MITSUBISHI ELECTRIC

Changes for the Better

CNC

MELDAS C6/C64/C64T

PROGRAMMING MANUAL

(MACHINING CENTER/TRANSFER MACHINE TYPE)

BNP-B2260B(ENG)

Introduction

This manual is a guide for using the MELDAS C6/C64/C64T.
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 this NC unit.

Details described in this manual

⚠ CAUTION

⚠ For items described in "Restrictions" or "Usable State", the instruction manual issued by the machine manufacturer 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 manufacturer before starting use.
⚠ Refer to the Instruction Manual issued by each machine manufacturer 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
MELDAS C6/C64/C64T Instruction Manual BNP-B2259

Precautions for Safety

Always read the specifications issued by the machine maker, 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 manufacturer 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 manufacturer before starting use.
⚠ Refer to the Instruction Manual issued by each machine manufacturer 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.

(Continued on next page)

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.

3. Items related to programming

- The commands with "no value after G" will be handled as "G00".
- " ; " "EOB" and " %" "EOR" are explanatory notations. The actual codes are "Line feed" and "%" for ISO, and "End of block" and "End of Record" for EIA.
- 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 manufacturer.
- When programming the multi-part system, take special care to the movements of the programs for other part systems.

Contents

Page

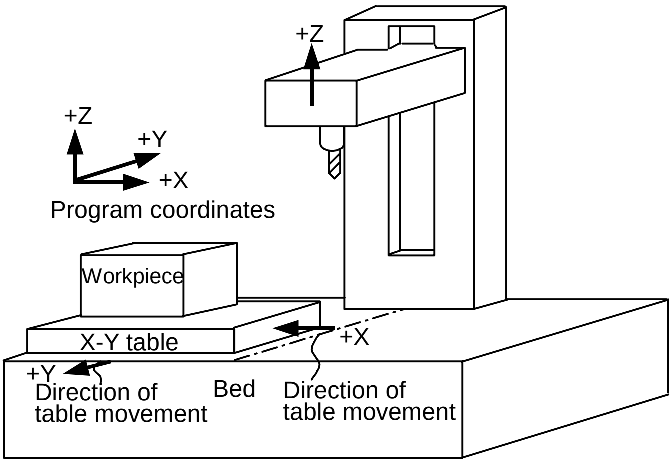
1. Control Axes

1.1 Coordinate word and control axis

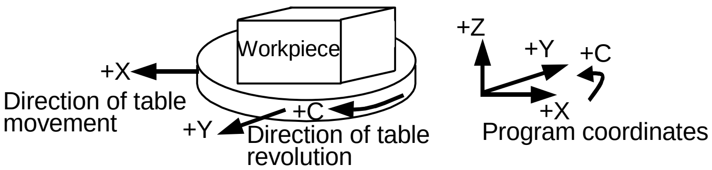
Function and purpose

In the standard specifications, there are 3 control axes, but, by adding an additional axis, up to 14 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

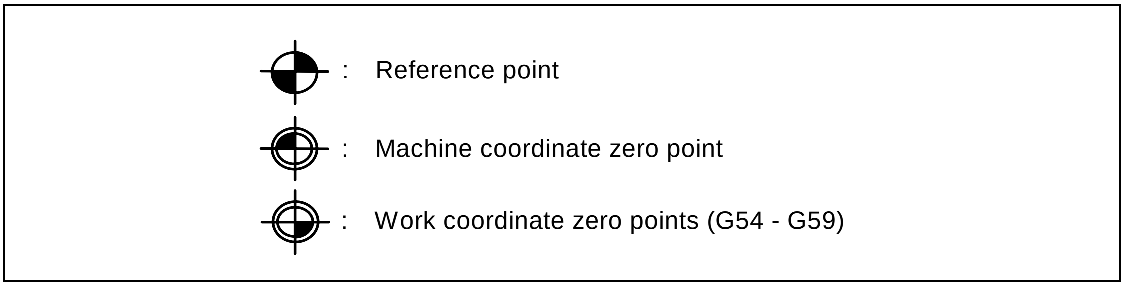


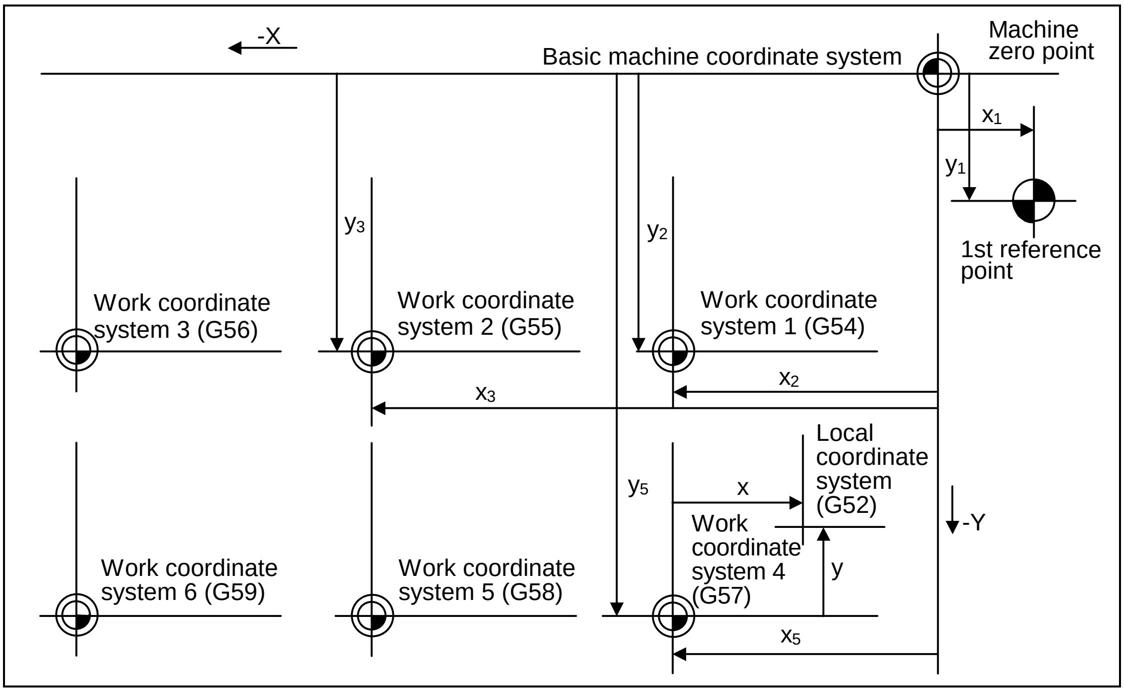
X-Y and revolving table



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. 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 command units can be selected from the following types for each axis with the parameters. The input setting units can be selected from the following types common to axes. (For further details on settings, refer to the Instruction Manual.)

	Input unit parameters	Linear axis				Rotation axis (°)
		Millimeter		Inch		
		Diameter command	Radius command	Diameter command	Radius command	
Input command unit	#1015 cunit = 10	0.001	0.001	0.0001	0.0001	0.001
	= 1	0.0001	0.0001	0.00001	0.00001	0.0001
Min. movement unit	#1003 iunit = B	0.0005	0.001	0.00005	0.0001	0.001
	= C	0.00005	0.0001	0.000005	0.00001	0.0001
Input setting unit	#1003 iunit = B	0.001	0.001	0.0001	0.0001	0.001
	= C	0.0001	0.0001	0.00001	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 explanatory notations. The actual codes are "Line feed" and "%" for ISO, and "End of block" and "End of Record" for EIA.

Detailed description

EOB/EOR keys and displays

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

(1) Significant data section (label skip function)

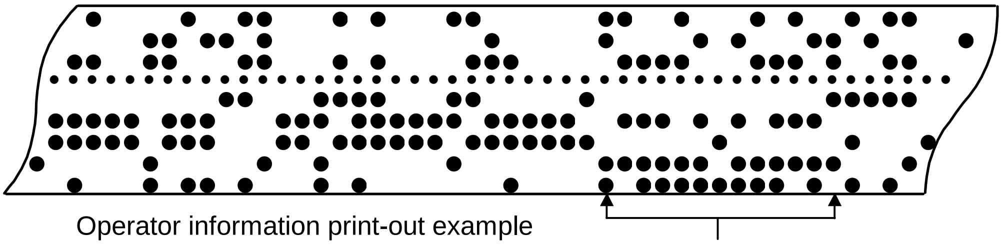
All data up to the first EOB (;), after the power has been turned on or after operation has been reset, are ignored during automatic operation based on tape, memory loading operation or during a search operation. In other words, the significant data section of a tape extends from the character or number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

(2) Control out, control in

When the ISO code is used, all data between control out "(" and control in ")" or ";" 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

LF CR G00X-85000Y-64000(CUTTER SP RETURN) LF



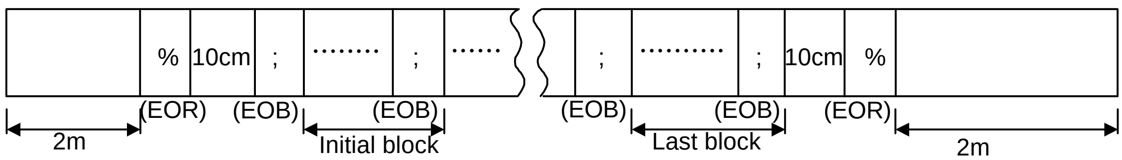
Operator information print-out example

Information in this section is ignored and nothing is executed.

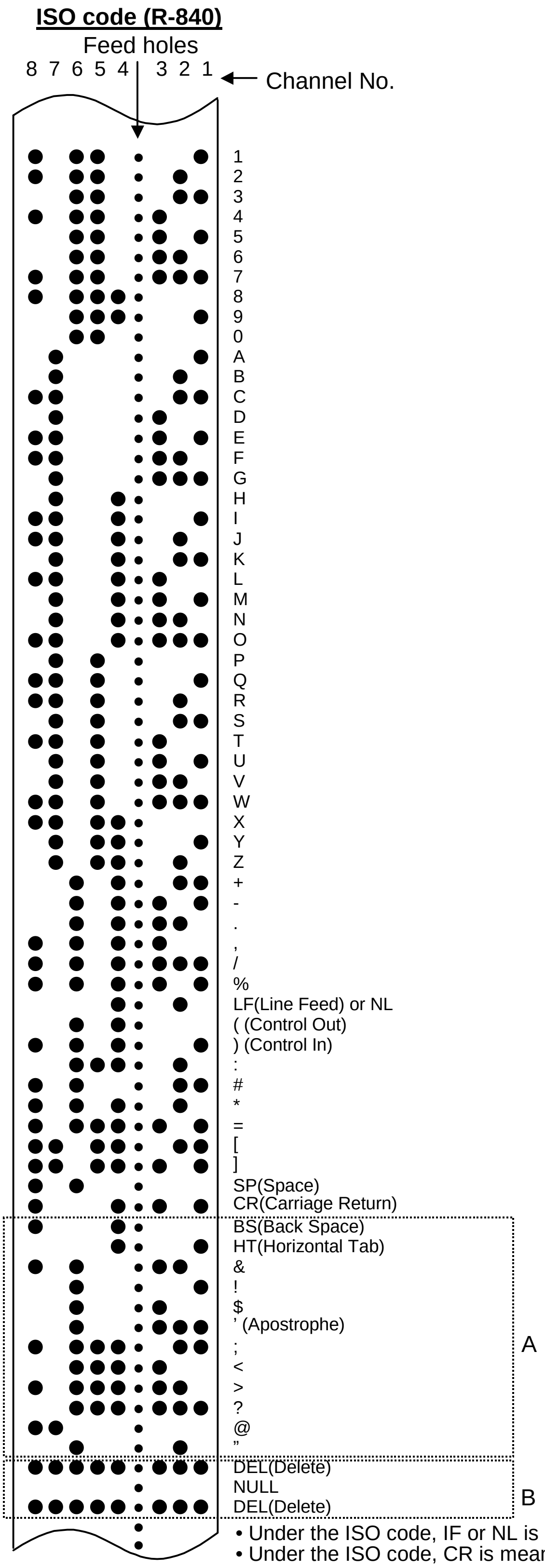
(3) EOR (%) code

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.



• 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

3.2 Program formats

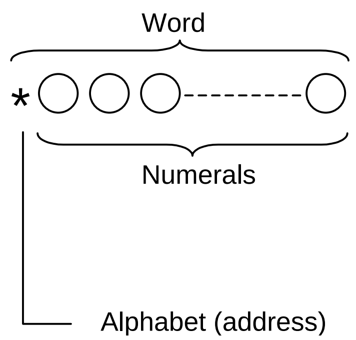
Function and purpose

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 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.

(Note 1) When there is no number following the alphabetic character in the actual program, the numeric value following the alphabetic character is handled as a 0.

(Example) G28XYZ; → G28X0Y0Z0;

Item			Metric command	Inch command
Program number			O8	
Sequence number			N5	
Preparatory function			G2/G21	
Movement axis	Input setting unit	0.01(°), mm	X+52 Y+52 Z+52 α+52	
	Input setting unit	0.001(°), mm/ 0.0001 inch	X+53 Y+53 Z+53 α+53	X+44 Y+44 Z+44 α+44
Additional axis	Input setting unit	0.01(°), mm	I+52 J+52 K+52	
	Input setting unit	0.001(°), mm/ 0.0001 inch	I+53 J+53 K+53	I+44 J+44 K+44
Dwell	Input setting unit	0.01(rev), mm	X53 P8	
	Input setting unit	0.001(rev), mm/ 0.0001 inch	X53 P8	X53 P8
Feed function	Input setting unit	0.01(°), mm	F53	
	Input setting unit	0.001(°), mm/ 0.0001 inch	F53	F44
Fixed cycle	Input setting unit	0.01(°), mm	R+52 Q52 P8 L4	
	Input setting unit	0.001(°), mm/ 0.0001 inch	R+53 Q53 P8 L4	R+44 Q+44 P8 L4
Tool offset			H3/D3	
Miscellaneous function			M8	
Spindle function			S6/S8	
Tool function			T8	
2nd miscellaneous function			A8/B8/C8	
Subprogram			P8H5L4	
Variable number			#5	

(Note 1) α represents one of the additional axes U, V, W, 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) The basic format is the same for any of the numerals input from the memory, MDI or setting display unit.

(Note 4) Numerals can be used without the leading zeros.

(Note 5) The program number is commanded with single block. It's necessary to command the program number in the head block of each program.

(Note 6) 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, too.

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:
- Reserved words
- Comment statements

Example of program

(1) Example of program for address check

(Example 1) When there are no numbers following an alphabetic character.
G28 X ; → 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.

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.

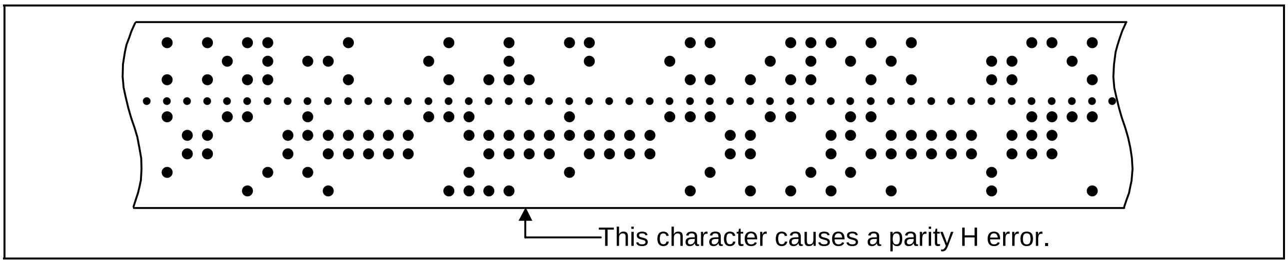
(1) Parity H

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



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
Δ 00	01	Positioning
* 01	01	Linear interpolation
02	01	Circular interpolation CW (clockwise)
03	01	Circular interpolation CCW (counterclockwise)
04	00	Dwell
06		
07		
08		
09	00	Exact stop check
10	00	Program parameter input/compensation input
11	00	Program parameter input cancel
12	00	Circular cut CW (clockwise)
13	00	Circular cut CCW (counterclockwise)
14		
15		
16		
* 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
* 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 position return 1
30.2	00	Tool position return 2
30.3	00	Tool position return 3
30.4	00	Tool position return 4
30.5	00	Tool position return 5
30.6	00	Tool position return 6
31	00	Skip function / Multi-step skip function
31.1	00	Multi-step skip function 1-1
31.2	00	Multi-step skip function 1-2
31.3	00	Multi-step skip function 1-3
32		
33	01	Thread cutting

G code	Group	Function
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
43	08	Tool length offset (+)
44	08	Tool length offset (-)
* 49	08	Tool length offset cancel
* 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	Workpiece coordinate system 1 selection
55	12	Workpiece coordinate system 2 selection
56	12	Workpiece coordinate system 3 selection
57	12	Workpiece coordinate system 4 selection
58	12	Workpiece coordinate system 5 selection
59	12	Workpiece coordinate system 6 selection
54.1	12	Workpiece coordinate system selection 48 sets expanded
60	00	Uni-directional positioning
61	13	Exact stop check mode
61.1	13	High-accuracy control mode
62	13	Automatic corner override
63	13	Tapping mode
* 64	13	Cutting mode
65	00	User macro call
66	14	User macro modal call A
66.1	14	User macro modal call B
* 67	14	User macro modal call cancel
70		User fixed cycle
71		User fixed cycle
72		User fixed cycle
73	09	Fixed cycle (step)
74	09	Fixed cycle (reverse tap)
75		User fixed cycle
76	09	Fixed cycle (fine boring)
77		User fixed cycle
78		User fixed cycle
79		User fixed cycle
* 80	09	Fixed cycle cancel
81	09	Fixed cycle (drill/spot drill)

G code	Group	Function
82	09	Fixed cycle (drill/counter boring)
83	09	Fixed cycle (deep drilling)
84	09	Fixed cycle (tapping)
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
* 91	03	Incremental command value
92	00	Machine coordinate system setting
93		
* 94	05	Asynchronous feed (per-minute feed)
Δ 95	05	Synchronous feed (per-revolution feed)
Δ 96	17	Constant surface speed control ON
* 97	17	Constant surface speed control OFF
* 98	10	Fixed cycle Initial level return
99	10	Fixed cycle R point level return
113	00	Spindle synchronous control OFF
114.1	00	Spindle synchronous control ON
115	00	• Start point designation synchronization (type1)
116	00	• Start point designation synchronization (type2)
117	00	• Miscellaneous function output during axis movement
100 ~ 255	00	User macro (G code call) Max. 10

(Note 1) A (∗) symbol indicates the G code to be selected in each group when the power is turned ON or when a reset is executed to initialize the modal.

(Note 2) A (Δ) symbol indicates the G code for which parameters selection is possible as an initialization status when the power is turned ON or when a reset is executed to initialize the modal. Note that inch/metric changeover can only be selected when the power is turned ON.

(Note 3) A (•) symbol indicates a function dedicated for multi-part system.

(Note 4) If two or more G codes from the same group are commanded, the last G code will be valid.

(Note 5) 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 manufacturer.

(Note 6) Whether the modal is initialized differs for each reset input.

(1) "Reset 1"
The modal is initialized when the reset initialization parameter (#1151 rstinit) is ON.

(2) "Reset 2 "and "Reset and Rewind"
The modal is initialized when the signal is input.

(3) Reset at emergency stop release
Conforms to "Reset 1".

(4) When an automatic reset is carried out at the start of individual functions, such as reference point return.
Conforms to "Reset and Rewind".

CAUTION

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

3.9 Precautions before starting machining

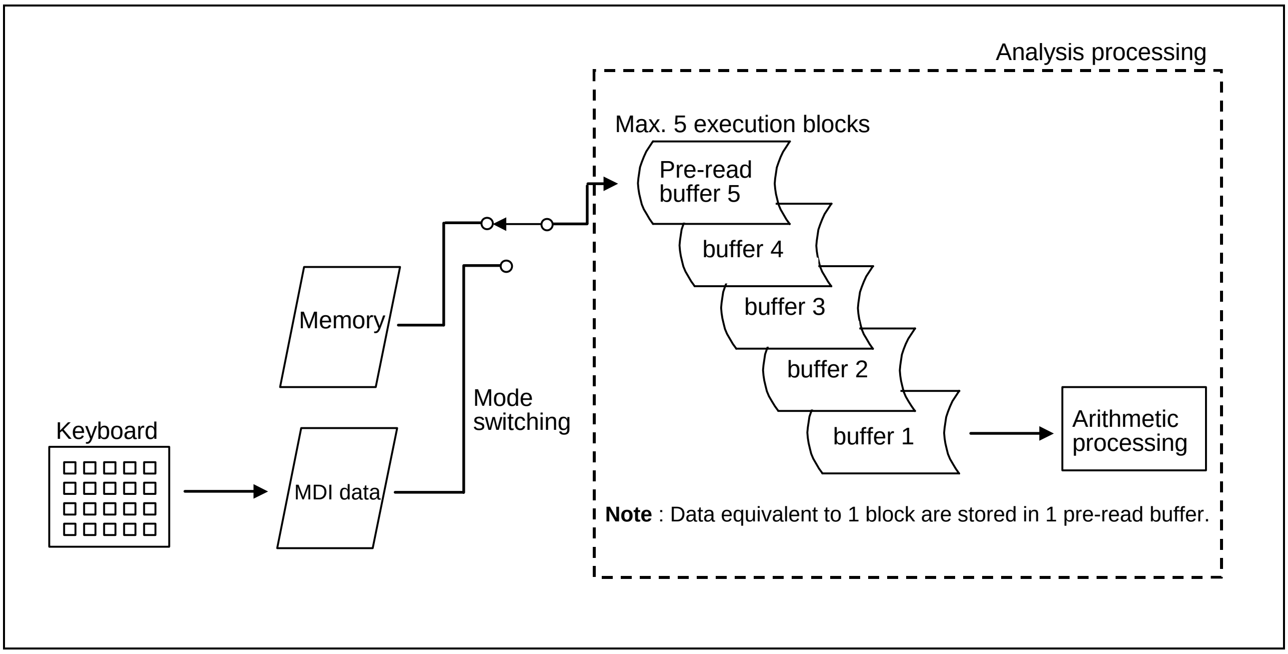
Precautions before starting machining

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 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.

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 -> 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.

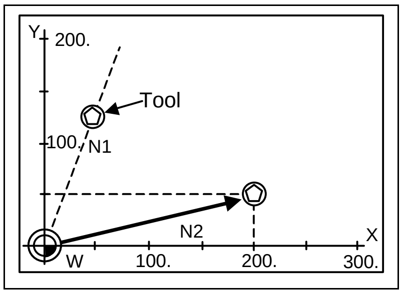
Command format

G90(G91) Xx1 Yy1 Zz1 αα1	
G90	:Absolute value command
G91	:Incremental command
α	:Additional axis

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;
```

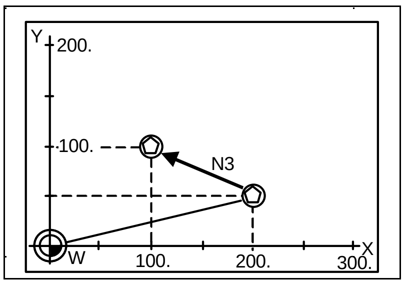
Using the command from the 0 point in the workpiece coordinate system, it becomes the same coordinate command value in either the absolute value mode or the incremental value mode.

(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.



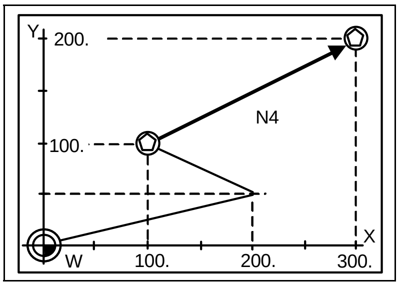
(G91)

```
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.

(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)

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;	0.100 mm	0.254 mm	0.0039 inch	0.0100 inch
Y	10	Y100;	0.100 mm	0.254 mm	0.0039 inch	0.0100 inch
Z	10	Z100;	0.100 mm	0.254 mm	0.0039 inch	0.0100 inch
X	1	X100;	0.0100mm	0.0254 mm	0.00039inch	0.00100inch
Y	1	Y100;	0.0100mm	0.0254 mm	0.00039inch	0.00100inch
Z	1	Z100;	0.0100mm	0.0254 mm	0.00039inch	0.00100inch

(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.

(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 ($0.05 \div 25.4 \fallingdotseq 0.002$).

(Note 2) 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.

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 1 (minimum input command unit) or type 2 (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.

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

(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.

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.

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	
	Invalid	Synchronous spindle No. at spindle synchronization	
E	Valid	Inch thread, number of ridges Precision thread lead	
F	Valid	Feed rate	
	Valid	Thread lead	
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)	
	Invalid	Basic spindle No. at spindle synchronization	
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)	
J	Valid	Arc center coordinates	
	Valid	Tool radius compensation vector components	
	Valid	Special fixed cycle's hole pitch or angle	
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	Program parameter input selection	L50
	Invalid	Program parameter input, 2-word type data	4 bytes
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	Valid	Dwell time	Parameter
	Invalid	Subprogram program call No.	
	Invalid	Dwell time at hole bottom of tap cycle	
	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	Skip signal command for multi-step skip	
	Invalid	Subprogram return destination sequence No.	
	Invalid	2nd, 3rd, 4th reference point return number	
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	
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	Synchronous spindle phase shift amount	
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	
	Valid	Dwell time	
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 αα ,Ii ; (α represents additional axis)	
x, y, z, α	: Represent coordinates, and could be either absolute values or incremental values, depending on the setting of G90/G91.
i	: In-position width. A decimal point command will result in a program error. This is valid only in the commanded block. A block that does not contain this address will follow the parameter "#1193 inpos" settings. The range is 1 to 999999 (μm).

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, the tool is always accelerated at the start point of the block and decelerated at the end point. Refer to (Note4) of "Example of program".

(3) If multiple axes are controlled, the next block will be executed after confirming that the position error amounts of all the moving axes become within the specified in-position width for each part system.

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

(5) Whether the tool moves along a linear or non-linear path is determined by parameter, but the positioning time does not change.

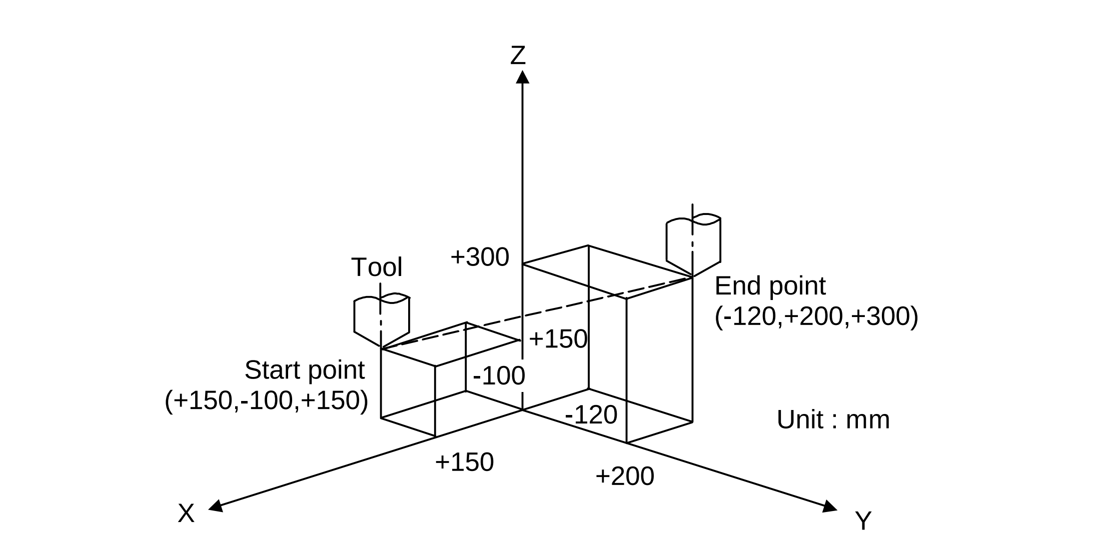
(a) Linear path..........: This is the same as linear interpolation (G01), and the speed is limited by the rapid traverse rate of each axis.
(b) Non-linear path...: The tool is positioned at the rapid traverse rate independently for each axis.

(6) Refer to "Operation during in-position check" for the programmable in-position check positioning command.

⚠CAUTION

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

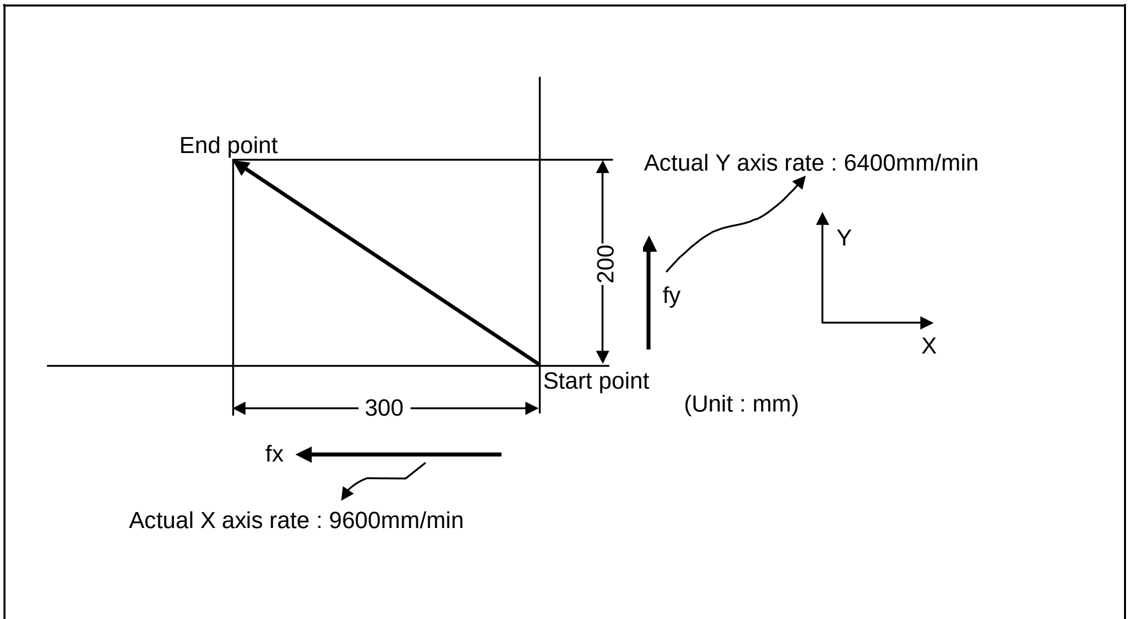
Example of program



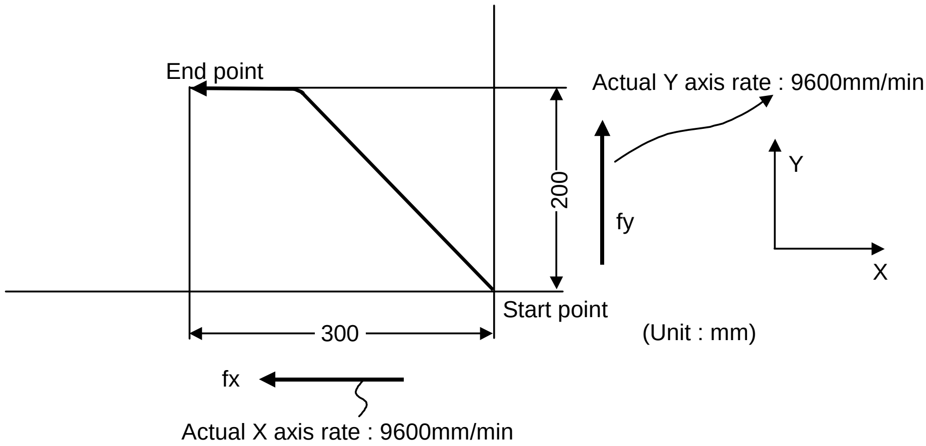
G91 G00 X-270000 Y300000 Z150000 ;	(For input setting unit: 0.001mm)

(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 manual.

(Note 4) Rapid traverse (G00) deceleration check

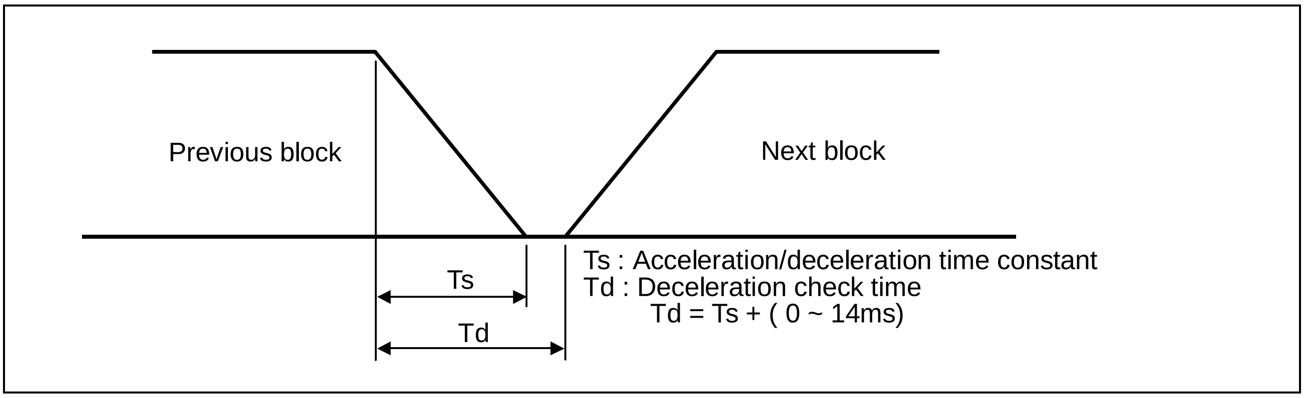
There are two methods for the deceleration check at rapid traverse; commanded deceleration method and in-position check method. Select a method with the parameter "#1193 inpos".

■ When “inpos” = “1”
Upon completion of the rapid traverse (G00), the next block will be executed after confirming that the remaining distances for each axis are below the fixed amounts. (Refer to the following drawing.)
The confirmation of the remaining distance should be done with the imposition width, L_R. L_R is the setting value for the servo parameter "#2224 SV024".

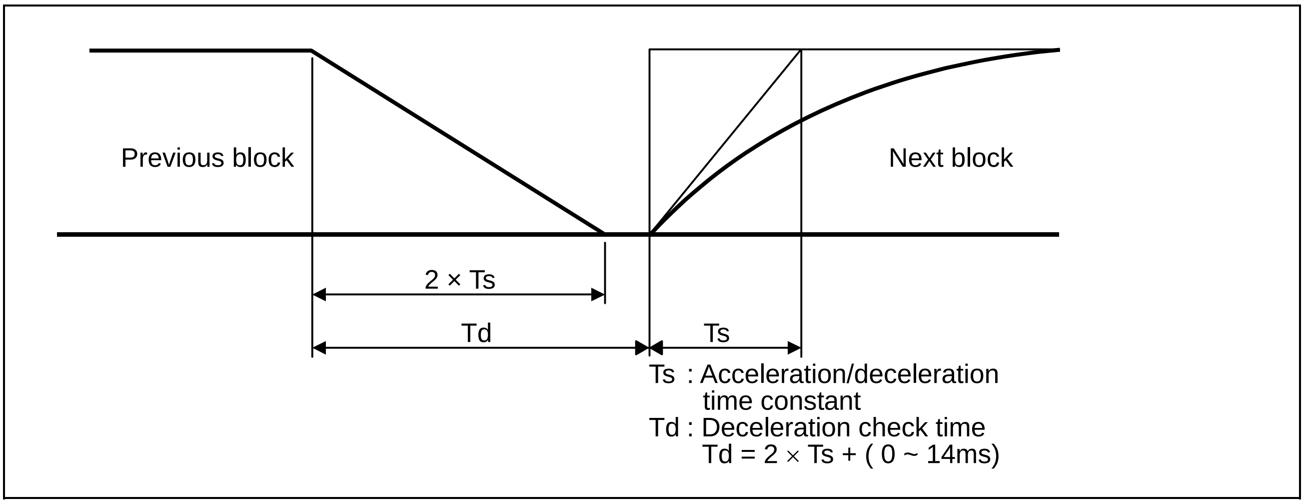
The purpose of checking the rapid traverse deceleration is to minimize the time it takes for positioning. The bigger the setting value for the servo parameter "#2224 SV024", the longer the reduced time is, but the remaining distance of the previous block at the starting time of the next block also becomes 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 SV024.

■ 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.

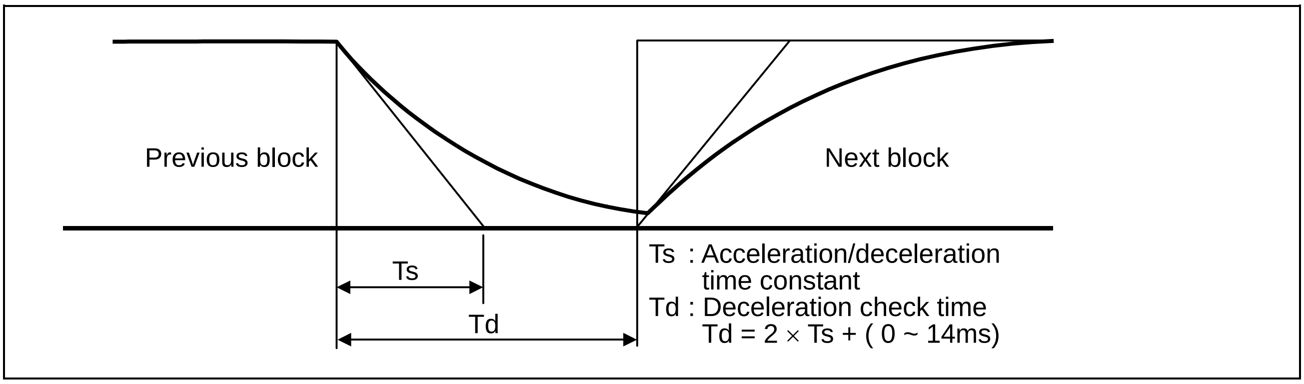
(a) Linear acceleration/linear deceleration Td = Ts + α



(b) Exponential acceleration/linear deceleration Td = 2 × Ts + α



(c) Exponential acceleration/exponential deceleration Td = 2 × Ts + α

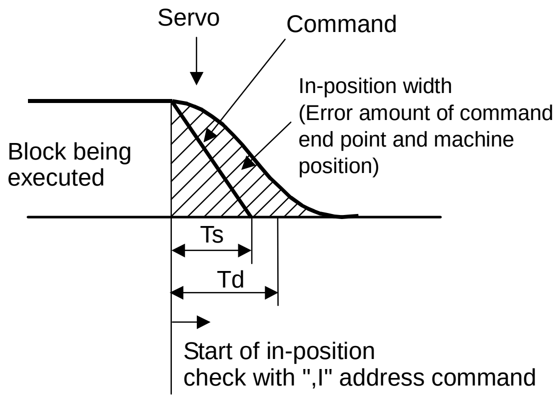
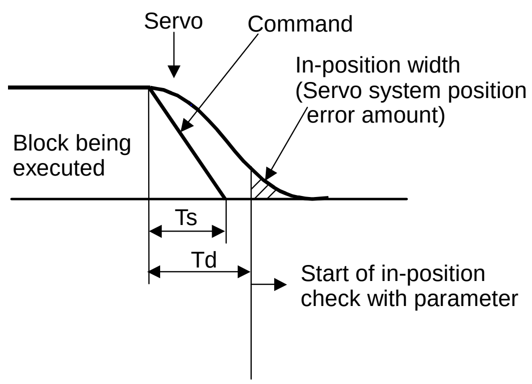


Where Ts is the acceleration time constant, α = 0 to 14ms
The time required for the deceleration check during rapid traverse is the longest among the rapid traverse deceleration check times of each axis determined by the rapid traverse acceleration/deceleration time constants and by the rapid traverse acceleration/deceleration mode of the axes commanded simultaneously.

Operation during in-position check

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 in each system 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.

