORATE WOODS PARKWAY, VERNON HILLS, IL. 60061, U.S.A.
TEL: +1-847-478-2500 (Se   FAX: +1-847-478-2650 (Se
California CNC Service Center
5665 PLAZA DRIVE, CYPRESS, CA. 90630, U.S.A.
TEL: +1-714-220-4796   FAX: +1-714-229-3818
Georgia CNC Service Center
2810 PREMIERE PARKWAY SUITE 400, DULUTH, GA., 30097, U.S.A.
TEL: +1-678-258-4500   FAX: +1-678-258-4519
New Jersey CNC Service Center
200 COTTONTAIL LANE SOMERSET, NJ. 08873, U.S.A.
TEL: +1-732-560-4500   FAX: +1-732-560-4531
Michigan CNC Service Satellite
2545 38TH STREET, ALLEGAN, MI., 49010, U.S.A.
TEL: +1-847-478-2500   FAX: +1-269-673-4092
Ohio CNC Service Satellite
62 W. 500 S., ANDERSON, IN., 46013, U.S.A.
TEL: +1-847-478-2608   FAX: +1-847-478-2690
Texas CNC Service Satellite
1000, NOLEN DRIVE SUITE 200, GRAPEVINE, TX. 76051, U.S.A.
TEL: +1-817-251-7468   FAX: +1-817-416-1439
Canada CNC Service Center
4299 14TH AVENUE MARKHAM, ON. L3R OJ2, CANADA
TEL: +1-905-475-7728   FAX: +1-905-475-7935
Mexico CNC Service Center
MARIANO ESCOBEDO 69 TLALNEPANTLA, 54030 EDO. DE MEXICO
TEL: +52-55-9171-7662   FAX: +52-55-9171-7698
Monterrey CNC Service Satellite
ARGENTINA 3900, FRACC. LAS TORRES, MONTERREY, N.L., 64720, MEXICO
TEL: +52-81-8365-4171   FAX: +52-81-8365-4171
Brazil MITSUBISHI CNC Agent Service Center
 (AUTOMOTION IND. COM. IMP. E EXP. LTDA.)
ACESSO JOSE SARTORELLI, KM 2.1 18550-000 BOITUVA – SP, BRAZIL
TEL: +55-15-3363-9900   FAX: +55-15-3363-9911

European FA Center (MITSUBISHI ELECTRIC EUROPE B.V.)

Germany CNC Service Center
GOTHAER STRASSE 8, 40880 RATINGEN, GERMANY
TEL: +49-2102-486-0   FAX:+49-2102486-591
South Germany CNC Service Center
KURZE STRASSE. 40, 70794 FILDERSTADT-BONLANDEN, GERMANY
TEL: +49-711-3270-010   FAX: +49-711-3270-0141
France CNC Service Center
25, BOULEVARD DES BOUVETS, 92741 NANTERRE CEDEX FRANCE
TEL: +33-1-41-02-83-13   FAX: +33-1-49-01-07-25
Lyon CNC Service Satellite

U.K CNC Service Center
TRAVELLERS LANE, HATFIELD, HERTFORDSHIRE, AL10 8XB, U.K.
TEL: +44-1707-282-846   FAX:-44-1707-278-992
Italy CNC Service Center
ZONA INDUSTRIALE VIA ARCHIMEDE 35 20041 AGRATE BRIANZA, MILANO ITALY
TEL: +39-039-60531-342   FAX: +39-039-6053-206
Spain CNC Service Satellite
CTRA. DE RUBI, 76-80 -APDO.420 08190 SAINT CUGAT DEL VALLES, BARCELONA SPAIN
TEL: +34-935-65-2236   FAX:
Turkey MITSUBISHI CNC Agent Service Center
 (GENEL TEKNIK SISTEMLER LTD. STI.)
DARULACEZE CAD. FAMAS IS MERKEZI A BLOCK NO.43 KAT2 80270 OKMEYDANI ISTANBUL, TURKEY
TEL: +90-212-320-1640   FAX: +90-212-320-1649
Poland MITSUBISHI CNC Agent Service Center (MPL Technology Sp. z. o. o)
UL SLICZNA 34, 31-444 KRAKOW, POLAND
TEL: +48-12-632-28-85   FAX:
Wroclaw MITSUBISHI CNC Agent Service Satellite (MPL Technology Sp. z. o. o)
UL KOBIERZYCKA 23, 52-315 WROCLAW, POLAND
TEL: +48-71-333-77-53   FAX:+48-71-333-77-53
Czech MITSUBISHI CNC Agent Service Center
 (AUTOCONT CONTROL SYSTEM S.R.O.)
NEMOCNICNI 12, 702 00 OSTRAVA 2 CZECH REPUBLIC
TEL: +420-596-152-426   FAX: +420-596-152-112

ASEAN FA Center (MITSUBISHI ELECTRIC ASIA PTE. LTD.)

Singapore CNC Service Center
307 ALEXANDRA ROAD #05-01/02 MITSUBISHI ELECTRIC BUILDING SINGAPORE   159943
TEL: +65-6473-2308   FAX: +65-6476-7439
Thailand MITSUBISHI CNC Agent Service Center (F. A. TECH CO., LTD)
898/19,20,21,22 S.V. CITY BUILDING OFFICE TOWER 1 FLOOR 12,14   RAMA III RD BANGPONGPANG, YANNAWA, BANGKOK 10120. THAILAND
TEL: +66-2-682-6522   FAX: +66-2-682-6020
Malaysia MITSUBISHI CNC Agent Service Center
 (FLEXIBLE AUTOMATION SYSTEM SDN. BHD.)
60, JALAN USJ 10/1B 47620 UEP SUBANG JAYA SELANGOR DARUL EHSAN MALAYSIA
TEL: +60-3-5631-7605   FAX: +60-3-5631-7636
JOHOR MITSUBISHI CNC Agent Service Satellite
 (FLEXIBLE AUTOMATION SYSTEM SDN. BHD.)
NO. 16, JALAN SHAHBANDAR 1, TAMAN UNGKU TUN AMINAH, 81300 SKUDAI, JOHOR MALAYSIA
TEL: +60-7-557-8218   FAX: +60-7-557-3404
Indonesia MITSUBISHI CNC Agent Service Center
 (PT. AUTOTEKNINDO SUMBER MAKMUR)
WISMA NUSANTARA 14TH FLOOR JL. M.H. THAMRIN 59, JAKARTA 10350 INDONESIA
TEL: +62-21-3917-144   FAX: +62-21-3917-164
India MITSUBISHI CNC Agent Service Center (MESSUNG SALES & SERVICES PVT. LTD.)
B-36FF, PAVANA INDUSTRIAL PREMISES M.I.D.C., BHOASRI PUNE 411026, INDIA
TEL: +91-20-2711-9484   FAX: +91-20-2712-8115
BANGALORE MITSUBISHI CNC Agent Service Satellite
 (MESSUNG SALES & SERVICES PVT. LTD.)
S 615, 6TH FLOOR, MANIPAL CENTER, BANGALORE 560001, INDIA
TEL: +91-80-509-2119   FAX: +91-80-532-0480
Delhi MITSUBISHI CNC Agent Parts Center (MESSUNG SALES & SERVICES PVT. LTD.)
1197, SECTOR 15 PART-2, OFF DELHI-JAIPUR HIGHWAY BEHIND 32ND MILESTONE   GURGAON 122001, INDIA
TEL: +91-98-1024-8895   FAX:
Philippines MITSUBISHI CNC Agent Service Center
 (FLEXIBLE AUTOMATION SYSTEM CORPORATION)
UNIT No.411, ALABAMG CORPORATE CENTER KM 25. WEST SERVICE ROAD SOUTH SUPERHIGHWAY, ALABAMG MUNTINLUPA METRO MANILA, PHILIPPINES 1771
TEL: +63-2-807-2416   FAX: +63-2-807-2417
Vietnam MITSUBISHI CNC Agent Service Center (SA GIANG TECHNO CO., LTD)
47-49 HOANG SA ST. DAKAO WARD, DIST.1 HO CHI MINH CITY, VIETNAM
TEL: +84-8-910-4763   FAX: +84-8-910-2593

China FA Center (MITSUBISHI ELECTRIC AUTOMATION (SHANGHAI) LTD.)