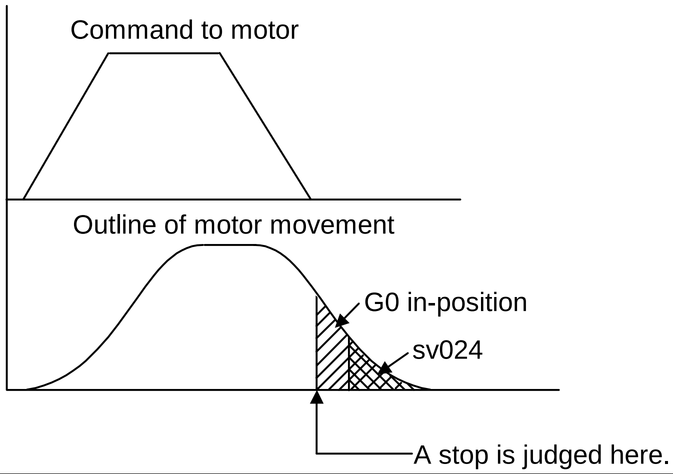
Differences between in-position check with this command and in-position check with parameter

<table>
<tr><th>In-position check with ",I" address command</th><th>In-position check with parameter</th></tr>
<tr><td>After starting deceleration of the command system, the position error amount and commanded in-position width are compared.</td><td>After starting deceleration of the command system, the servo system's position error amount and the parameter setting value (in-position width) are compared.</td></tr>
<tr><td>
</td><td>
</td></tr>
<tr><td colspan="2">Ts : Acceleration/deceleration time constant
Td : Deceleration check time
Td = Ts + (0 to 14ms)</td></tr>
</table>

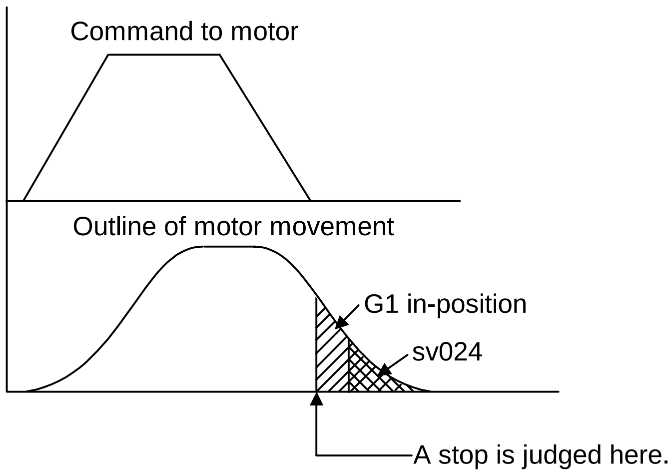
In-position width setting

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



In-position check using the “G1inps” value



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) When the in-position width check is carried out, the in-position width command in the program takes place the in-position width set with the parameters such as SV024, G0inps, or G1inps.

(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.

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

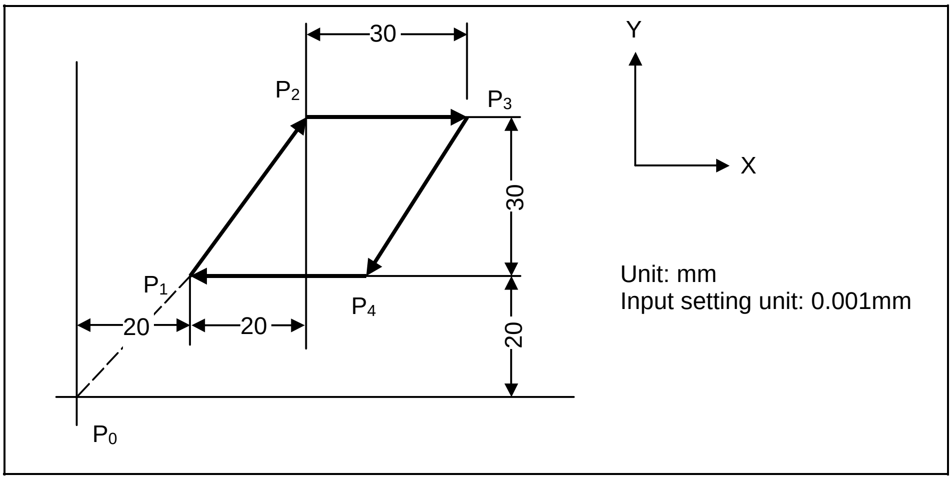
G01 Xx Yy Zz αα Ff ,Ii ; (α represents additional axis)	
x, y, z, α	:Coordinate values and may be an absolute position or incremental position depending on the G90/G91 state.
f	:Feedrate (mm/min or °/min)
i	:In-position width. A decimal point command will result in a program error. This is valid only in the commanded block. A block that does not contain this address will follow the parameter "#1193 inpos" settings. The range is 1 to 999999 (μm).

Detailed description

(1) 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.
(2) The feedrate for a rotary axis is commanded by °/min (decimal point position unit). (F300 = 300°/min)
(3) 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 $P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_1$ at 300 mm/min feedrate
$P_0 \rightarrow P_1$ is for tool positioning



G90	G00	X20000	Y20000 ;	$P_0 \rightarrow P_1$
	G01	X20000	Y30000 F300	$P_1 \rightarrow P_2$
		X30000	;	$P_2 \rightarrow P_3$
		X-20000	Y-30000 ;	$P_3 \rightarrow P_4$
		X-30000 ;		$P_4 \rightarrow P_1$

Programmable in-position width command for linear interpolation

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__ ;
```

- I__ : In-position width
- F__ : Feedrate
- X__ Y__ Z__ : Linear interpolation coordinate value of each axis

(Note 1) Refer to the 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

Table 1 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 Table 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 Table 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) I f 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 Table 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 Table 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 Table 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

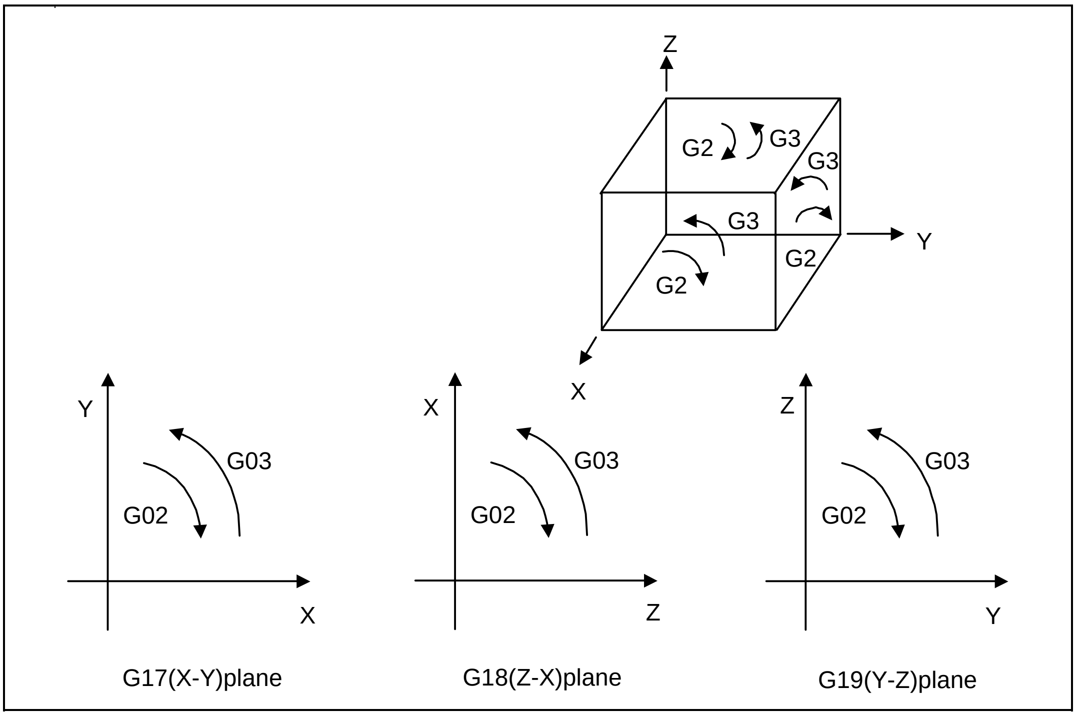
G02 (G03) Xx Yy Ii Jj Kk Ff;	
G02	: Clockwise (CW)
G03	: Counterclockwise (CCW)
Xx, Yy	: End point
Ii, Jj	: Arc center
Ff	: Feedrate

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.

Detailed description

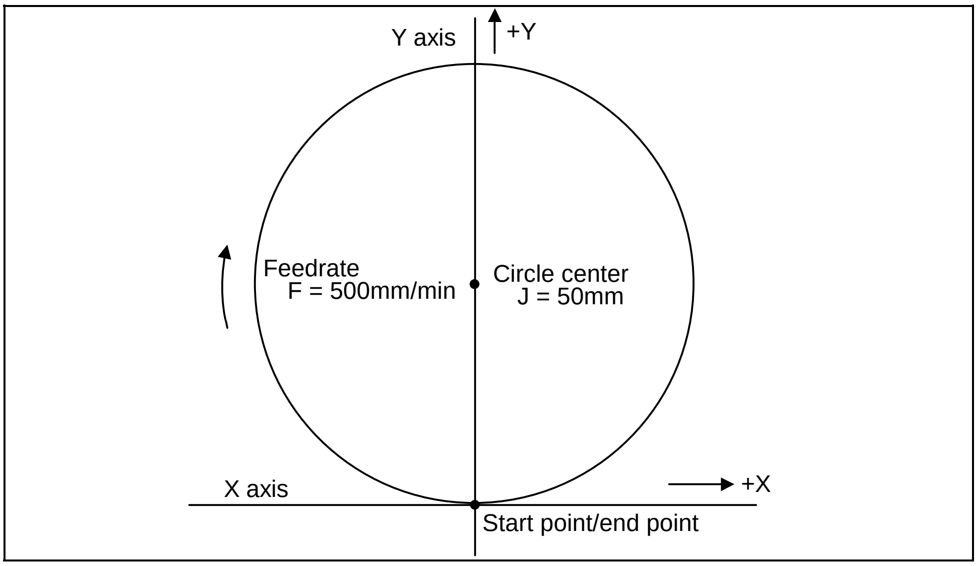
(1) G02 (or G03) is retained until another G command (G00, G01 or G33) in the 01 group that changes its mode is issued.
The arc rotation direction is distinguished by G02 and G03.
G02 Clockwise (CW)
G03 Counterclockwise (CCW)



(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

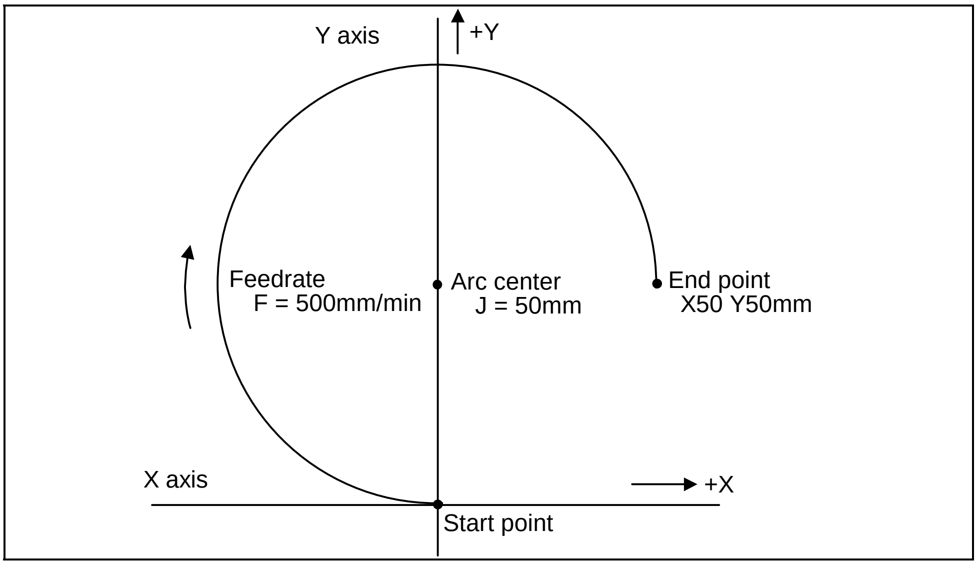
Example of program

(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)

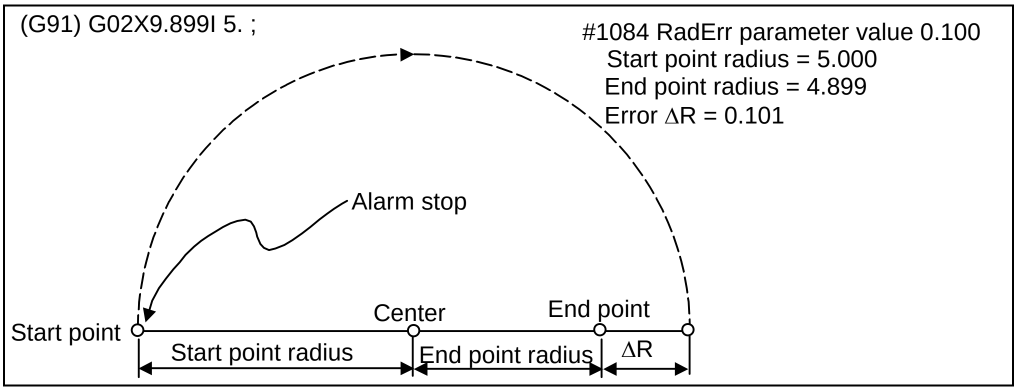
Precautions for circular interpolation

(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.

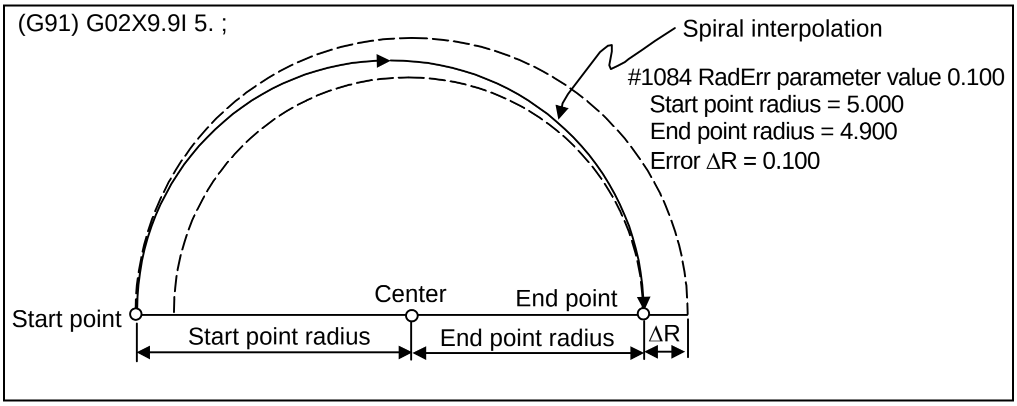
(2) When all the end point coordinates are omitted or when the end point is the same position as the start point, a 360° arc (full circle) is commanded when the center is commanded using I, J and K.

(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".



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



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.

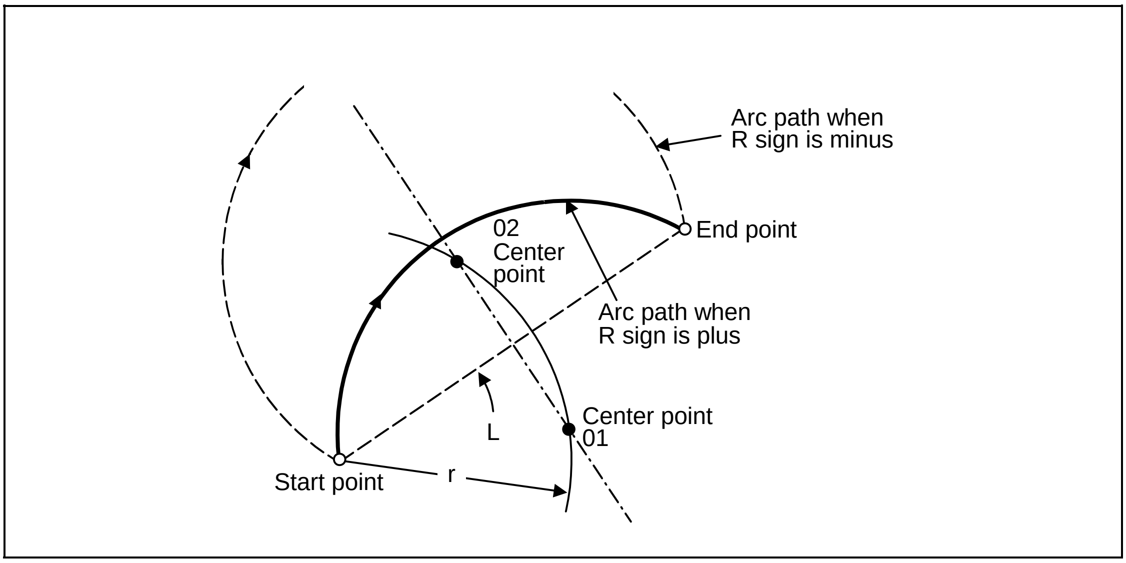
Command format

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) ≤ 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.

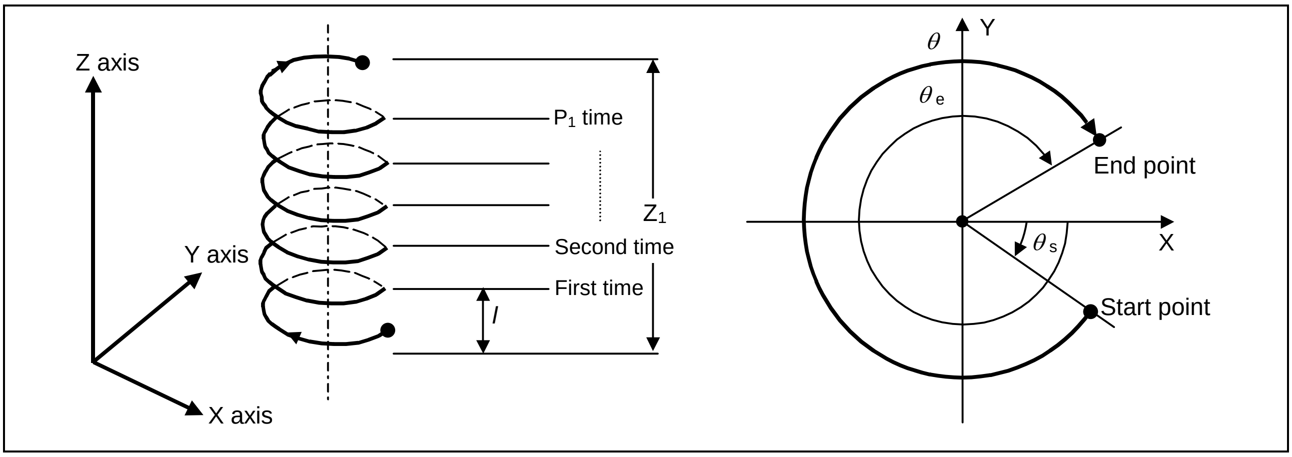
Command format

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.

Detailed description



(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.

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

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

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.

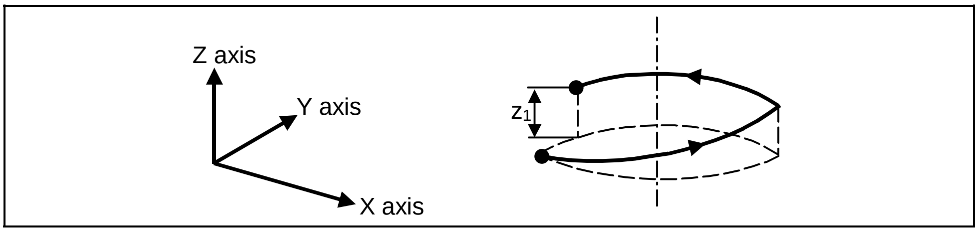
(5) Plane selection
The helical interpolation arc plane selection is determined with the plane selection mode and axis address as for the circular interpolation. For the helical interpolation command, the plane where circular interpolation is executed is commanded with the plane selection G code (G17, 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

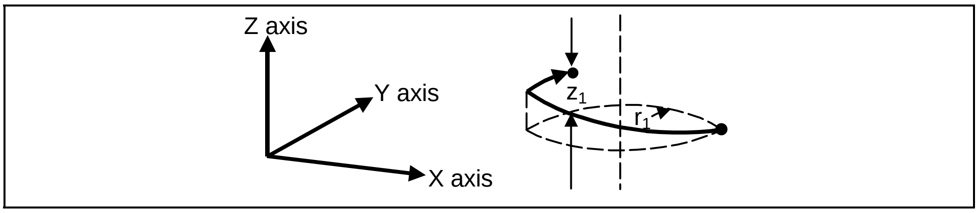
(Example 1)



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

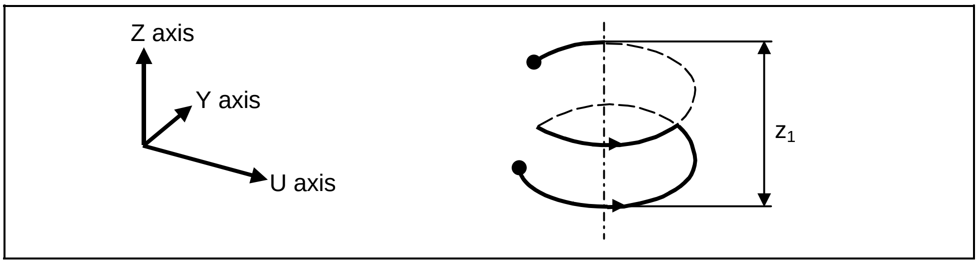
(Note) If pitch No. is 0, address P can be omitted.

(Example 2)



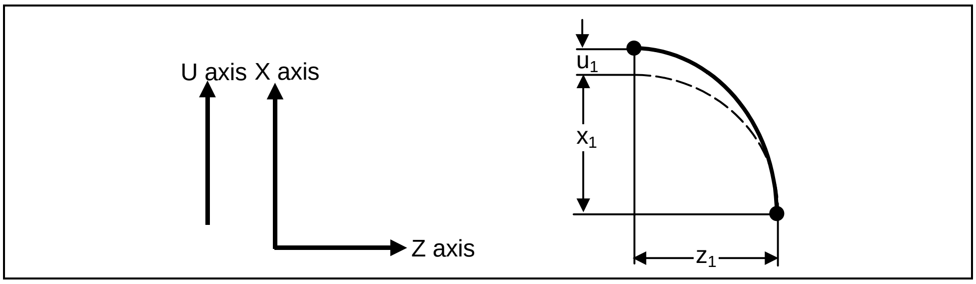
G17 ;	XY plane
G02 Xx1 Yy1 Zz1 Rr1 Ff1 ;	XY plane arc, Z axis linear

(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 Ii1 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.

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. Multiple thread screws, etc., can also be machined by designating the thread cutting angle.

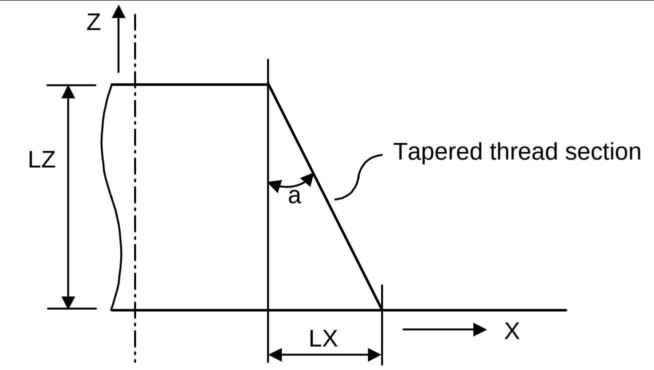
Command format

G32 Zz Ff$_1$ Qq ;	(Normal lead thread cutting commands)
Zz	: Thread cutting direction axis address (X, Y, Z, a) and thread length
Ff	: Lead of long axis (axis which moves most) direction.
Qq	: Thread cutting start shift angle, (0 to 360°)

G33 Zz Ee$_1$ Qq ;	(Precision lead thread cutting commands)
Zz	: Thread cutting direction axis address (X, Y, Z, α) and thread length
Ee	: Lead of long axis (axis which moves most) direction
Qq	: Thread cutting start shift angle, (0 to 360°)

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" 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

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	0.00001 to 99.9999	0.000001 to 99.99999	0.1 to 2559999.999

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.=1.000000)	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 39370078	0.0255 to 9999.9999	0.00001 to 3937009	0.000001 to 3937009	0.25401 to 999.9999

(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.

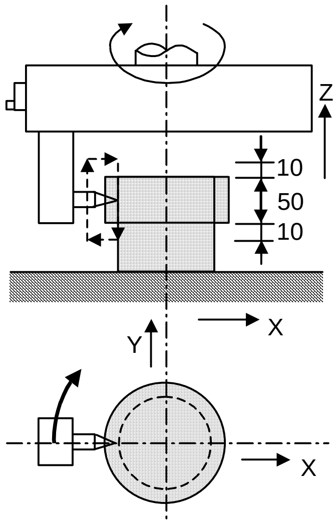
(3) The thread cutting will start by the one rotation synchronous signal from the encoder installed on the spindle.
(4) The spindle speed should be kept constant throughout from the rough cutting until the finishing.
(5) If the feed hold function is employed during thread cutting to stop the feed, the thread ridges will lose their shape. For this reason, feed hold does not function during thread cutting.
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 feed rate 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 to the required thread length.
(9) The spindle speed is subject to the following restriction :

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

Where R ≤ Permissible speed of encoder (r/min)
R : Spindle 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) The thread cutting start angle is designated with an integer or 0 to 360.

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.
N11 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, α) 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 to 360°.

Detailed description

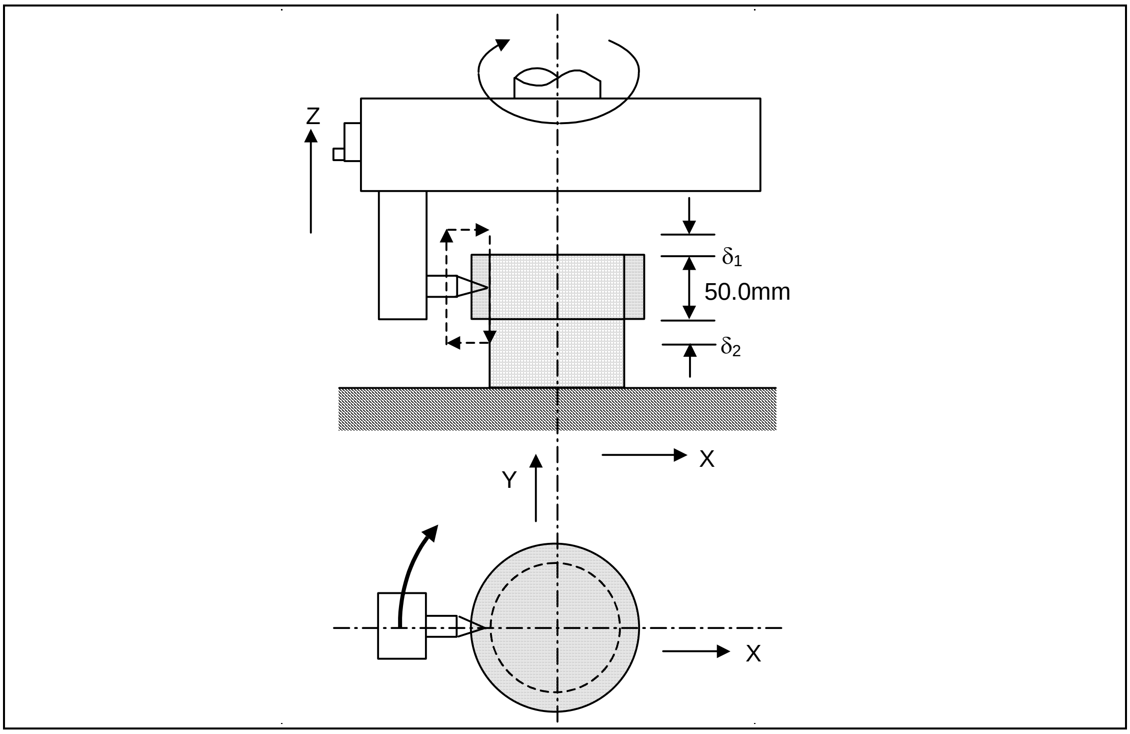
(1) The number of ridges in the long axis direction is assigned as the number of ridges per inch.

(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.

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

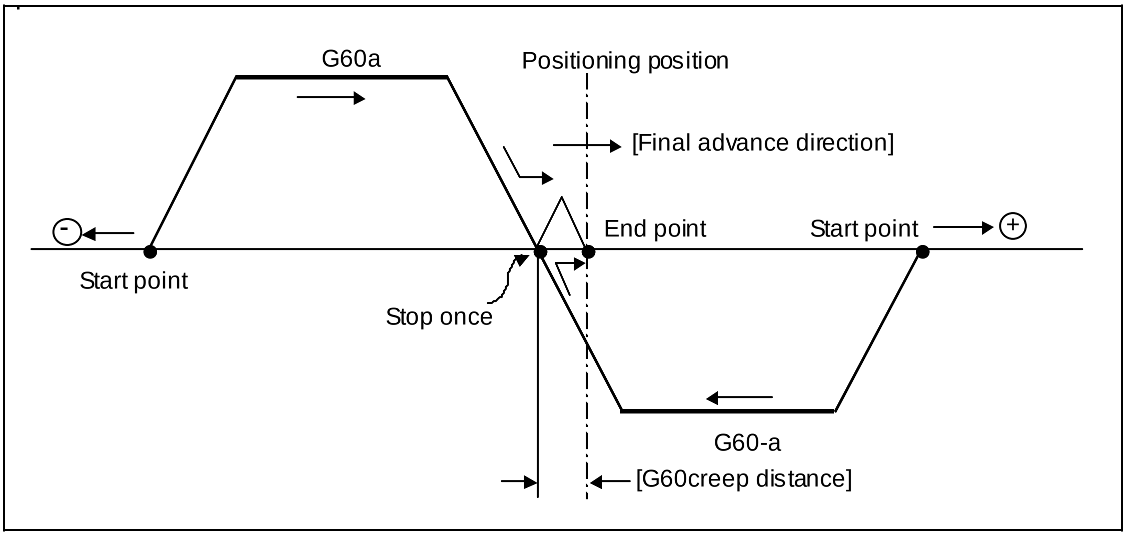
The G60 command can position the tool at a high degree of precision without backlash error by locating the final tool position from a single determined direction.

Command format

G60 Xx Yy Zz $\alpha\alpha$;
α : Additional 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.

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 1,000,000 mm/min for input setting units of 1μm. 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	Remarks
G1 X100. Y100. F200 ; G1 X100. Y100. F123.4 ; G1 X100. Y100. F56.789 ;	200.0mm/min 123.4mm/min 56.789mm/min	F200. or F200.000 gives the same rate.

Speed range that can be commanded (when input setting unit is 1μm or 10μm)

Command mode	F command range	Feed rate command range	Remarks
mm/min	0.001 to 1000000.000	0.001 to 1000000.000 mm/min	
inch/min	0.0001 to 39370.0787	0.0001 to 39370.0787 inch/min	
°/min	0.001 to 1000000.000	0.01 to 1000000 °/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 only in a G01, G02 or G03 modal.
The F1-digit command can also be used for fixed cycle.

Detailed description

Set the corresponding speed of F1 to F5 with the base specification parameters "#1185 spd_F1" to "#1189 spd_F5" respectively.
Operation alarm "104" will occur when the feedrate is 0.

(1) Operation method

(a) Make the F1-digit command valid. (Set the base specification parameter "#1079 F1digt" to 1.)

(b) Set F1 to F5. (Base specification parameter "1185 spd_F1" to "#1189 spd_F5")

(2) Special notes

(a) 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

(b) F1 to F5 are invalid in the G00 mode and the rapid traverse rate is established instead.

(c) If F0 is used in the G02 or G03 mode, a program error (P121) will result.

(d) 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.

(e) 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.

(f) 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.

(g) During a F1-digit command, the F1-digit number and F1-digit command signal are output as the PLC signals.

(3) 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.

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.

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 60,000mm/min (or 2,362 inch/min, 60,000 °/min) 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

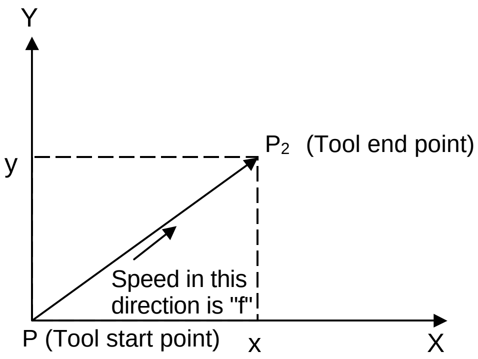
Function and purpose

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.

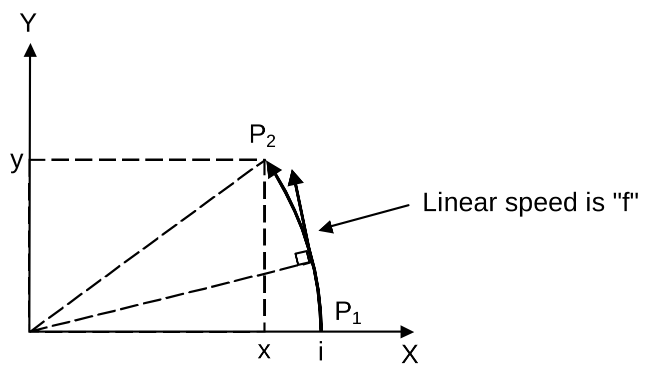


$$\text{Feedrate for X axis} = f \times \frac{x}{\sqrt{x^2 + y^2}}$$

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

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.

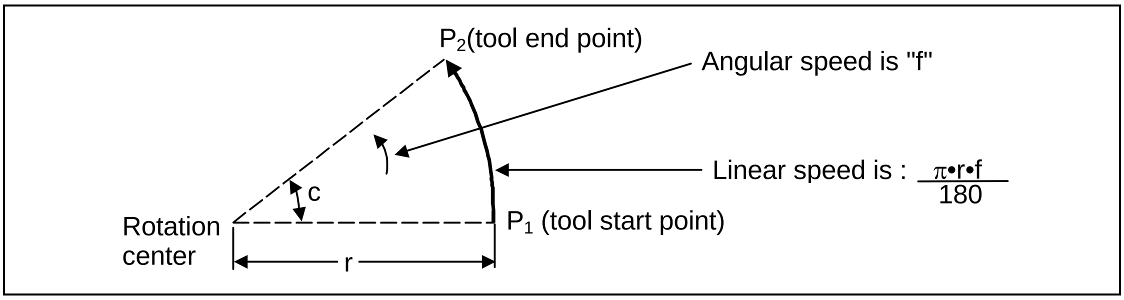


(Example) When the feedrate is designated as "f" and the linear axes (X and Y) are to be controlled using the circular interpolation function.
In this case, the feed rate of the X and Z axes will change along with the tool movement. However, the combined speed will always be maintained at the constant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



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

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

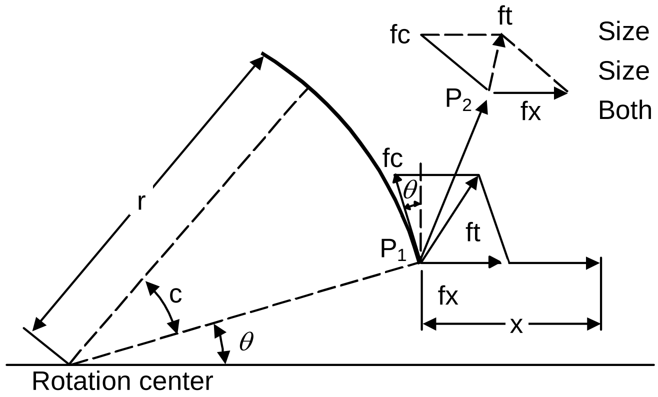
Therefore, the feedrate to be designated in the program must be :

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

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.

(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:

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

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

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

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

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:

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

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

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:

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

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

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

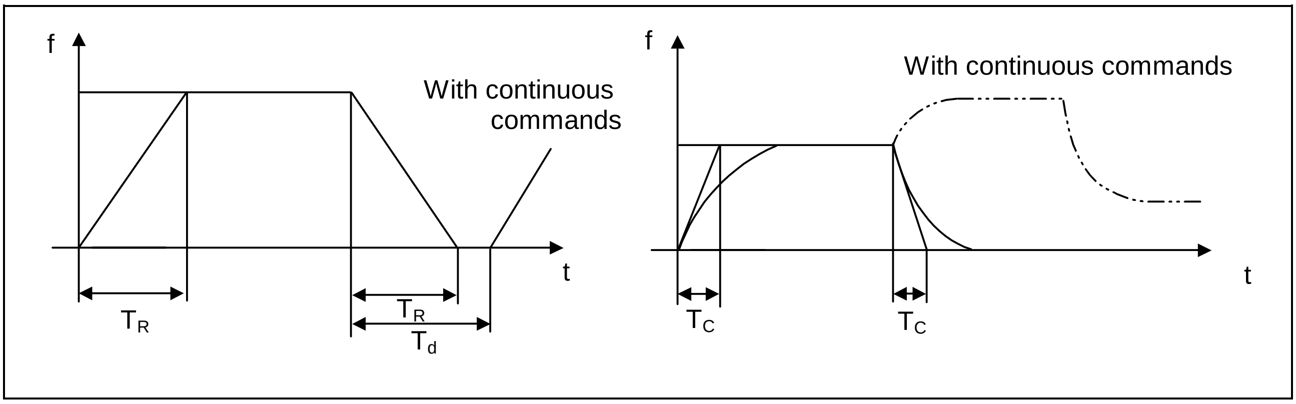
$$f = ft \times \frac{\sqrt{x^2 + c^2}}{\sqrt{x^2 - x \bullet c \bullet r\sin\left(\frac{\pi}{180}\theta\right)\frac{\pi}{90} + \left(\frac{\pi \bullet r \bullet c}{180}\right)^2}} \quad \cdots\cdots (7)$$

"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 Automatic acceleration/deceleration

Function and purpose

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



Rapid traverse acceleration/deceleration Pattern
(T_R = Rapid traverse time constant)
(T_d = Deceleration check time)

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

With rapid traverse and manual feed, the following block is executed after the command pulse of the present block has become "0" and the tracking error of the acceleration/deceleration circuit has become "0". However, with cutting feed, the following block is executed as soon as the command pulse of the present block becomes "0" although an external signal (error detect) can detect that the tracking error of the acceleration/deceleration circuit has reached "0" and the following block can be executed. When the in-position check has been made valid (selected by parameter "#1193 inpos") during the deceleration check, it is first confirmed that the tracking error of the acceleration/deceleration circuit has reached "0", then it is checked that the position deviation is less than the parameter setting value "#2204 SV024", and finally the following block is executed. It depends on the machine as to whether the error detect function can be activated by a switch or M function and so reference should be made to the instructions issued by the machine maker.

7.7 Speed clamp

Function and purpose

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

(Note) Speed clamping is not applied to synchronous feed and thread cutting.

7.8 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.
Either the deceleration check time or in-position state is selected with parameter "#1193 inpos". In-position check is valid when "#1193 inpos" is set to 1.
The in-position width is set with parameter "#2224 sv024" on the servo parameter screen by the machine manufacturer.

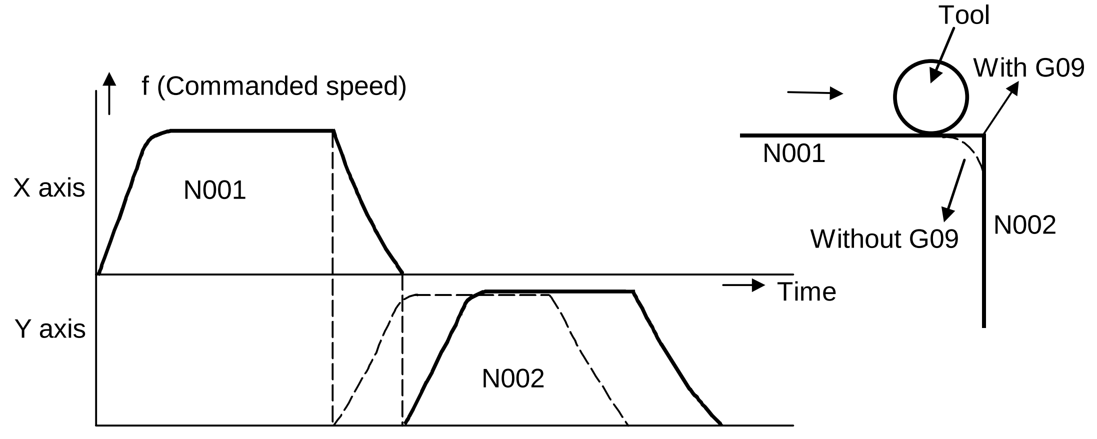
Command format

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

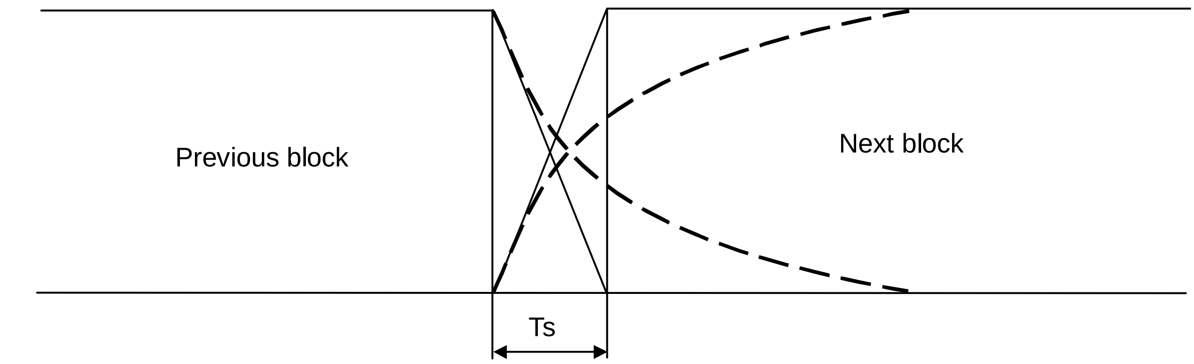


Fig. 2 Continuous cutting feed command

(2) With cutting feed in-position check

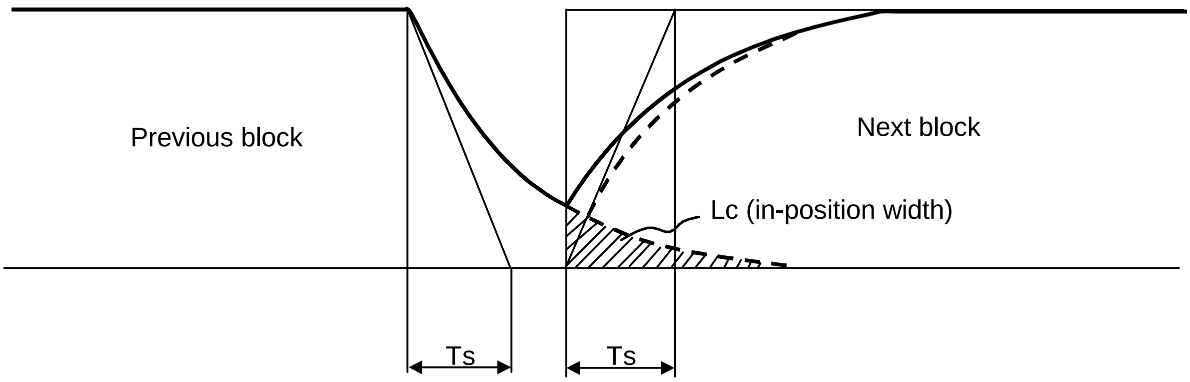
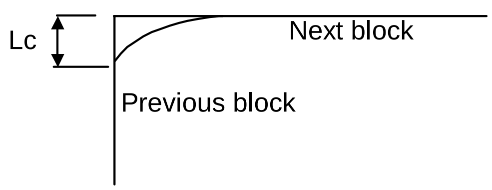


Fig. 3 Block joint with cutting feed in-position check

In Figs. 2 and 3:

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

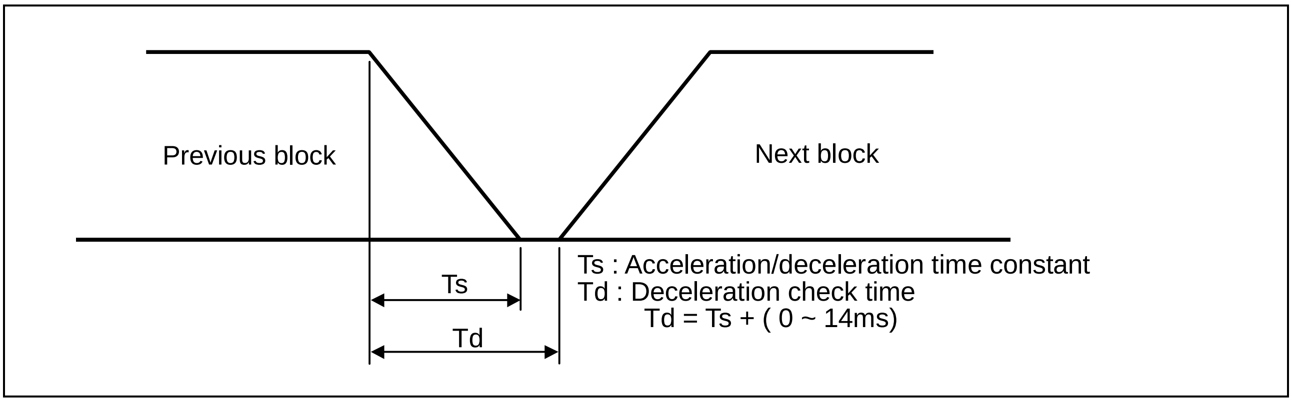
As shown in Fig. 3, the in-position width "Lc" can be set into the servo parameter "#2224 SV024" as the remaining distance (shaded area in Fig. 3) of the previous block when the next block is started.
The in-position width is designed to reduce the roundness at the workpiece corners to below the constant value.



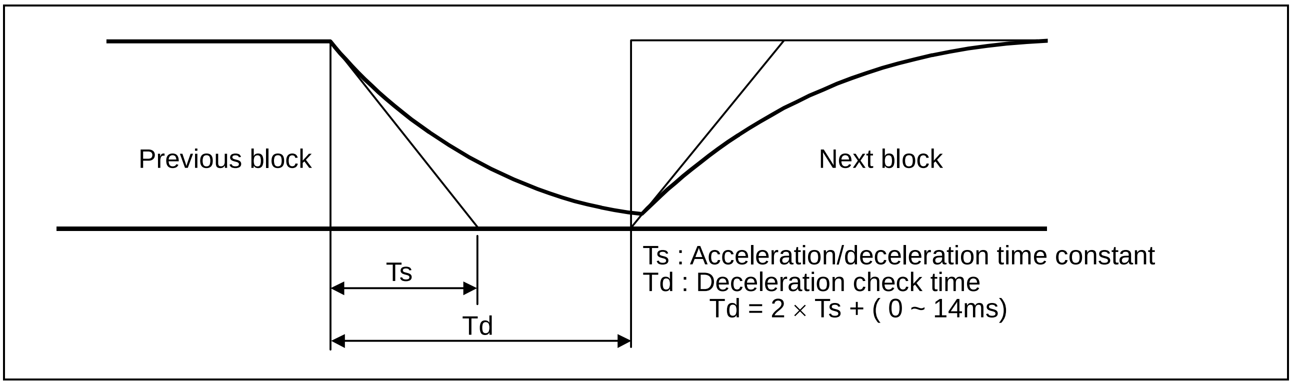
To eliminate corner roundness, set the servo parameter "#2224 SV024" to zero and perform an in-position check or assign the dwell command (G04) between blocks.

(3) **With deceleration check**

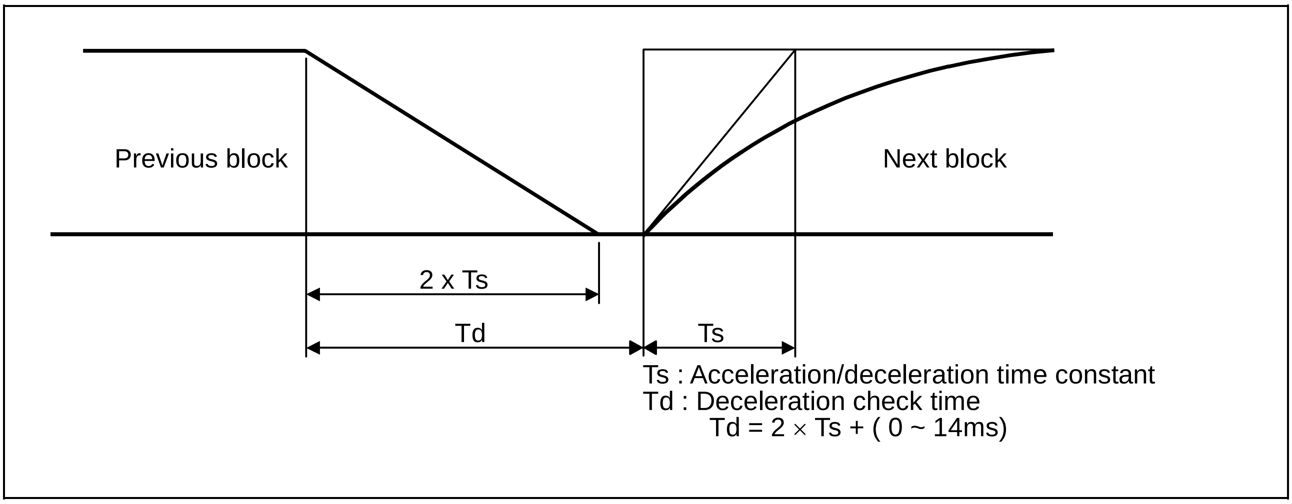
(a) **With linear acceleration/deceleration**



(b) **With exponential acceleration/deceleration**



(c) **With exponential acceleration/linear deceleration**



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

(Note 1) To execute exact stop check in a fixed cycle cutting block, insert command G09 into the fixed cycle subprogram.

7.9 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.

7.10 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), or 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.

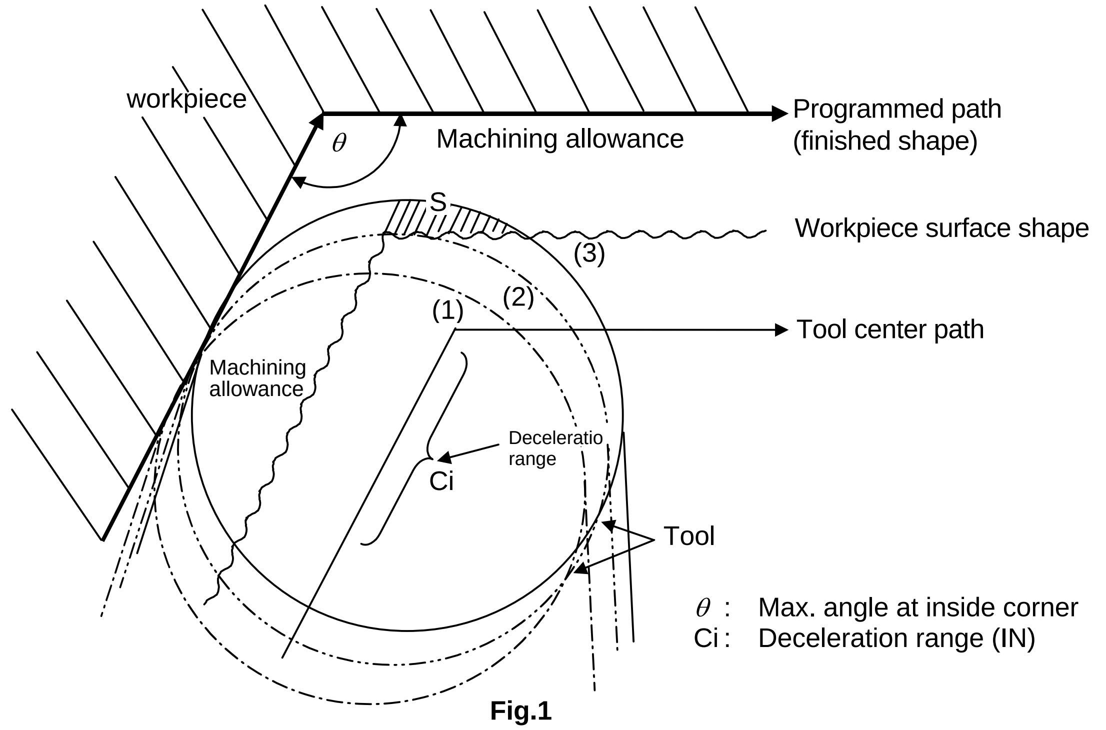


Fig.1

(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.

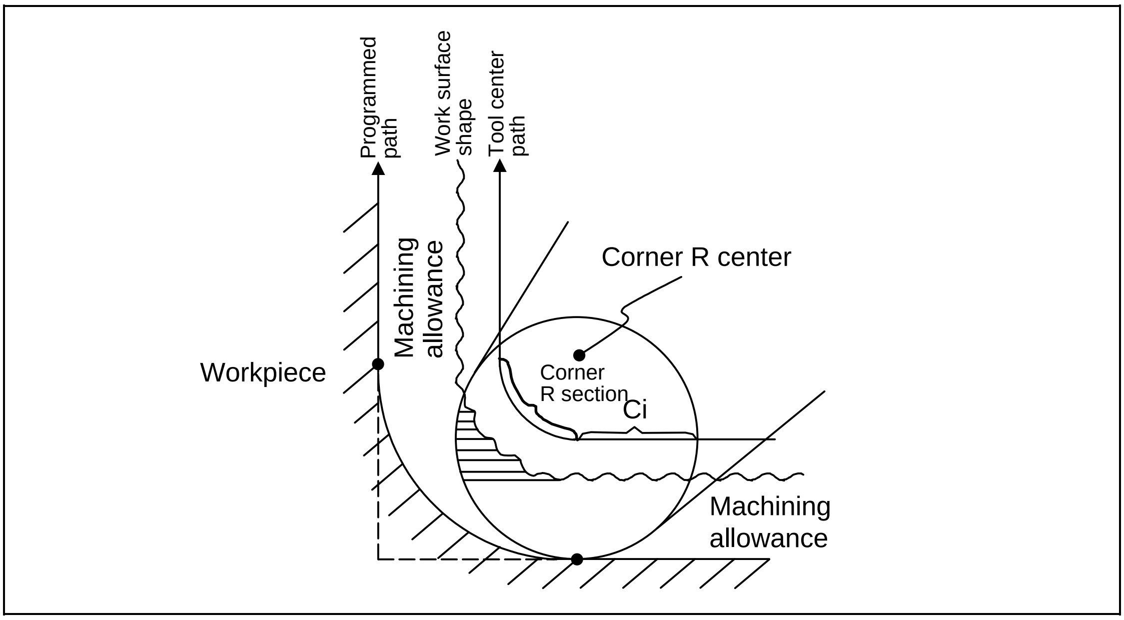
(2) Parameter setting

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.

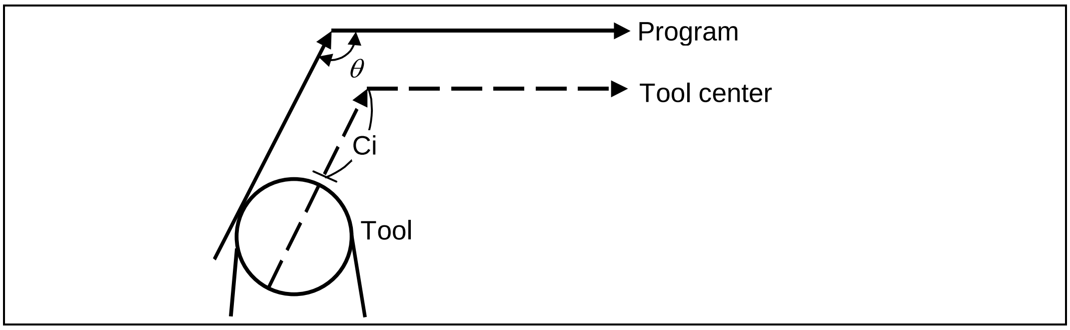
Automatic corner R



(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.)

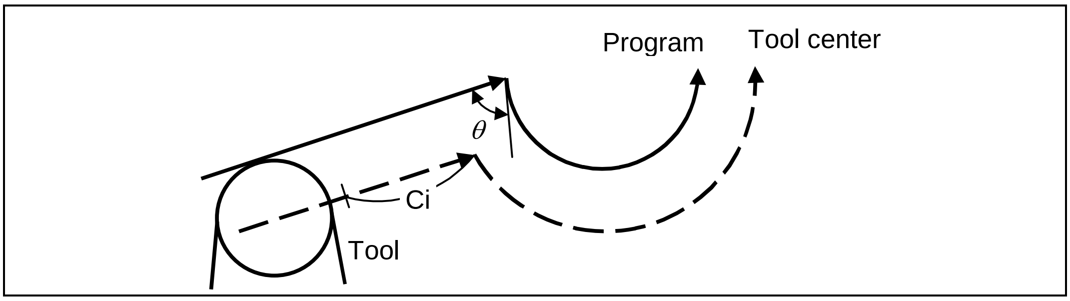
Application example

(1) Line – line corner



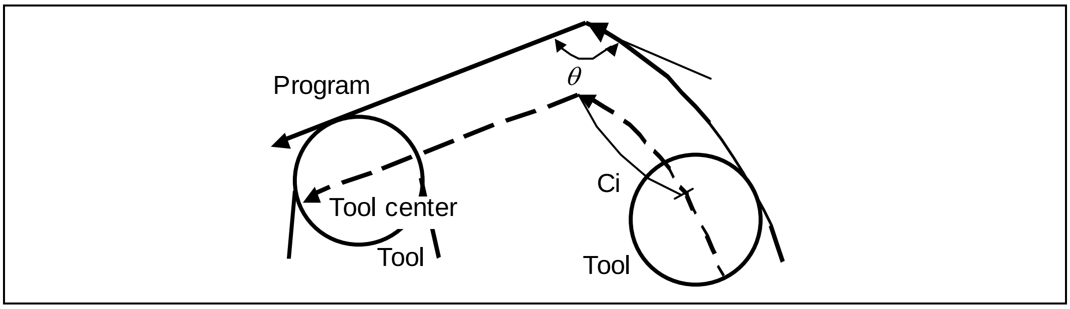
The override set in the parameter is applied at Ci.

(2) Line – arc (outside) corner



The override set in the parameter is applied at Ci.

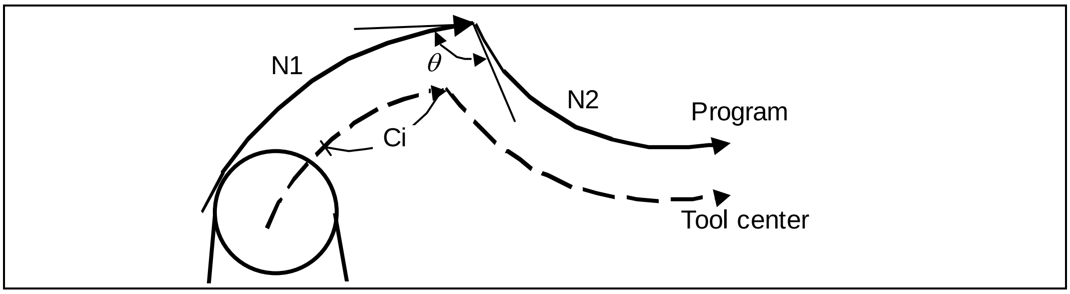
(3) Arc (inside offset) – line corner



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

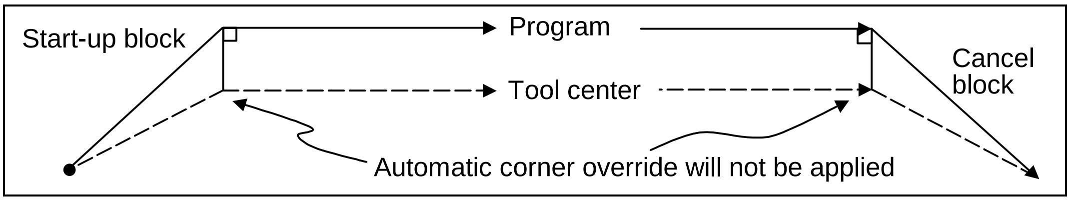
Function	Override at corner
Cutting feed override	Automatic corner override is applied after cutting feed override has been applied.
Override cancel	Automatic corner override is not canceled by override cancel.
Speed clamp	Valid after automatic corner override
Dry run	Automatic corner override is invalid.
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

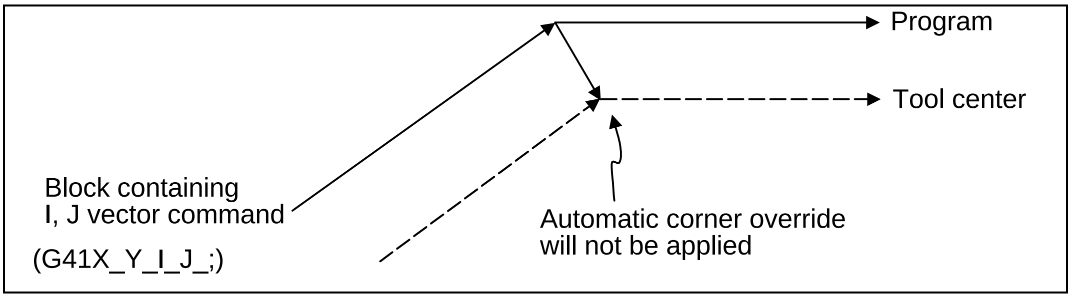
(1) Automatic corner override is valid only in the G01, G02, and G03 modes; it is not effective in the G00 mode. When switching from the G00 mode to the G01 (or G02 or G03) mode at a corner (or vice versa), automatic corner override will not be applied at that corner in the G00 block.

(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 radius compensation is started or canceled.



(4) Automatic corner override will not be applied on a corner where the tool radius compensation I, J 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.

7.11 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 mode (G61.1), automatic corner override (G62),or cutting mode (G64) command.

Command format

G63 ;

7.12 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 check mode (G61), high-accuracy control mode (G61.1), automatic corner override (G62), or tapping mode (G63) command.
This cutting mode is established in the initialized status.

Command format

G64 ;

8. Dwell

The G04 command can delay the start of the next block. The dwell remaining time can be canceled by adding the multi-step skip function.

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) The dwell time command range is as follows.
0.001 ~ 99999.999 (s)

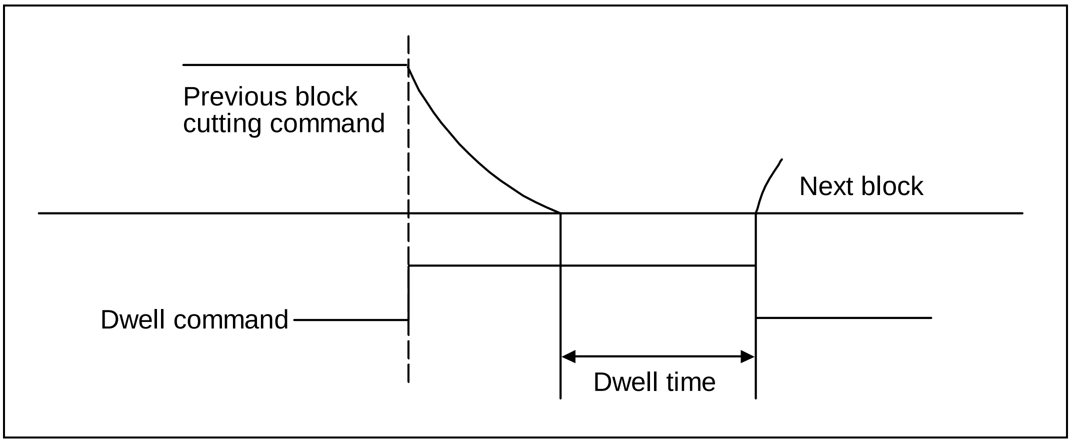
(3) The dwell time setting unit applied when there is no decimal point can be made 1s by setting 1 in the parameter "#1078 Decpt2". This is effect only for X and P for which the decimal command is valid.

(4) 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.

(5) The dwell is valid during the interlock.

(6) The dwell is valid even for the machine lock.

(7) 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.



Example of program

Command	Dwell time [sec]	
	#1078 Decpt2 = 0	**#1078 Decpt2 = 1**
G04 X500 ;	0.5	500
G04 X5000 ;	5	5000
G04 X5. ;	5	5
G04 X#100 ;	1000	1000
G04 P5000 ;	5	5000
G04 P12.345 ;	12.345	12.345
G04 P#100 ;	1000	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) If the input setting unit is 0.0001inch, the X before G04 will be multiplied by 10. For example for "X5. G04 ;", the dwell time will be 50 sec.

Precautions

(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 eight 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, and M30 functions are used, the next block is not read into the pre-read buffer due to pre-read inhibiting.
An M function can be specified together with other commands in the same block, and when such a function is specified together with a movement command in the same block, there are two possible sequences in which the commands are executed. 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 8 M functions used for specific purposes will now be described.

Program stop : M00

When the NC has read this function, it stops reading the next block. 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.

Optional stop : M01

If the M01 command is read when the optional stop switch on the machine operation board is ON, reading of the next block will stop and the same effect as with the M00 function will apply.

(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.

(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.

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 a miscellaneous 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. Indexing operation other than M02/M03 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 maker determines which codes correspond to which positions.
When the A, B and C functions are commanded in the same block as movement commands, there are 2 sequences in which the commands are executed, as below. 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.

Additional axis name / **Secondary miscellaneous function**	**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
(2) Geometric commands

10. Spindle Functions

10.1 Spindle functions (S2-digits BCD) ..... During standard PLC specifications

Function and purpose

The spindle functions are also known simply as S functions and they assign the spindle rotation speed. In this controller, they are assigned with a 2-digit number following the S code ranging from 0 to 99, and 100 commands can be designated. In actual fact, however, it depends on the machine specifications as to how many of these 100 functions are used and which numbers correspond to which functions, and thus reference should be made to the instruction issued by the machine manufacturer. When a number exceeding 2 digits is assigned, the last 2 digits will be valid.
When S functions are commanded in the same block as movement commands, there are 2 sequences in which the commands are executed, as below. 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.

Processing and completion sequences are required for all S commands from S00 to S99.

10.2 Spindle functions (S6-digits Analog)

Function and purpose

When the S6-digits function is added, commands with a 6-dight number following the S code can be designated. Other commands conform to the S2-digits function.
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 rotation speed, maximum rotation speed, shift rotation speed and tapping rotation speed
(2) Parameters corresponding to all gears .... Orientation rotation speed, minimum rotation speed

10.3 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.

10.4 Multiple spindle control I

10.4.1 Multiple spindle control

Function and purpose

Spindle rotation command for up to 7 spindles is provided.
Although the S∗∗∗∗∗ command is normally used to designate the spindle rotation speed, the Sn=∗∗∗∗∗ command is also used for multiple spindle control.
S commands can be issued from the machining program of any part systems.
Number of usable spindles differ the machine model, confirm the specifications of the model used.

Command format

Sn=∗∗∗∗∗ ;	S6-digit binary data.
n	Designate the spindle number with one numeric character.
∗∗∗∗∗	Rotation speed or constant surface speed command value.

Detailed description

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

(Example)

S1 = 3500 ; 1st spindle 3500(r/min) command
S2 = 1500 : 2nd spindle 1500(r/min) command
S3 = 2000 ; 3rd spindle 2000(r/min) command
S4 = 2500 : 4th spindle 2500(r/min) command
S5 = 2000 ; 5th spindle 2000(r/min) command
S6 = 3000 : 6th spindle 3000(r/min) command
S7 = 3500 ; 7th spindle 3500(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 Sn=∗∗∗∗∗ command can be used together.
The spindle targeted for the S∗∗∗∗∗ command is normally the 1st spindle, however, the S∗∗∗∗∗ command can be used for 2nd or following spindle according to the spindle selection command.

(5) The commands for each spindle can be commanded from the machining program of any part systems.
The spindles will rotate with the speed commanded last.
If the S commands are issued from two or more part systems, the command from the part system of largest No. will be valid.

(6) As for C6 T-type and L-type, C64 T-type, and C64T T-type, the multiple spindles control can not be used in a part system. A program error (P33) will occur when the Sn=∗∗∗∗∗ command is issued. Refer to "10.4.2 Spindle selection command" for details.

10.4.2 Spindle selection command

Function and purpose

This function controls which spindle's rotation the cutting follows, in addition, designates the spindle to be selected when "S∗∗∗∗∗∗" command is issued.

Command format

G43.1;	**Selected spindle (nth spindle) control mode ON (Selected with parameter)**
G44.1;	**2nd spindle control mode ON**

Detailed description

(1) G43.1 and G44.1 are modal G codes.

(2) The spindle control mode entered when the power is turned ON or reset depends on the parameter setting.
Designate the spindle No. to be selected in G43.1 modal with the parameter (basic specifications parameter "#1199 Sselect").
This parameter is provided for every part system to set as follows.

#	Items		Details	Setting range (unit)
1199	Sselect	Select initial spindle control	Select the initial condition of spindle control when power is turned ON or reset.	0: Selected spindle control mode (G43.1) 1: 2nd spindle control mode (G44.1)
21049	SPname		Designate the spindle No. selected for the G43.1 modal in each part system.	0: 1st spindle 4: 4th spindle 1: 1st spindle 5: 5th spindle 2: 2nd spindle 6: 6th spindle 3: 3rd spindle 7: 7th spindle

Reset the NC after changing "#1199 Sselect " and "#21049 SPname" parameters. It is no use to turn the power OFF once and ON again.

(3) As for C6 L-type, T-type, C64 T-type and C64T T-type, there are following restrictions;
· A program error (P34) will occur if G44.1 command is issued.
· No data can be set to "#1199 Sselect". "0" is set when the NC power is turned ON.
· Only one spindle than is selected with "#21049 SPname" can be commanded as "S∗∗∗∗∗" in each part system.
· A program error (P33) will occur if the "S0=∗∗∗∗∗" command is issued.

(4) If the S command is issued in the same as the spindle selection commands (G43.1, and G44.1), which spindle the S command is valid for depends on the order that G43.1, G44.1, and S command are issued.
When S command precedes the G codes, it follows the G43.1 / G44.1 mode before S command is issued.
When G codes precede, it follows the G43.1 / G44.1 mode issued in the same block.

(5) G43.1 and G44.1 commands can be issued from every part system.

Relation with other functions

(1) The following functions change after the spindle selection command.

(a) Per rotation command (synchronous feed)
Even if F is commanded in the G95 mode, the per rotation feedrate for the selected spindle (nth spindle) will be applied during G43.1 mode and for the 2nd spindle during G44.1 mode.

(b) S commands (S∗∗∗∗∗, Sn=∗∗∗∗∗), constant surface speed control, thread cutting

Function	G43.1 mode	G44.1 mode
S command during G97/G96 constant surface speed control Upper limit / Lower limit of spindle rotation speed command during constant surface speed control (G92 S_ Q) Thread cutting	Command control for the selected spindle (nth spindle). (Note 1)	Command control for the 2nd spindle.

(Note 1) The spindle selected during G43.1 mode depends on the parameter "#21049 SPname".

(2) The Sn=∗∗∗∗∗ command can be used to command the other spindle even if it is commanded during G43.1 or G44.1 mode.
Note that the rotation speed designation will be applied for such command even if the G96 mode is ON.

(Example) When "SPname" = 0;

	Rotation speed	
G43.1;	1st spindle	2nd spindle
G97 S1000; :	1000(r/min)	0(r/min)
S2 = 2000; :		2000(r/min)
G96 S100; :	100(m/min)	
S2 = 2500; :		2500(r/min)
G44.1 S200; :	(Note 2)	200(m/min)
S1 = 3000; :	3000(r/min)	
G97 S4000; :		4000(r/min)

(Note 2) The constant surface speed control will be switched to the 2nd spindle by G44.1 command. Therefore, the 1st spindle retains its rotation speed as that of "G44.1 S200;" command.
The 1st spindle rotation speed will be 3000 (r/min) when "S1=3000;" command is issued.

10.5 Constant surface speed control; G96, G97

10.5.1 Constant surface speed control

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	: Surface speed (1 to 99999999 m/min)
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 ; ≀	The spindle speed is controlled so that the peripheral speed is 200m/min.
G97 G01 X50. Z100. F300 S500 ; ≀	The spindle speed is controlled to 500r/min.
M02 ;	The modal returns to the initial setting.

(4) Constant surface speed control can be commanded on the selected spindle (nth spindle) / the 2nd spindle.
Select which spindle (the selected spindle or 2nd one) the commands are made to by the spindle selection G codes (G43.1 and G44.1).
Select which spindle (the selected spindle or 2nd one) is valid as the initial state with the parameter (base specifications parameter "#1199 Sselect").

(5) Select whether calculating the surface speed at rapid traverse command is performed constantly or only at the block end poing.

10.6 Spindle clamp speed setting; G92

Function and purpose

The maximum clamp speed of the spindle can be assigned by address S following G92 and the minimum clamp speed by address Q.

Command format

G92 Ss Qq;	
Ss	: Maximum clamp speed
Qq	: Minimum clamp speed

Detailed description

(1) Besides this command, parameters can be used to set the rotational speed range up to 4 stages in 1 r/min units to accommodate gear selection between the spindle and spindle motor. The lowest upper limit and highest lower limit are valid among the rotational speed ranges based on the parameters and based on G92 Ss Qq ;

(2) Set in the parameters "#1146 Sclamp" and "#1227 aux11/bit5" whether to carry out rotation speed clamp only in the constant surface speed mode or even when the constant surface speed is canceled.

(Note) G92S command and speed clamp operation

<table>
<tr><th colspan="2" rowspan="2"></th><th colspan="2">Sclamp = 0</th><th colspan="2">Sclamp = 1</th></tr>
<tr><th>aux11/bit5 = 0</th><th>aux11/bit5 = 1</th><th>aux11/bit5 = 0</th><th>aux11/bit5 = 1</th></tr>
<tr><td rowspan="2">Command</td><td>In G96</td><td colspan="2">Rotation speed clamp command</td><td>Rotation speed clamp command</td><td>Rotation speed clamp command</td></tr>
<tr><td>In G97</td><td colspan="2">Spindle rotation speed command</td><td>Rotation speed clamp command</td><td>Rotation speed clamp command</td></tr>
<tr><td rowspan="2">Operation</td><td>In G96</td><td colspan="2">Rotation speed clamp execution</td><td>Rotation speed clamp execution</td><td>Rotation speed clamp execution</td></tr>
<tr><td>In G97</td><td colspan="2">No rotation speed clamp</td><td>Rotation speed clamp execution</td><td>No rotation speed clamp</td></tr>
</table>

10.7 Spindle synchronous control I; G114.1

Function and purpose

In a machine having two or more 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".
With the spindle synchronous control function I, designation of spindles and controls start / stop of synchronization are commanded using G codes in the machining program.

Command format

(1) Spindle synchronous control ON (G114.1)

This command designates the basic spindle and synchronous spindle, and synchronizes the two designated spindles. By commanding the synchronous spindle phase shift amount, the phases of the basic spindle and synchronous spindle can be aligned.

G114.1 H_ D_ R_ A_ ;	
H_	Basic spindle selection
D_	Synchronous spindle selection
R_	Spindle synchronization phase shift amount
A_	Spindle synchronization acceleration/deceleration time constant

(2) Spindle synchronous control cancel (G113)

This command cancels the synchronous state of the two spindles rotating in synchronization with the spindle synchronous command.

G113 ;

Add-ress	Meaning of address	Command range (unit)	Remarks
H	Basic spindle selection Select the No. of the spindle to be used as the basic spindle from the two spindles.	1 to 7 1: 1st spindle 2: 2nd spindle : 7: 7th spindle	• A program error (P35) will occur if a value exceeding the command range or spindle No. without specifications is commanded. • A program error (P33) will occur if there is no command. • A program error (P610) will occur if a spindle not serially connected is commanded.

Add-ress	Meaning of address	Command range (unit)	Remarks
D	Synchronous spindle selection Select the No. of the spindle to be synchronized with the basic spindle from the two spindles.	1 to 7 or –1 to –7 1: 1st spindle 2: 2nd spindle : 7: 7th spindle	• A program error (P35) will occur if a value exceeding the command range or spindle No. without specifications is commanded. • A program error (P33) will occur if there is no command. • A program error (P33) will occur if the same spindle as that commanded for the basic spindle selection is designated. • The rotation direction of the synchronous spindle in respect to the basic spindle is commanded with the D sign. • A program error (P610) will occur if a spindle not serially connected is commanded.
R	Synchronous spindle phase shift amount Command the shift amount from the Z-phase point (one rotation signal) of the synchronous spindle.	0 to 359.999 (°) or 0 to 35999 (° × 10^{-3})	• A program error (P35) will occur if a value exceeding the command range is commanded. • The commanded shift amount is effective in the clockwise direction of the basic spindle. • The commanded shift amount's minimum resolution is as follows: For semi-closed (Only gear ratio 1:1) 360/4096 (°) For full closed (360/4096) ∗ K (°) K: Spindle and encoder gear ratio • If there is no R command, the phases will not be aligned.
A	Spindle synchronization acceleration/decele-ration time constant Command the acceleration/decele-ration time constant for when the spindle synchronous command rotation speed changes. (Command this to accelerate or decelerate at a speed slower than the time constant set in the parameters.)	0.001 to 9.999 (s) or 1 to 9999 (ms)	• A program error (P35) will occur if a value exceeding the command range is commanded. • If the commanded value is smaller than the acceleration/deceleration time constant set with the parameters, the value set in the parameters will be applied.

Rotation and rotation direction

(1) The rotation speed and rotation direction of the basic spindle and synchronous spindle during spindle synchronous control are the rotation speed and rotation direction commanded for the basic spindle. Note that the rotation direction of the synchronous spindle can be reversed from the basic spindle through the program.
(2) The basic spindle's rotation speed and rotation direction can be changed during spindle synchronous control.
(3) The synchronous spindle's rotation command is also valid during spindle synchronous control. When spindle synchronous control is commanded, if neither a forward run command nor reverse run command is commanded for the synchronous spindle, the synchronization standby state will be entered without starting the synchronous spindle's rotation. If the forward run command or reverse run command is input in this state, the synchronous spindle will start rotation. The synchronous spindle's rotation direction will follow the direction commanded in the program.
If spindle stop is commanded for the synchronous spindle during spindle synchronization control (when both the forward run and reverse run commands are turned OFF), the synchronous spindle rotation will stop.
(4) The rotation speed command (S command) and constant surface speed control are invalid for the synchronous spindle during spindle synchronous control. Note that the modal is updated, so these will be validated when the spindle synchronization is canceled.
(5) The constant surface speed can be controlled by issuing a command to the basic spindle even during spindle synchronous control.

Rotation synchronization

(1) When rotation synchronization control (command with no R address) is commanded with the G114.1 command, the synchronous spindle rotating at a random rotation speed will accelerate or decelerate to the rotation speed commanded beforehand for the basic spindle, and will enter the rotation synchronization state.

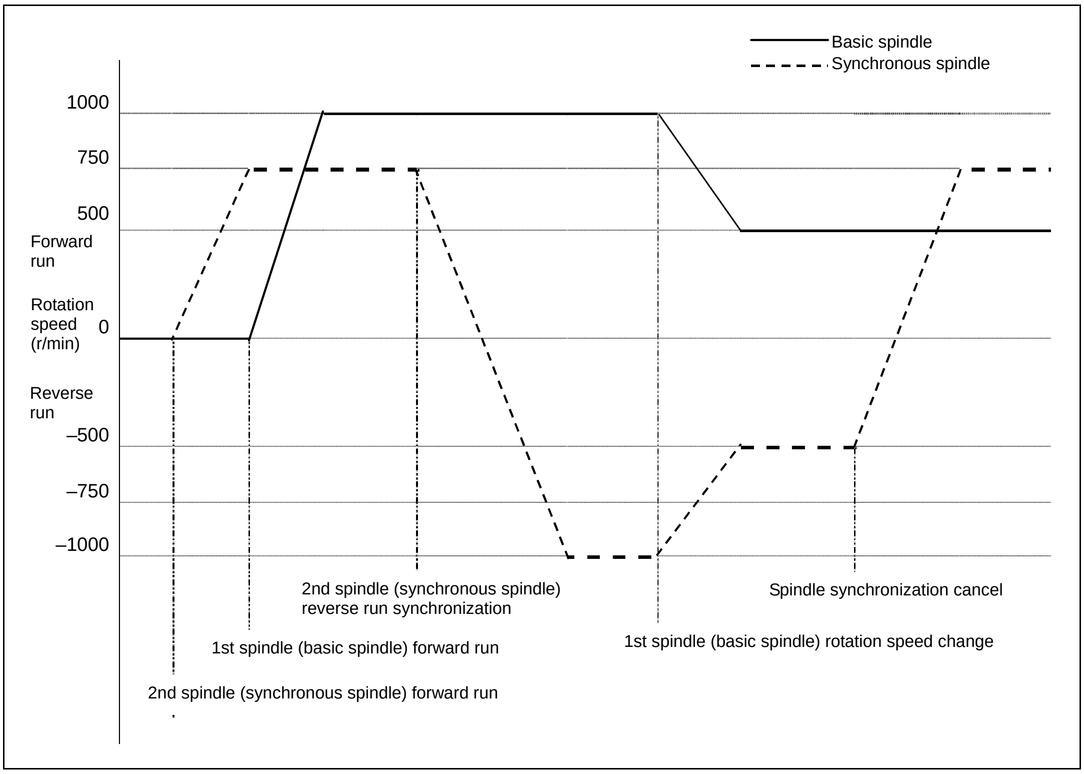
(2) If the basic spindle's commanded rotation speed is changed during the rotation synchronization state, acceleration/deceleration will be carried out while maintaining the synchronization state following the spindle acceleration/deceleration time constants set in the parameters, and the commanded rotation speed will be achieved.

(3) In the rotation synchronization state, the basic spindle can be controlled to the constant surface speed even when two spindles are grasping one workpiece.

(4) Operation will take place in the following manner.