China CNC Service Center
2/F., BLOCK 5 BLDG.AUTOMATION INSTRUMENTATION PLAZA, 103 CAOBAO RD.   SHANGHAI 200233, CHINA
TEL: +86-21-6120-0808   FAX: +86-21-6494-0178
Shenyang CNC Service Center
TEL: +86-24-2397-0184   FAX: +86-24-2397-0185
Beijing CNC Service Satellite
9/F, OFFICE TOWER1, HENDERSON CENTER, 18 JIANGUOMENNEI DAJIE,   DONGCHENG DISTRICT, BEIJING 100005, CHINA
TEL: +86-10-6518-8830   FAX: +86-10-6518-8030
China MITSUBISHI CNC Agent Service Center
 (BEIJING JIAYOU HIGHTECH TECHNOLOGY DEVELOPMENT CO.)
RM 709, HIGH TECHNOLOGY BUILDING NO.229 NORTH SI HUAN ZHONG ROAD, HAIDIAN DISTRICT , BEIJING 100083, CHINA
TEL: +86-10-8288-3030   FAX: +86-10-6518-8030
Tianjin CNC Service Satellite
RM909, TAIHONG TOWER, NO220 SHIZILIN STREET, HEBEI DISTRICT, TIANJIN, CHINA 300143
TEL: -86-22-2653-9090   FAX: +86-22-2635-9050
Shenzhen CNC Service Satellite
RM02, UNIT A, 13/F, TIANAN NATIONAL TOWER, RENMING SOUTH ROAD, SHENZHEN, CHINA 518005
TEL: +86-755-2515-6691   FAX: +86-755-8218-4776
Changchun Service Satellite
TEL: +86-431-50214546   FAX: +86-431-5021690
Hong Kong CNC Service Center
UNIT A, 25/F RYODEN INDUSTRIAL CENTRE, 26-38 TA CHUEN PING STREET, KWAI CHUNG, NEW TERRITORIES, HONG KONG
TEL: +852-2619-8588   FAX: +852-2784-1323

Taiwan FA Center (MITSUBISHI ELECTRIC TAIWAN CO., LTD.)

Taichung CNC Service Center
NO.8-1, GONG YEH 16TH RD., TAICHUNG INDUSTIAL PARK TAICHUNG CITY, TAIWAN R.O.C.
TEL: +886-4-2359-0688   FAX: +886-4-2359-0689
Taipei CNC Service Satellite
TEL: +886-4-2359-0688   FAX: +886-4-2359-0689
Tainan CNC Service Satellite
TEL: +886-4-2359-0688   FAX: +886-4-2359-0689

Korean FA Center (MITSUBISHI ELECTRIC AUTOMATION KOREA CO., LTD.)

Korea CNC Service Center
DONGSEO GAME CHANNEL BLDG. 2F. 660-11, DEUNGCHON-DONG KANGSEO-KU SEOUL, 157-030 KOREA
TEL: +82-2-3660-9607   FAX: +82-2-3663-0475

Notice

Every effort has been made to keep up with software and hardware revisions in the contents described in this manual. However, please understand that in some unavoidable cases simultaneous revision is not possible.
Please contact your Mitsubishi Electric dealer with any questions or comments regarding the use of this product.

Duplication Prohibited

This manual may not be reproduced in any form, in part or in whole, without written permission from Mitsubishi Electric Corporation.

MODEL	EZMotion-NC E60/E68 Series

MODEL CODE	008—400

Manual No.	IB-1500174(ENG)-A

MITSUBISHI ELECTRIC CORPORATION

Specifications subject to change without notice.

Printed in Japan on recycled paper.

(0603) MEE