```
M23 S2=750 ;        ... Forward rotate 2nd spindle (synchronous spindle) at 750
:                       r/min (speed command)
M03 S1=1000 ;       ... Forward rotate 1st spindle (basic spindle) at 1000 r/min
:                       (speed command)
G114.1 H1 D-2 ;     ... Synchronize 2nd spindle (synchronous spindle) to 1st
:                       spindle (basic spindle) with reverse run
S1=500 ;            ... Change 1st spindle (basic spindle) rotation speed to 500
:                       r/min
G113 ;              ... Cancel spindle synchronization
```

<Operation>



Phase synchronization

(1) When phase synchronization (command with R address) is commanded with the G114.1 command, the synchronous spindle rotating at a random rotation speed will accelerate or decelerate to the rotation speed commanded beforehand for the basic spindle, and will enter the rotation synchronization state.
Then, the phase is aligned so that the rotation phase commanded with the R address is reached, and the phase synchronization state is entered.

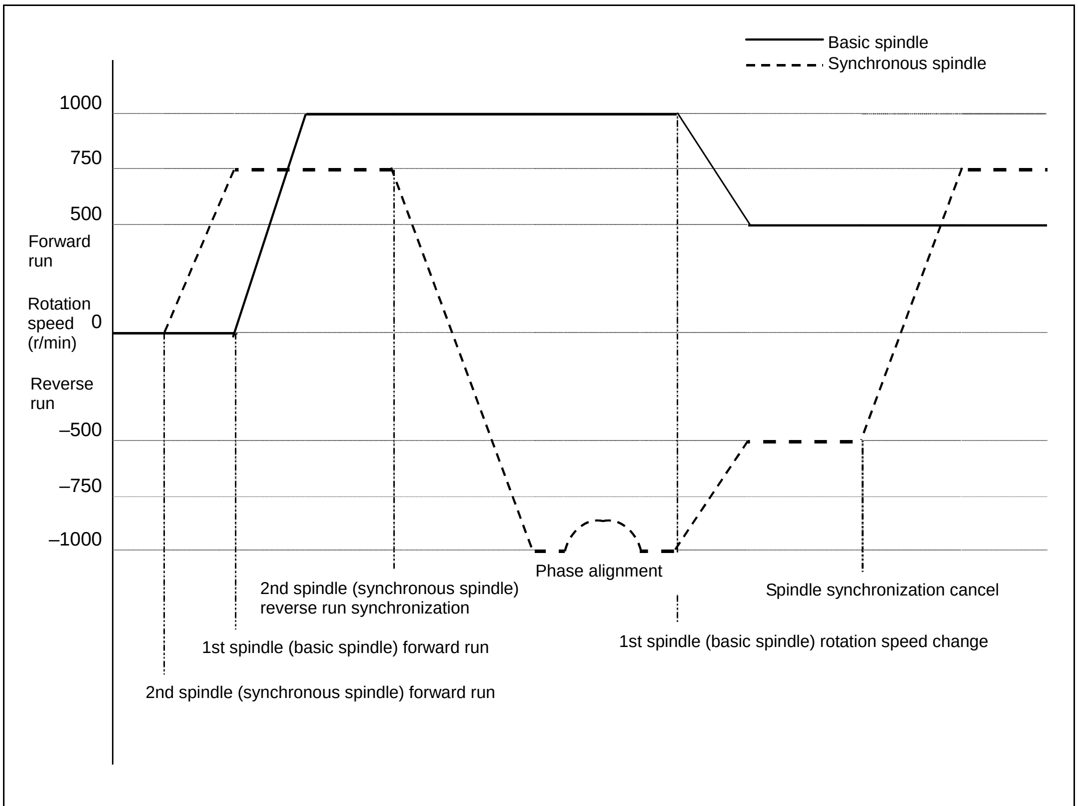
(2) If the basic spindle's commanded rotation speed is changed during the phase synchronization state, acceleration/deceleration will be carried out while maintaining the synchronization state following the spindle acceleration/deceleration time constants set in the parameters, and the commanded rotation speed will be achieved.

(3) In the phase synchronization state, the basic spindle can be controlled to the constant surface speed even when two spindles are grasping one workpiece.

(4) Operation will take place in the following manner.

```
M23 S2=750 ;            ... Forward rotate 2nd spindle (synchronous spindle) at 750
:                           r/min (speed command)
M03 S1=1000 ;           ... Forward rotate 1st spindle (basic spindle) at 1000 r/min
:                           (speed command)
G114.1 H1 D-2 Rxx ;     ... Synchronize 2nd spindle (synchronous spindle) to 1st
:                           spindle (basic spindle) with reverse run
:                           Shift phase of synchronous spindle by R command value
S1=500 ;                ... Change 1st spindle (basic spindle) rotation speed to 500
:                           r/min
G113 ;                  ... Cancel spindle synchronization
```

<Operation>



Cautions on programming

(1) To enter the rotation synchronization mode while the basic spindle and synchronous spindle are chucking the same workpiece, turn the basic spindle and synchronous spindle rotation commands ON before turning the spindle synchronous control mode ON.

$1 (1st part system)		$2 (2nd part system)	
:		:	
M6 ;	1st spindle chuck close	:	
:		M25 S2=0 ;	2nd spindle stops at S=0
:		:	
!2 ;	----------------------	!1 ;	Waiting between part systems
M5 S1=0 ;	1st spindle stops at S=0	M15 ;	2nd spindle chuck close
:		M24 ;	2nd spindle rotation command ON
M3 ;	1st spindle rotation command ON	:	
!2 ;	----------------------	!1 ;	Waiting between part systems
:		G114.1 H1 D-2 ;	Rotation synchronization
:		:	mode ON
S1=1500 ;	Synchronous rotation at S=1500	:	
:		:	
S1=0 ;	Both spindles stop		
G113 ;	Synchronization mode OFF		

(2) To chuck the same workpiece with the basic spindle and synchronous spindle in the phase synchronization mode, align the phases before chucking.

$1		$2	
:		:	
M6 ;	1st spindle chuck close	:	
:		:	
M3 S1=1500 ;	1st spindle rotation command ON	:	
:		G114.1 H1 D-2 R0 ;	Phase synchronization
:		:	mode ON
:		M24 ;	2nd spindle rotation command ON
:		:	
:		M15 ;	2nd spindle chuck close (Note 1)
:		:	

(Note 1) Close the chuck after confirming that the spindle phase synchronization complete signal (X42A) has turned ON (phase alignment complete).

⚠ **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.

Precautions and restrictions

(1) To carry out the spindle synchronization, it is required to command spindle rotation for both basic spindle and synchronous spindle. Note that the rotating direction of the synchronous spindle follows the rotating direction of the basic spindle and rotating direction designation by "D" address.

(2) The spindle rotating with spindle synchronous control will stop when emergency stop is applied.

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

(4) Orientation of the basic spindle and synchronous spindle is not possible during the spindle synchronous control mode. To carry out orientation, cancel the spindle synchronous control mode first.

(5) The rotation speed command (S command) is invalid for the synchronous spindle during the spindle synchronous control mode. Note that the modal will be updated, so this will be validated when spindle synchronous control is canceled.

(6) The constant surface speed control is invalid for the synchronous spindle during the spindle synchronization control mode. Note that the modal will be updated, so this will be validated when spindle synchronization is canceled.

(7) The rotation speed command (S command) and constant surface speed control for the synchronous spindle will be validated when spindle synchronous control is canceled. Thus, the synchronous spindle may carry out different operations when this control is canceled.

(8) An attention should be made that if the phase synchronization command is executed with the phase error not obtained by the phase shift calculation request signal, the phase shift amount will not be obtained correctly.

(9) The spindle rotation speed command (S command) and the constant surface speed control for the synchronous spindle will become valid when the spindle synchronous control is canceled. Thus, special attention should be made because the synchronous spindle may do different action than before when the spindle synchronous control is canceld.

(10) If the phase synchronization command (command with R address) is issued while the phase shift calculation request signal is ON, an operation error (1106) will occur.

(11) If the phase shift calculation request signal is ON and the basic spindle or synchronous spindle is rotation while rotation synchronization is commanded, an operation error (1106) will occur.

(12) If the phase synchronization command R0 (<Ex.> G114.2 H1 D-2 R0) is commanded while the phase offset request signal is ON, the basic spindle and synchronous spindle phases will be aligned to the phase error of the basic spindle and synchronous spindle saved in the NC memory.

(13) If a value other than the phase synchronization command R0 (<Ex.> G114.1 H1 D-2 R000) is commanded while the phase offset request signal is ON, the phase error obtained by adding the value commanded with the R address command to the phase difference of the basic spindle and synchronous spindle saved in the NC memory will be used to align the basic spindle and synchronous spindle.

(14) The phase offset request signal will be ignored when the phase shift calculation request signal is ON.

(15) The phase error of the basic spindle and synchronous spindle saved in the NC is valid only when the phase shift calculation signal is ON and for the combination of the basic spindle selection (H_) and synchronous spindle (D_) commanded with the rotation synchronization command (no R address).
For example, if the basic spindle and synchronous spindle phase error is saved as "G114.1 H1 D-2 ;", the saved phase error will be valid only when the phase offset request signal is ON and "G114.1 H1 D_2 R∗∗∗ ;" is commanded. If "G114.1 H2 D-1 R∗∗∗ ;" is commanded in this case, the phase shift amount will not be calculated correctly.

(16) The phase error of the basic spindle and synchronous spindle saved in the NC is retained until the spindle synchronization phase shift calculation, in other words, until the rotation synchronous control command completes with the phase shift calculation request signal is ON.

(17) Synchronous tapping can not be used during spindle synchronous mode.

(18) When the spindle synchronous control commands are being issued with the PLC I/F method (#1300 ext36/bit7 OFF), a program error (P610) will occur if the spindle synchronous control is commanded with G114.1/G113.

10.8 Spindle synchronization control II

Function and purpose

In a machine having two or more spindles, this function controls the rotation speed and phase of one spindle (synchronous spindle) in synchronization with the rotation of the other spindle (basic spindle).
The function is used if a workpiece grasped by the basic spindle is to be grasped by a synchronous spindle, or if the spindle rotation speed has to be changed when one workpiece is grasped by both spindles.
With the spindle synchronous control II, selection of the spindles and synchronization start, etc., are all designated from the PLC.

Basic spindle and synchronous spindle selection

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

Device No.	Signal name	Abbrev.	Explanation
R157	Basic spindle selection	–	Select a serially connected spindle to be controlled as the basic spindle. (0: 1st spindle), 1: 1st spindle, 2: 2nd spindle, … , 7: 7th 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.
R158	Synchronous spindle selection	–	Select a serially connected spindle to be controlled as the synchronous spindle. (0: 2nd spindle), 1: 1st spindle, 2: 2nd spindle, … , 7: 7th spindle **(Note 3)** Spindle synchronous 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

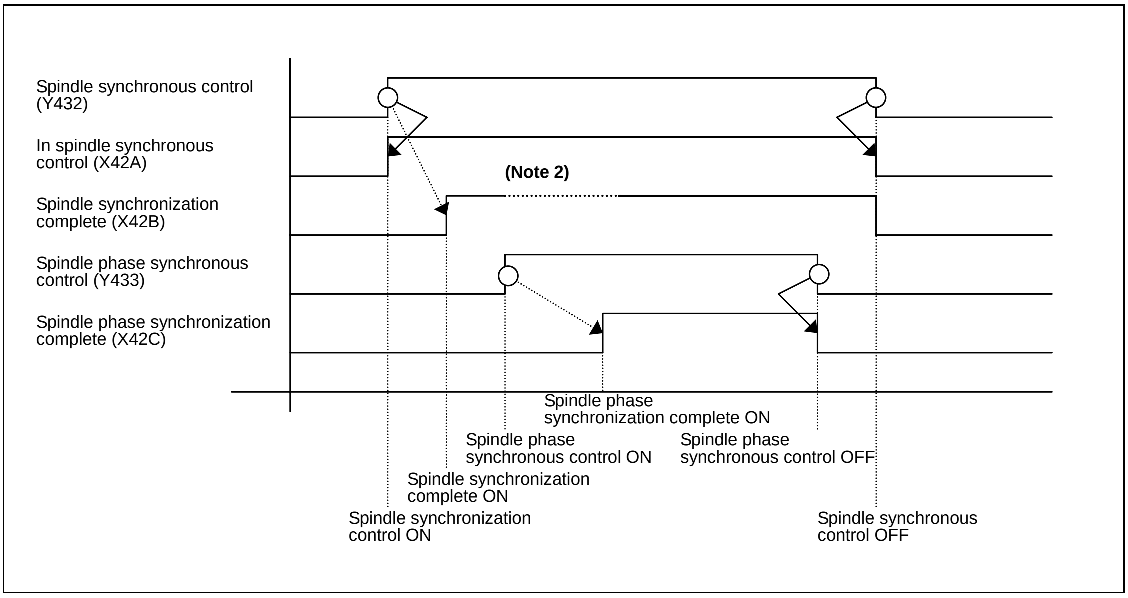
The spindle synchronous control mode is entered by inputting the spindle synchronous control signal (SPSYC). The synchronous spindle will be controlled in synchronization with the rotation speed commanded for the basic spindle during the spindle synchronous control mode.
When the difference of the basic 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
Y432	Spindle synchronous control	SPSYC	The spindle synchronous control mode is entered when this signal turns ON.
X42A	In spindle synchronous control	SPSYN1	This notifies that the mode is the spindle synchronous control.
X42B	Spindle rotation speed synchronization complete	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 synchronous control mode. This signal turns OFF when the spindle synchronous control mode is canceled, or when an error exceeding the spindle rotation speed reach level setting value occurs during the spindle synchronous control mode.
Y434	Spindle synchronization rotation direction designation	SPSDR	Designate the basic spindle and synchronous spindle rotation directions for spindle synchronous 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 synchronous 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
Y433	Spindle phase synchronous control	SPPHS	Spindle phase synchronization starts when this signal is turned ON during the spindle synchronous control mode. **(Note 1)** If this signal is turned ON in a mode other than the spindle synchronous control mode, it will be ignored.
X42C	Spindle phase synchronization complete	FSPPH	This signal is output when the spindle synchronization phase reach level is reached after starting spindle phase synchronization.
R159	Phase shift amount setting	–	Designate the synchronous spindle's phase shift amount. Unit: 360°/4096

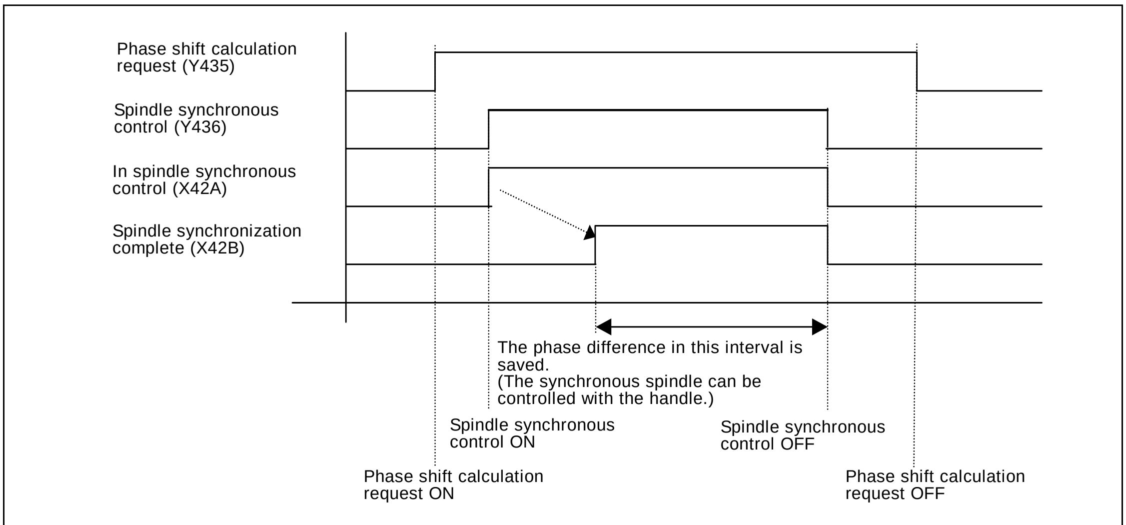


(Note 2) Turns OFF temporarily to change the rotation speed during phase synchronization.

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
Y435	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.
Y436	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.
R55	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.
R59	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 synchronous control.

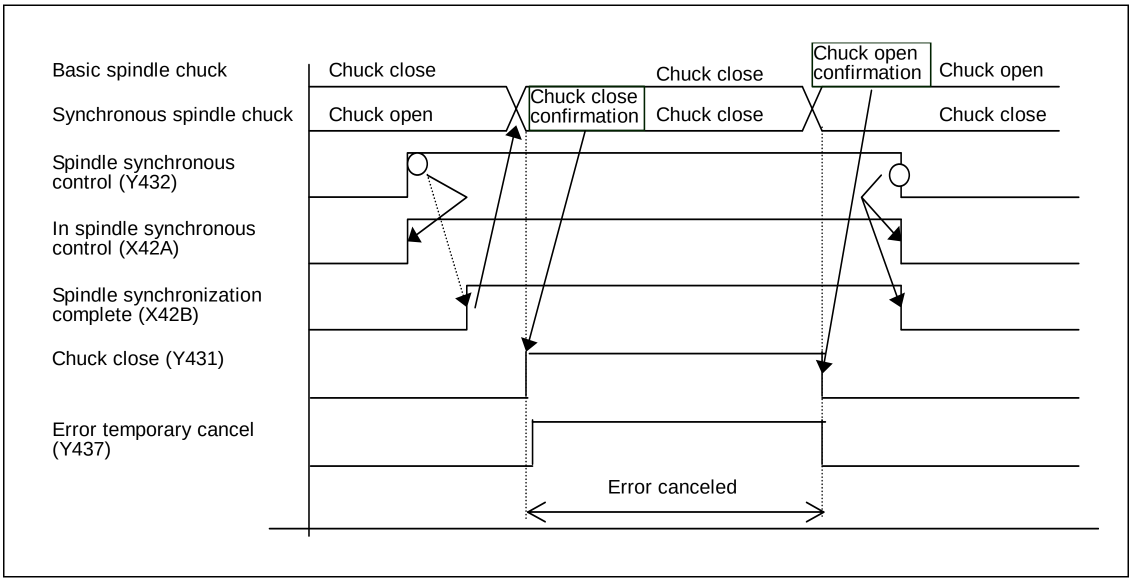


(Note 4) The phases cannot be aligned while calculating the phase shift.
(Note 5) The synchronous spindle cannot be rotated with the handle when the manual operation mode is set to the handle mode.

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
Y431	Chuck close	SPCMPC	This turns ON when the chuck of both spindles are closed. This signal is ON while the basic spindle and the synchronous spindle grasp the same workpiece.
X42D	Chuck close confirmation	SPCMP	This turns ON when the chuck close signal is received during the spindle synchronous 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
Y437	Error temporary cancel	SPDRP0	The error is canceled when this signal is ON.

(Note 1) Even if the chuck close signal (Y431) is OFF, the error will be canceled while this signal (Y437) is ON.

(Note 2) Turn this signal ON after the both chucks of basic spindle side and synchronous spindle side are closed to grasp the workpiece.
Turn this signal OFF if even one chuck is opened.

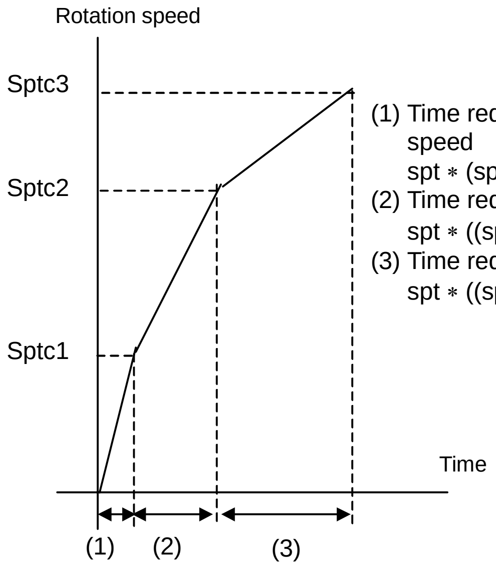
Phase error monitor

The phase error can be monitored during spindle phase synchronization.

Device No.	Signal name	Abbrev.	Explanation
R56	Phase error monitor	–	The phase error during spindle phase synchronous control is output as a pulse unit.
R57	Phase error monitor (lower limit value)	–	The lower limit value of the phase error during spindle phase synchronous control is output as a pulse unit.
R59	Phase error monitor (upper limit value)	–	The upper limit value of the phase error during spindle phase synchronous 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

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 speed is commanded 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. Note that the modal is rewritten, thus, 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. However, note that the modal is rewritten and it will be valid after spindle synchronization is canceled.

(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 and incorrect operation results.

(10) The spindle rotation speed command (S command) and the constant surface speed control for the synchronous spindle will become valid when the spindle synchronous control is canceled. Thus, special attention should be made because the synchronous spindle may do different action than before when the spindle synchronous control is canceled.

(11) The spindle Z-phase encoder position parameter (sppst) is invalid (ignored) when phase offset is carried out.
This parameter will be valid when the phase offset request signal is OFF.

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

(13) 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.

(14) The phase offset request signal is ignored when the phase shift calculation request signal (Y435) is ON.

(15) "M01 OPERATION ERROR 1106" will occur when a spindle No. out of specifications is designated in the R registers to set the basic spindle and the synchronous spindle, or when the spindle synchronous control signal (Y432) is turned ON with R resister value illegal.

(16) 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.)

(17) Synchronous tapping can not be used during spindle synchronous mode.

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 commanded one block. The output signal is an 8-digit BCD signal and start signal.
When the T functions are commanded in the same block as movement commands, there are 2 sequences in which the commands are executed, as below. 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.

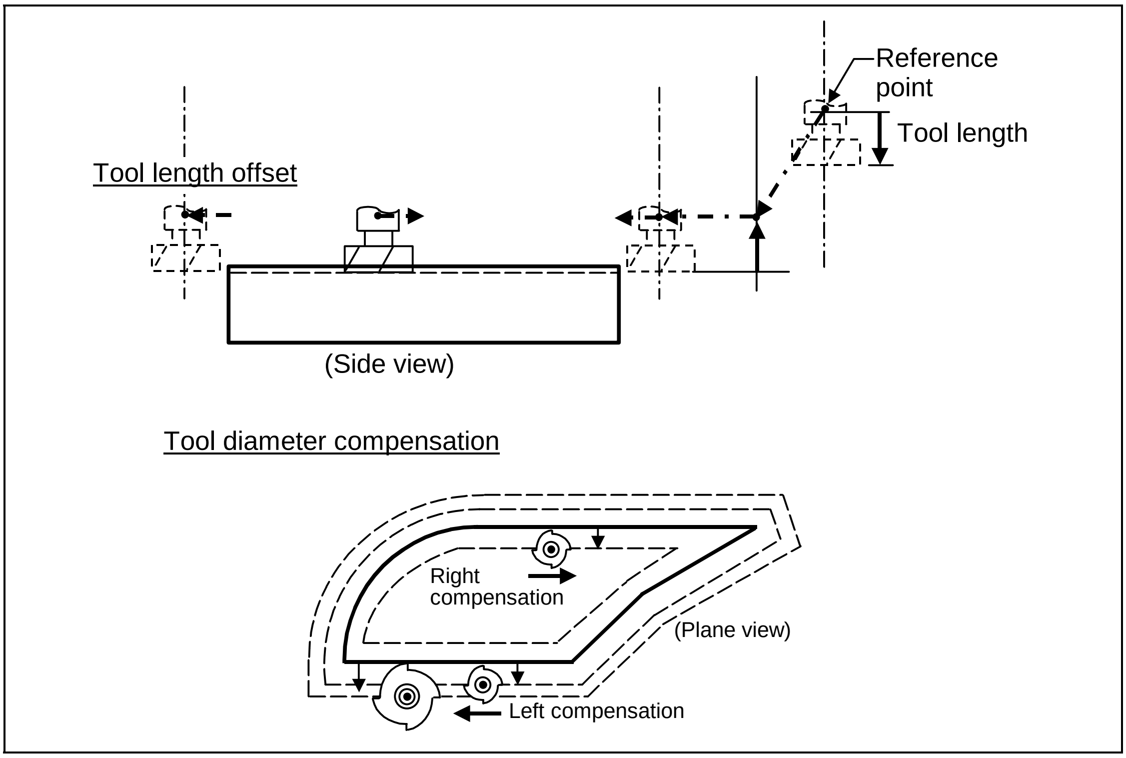
Processing and completion sequences are required for all T commands.

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.

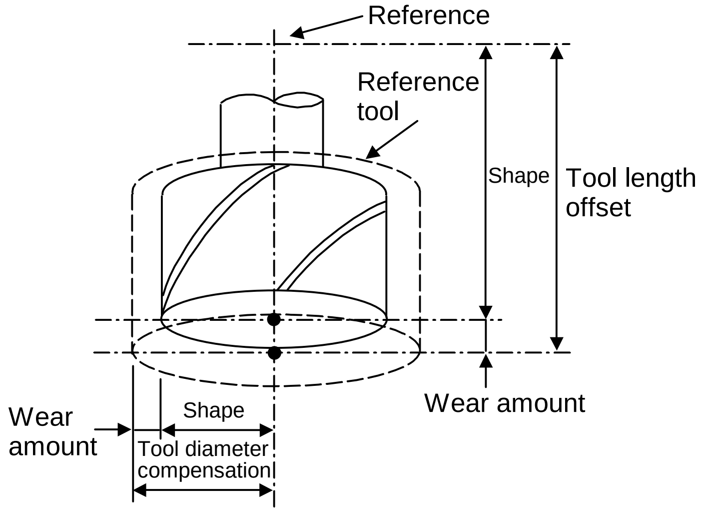


Tool offset memory

There are two types of tool offset memories for setting and selecting the tool offset amount. (The type used is determined by the machine maker 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

One 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 40 sets: Designate with the H01 to H40 (D01 to D40) 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.

Input setting unit	Shape offset amount		Wear offset amount	
	Metric system	Inch system	Metric system	Inch system
#1015 cunit=100	±99999.99mm	±9999.999 inch	±9999.99 mm	±999.999 inch
#1015 cunit=10	±9999.999mm	±999.9999 inch	±999.999 mm	±99.9999 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.

	Z axis movement amount		
G43 Zz Hn_1 ;	z +	(ℓh_1)	Offset in + direction by tool offset amount
G44 Zz Hh_1 ;	z –	(ℓh_1)	Offset in – direction by tool offset amount
G49 Zz ;	z –(+)	(ℓh_1)	Offset amount cancel.
		ℓh_1 :	Offset amount for offset No. h_1

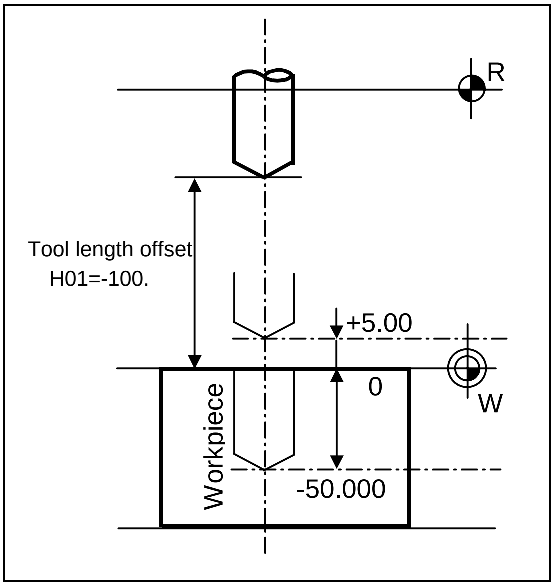
Regardless of the absolute value command or incremental value command, the actual end point will be the point offset by the offset amount designated for the programmed movement command end point coordinate value.
The G49 (tool length offset cancel) mode is entered when the power is turned ON or when 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  ;
```



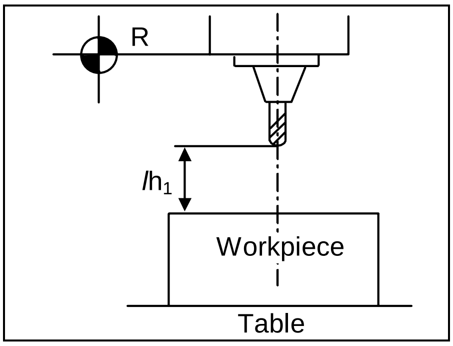
(2) Offset No.

(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.

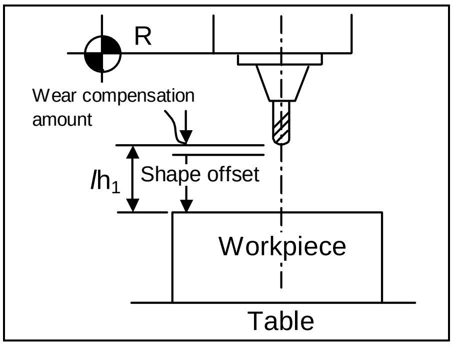


Type 2

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  Zz1  Hh1  ;...........Tool length offset is executed with h1.
          :
G45  Xx1  Yy1  Hh6 ;
          :
G49  Zz2  ; ...................The tool length offset is canceled.
          :
G43  Zz2  ; ...................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  Zz1  Hh1 ; ........... Becomes the z1 + (ℓh1) movement.
          :
G43  Zz2  Hh2 ; ........... Becomes the z2 + (ℓh2 - lh1) movement.
          :
```

The same applies for the G44 command in the G44 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  αα1  Hh3 ; ................+ offset to additional offset
      :
G49  αα1 ;
      :
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 during the radius compensation mode.
G39X___Y___ ;	**: Corner changeover**	

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.

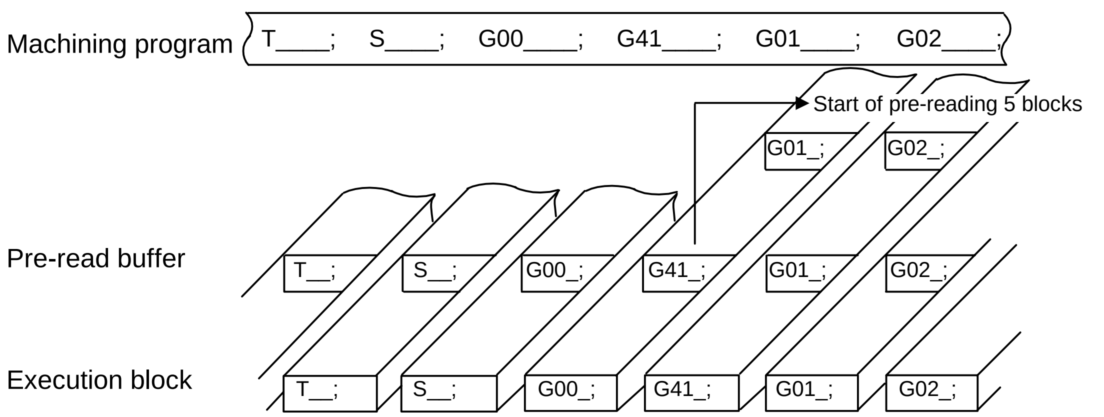
Tool radius compensation start (start-up)

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

(1) A movement command is issued after the G41or G42 command has been issued.
(2) The tool radius compensation offset No. is 0 < D ≤ 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.

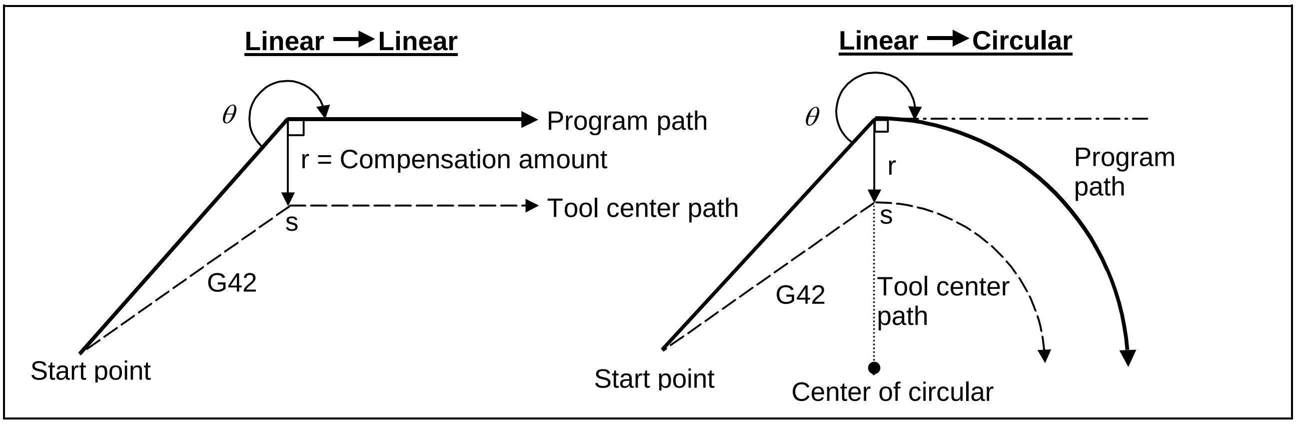
Control mode transition diagram



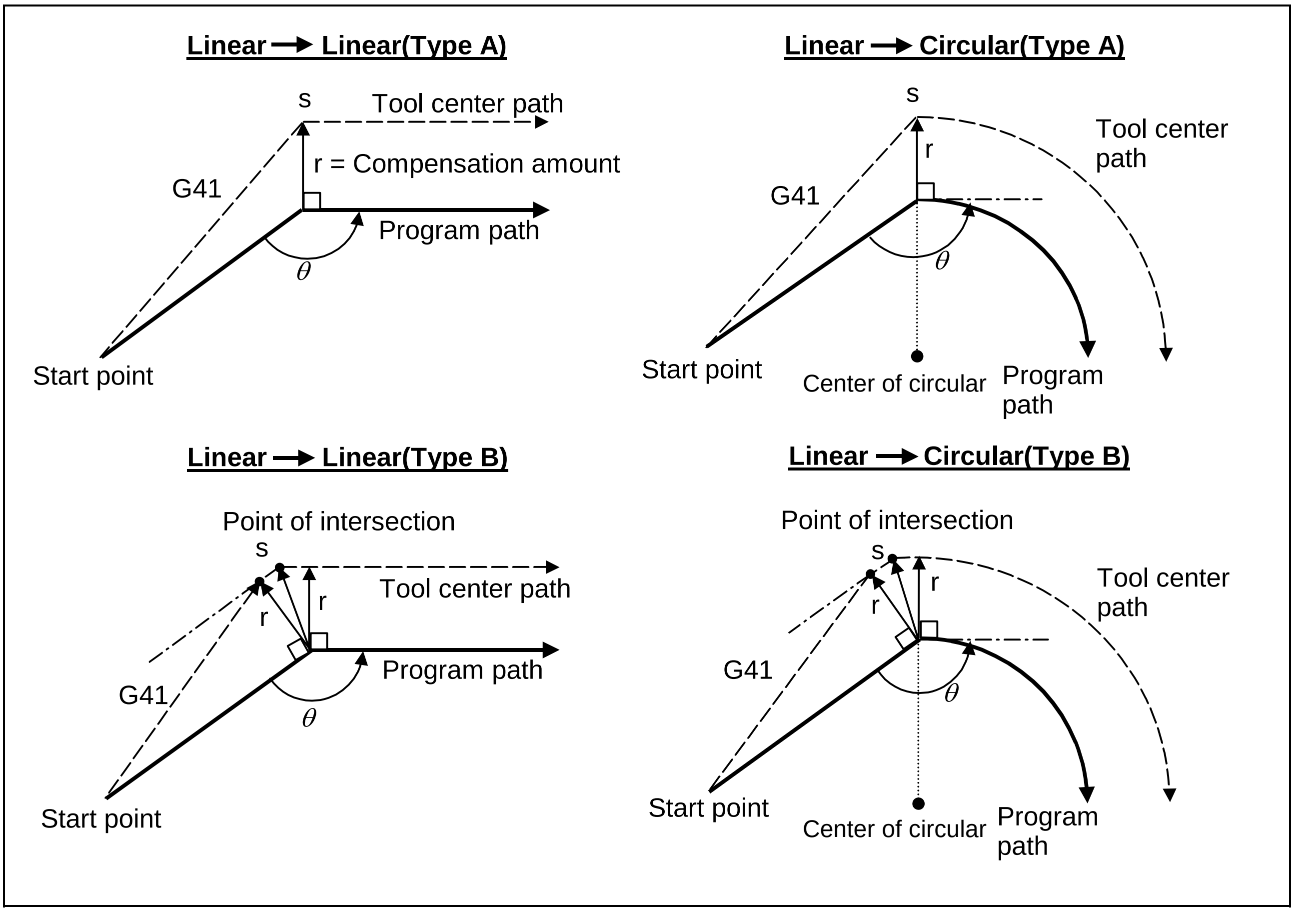
There are two ways of starting the compensation operation: type A and type B.
The type can be selected with bit 2 of parameter "#1229 set 01". This type is used in common with the compensation cancel type.
In the following explanatory figure, "S" denotes the single block stop point.

Start of movement for tool radius compensation

(1) For inner side of corner

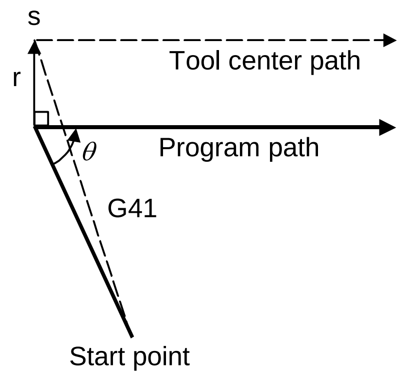


(2) For outer side of corner (obtuse angle) [90°≤0<180°]

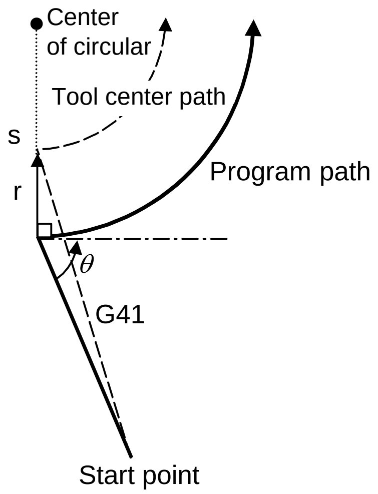


(3) For outer side of corner (obtuse angle) [0<90°]

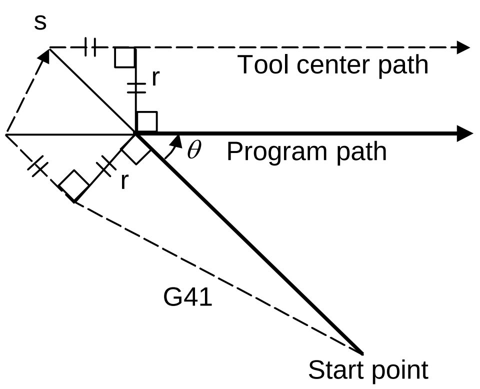
Linear → Linear(Type A)



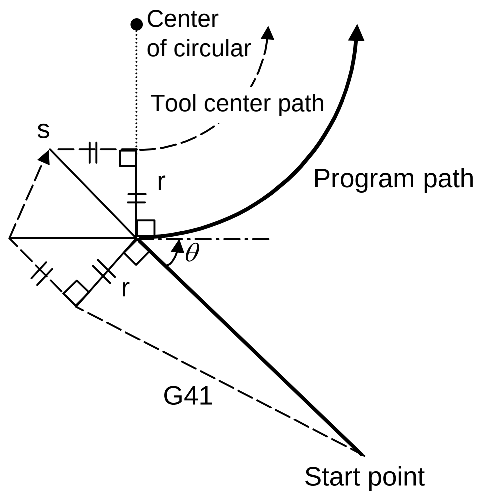
Linear → Circular(Type A)



Linear → Linear(Type B)



Linear → Circular(Type B)

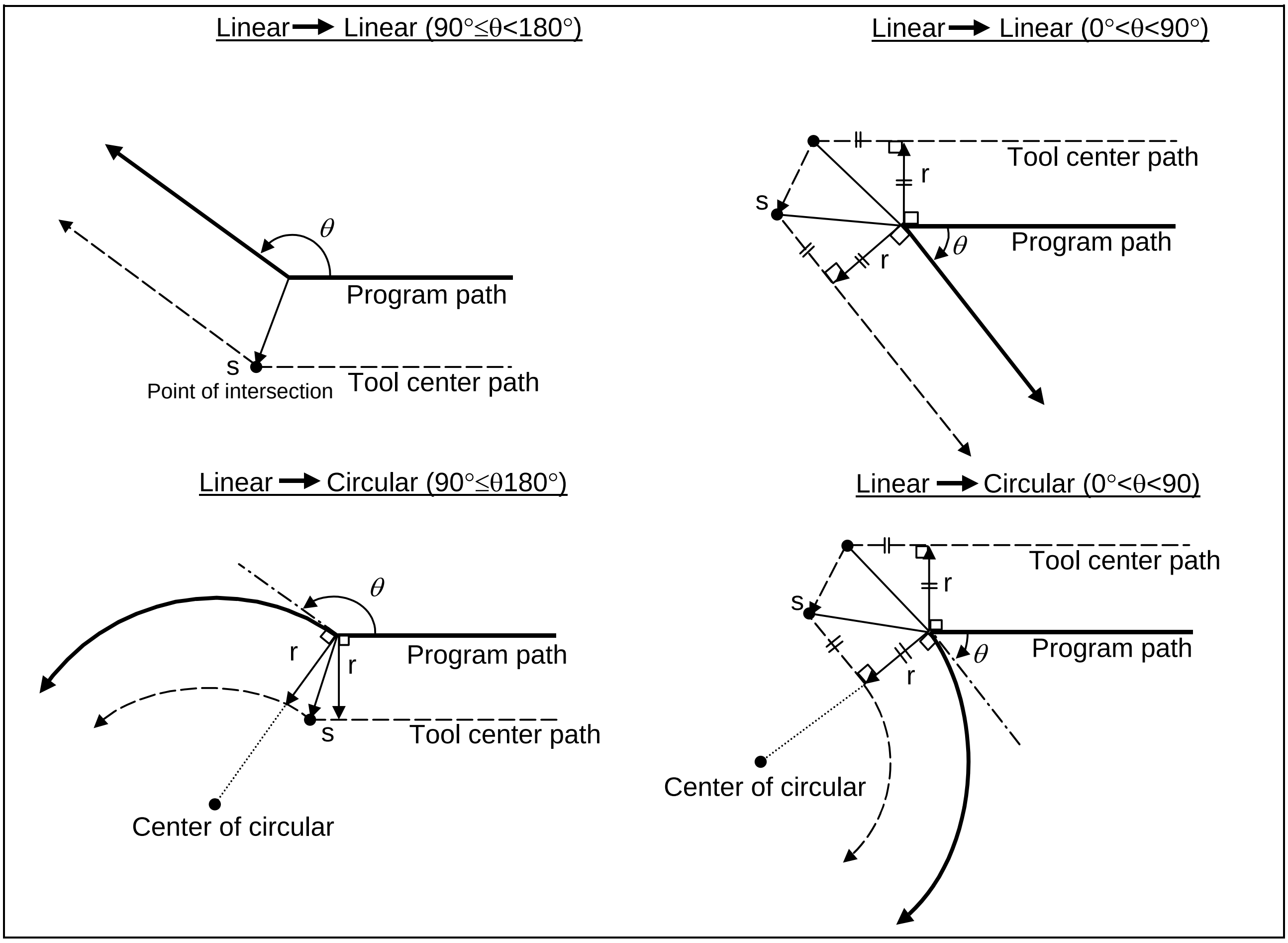


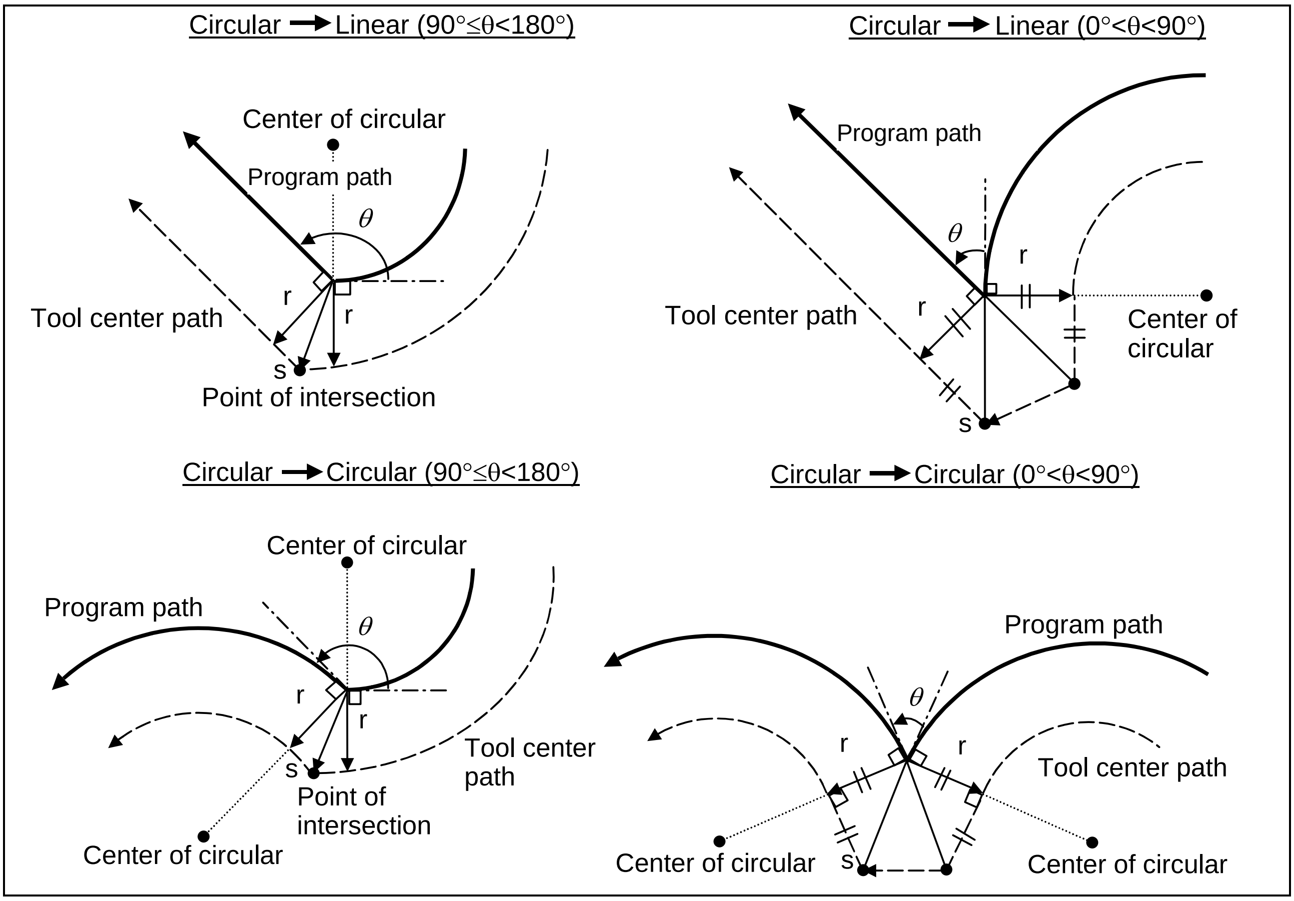
(Note 1) Where is no axis movement command in the same block as G41 or G42, compensation is performed perpendicularly to the next block direction.

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.

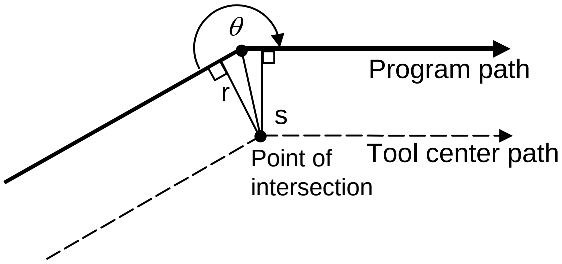
(1) Machining an outer wall



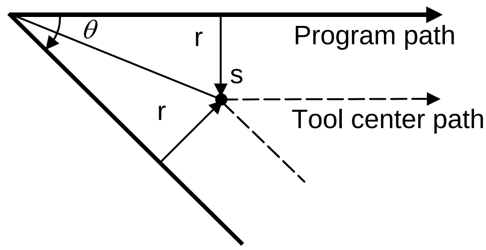

Circular → Linear (90°≤θ<180°)
Center of circular
Program path
θ
r
r
Tool center path
s
Point of intersection
Circular → Linear (0°<θ<90°)
Program path
θ
r
r
Tool center path
Center of circular
s
Circular → Circular (90°≤θ<180°)
Center of circular
Program path
θ
r
r
s
Tool center path
Point of intersection
Center of circular
Circular → Circular (0°<θ<90°)
Program path
θ
r
r
Tool center path
Center of circular
s
Center of circular

(2) **Machining an inner wall**

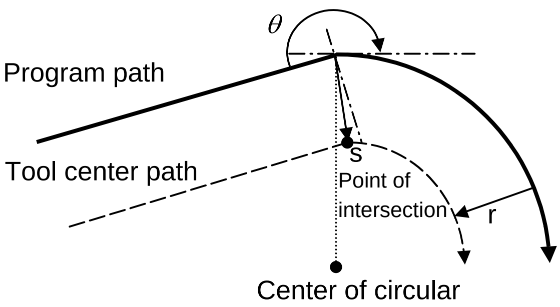
Linear → Linear (Obtuse angle)



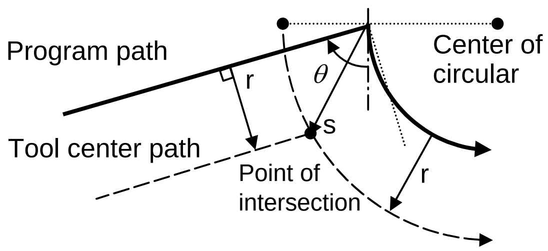
Linear → Linear (Acute angle)



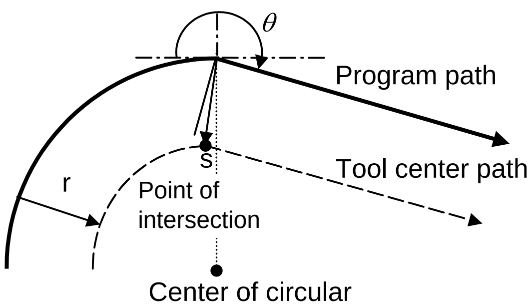
Linear → Circular (Obtuse angle)



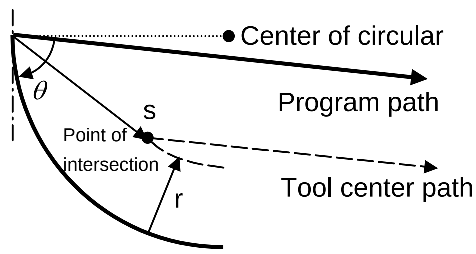
Linear → Circular (Acute angle)

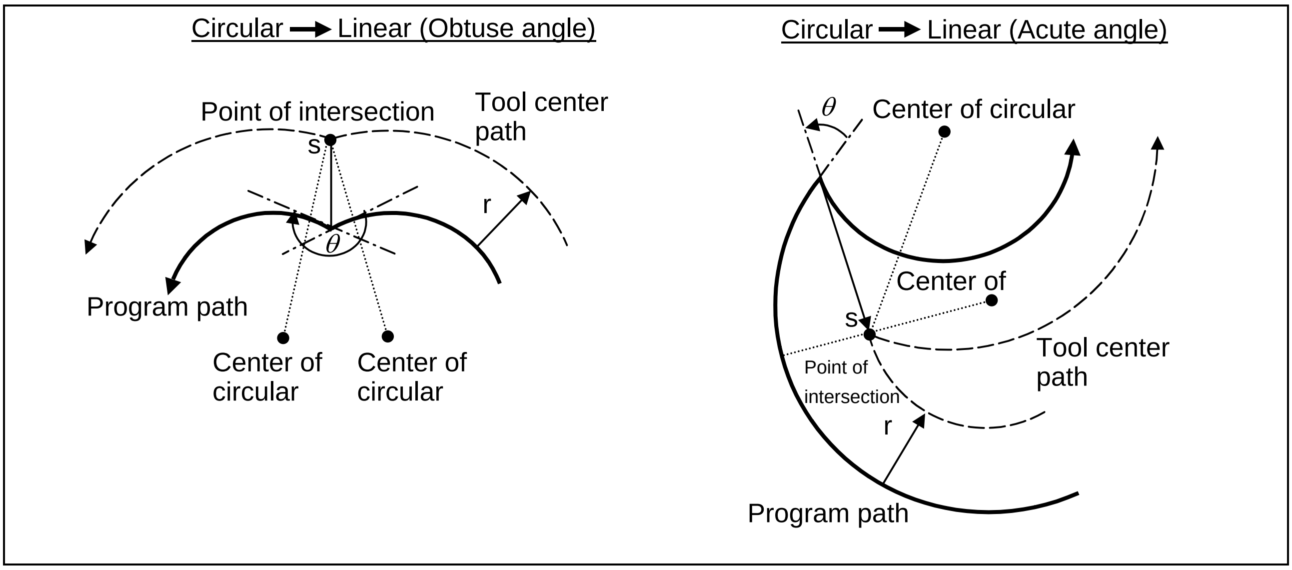


Circular → Linear (Obtuse angle)



Circular → Linear (Acute angle)

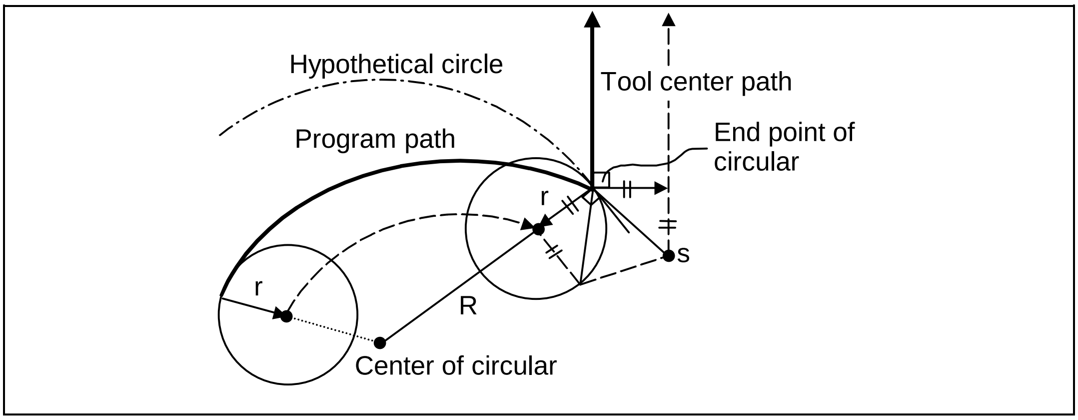




(3) When the arc end point is not on the arc

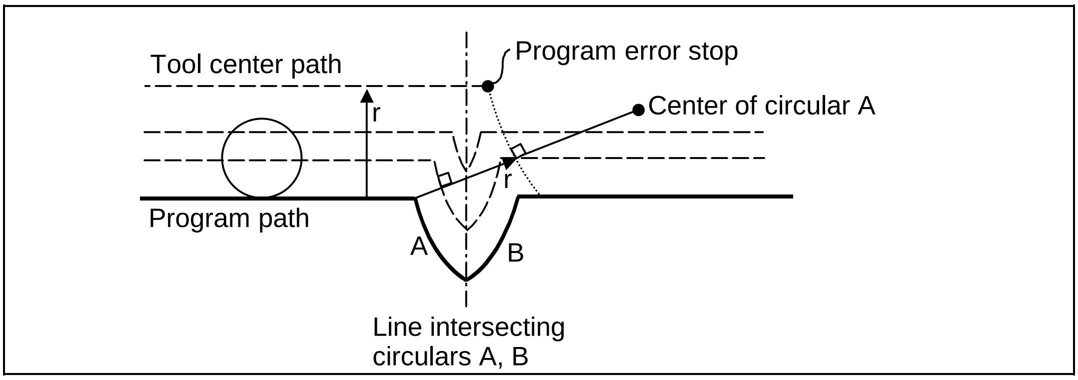
For spiral arc .... A spiral arc will be interpolated from the start to end point of the arc.

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.

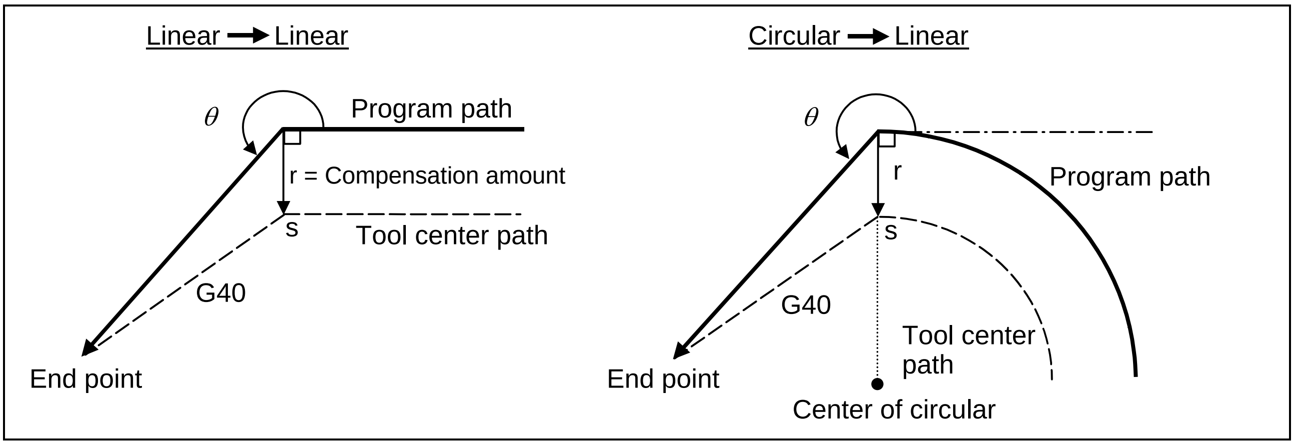


Tool radius compensation cancel

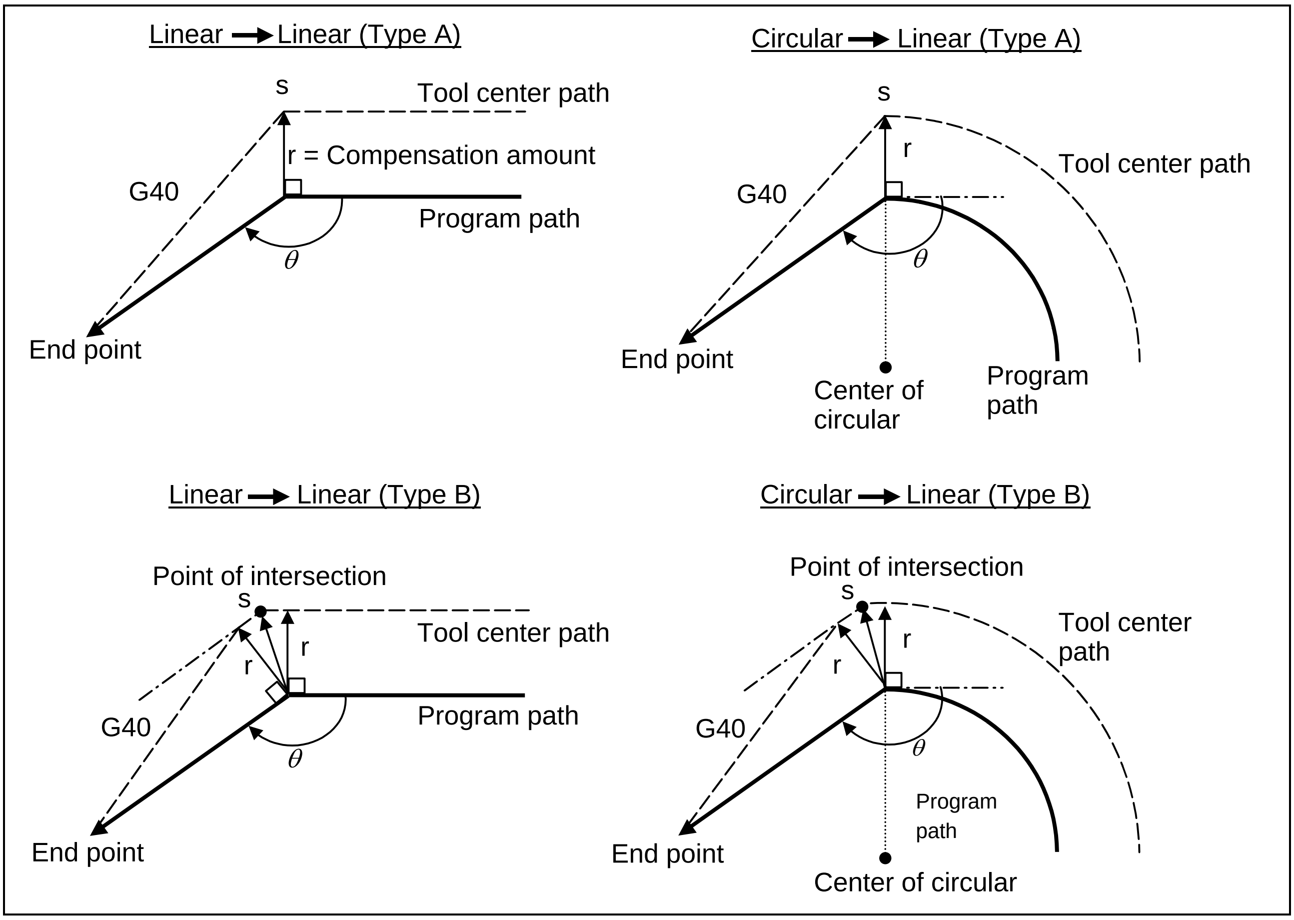
If either of the following conditions is met in the tool radius compensation mode, the compensation will be canceled. However, the movement command must be a command which is not a circular command.
If the compensation is canceled by a circular command, program error (P151) results.
(1) The G40 command 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

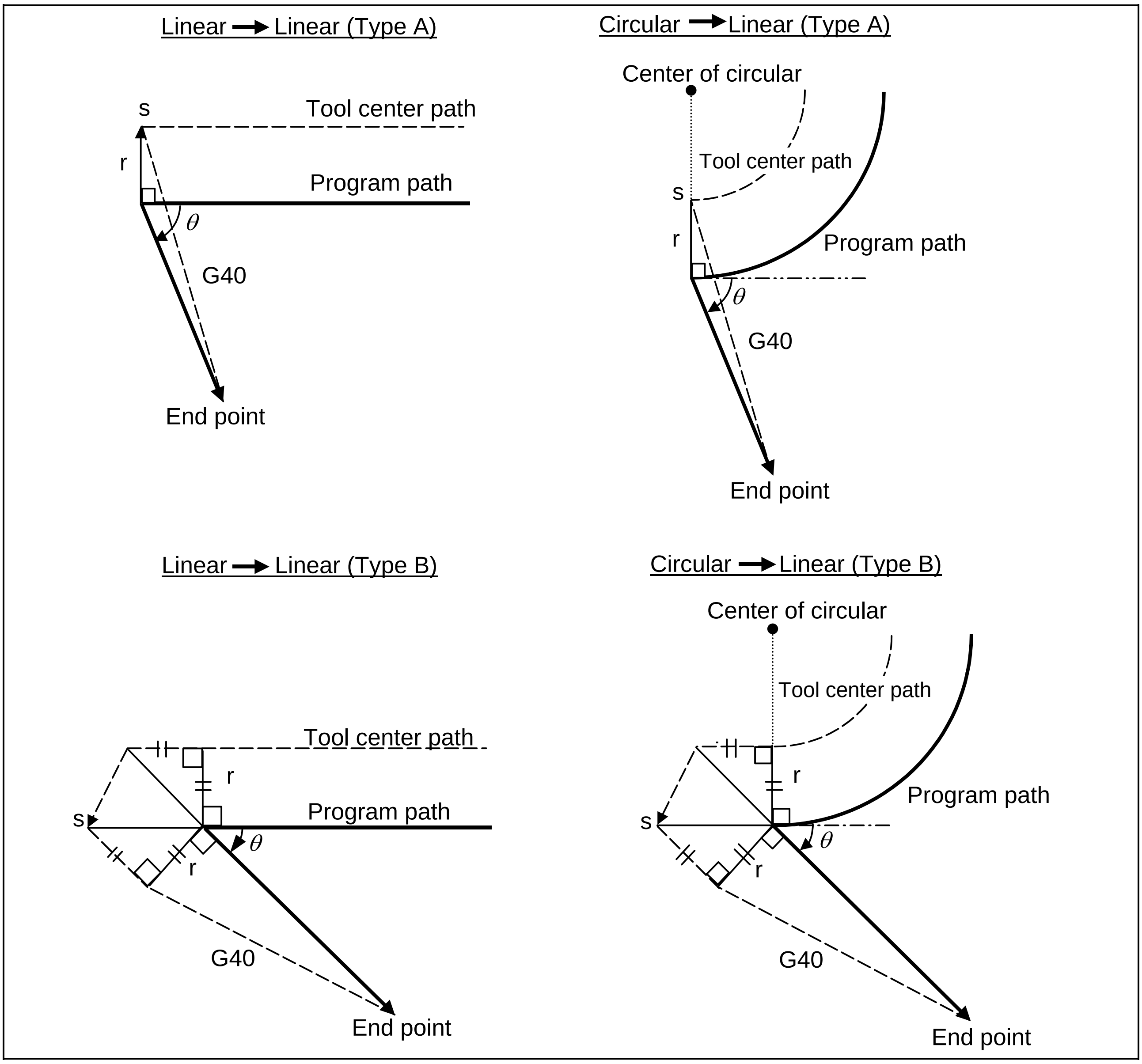
(1) For inner side of corner



(2) **For outer side of corner (obtuse angle)**



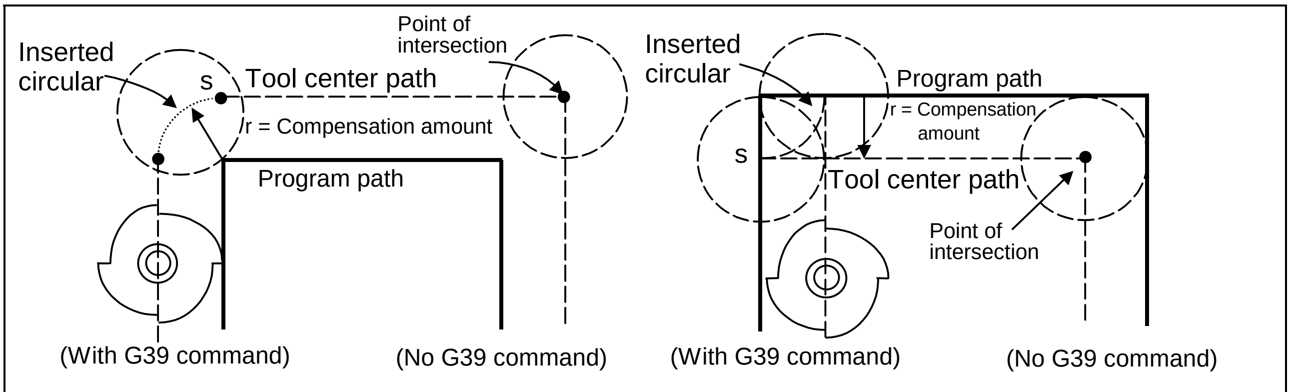
(3) For outer side of corner (acute angle)



12.3.2 Other operations during tool radius compensation

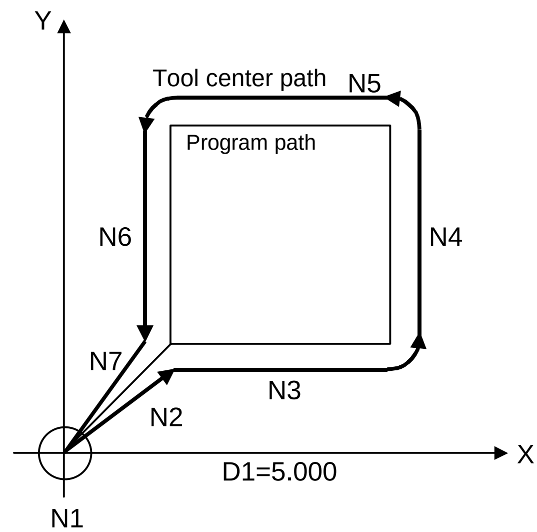
Insertion of corner arc

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.

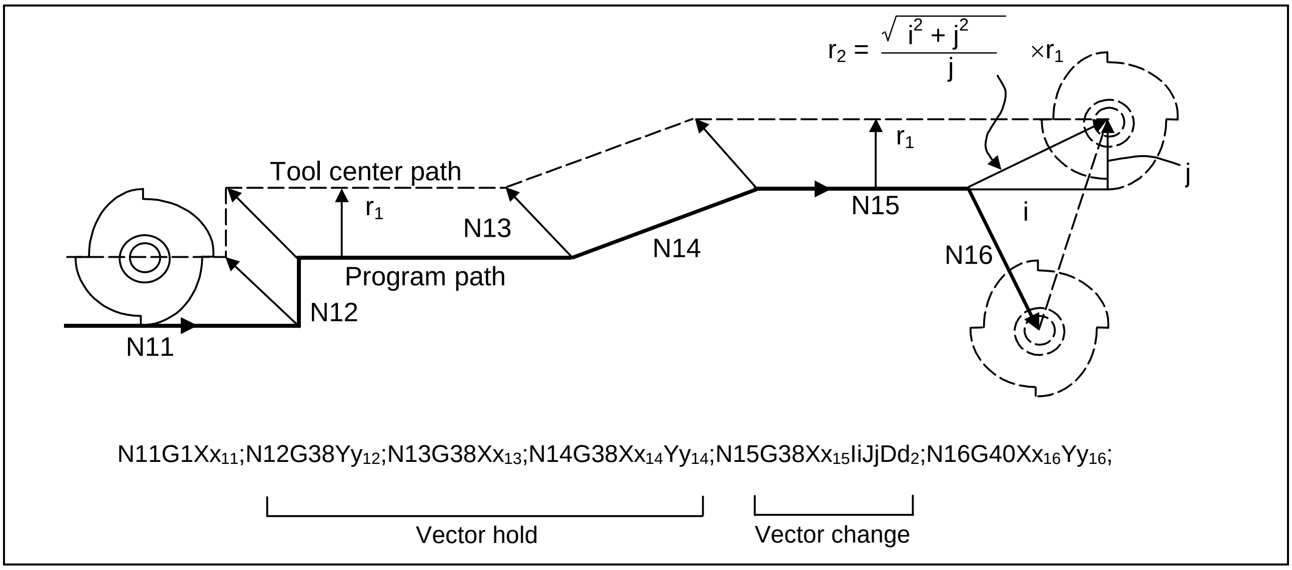
G38 Xx Yy ;

This can be used for pick feed, etc.

(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.)

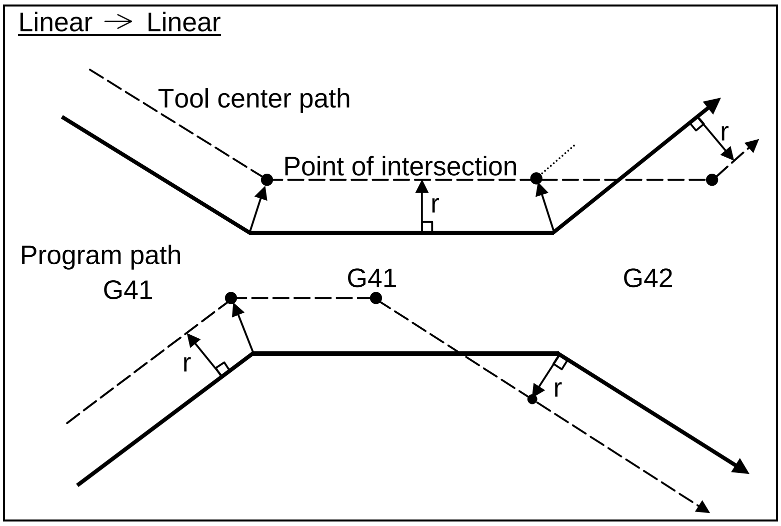


Changing the compensation direction during tool diameter compensation

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

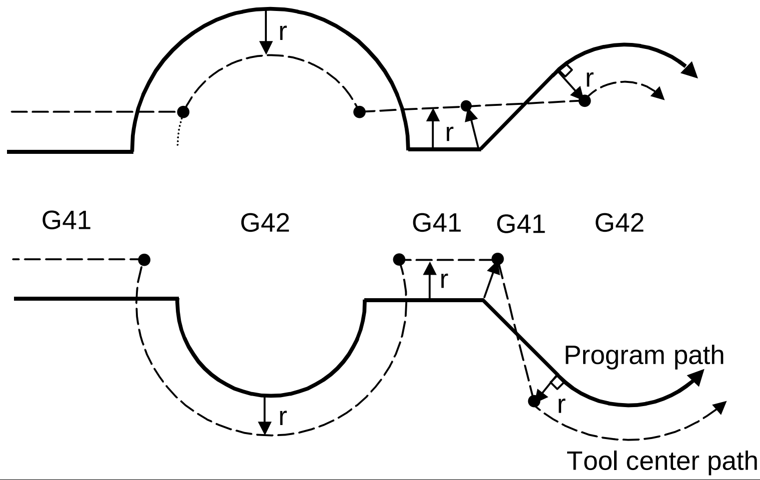
G code \ Compensation amount sign	+	–
G41	Left-hand compensation	Right-hand compensation
G42	Right-hand compensation	Left-hand compensation

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.

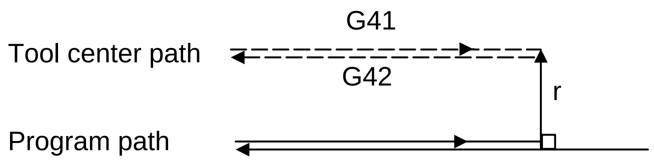


(If there is no point of intersection when the compensation direction is changed.)

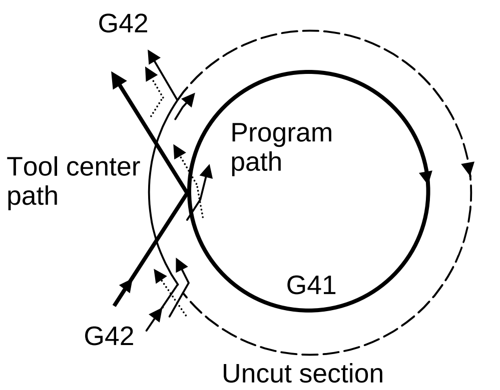
Linear ↔ Circular



Linear return



Arc exceeding 360° due to compensation



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

a. With offset direction selection based on G41/G42

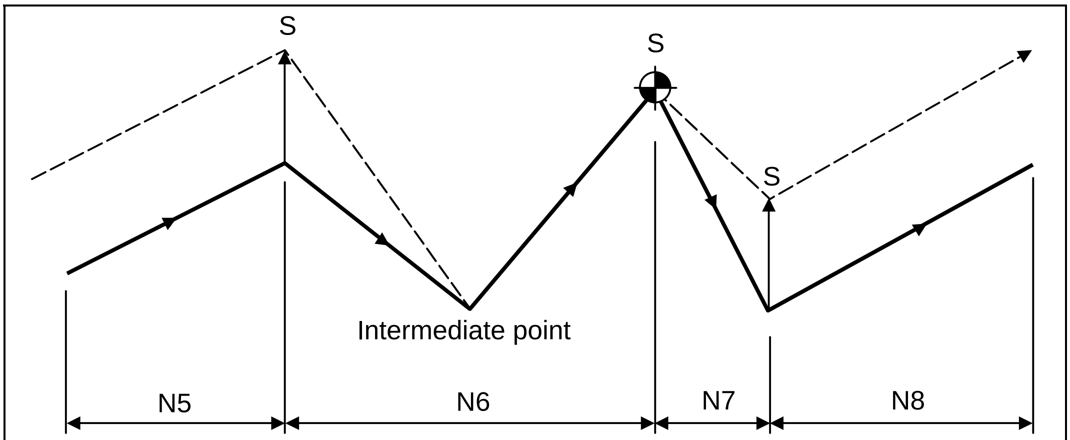
b. I, J, K was commanded in G40.

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.

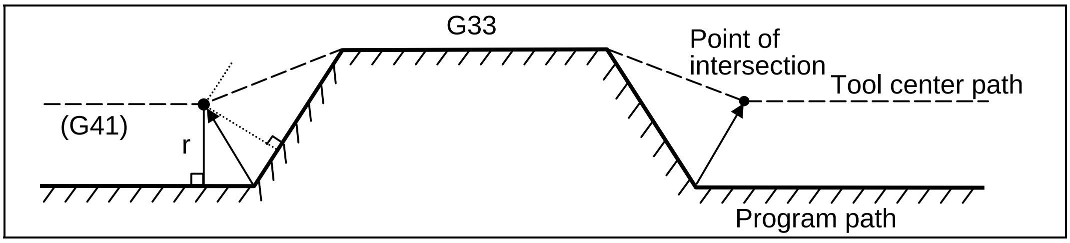
(1) Reference point return command




```
(G41)      ~
N5   G91  G01  X60.  Y30.  ;
N6   G28       X50.  Y-40. ; ← Temporarily no compensation vectors at
N7             X30.  Y-60. ;   intermediate point.
N8             X70.  Y40.  ;   (Reference point when there is no
           ~                   intermediate point)
```

(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

No movement

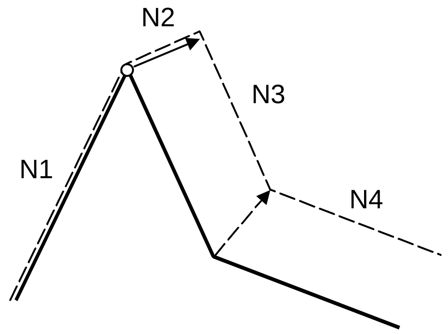
j. G91 X0 ; .... Zero movement amount ..... Movement amount is zero

M00, M01, M02 and M30 are handled as pre-read inhibit M codes.

(1) When command is assigned at start of the compensation

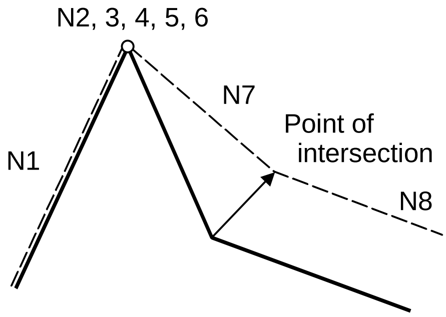
Perpendicular compensation will be applied on the next movement block.

```
N1  X30.   Y60.   ;
N2  G41    D10    ;  ← Block without movement
N3  X20. Y-50.    ;
N4  X50. Y-20.    ;
```

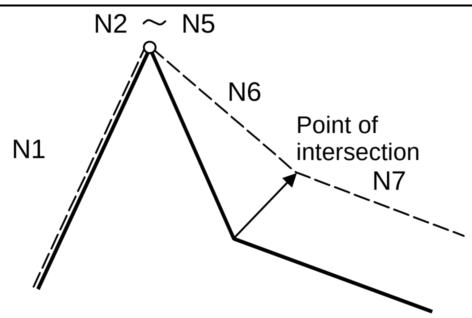


Compensation vector cannot be generated when 4 or more blocks continue without movement or when a pre-reading prohibit M code is issued.

```
N1  X30.   Y60.   ;
N2  G41    D10 ;      }
N3  G4     X1000 ;    }
N4  F100 ;            }  Block without movement
N5  S500 ;            }
N6  M3 ;              }
N7  X20. Y-50. ;
N8  X50. Y-20. ;
```



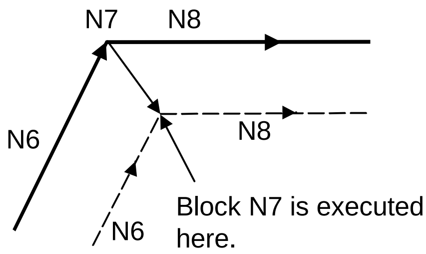

```
N1  G41   X30.   Y60.   D10 ;
N2  G4    X1000 ;   }
N3  F100 ;          }  Block without movement
N4  S500 ;          }
N5  M3 ;            }
N6  X20.   Y-50. ;
N7  X50.   Y-20. ;
```



(2) When command is assigned in the compensation mode

When the blocks without movement follows up to 3 blocks in succession in the compensation mode and there is no pre-reading prohibit M code is issued, the intersection point vectors will be created as usual.

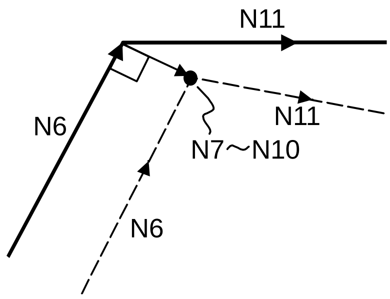
```
N6  G91 X100.   Y200. ;
N7  G04 X   P1000 ;........ Block without movement
N8  X200. ;
```



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.

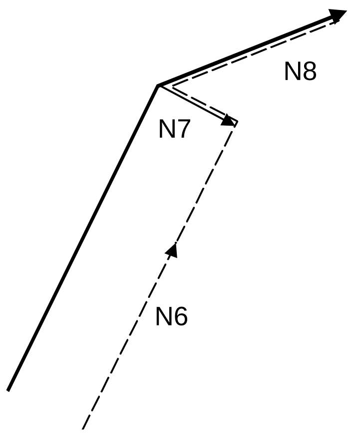
```
N6    X100.      Y200. ;
N7    G4   X1000 ;  }
N8    F100 ;        }
N9    S500 ;        }  Block without movement
N10   M4 ;          }
N11   W100. ;       }
```

In this case, a cut results.



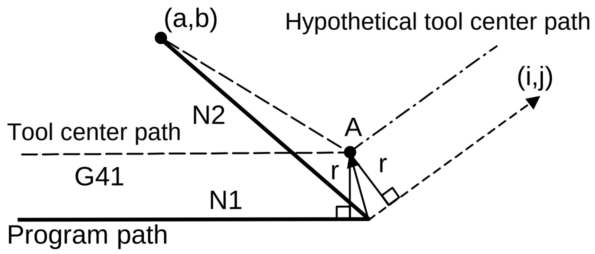
(3) When commanded together with compensation cancel

```
N6   X100. Y200.   ;
N7   G40   M5 ;
N8   X100. Y50.   ;
```



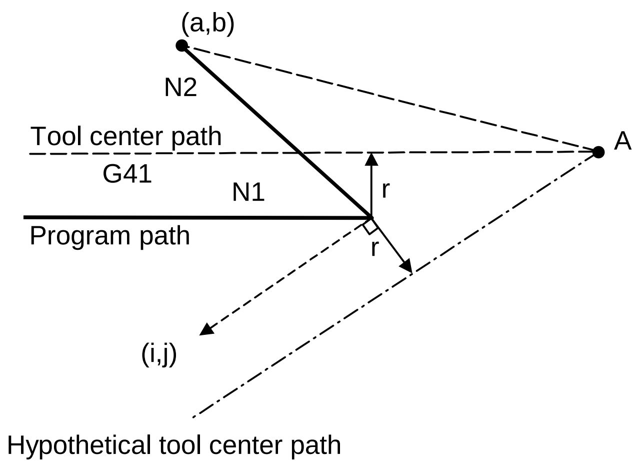
When I, J, K are commanded in G40

(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.



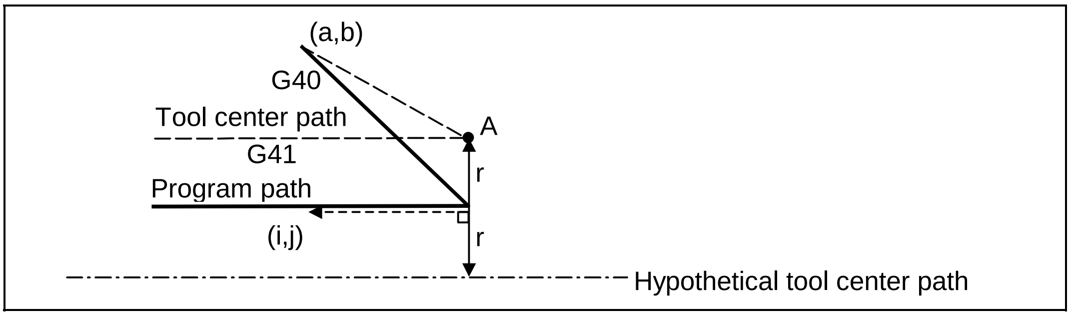
N1 (G41) G1X_ ;
N2 G40XaYbIiJj;

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.

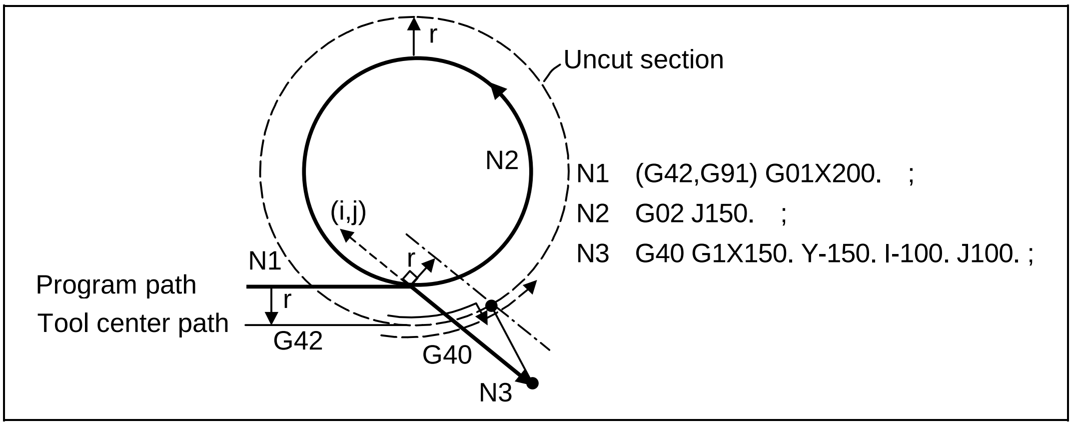


When the I and j symbols in the above program example are incorrect

If the compensation vector obtained with point of intersection calculation is extremely large, a perpendicular vector will be created in the block before G40.

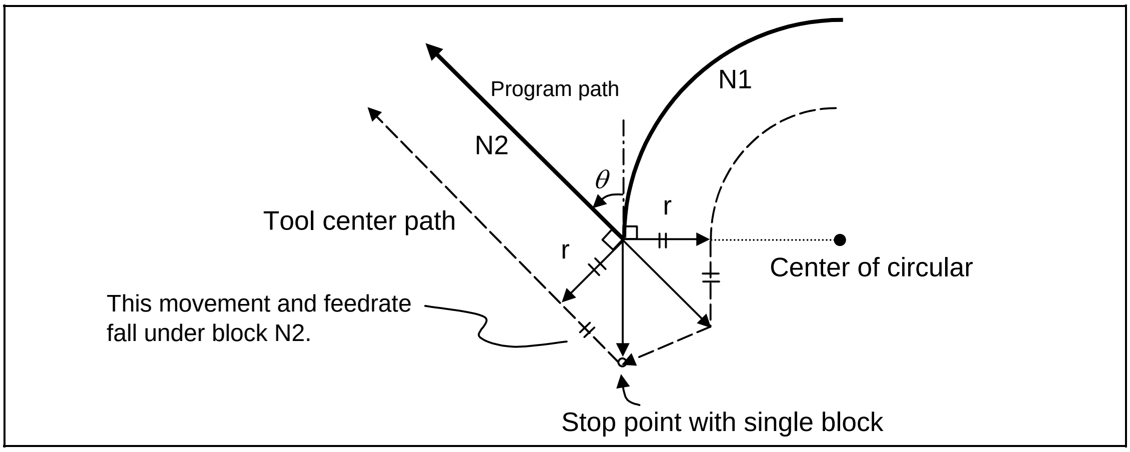


(2) If the arc is 360° or more due to the details of I, J and K at G40 after the arc command, an uncut section will occur.



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

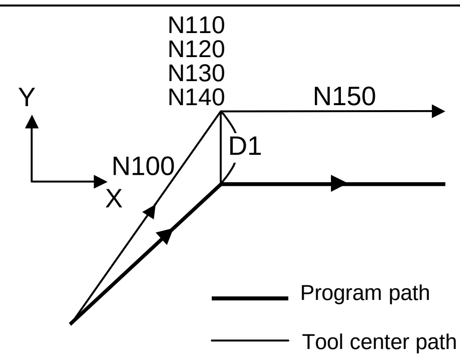
G17 (XY plane) G41/G42 X__ Y__ I__ J__ ;
G18 (ZX plane) G41/G42 X__ Z__ I__ K__ ;
G19 (YZ plane) G41/G42 Y__ Z__ J__ K__ ;

Assign an linear command (G00, G01) in a movement mode.

I, J type vectors (G17 XY plane selection)

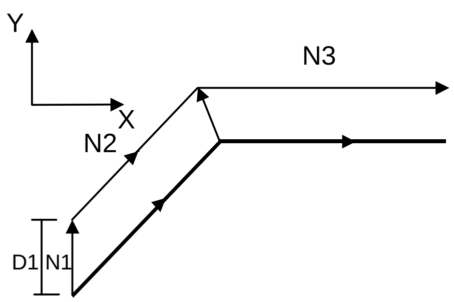
The new I, J type vector (G17 plane) created by this command is now described. (Similar descriptions apply to vector I, K for the G18 plane and to J, K for the G19 plane.)
As shown in the figures, the vectors with a size equivalent to the offset amount are made to serve as the I, J type compensation 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



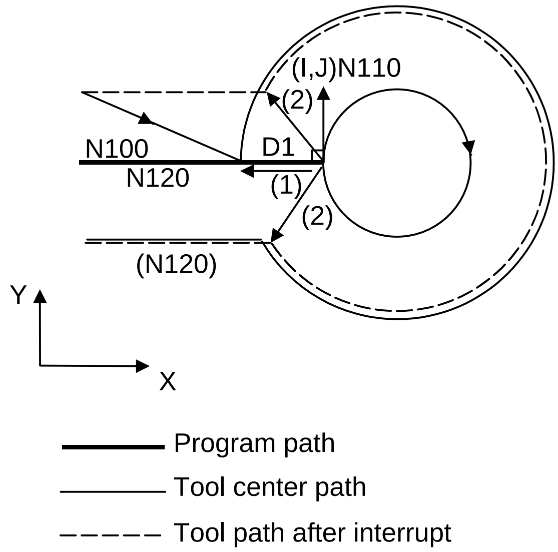

```
(G40)
N100  G91  G41  X100. Y100.  I150. D1 ;
N110  G04  X1000 ;
N120  G01  F1000 ;
N130  S500 ;
N140  M03 ;
N150  X150. ;
```

(2) When there are no movement commands at the compensation start.




```
(G40)
N1  G41  I150. D1 ;
N2  G91  X100. Y100. ;
N3  X150. ;
```

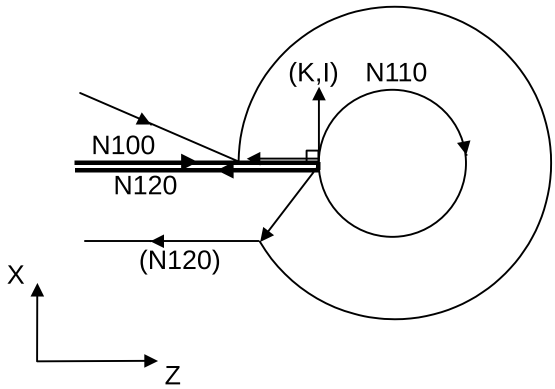
(3) When I, J has been commanded in the G41/G42 mode (G17 plane)




```
        ~
     (G17   G41   G91)
N100   G41   G00X150.   J50. ;
N110   G02   I150. ;
N120   G00   X–150. ;
        ~
```

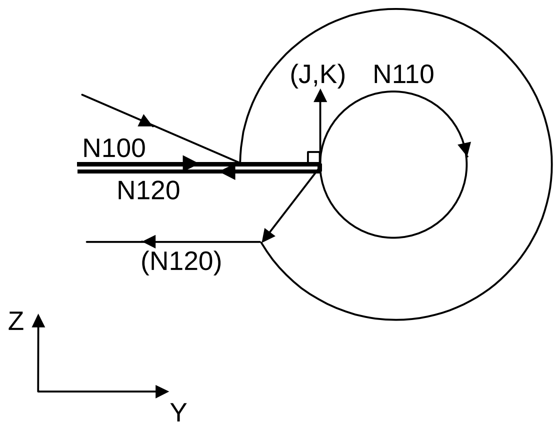
(1) I, J type vector
(2) Intersection point calculation type vector

(Reference)
(a) G18 plane



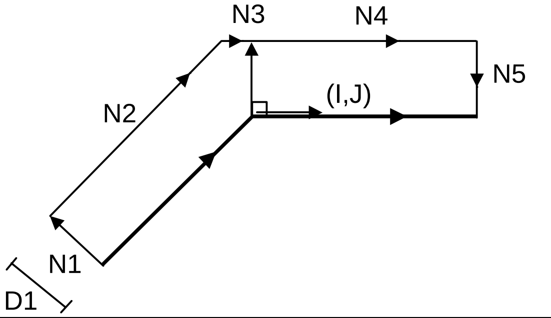

```
        ~
     (G18   G41   G91)
N100   G41   G00 Z150.   I50. ;
N110   G02   K50. ;
N120   G00   Z–150. ;
        ~
```

(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



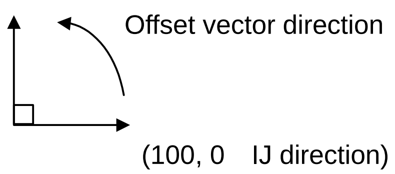

```
N1  G41  D1  G01  F1000 ;
N2  G91  X100. Y100. ;
N3  G41  I50. ;
N4  X150. ;
N5  G40 ;
```

Direction of offset vectors

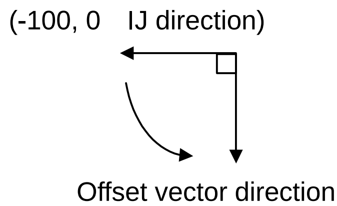
(1) In G41 mode

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.



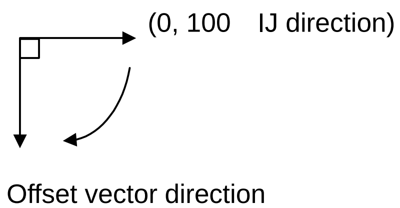
(Example 2) With I-100.



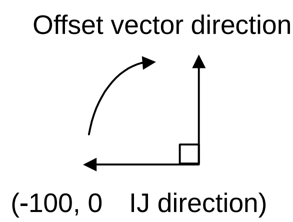
(2) In G42 mode

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.

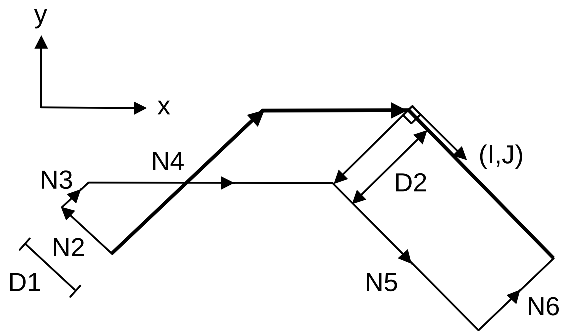


(Example 2) With I-100.



Selection of offset modal

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

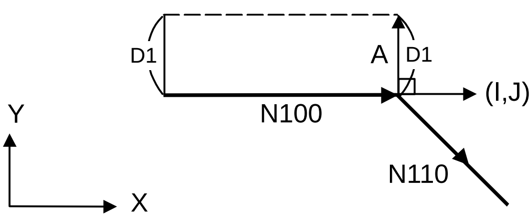



```
N1   G28   X0 Y0 ;
N2   G41   D1   F1000 ;
N3   G01   G91   X100.   Y100. ;
N4   G42   X100.   I100.   J-100.   D2 ;
N5   X100.   Y-100. ;
N6   G40 ;
N7   M02 ;
%
```

Offset amount for offset vectors

The offset amounts are determined by the offset number (modal) in the block with the I, J designation.

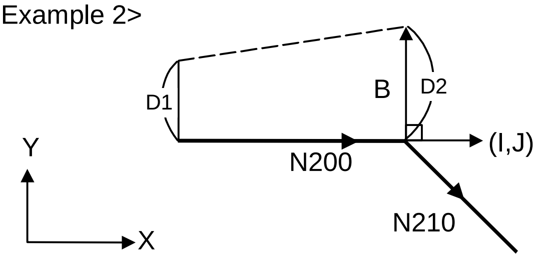
< Example 1>




```
(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>



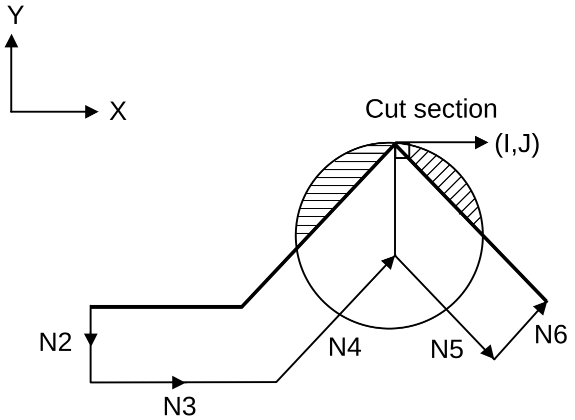

```
(G41   D1   G91)
        ≀
N200   G41   X150.   I50.   D2 ;
N210   X100.   Y-100. ;
        ≀
```

Vector B is the offset amount entered in offset number modal D2 in the N200 block.

Precautions

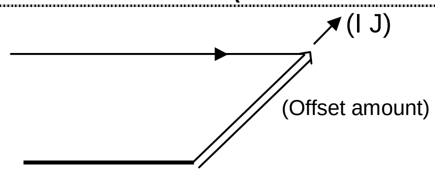
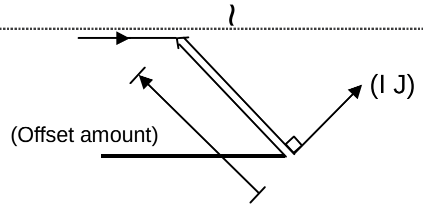
(1) Issue the I, J type vector in a linear mode (G0, G1). If it is issued in an arc mode at the start of compensation, program error (P151) will result.
An IJ designation in an arc mode functions as an arc center designation in the offset mode.

(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.



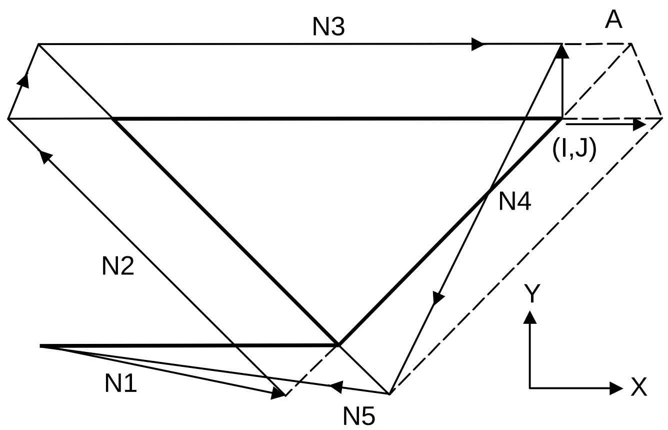

```
N1  G28  X0Y0 ;
N2  G42  D1   F1000 ;
N3  G91  X100. ;
N4  G42  X100.   Y100. I10. ;
N5  X100.   Y-100. ;
N6  G40 ;
N7  M02 ;
```

(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.

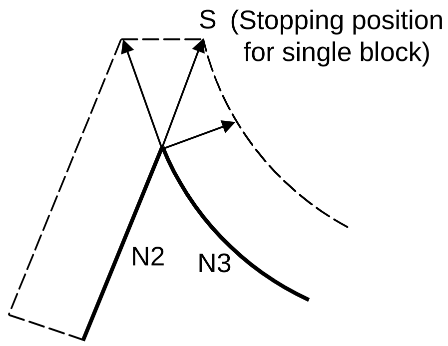
12.3.4 Interrupts during tool radius compensation

MDI interrupt

Tool radius compensation is valid in any automatic operation mode-whether memory or MDI operation.
An interrupt based on MDI will give the result as in the figure below after block stop during 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;

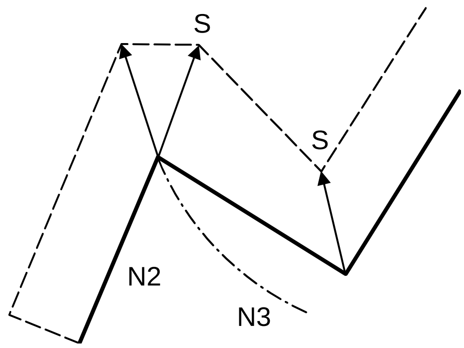


(2) Interrupt with movement

The offset vectors are automatically re-ca
lculated at the movement block after interrupt.

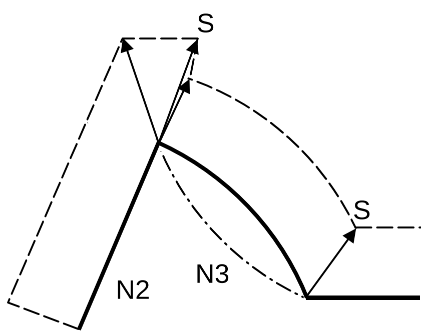
With linear interrupt

N1 G41D1;
N2 X20. Y50. ; MDI interrupt
N3 G3 X40.Y-40. R70. ; X50. Y-30. ;
X30. Y50. ;



With circular interrupt

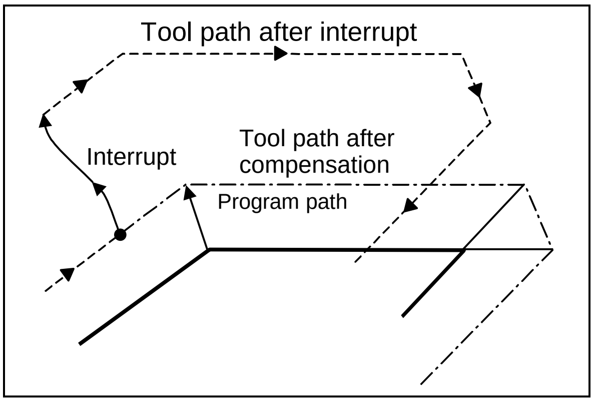
N1 G41 D1 ;
N2 X20. Y50. ; MDI interrupt
N3 G3 X40. Y-40. R70.; G2 X40. Y-40. R70. ;
G1 X4. ;



Manual interrupt

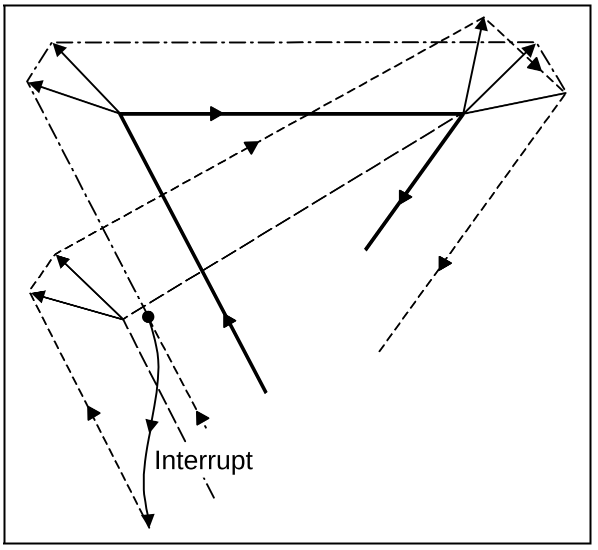
(1) Interrupt with manual absolute OFF.

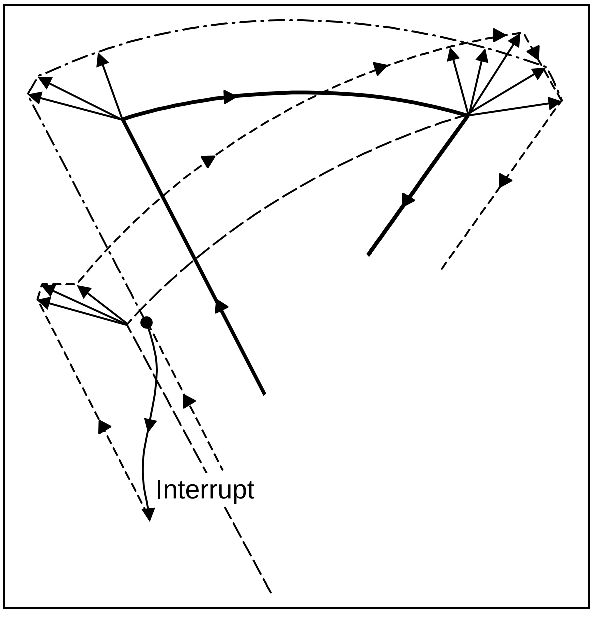
The tool path is shifted by an amount equivalent to the interrupt amount.



(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.





12.3.5 General precautions for tool radius compensation

Precautions

(1) Designating the offset amounts

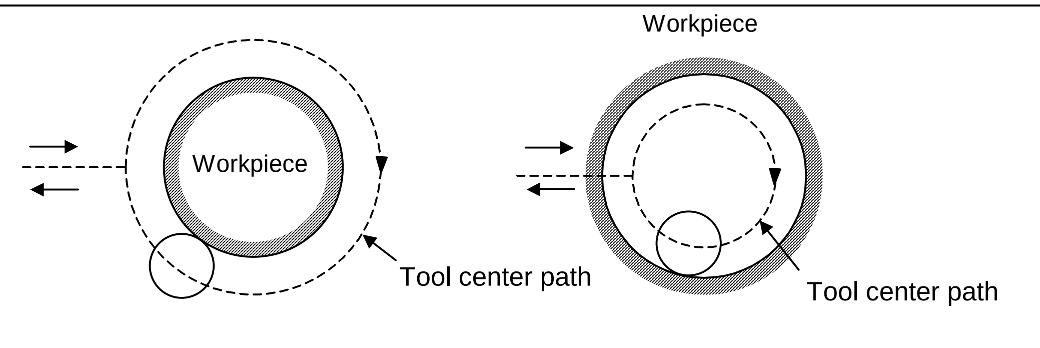
The offset amounts can be designated with the D code by designating an offset amount No. Once designated, the D code is valid until another D code is commanded. If an H code is designated, the program error (P170) No COMP No will occur.
Besides being used to designate the offset amounts for tool radius 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 (–)
(a)

G41 offset amount (–) or G42 offset amount (+)
(b)

12.3.6 Changing of offset No. during compensation mode

Function and purpose

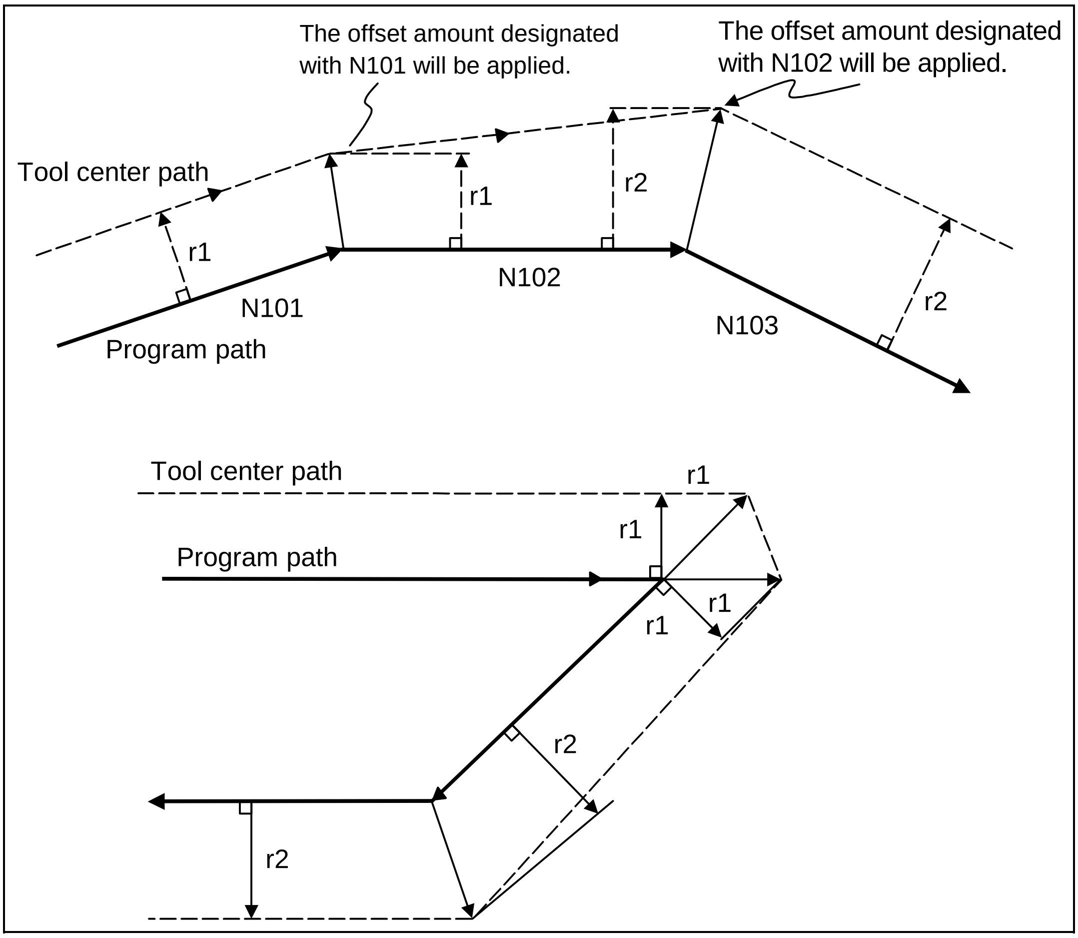
As a principle, the offset No. must not be changed during the compensation mode. If changed, the movement will be as shown below.
When offset No. (offset amount) is changed:

G41 G01 .... Dr1 ;

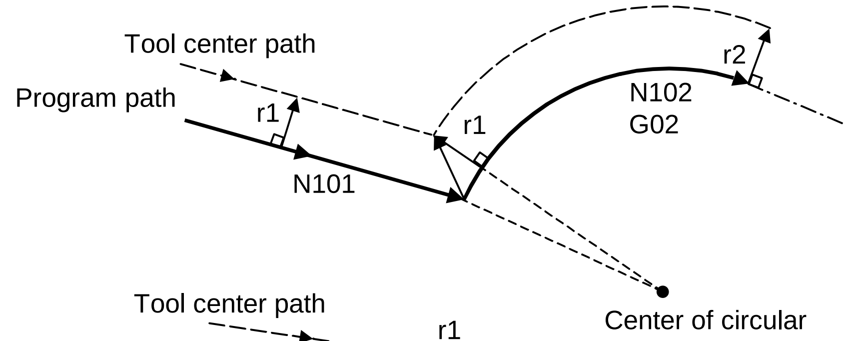
α = 0, 1, 2, 3

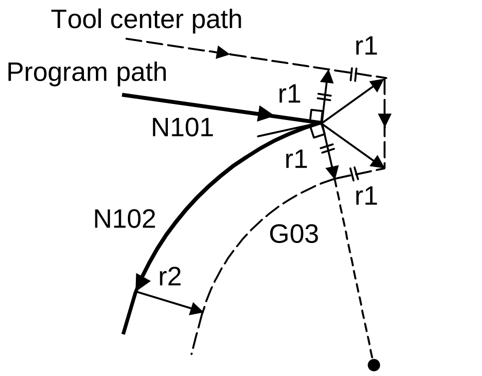
N101	G0α	Xx1	Yy1	;	
N102	G0α	Xx2	Yy2	Dr2 ;	.... Offset No. changed
N103		Xx3	Yy3	;	

(1) During linear → linear

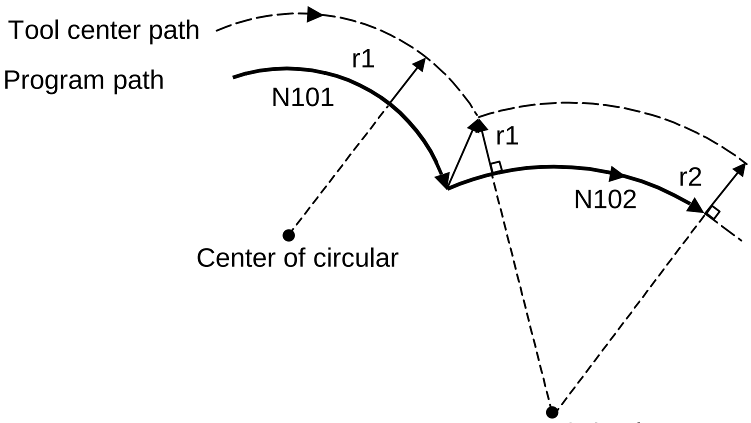


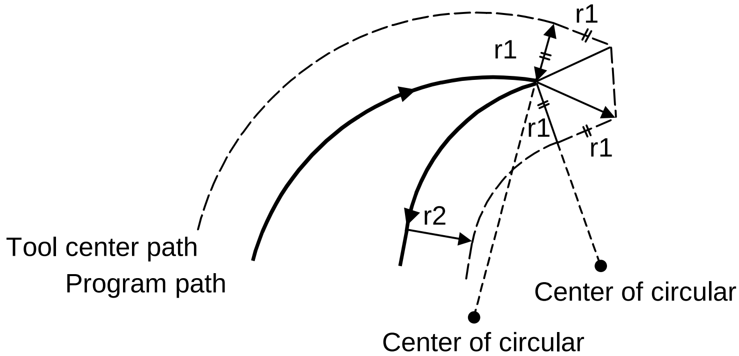
(2) Linear → circular





(3) Circular → circular





12.3.7 Start of tool radius compensation and Z axis cut in operation

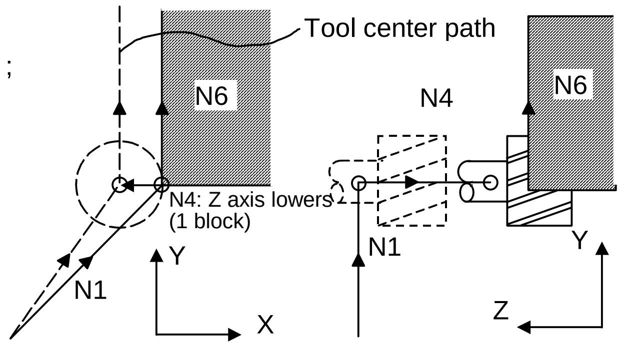
Function and purpose

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

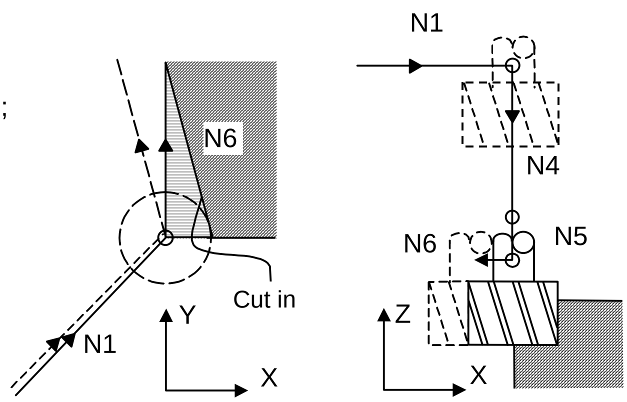
When the following type of program is created:

```
N1   G91 G00 G41 X 500. Y 500. D1 ;
N2   S1000    ;
N3   M3    ;
N4   G01    Z-300.    F1    ;
N6              Y 100.    F2    ;
                 •
                 •
                 •
                 •
```



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

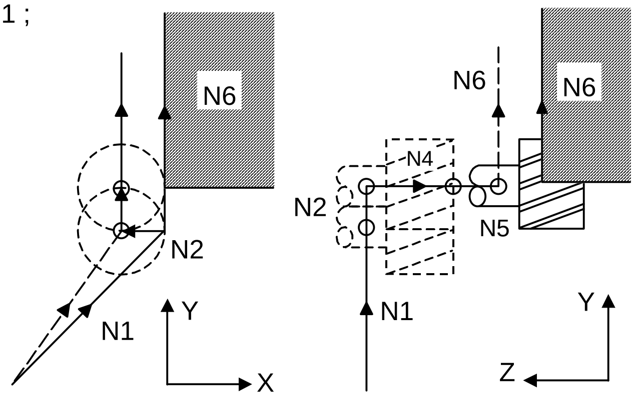
```
N1   G91 G00 G41 X 500. Y 500. D1;
N2   S1000 ;
N3   M3 ;
N4         Z-250. ;
N5   G01 Z-50.      F1 ;
N6         Y 100.    F2 ;
```



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.

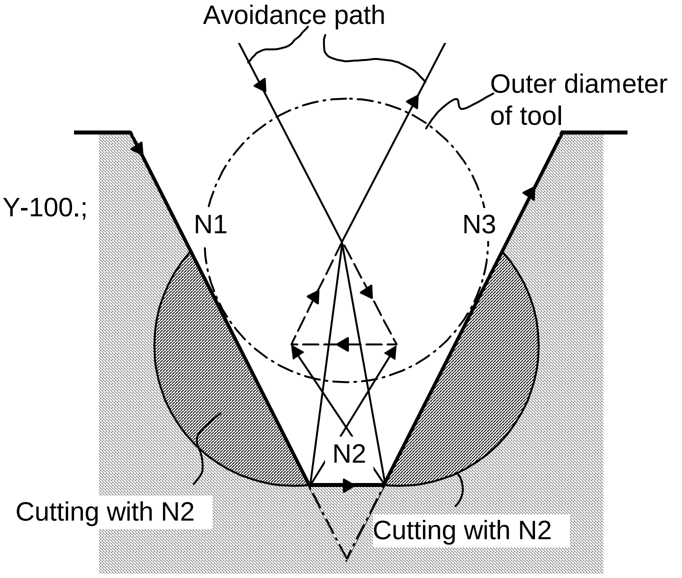
Function	Parameter	Operation
Interference check alarm function	#8102 : OFF #8103 : OFF	A program error results before the execution of the block in which the cut arises, and operation stops.
Interference check avoidance function	#8102 : ON #8103 : OFF	The tool path is changed so that workpiece is not 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

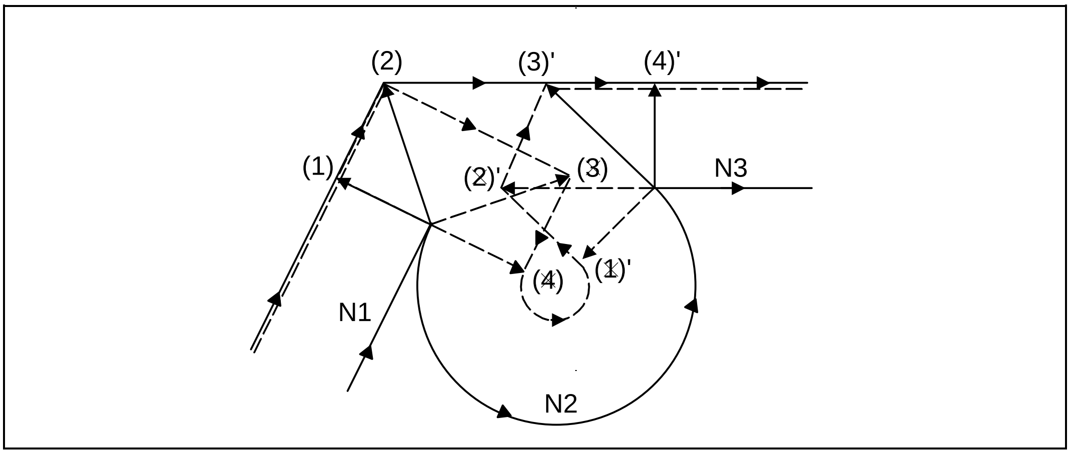
The alarm occurs before N1 is executed and so, using the edit function, N1 can be changed as below and machining can be continued :
N1 G90 G1 X20. Y–40. ;

(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.



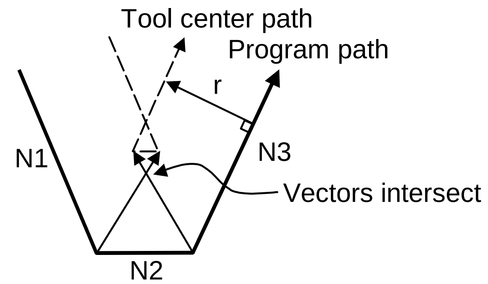
Example of interference check

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.

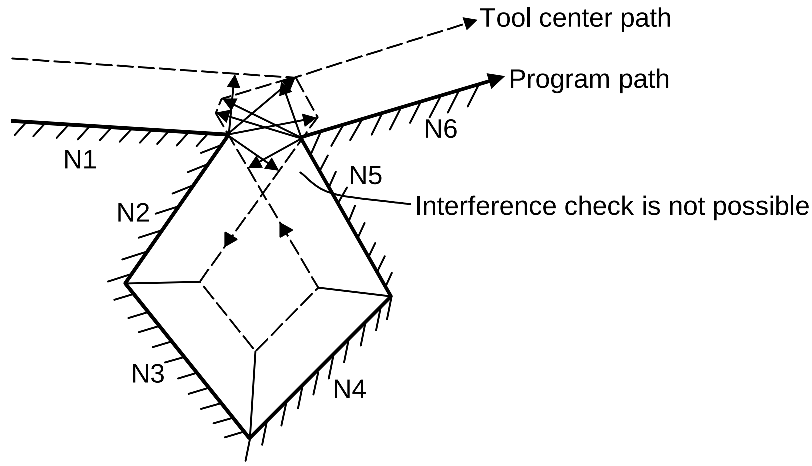
Conditions viewed as interference

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.



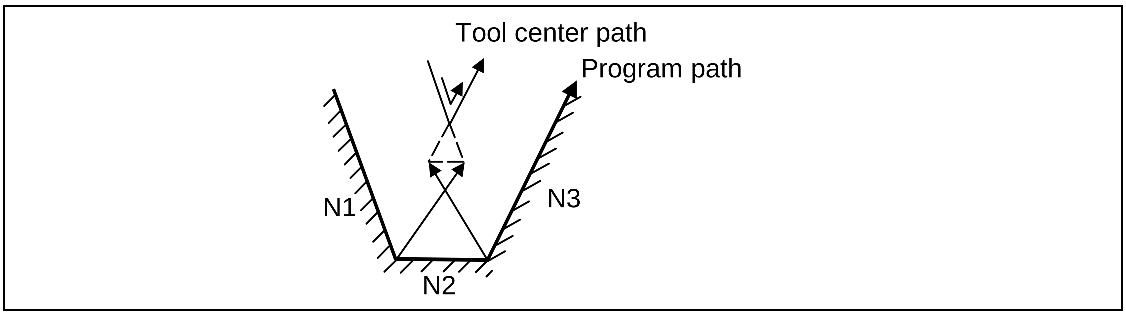
When interference check cannot be executed

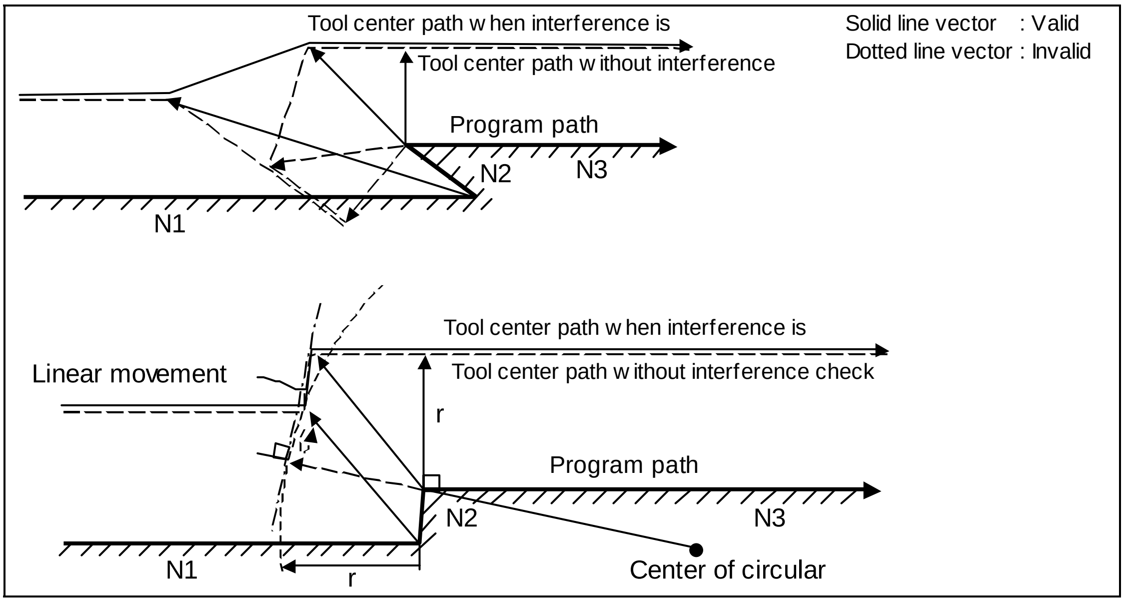
(1) When three of the movement command blocks cannot be pre-read
(When there are three or more blocks in the five pre-read blocks that do not have movement)
(2) When there is an interference following the fourth movement block

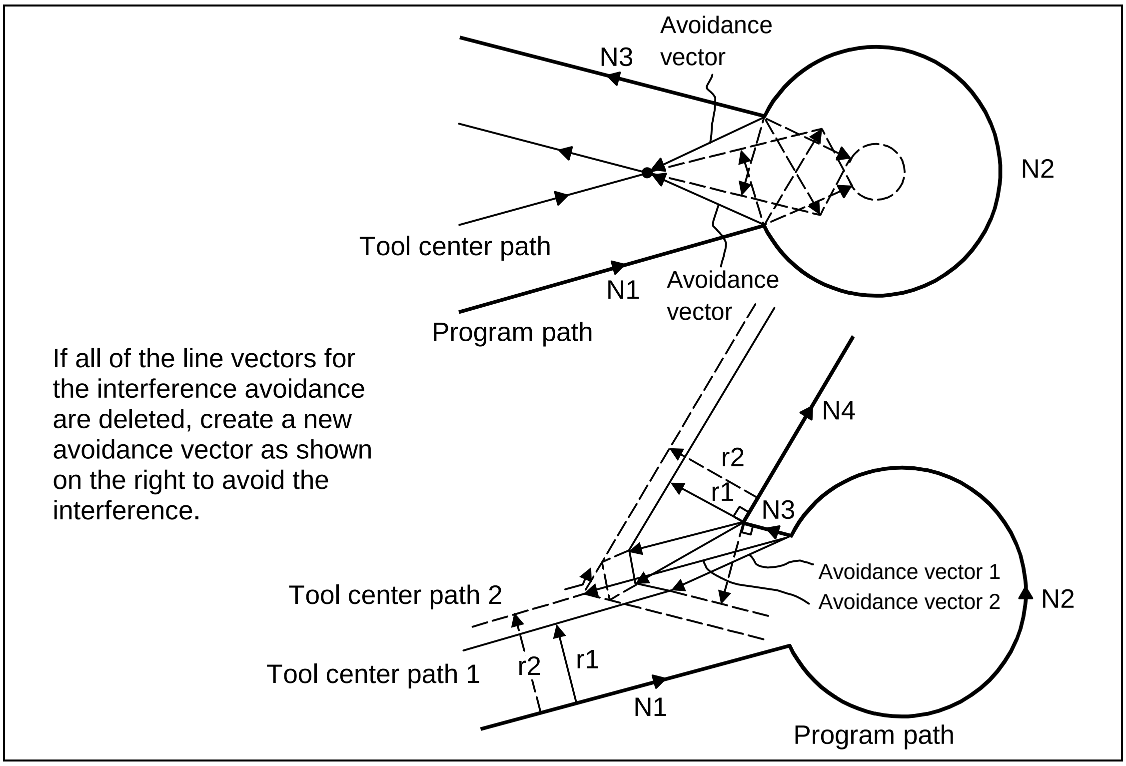


Operation during interference avoidance

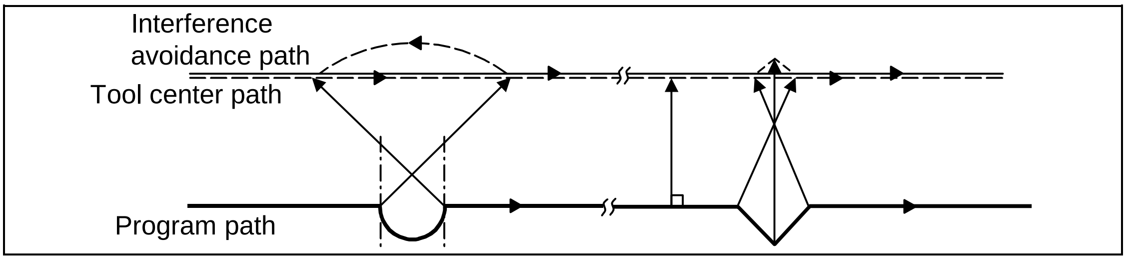
The movement will be as shown below when the interference avoidance check is used.







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



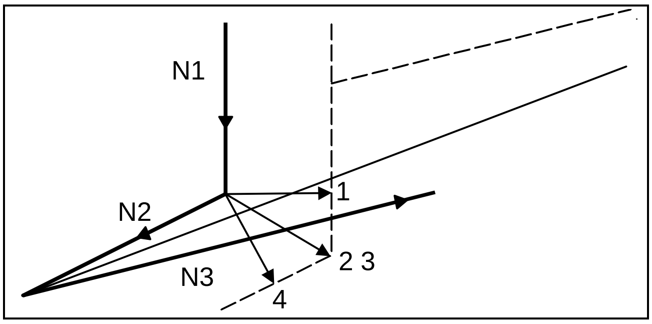
Interference check alarm

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.

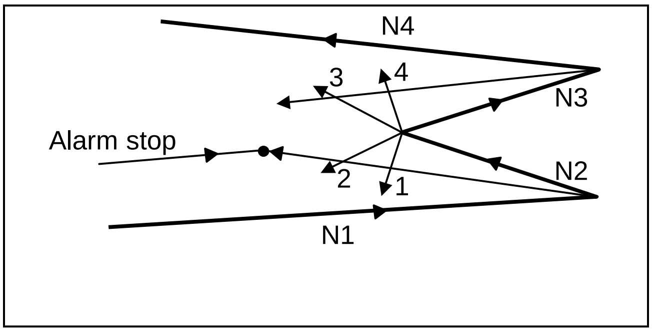
When, as shown in the figure, vectors 1 through 4 at the end point of the N1 block have all been deleted, program error (P153) results prior to N1 execution.



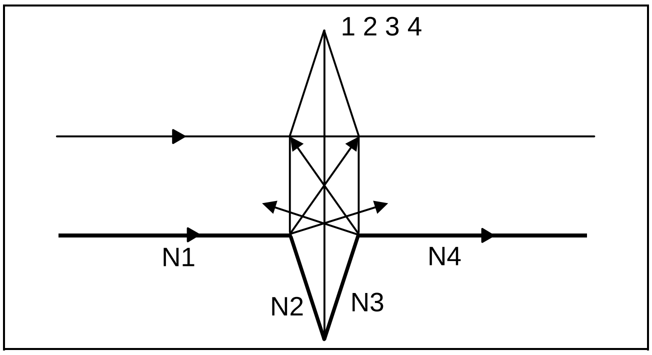
(2) When the interference check avoidance function has been selected

(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.

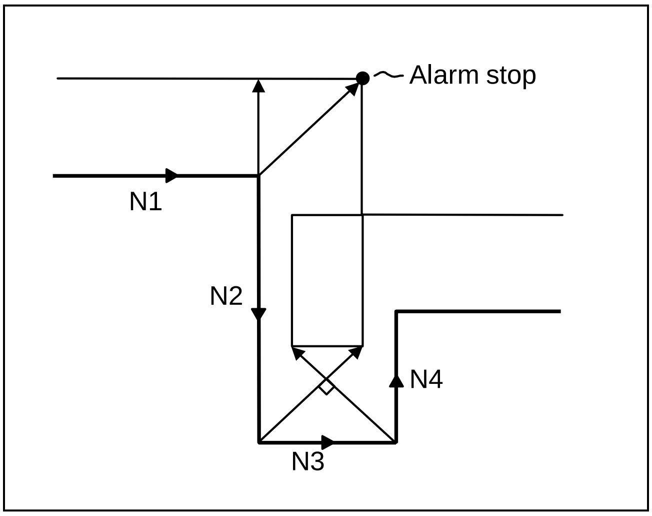


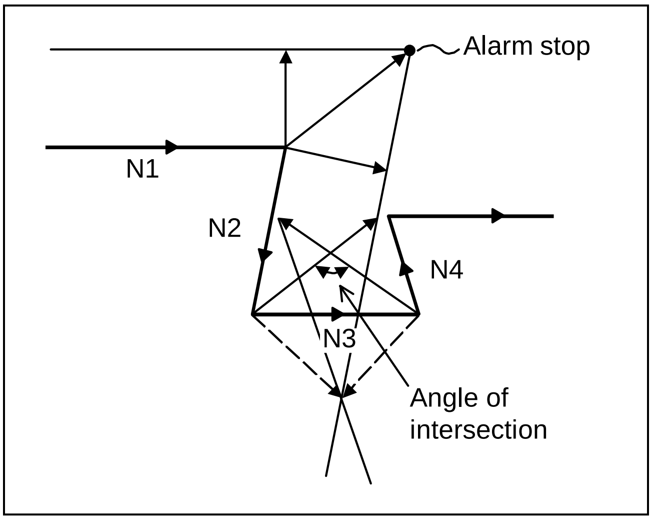
(ii) In a case such as that shown in the figure, the tool will move in the reverse direction at N2. Program error (P153) occurs after N1 execution.



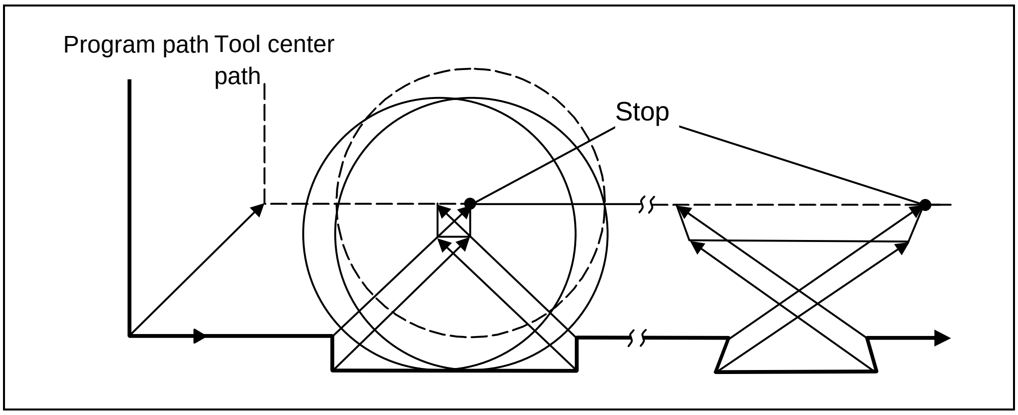
(b) When avoidance vectors cannot be created

(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 a value other than the above is set or 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__ ;	**Tool length compensation shape offset**
G10 L11 P__R__ ;	**Tool length compensation wear compensation**
G10 L12 P__R__ ;	**Tool radius shape offset**
G10 L13 P__R__ ;	**Tool radius wear compensation**

(3) Offset input cancel

```
G11 ;
```

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 fixed cycles and subprogram call commands. This will cause malfunctioning and program errors.

Example of program

(1) Input the offset amount.

```
• • • • • • ; 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)

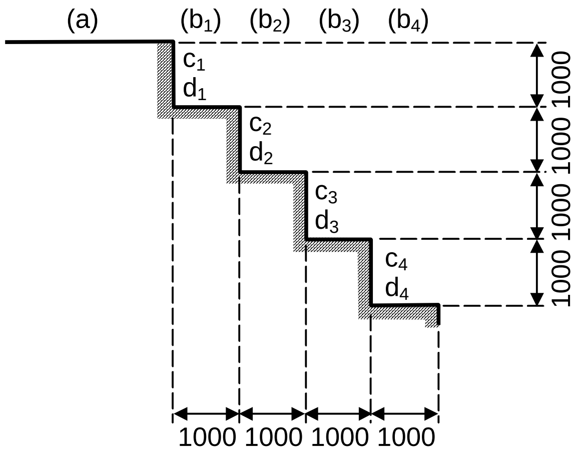
(Example 2) Assume that H10 = –1000 is already set.

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 ;

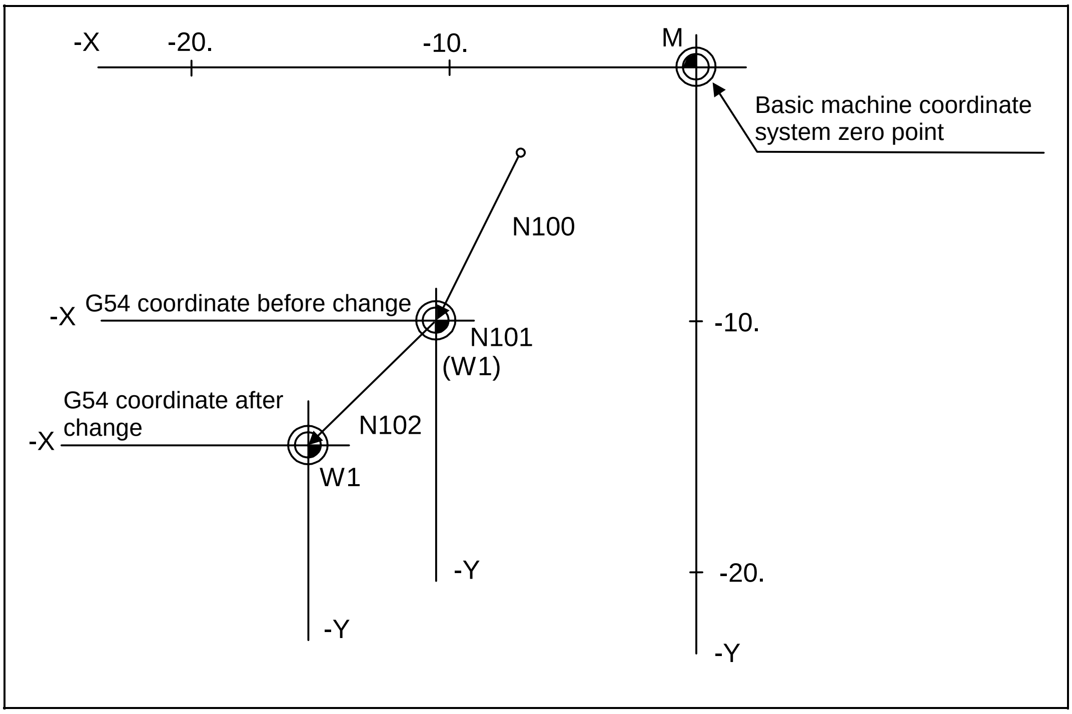
Subprogram

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 → X = +5.000
Y = 0 → Y = +5.000

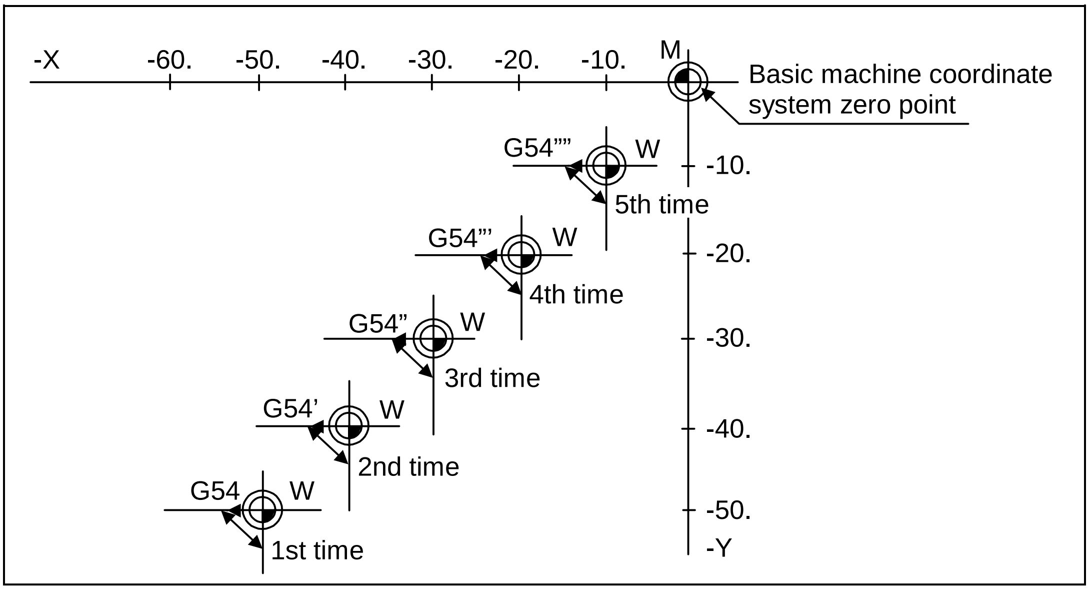
When workpiece coordinate system offset amount is set in G54 to G59

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.

13. Program Support Functions

13.1 Canned cycles

13.1.1 Standard canned cycles; G80 to G89, G73, G74, G76

Function and purpose

These standard canned 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 canned cycles.

By editing the standard canned cycle subprogram, the canned cycle sequence can be changed by the user. The user can also register and edit an original canned cycle program. For the standard canned cycle subprogram, refer to the list of the canned cycle subprogram in the appendix of the operation manual. The list of canned 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 canned 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.

Command format

G8Δ (G7Δ) X__ Y__ Z__ R__ Q__ P__ F__ L__ S__ , S __ ,R __ ,I__ ,J__;	
G8Δ (G7Δ)	: Hole machining mode
X__ Y__ Z__	: Hole positioning data
R__ Q__ P__ F__	: Hole machining data
L__	: Number of repetitions
S__	: Spindle rotation speed
, S__	: Spindle rotation speed at during retract
, 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, synchronization changeover (or spindle rotation speed at during retract), positioning axis in-position width and drilling axis in-position width.

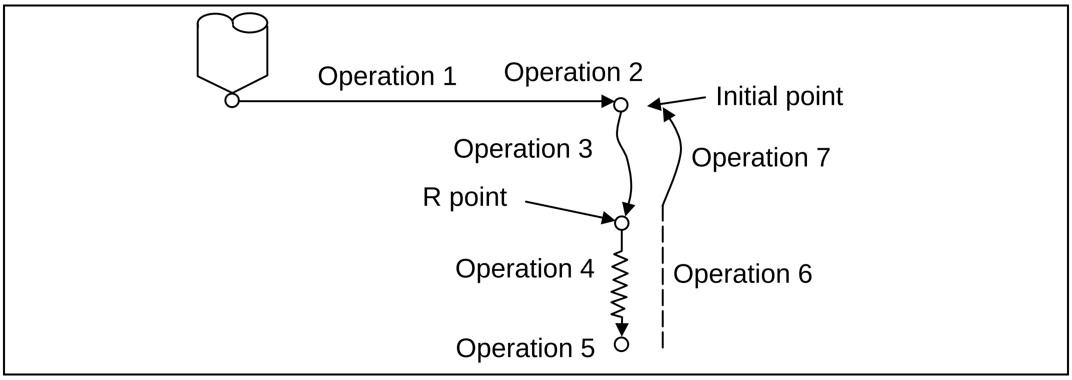
Detailed description

(1) Data outline and corresponding address

(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	: Machining data actually used for machining (modal)
(d) No. of repetitions	: Number of times to carry out drilling machining (unmodal)
(e) Synchronization changeover	: Command for selecting synchronous/asynchronous tapping during G84/G74 tapping (modal)

(2) If M00 or M01 is commanded in the same block as the canned cycle or during the canned cycle mode, the canned cycle will be ignored. Instead, M00 and M01 will be output after positioning. The canned 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 canned 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.

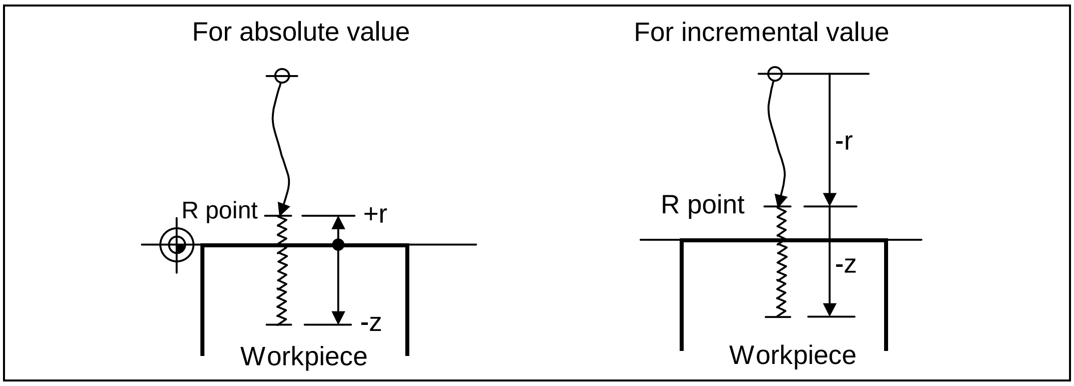
Whether the canned cycle is to be completed at operation 6 or 7 can be selected by the following G commands.
G98 .... Initial level return
G99 .... R point level return

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 canned cycled.

(4) Canned 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
L	Designation of number of repetitions. 0 to 9999
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 retract
, R	Synchronous/asynchronous tap cycle selection

(5) Difference between absolute value command and incremental value command



(6) Feed rate for tapping cycle and tapping retract
The feed rates for the tapping cycle and tapping retract are as shown below.

(a) Selection of synchronous tapping cycle/asynchronous tapping cycle

<table>
<tr><th>Program G84×××, Rxx</th><th>Control parameter
Synchronous tapping</th><th>Synchronous/
asynchronous</th></tr>
<tr><td>, R00</td><td>—</td><td rowspan="2">Asynchronous</td></tr>
<tr><td rowspan="2">, Rxx
No designation</td><td>OFF</td></tr>
<tr><td>ON</td><td rowspan="2">Synchronous</td></tr>
<tr><td>, R01</td><td>—</td></tr>
</table>

– is irrelevant to the setting

(b) Selection of asynchronous tapping cycle feed rate

<table>
<tr><th>G94/G95</th><th>Control parameter
F1-digit value</th><th>F command value</th><th>Feed designation</th></tr>
<tr><td rowspan="3">G94</td><td>OFF</td><td>F designation</td><td rowspan="2">Per-minute feed</td></tr>
<tr><td rowspan="2">ON</td><td>Other than F0 to F8</td></tr>
<tr><td>F0 to F8 (no decimal point)</td><td>F1-digit feed</td></tr>
<tr><td>G95</td><td>—</td><td>F designation</td><td>Per-revolution feed</td></tr>
</table>

– is irrelevant to the setting

(c) Spindle rotation speed during retract of synchronous tapping cycle

Address	Meaning of address	Command range (unit)	Remarks
,S	Spindle rotation speed during retract	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 retract. If the spindle rotation speed is not 0 during retract, the tap retract override value will be invalid.

Positioning plane and hole drilling axis

The canned 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.

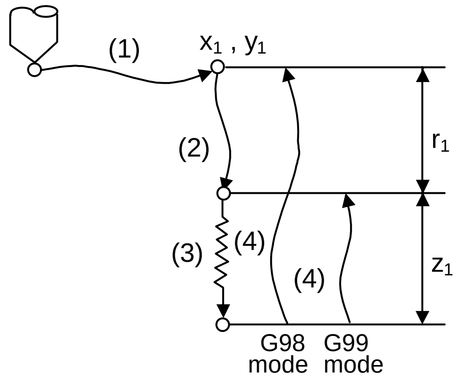
(Example 1) When G17 (XY plane) is selected, and the axis parallel to the Z axis is set as the W axis.

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 canned cycle canceled state.

In the following explanations on the movement in each canned 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.

(a) G81 (Drilling, spot drilling)
Program
G81 Xx1 Yy1 Zz1 Rr1 Ff1 ,Ii1 ,Jj1;

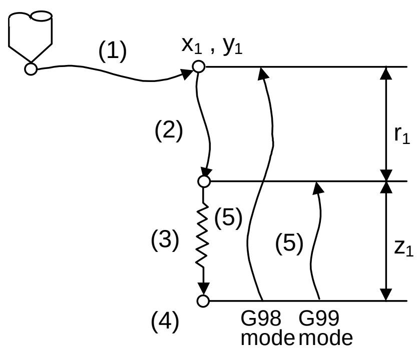


(1) G0 Xx_1 Yy_1
(2) G0 Zr_1
(3) G1 Zz_1 Ff_1
(4) { G98 mode G0Z – (z_1+r_1)
G99 mode G0Z – z_1

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

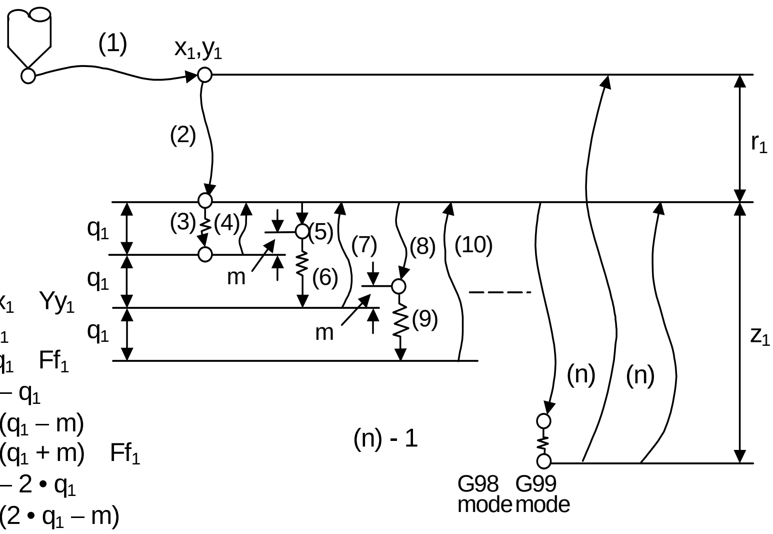
The operation stops at after the (1), (2) and (5) commands during single block operation.

(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 n". Program so that q_1>m.

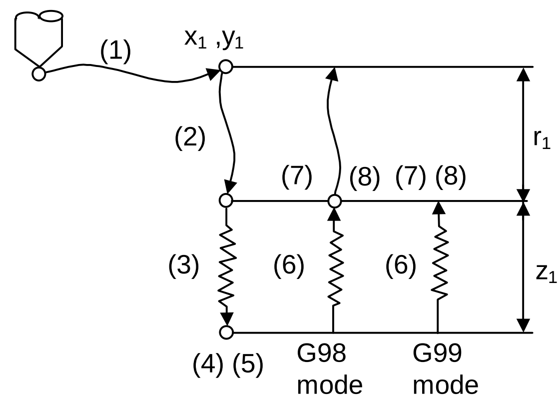
The operation stops at after the (1), (2) and (n) commands during single block operation.

(d) G84 (Tapping cycle)

Program

G84 Xx_1 Yy_1 Zz_1 Rr_1 Ff_1 Pp_1 Ss_1 ,Ss_2 ,Rr_2 ,Ii_1 ,Jj_1;

P : Dwell designation



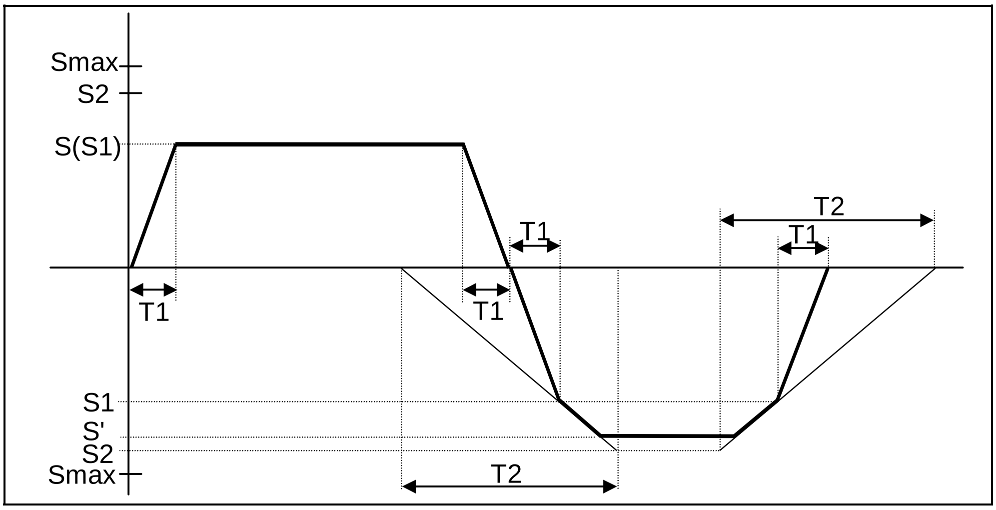
(1) G0 Xx_1 Yy_1
(2) G0 Zr_1
(3) G1 Zz_1 Ff_1
(4) G4 Pp_1
(5) M4 (Spindle reverse rotation)
(6) G1 $Z - z_1$ Ff_1
(7) G4 Pp_1
(8) M3 (Spindle forward rotation)
(9) G98 mode G0Z – r_1
G99 mode No movement

Operation pattern	i1	j1
(1)	Valid	–
(2)	–	Invalid
(3)	–	Invalid
(4)	–	–
(5)	–	–
(6)	–	Invalid
(7)	–	–
(8)	–	–
(9)	–	Valid

- When r2 = 1, the synchronous tapping mode will be entered, and when r2 = 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.

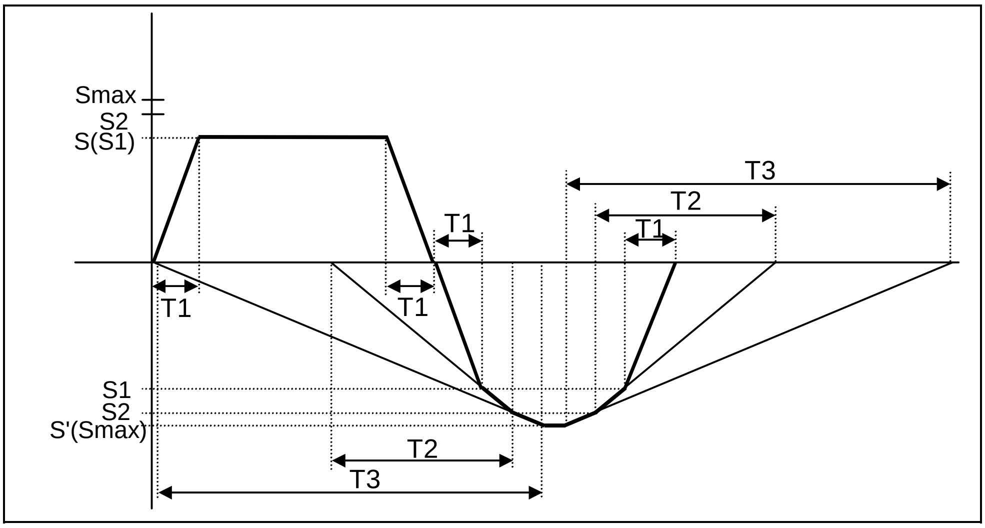
This function allows spindle acceleration/deceleration pattern to be approached to the speed loop acceleration/deceleration pattern by dividing the spindle and drilling axis acceleration/deceleration pattern into up to three stages during synchronous tapping.
The acceleration/deceleration pattern can 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



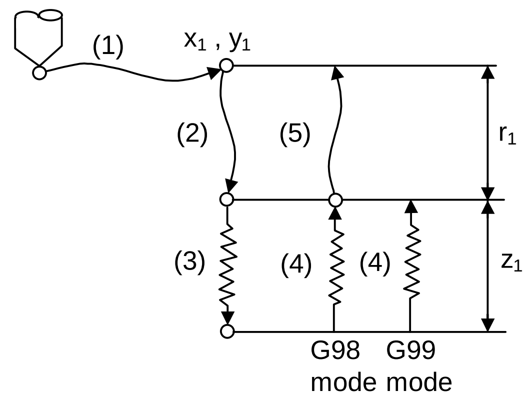
S	: Command spindle rotation speed
S'	: Spindle rotation speed during return
S1	: Tap rotation speed (spindle base specification parameters #3013 to #3016)
S2	: Synchronous tap changeover spindle rotation speed 2 (spindle base specification parameters #3037 to #3040)
Smax	: Maximum rotation speed (spindle 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)

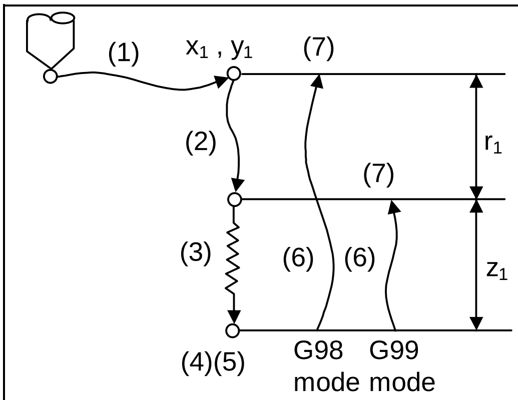
(e) G85 (Boring)
Program
G85 Xx_1 Yy_1 Zz_1 Rr_1 Ff_1 ;



(1) G0 Xx_1 Yy_1
(2) G0 Zr_1
(3) G1 Zz_1 Ff_1
(4) G1 $Z - z_1$ Ff_1
(5) G98 mode G0Z $- r_1$
G99 mode No movement

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) G98 mode G0Z $- (z_1+r_1)$
G99 mode G0Z $- z_1$
(7) M3 (Spindle forward rotation)

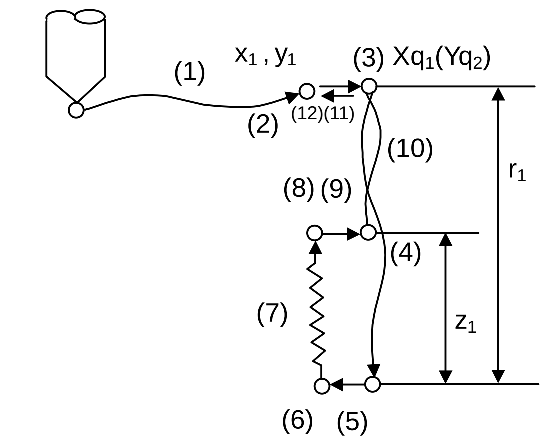
The operation stops at after the (1), (2) and (7) commands during single block operation.

(g) G87 (Back boring)

Program

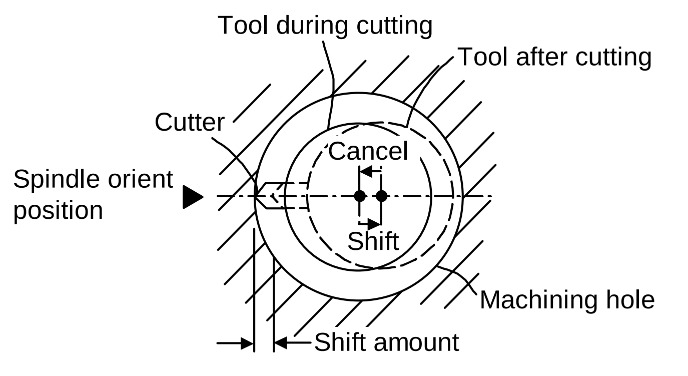
G87 Xx_1 Yy_1 Zz_1 Rr_1 Iq_1 Jq_2 Ff_1 ;

(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.



(1)	G0 Xx_1 Yy_1
(2)	M19 (Spindle orient)
(3)	G0 Xq_1 (Yq_2) (Shift)
(4)	G0 Zr_1
(5)	G1 X–q_1 (Y–q_2) Ff_1 (Shift)
(6)	M3 (Spindle forward rotation)
(7)	G1 Zz_1 Ff_1
(8)	M19 (Spindle orient)
(9)	G0 Xq_1 (Yq_2) (Shift)
(10)	G98 mode G0Z – (z_1+r_1) G99 mode G0Z – (r_1+z_1)
(11)	G0 X–q_1 (Y–q_2) (Shift)
(12)	M3 (Spindle forward rotation)

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.



For G17 : I, J
For G18 : K, I
For G19 : J, K

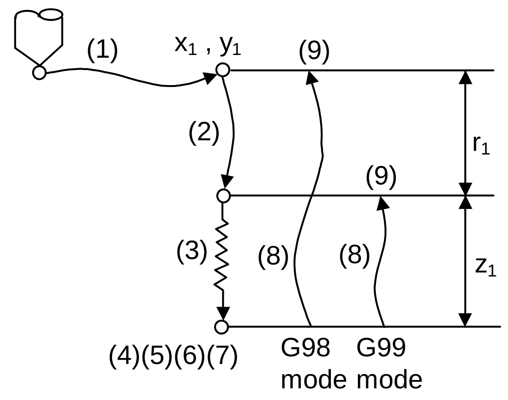
The shift amount is executed with linear interpolation, and the feed rate follows the F command.
Command I, J, and K with incremental values in the same block as the hole position data.
I, J and K will be handled as modals during the canned cycle.

(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 canned 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 ;



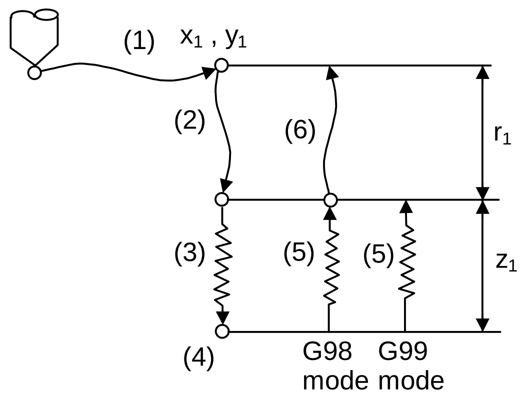
(1) G0 Xx_1 Yy_1
(2) G0 Zr_1
(3) G1 Zz_1 Ff_1
(4) G4 Pp_1
(5) M5 (Spindle stop)
(6) Stop when single block stop switch is ON.
(7) Automatic start switch ON
(8) G98 mode G0Z – (z_1+r_1)
    G99 mode G0Z – z_1
(9) M3 (Spindle forward rotation)

The operation stops at after the (1), (2), (6) and (9) commands during single block operation.

(i) G89 (Boring)

Program

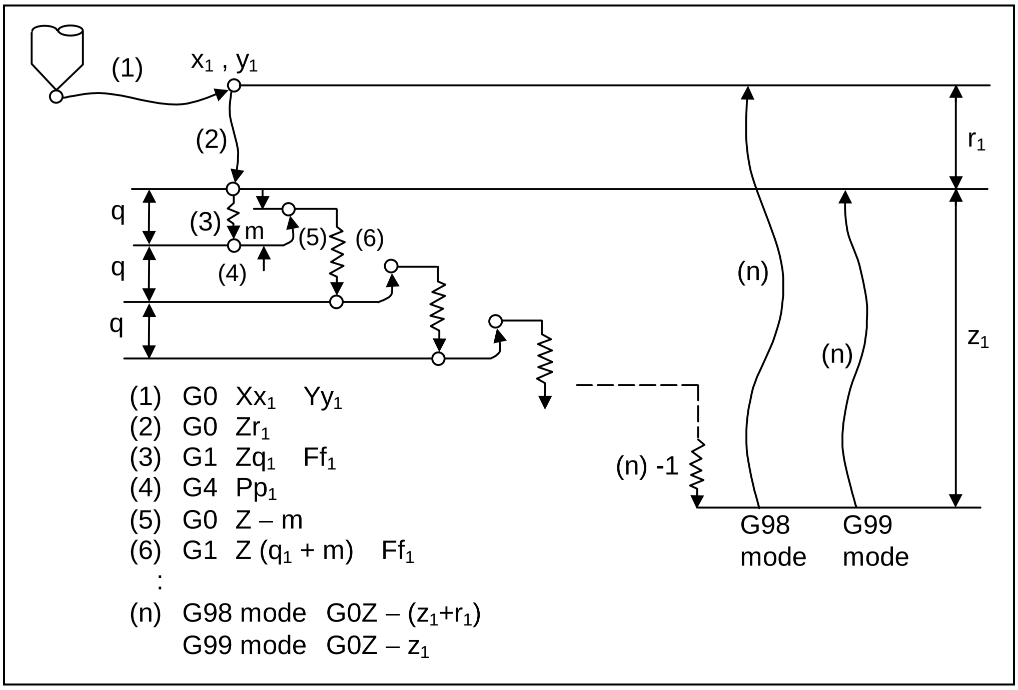
G89 Xx_1 Yy_1 Zz_1 Rr_1 Ff_1 Pp_1 ;



(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) G98 mode G0Z – r_1
    G99 mode No movement

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 ;

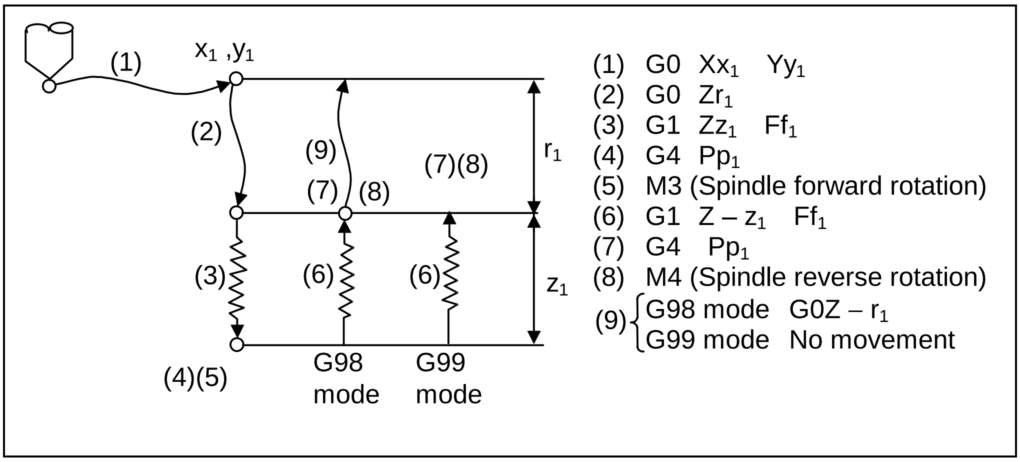


When executing a second and following cutting in the G73 as shown above, the movement will return several m mm with rapid traverse and then will change to cutting feed.
The return amount m will differ according to the parameter "#8012 G73 n".
The operation stops at after the (1), (2) and (n) commands during single block operation.

(k) G74 (Reverse tapping cycle)

Program

G74 Xx_1 Yy_1 Zz_1 Rr_1 Pp_1 Ss_1 ,Ss_2 Rr_2 ,Ii_1 ,Jj_1;



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.

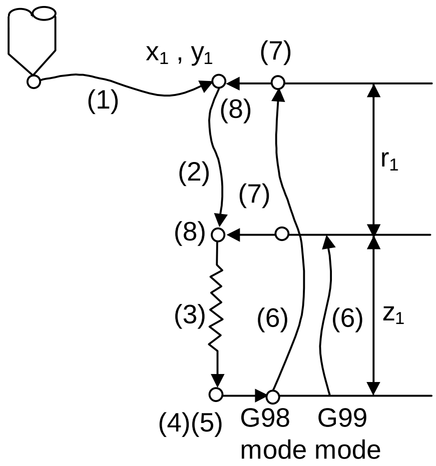
This function allows spindle acceleration/deceleration pattern to be approached to the speed loop acceleration/deceleration pattern by dividing the spindle and drilling axis acceleration/deceleration pattern into up to three stages during synchronous tap.

Refer to the item "d) G84 (Tapping cycle)" for details of multi-stages of the spindle acceleration/deceleration pattern.

(l) G76 (Fine boring)

Program

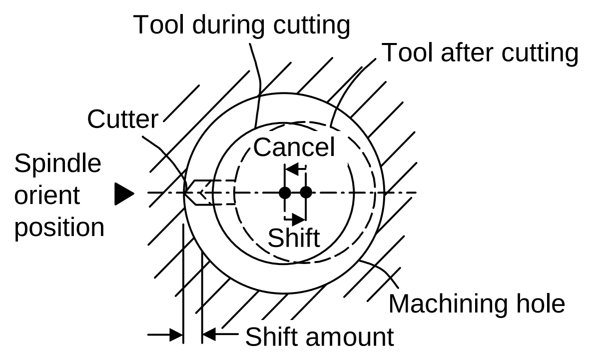
G76 Xx_1 Yy_1 Zz_1 Rr_1 Iq_1 Jq_2 Ff_1 ;



(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) G98 mode G0Z – (z_1+r_1)
G99 mode G0Z – z_1
(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 canned 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 canned cycle, and will also be used as the G83, G87 and G73 cutting amount.

Precautions for using canned cycle

(1) Before the canned 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 canned 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 canned 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 canned 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 canned cycle, the following will occur.

(Where, 00 to 03 and 33 are m, and the canned 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 canned 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 canned cycle control axis, the canned 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 canned 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___;

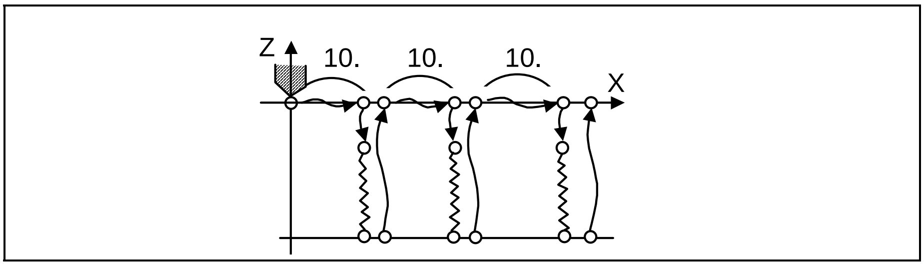
```
               Execute      Record only code having an address
```

(8) When the canned cycle is executed, only the modal command commanded in the canned cycle program will be valid in the canned cycle subprogram. The modal of the program that called out the canned cycle will not be affected.

(9) Other subprograms cannot be called from the canned cycle subprogram.

(10) Decimal points in the movement command will be ignored during the canned 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 ;



13.1.2 Initial point and R point level return; G98, G99

Function and purpose

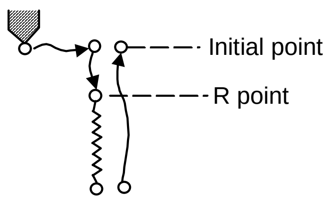
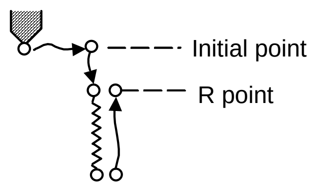
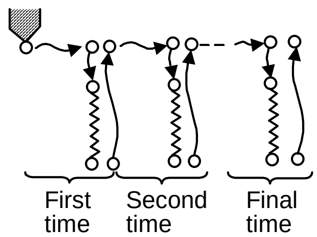
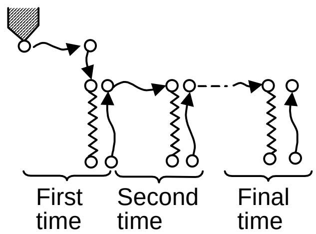
Whether to use R point or initial level for the return level in the final sequence of the canned cycle can be selected.

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;	 Initial level return is executed for all times.	

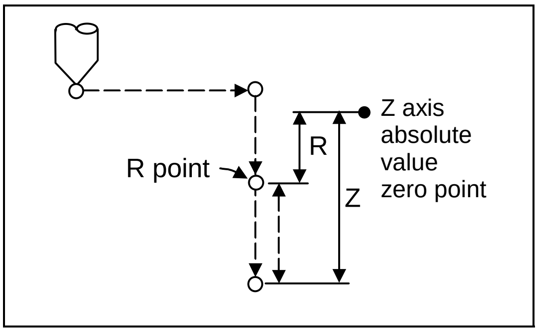
Example of program

(Example 1)

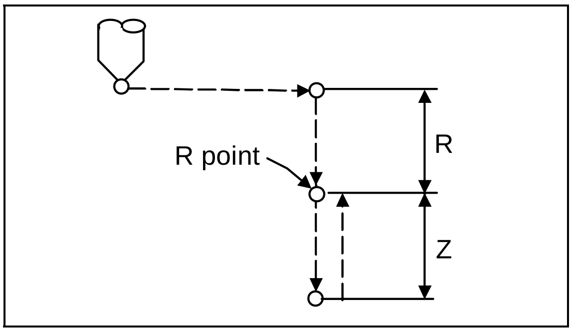
G82 Zz_1 Rr_1 Pp_1 Ff_1 L0 ;	Record only the hold machining data (Do not execute)
Xx_1 Yy_1 ;	Execute hole drilling operation with G82 mode

The No. of canned cycle repetitions is designated with L. If L1 is designated or L not designated, the canned 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 canned 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 canned cycle sequence
N003 Xx_2 Yy_2 ;	Change of positioning point, and execution of canned 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 canned cycle for l_5 times
N011 G98 Xx_6 Yy_6 Zz_6 Rr_6 ;	Change of positioning point, and execution of canned cycle
N012 Ww_1 ;	Execute W axis according to 01 group modal before N001, and then execute canned cycle

13.1.3 Setting of workpiece coordinates in canned 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 canned cycle; G34, G35, G36, G37.1

Function and purpose

The special canned cycle is used with the standard canned cycle. Before using the special canned cycle, program the canned cycle sequence selection G code and hole machining data to record the hole machining data. (If there is no positioning data, the canned cycle will not be executed, and only the data will be recorded.)
Even after the special canned cycle is executed, the recorded standard canned cycled will be held until canceled.
If the special canned cycle is designated when not in the canned cycle mode, only positioning will be executed, and the hole drilling operation will not be done.

Bolt hole circle (G34)

G34 X x_1 Y y_1 I r J θ K n ;

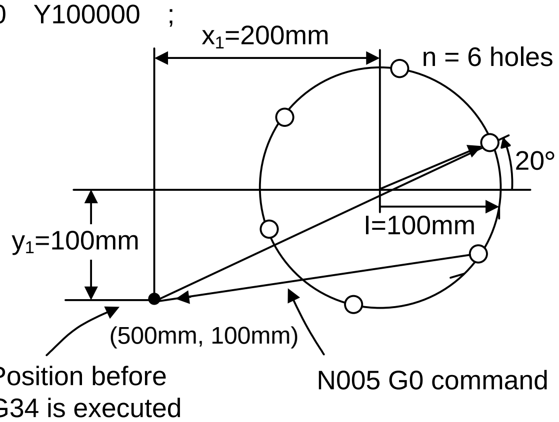
X, Y	:Positioning of bolt hole cycle center. This will be affected by G90/G91.
I	:Radius r of the circle. The unit follows the input setting unit, and is given with a positive number.
J	:Angle θ of the point to be drilled first. The CCW direction is positive. (The decimal point position will be the degree 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 canned 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  ;
```



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.

Line at angle (G35)

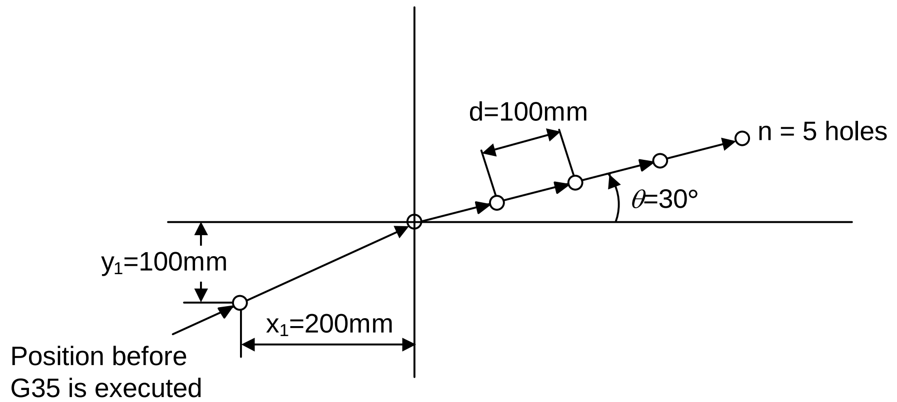
G35 X x1 Y y1 I d J θ K n ;	
X, Y	:Designation of start point coordinates. This will be affected by G90/G91.
I	:Interval d. The unit follows the input setting unit. If d is negative, the drilling will take place in the direction symmetrical to the point that is the center of the start point.
J	:Angle 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 canned 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   ;
```



(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 canned cycle will be ignored, and the G35 command will be executed.

Arc (G36)

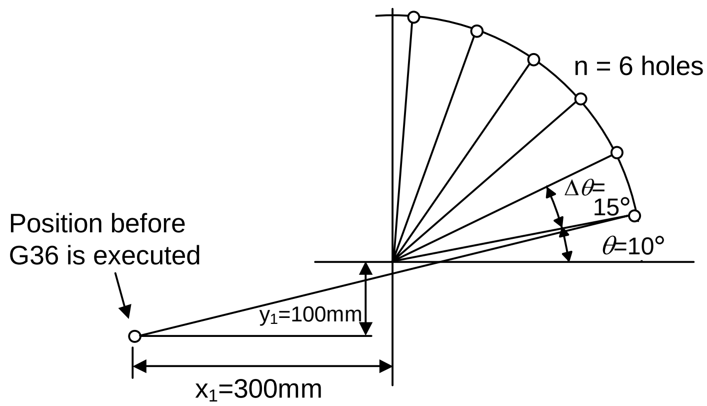
G36 X x1 Y y1 I r J θ P Δθ K n ;	
X, Y	:Center coordinates of arc. This will be affected by G90/G91.
I	:Radius r of arc. The unit follows the input setting unit, and is given with a positive No.
J	:Angle θ of the point to be drilled first. The CCW direction is positive. (The decimal point position will 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 canned 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 ;
```



Grid (G37.1)

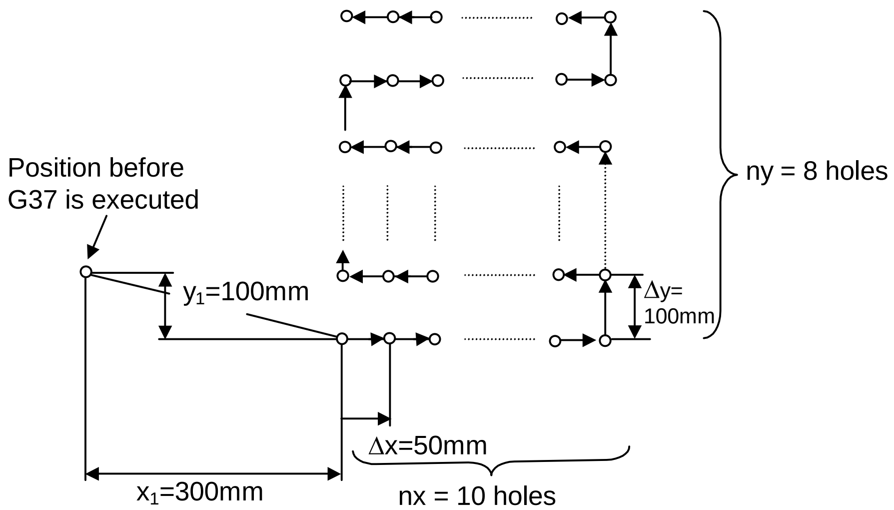
G37.1 X x1 Y y1 I Dx P nx J Dy K ny ;	
X, Y	:Designation of start point coordinates. This will be affected by G90/G91.
I	:Interval Dx of the X axis. The unit will follow the input setting unit. If Dx is positive, the interval will be in the forward direction looking from the start point, and when negative, 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 canned 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 ;
```



(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 G, L, N, X, Y, I, P, J, K, F, M, S or B is programmed in the same block as the G37.1 command, that address will be ignored.
(Example) G37.1 Xx_1 Yy_1 Ii_1 Pp_1 Jj_1 Kk_1 $\underline{Qq_1}$;
Ignore

(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.

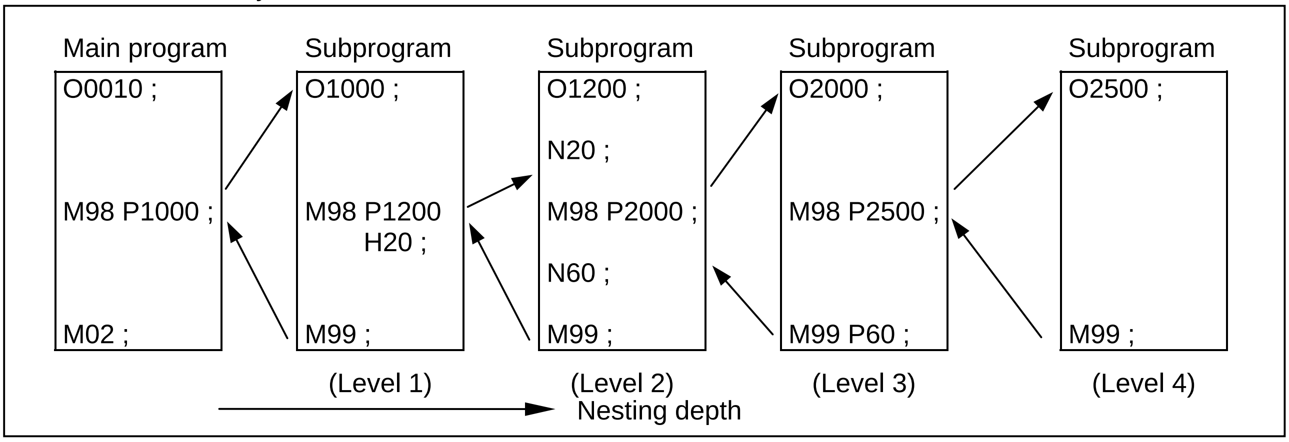
(Note 4) If there is a G72 to G89 command in the same block as the G37.1 command, the canned cycle will be ignored, and the G37.1 command will be executed.

13.3 Subprogram control; M98, M99

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.



The table below shows the functions which can be executed by adding and combining subprogram control functions and canned cycle functions.

	Case 1	Case 2	Case 3	Case 4
1. Subprogram control 2. Canned cycles	No No	Yes No	Yes Yes	No Yes
Function				
1. Memory operation	○	○	○	○
2. Subprogram call	×	○	○	×
3. Subprogram variable designation (Note 2)	×	○	○	×
4. Subprogram nesting level call (Note 3)	×	○	○	×
5. Canned cycles	×	×	○	○
6. Canned 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.

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_ H_ Q_ R_ L_ ;	
M99	Subprogram return command
P_	Sequence number of return destination (return to the block that follows the calling block if omitted)
H_	Program number of return destination (return to the main program at calling if omitted)
Q_	Sequence number to start searching of return destination (the block that follows the calling block will be handled as the search start position if omitted)
R_	Sequence number to finish searching of return destination (the block that precedes the calling block will be handled as the search finish position if omitted)
L_	Number of times after repetition number has been changed ("-1" 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.

OΔΔΔΔΔΔΔΔ	;	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.
(3) 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      ;
%
```

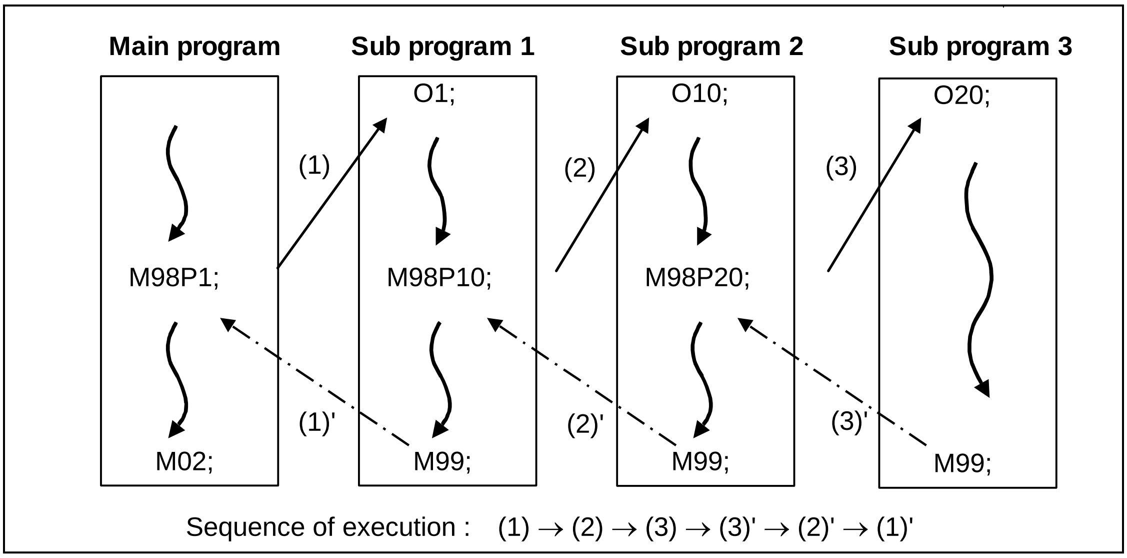
(4) Main programs can be entered in the memory or program by MDI operation but subprograms must be entered in the memory.
(5) Besides the M98 command, subprogram nesting is subject to the following commands:
- G65 Macro call
- G66 Modal call
- G66.1 Modal call
- G code call
- Miscellaneous function call (M, S, T, etc.)
- Macro interrupt
- MDI interrupt
- Automatic tool length measurement
- Multi-step skip function

(6) Subprogram nesting is not subject to the following commands which can be called even beyond the 8th nesting level.
- Canned cycles

(7) When the subprogram is to be repeatedly used, it will be repeatedly executed for I_1 times provided that "M98 Pp_1 Ll_1 ;" is programmed.

Example of program

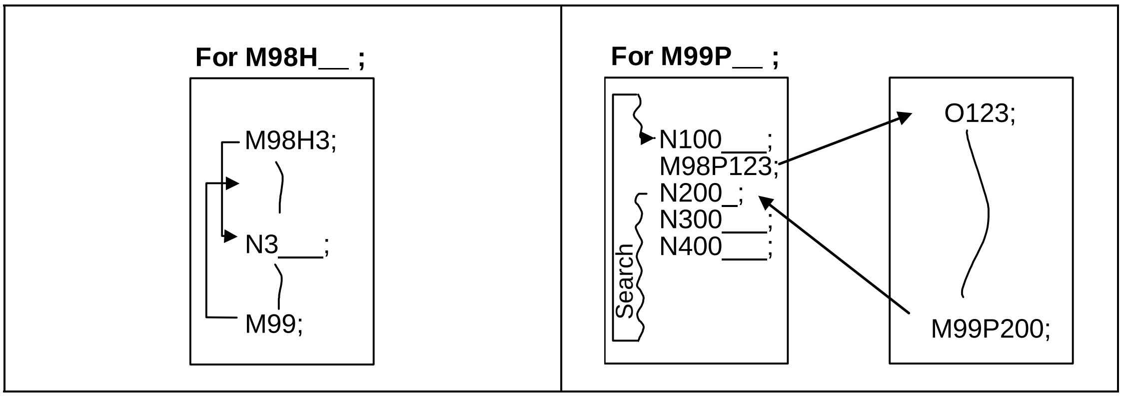
When there are 3 subprogram calls (known as 3 nesting levels)



(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.



Precautions

(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 BTR 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__; .

13.4 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	No. of variable sets option	Common variables 1 (Common to part systems)	Common variables 2 (Provided per part system)	Can be used in common throughout main, sub and macro programs.
	50 + 50 × number of part systems	#500 to #549 (50 sets)	#100 to #149 (50 sets)	
	100 + 100 × number of part systems	#500 to #599 (100 sets)	#100 to #199 (100 sets)	
	200 + 100 × number of part systems	#500 to #699 (200 sets)	#100 to #199 (100 sets)	
Local variables		1 to 33		Can be used for local variables in macro programs.
System variables		1000 to		Application is fixed by system.
Canned cycle variables		1 to 32		Local variables in canned 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".

(Note 3) The common variables are divided into the following two types.

Common variables 1 : Used in common through all part systems

Common variables 2 : Used in common in the programs of the part system

(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

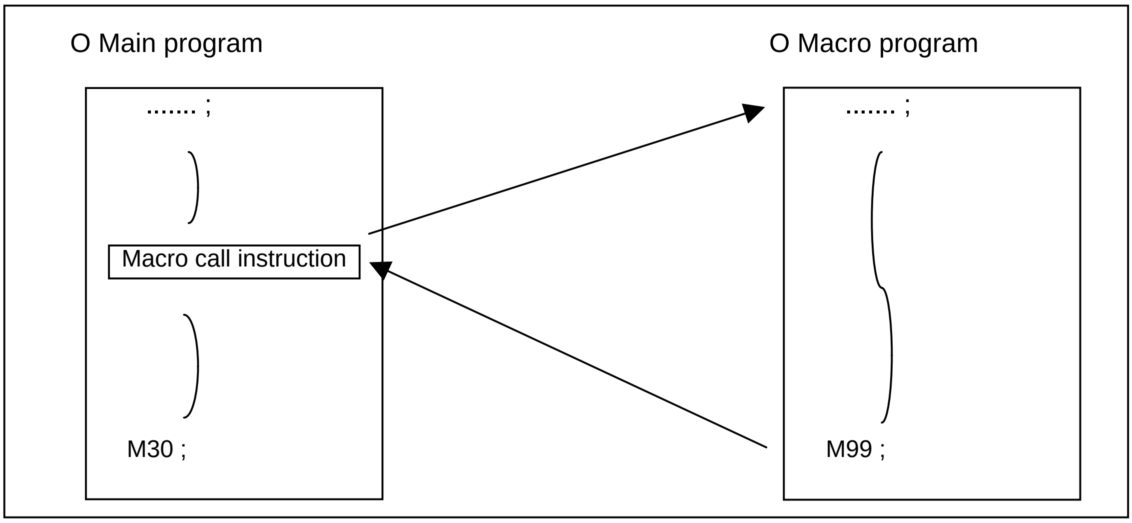
(Note 6) Variable quotations are always regarded as having a decimal point at the end.
When #100 = 10, then X#100 ; is treated as X10.

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.



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 (called after the movement command)
G66.1	User macro Modal call B (called after the every block)
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.

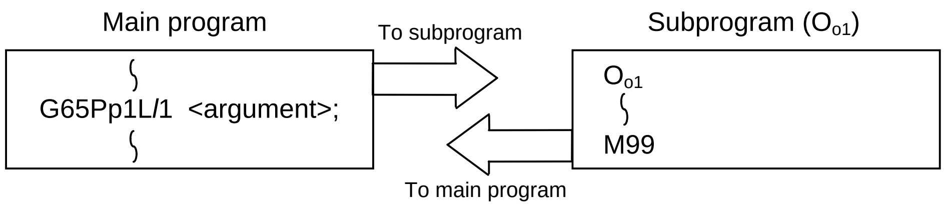
(2) The G66 (or G66.1) and G67 commands must be paired in the same program.

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



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
A	# 1
B	# 2
C	# 3
I1	# 4
J1	# 5
K1	# 6
I2	# 7
J2	# 8
K2	# 9
I3	#10
J3	#11
K3	#12
I4	#13
J4	#14
K4	#15
I5	#16

Argument designation II address	Variable within macro
J5	#17
K5	#18
I6	#19
J6	#20
K6	#21
I7	#22
J7	#23
K7	#24
I8	#25
J8	#26
K8	#27
I9	#28
J9	#29
K9	#30
I10	#31
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

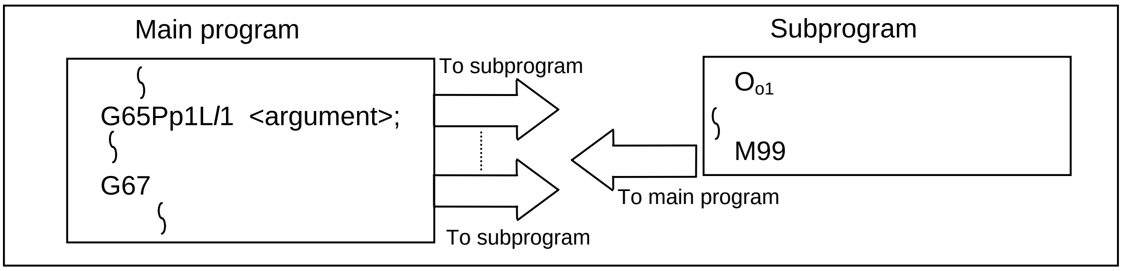
If addresses corresponding to the same variable are commanded when both types I and II are used to designate arguments, the latter address is valid.

(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 : ~~3.3~~ 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 (called after the movement command)



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 $\ell 1$ times with each call.
The <argument> is the same as for a simple call.

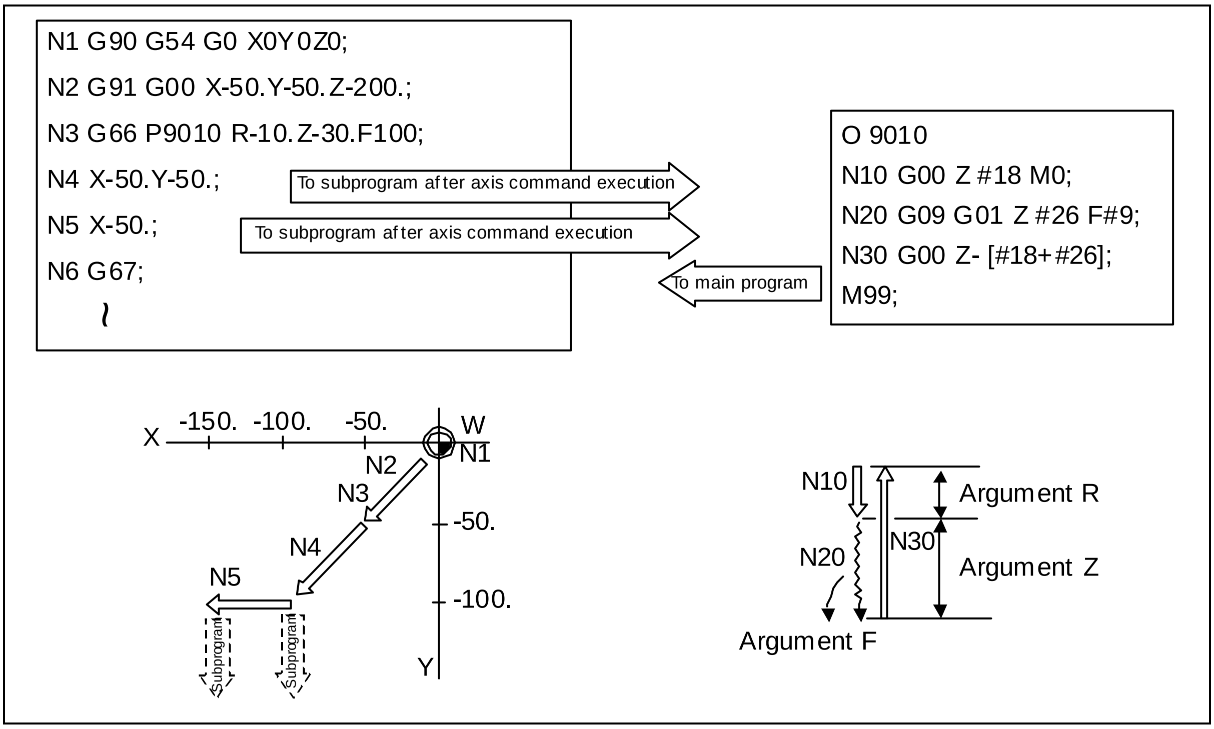
Format

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.

Modal call B (called after the every 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 number of times designated with “L” address.

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. M98P∆∆∆∆ ;
   b. G65P∆∆∆∆∆ <argument> ;
   c. G66P ∆∆∆∆∆ <argument> ;
   d. G66.1P∆∆∆∆∆ <argument> ;

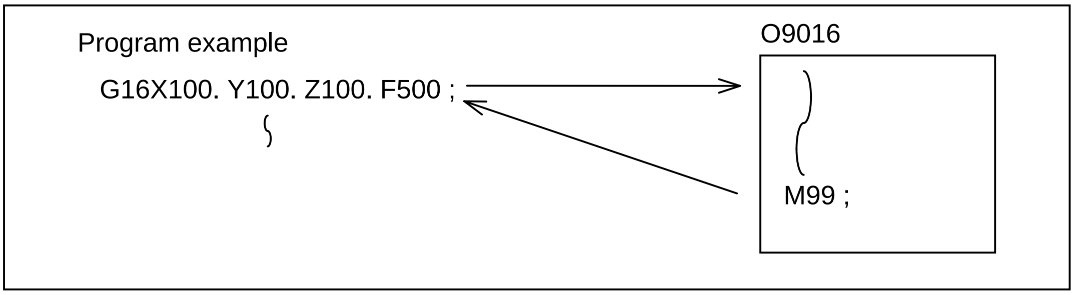
   When the parameters corresponding to c and d above are set, issue the cancel command (G67) either in the user macro or after the call code has been commanded so as to cancel the modal call.

(2) The correspondence between the "XX" 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.



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
 M [02]    21     0          8001
 M [03]  9999     0     199999999
 M [04]  9999     0     199999999
 M [05]  9999     0     199999999
            :     :         :
            :     :         :
 M [10]  9999     0     199999999
```

Setting to call O8000 with type 0 (M98 type) during M20 command

Setting to call O8001 with type 0 (M98 type) during M21 command

Set parameters not being used as 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.

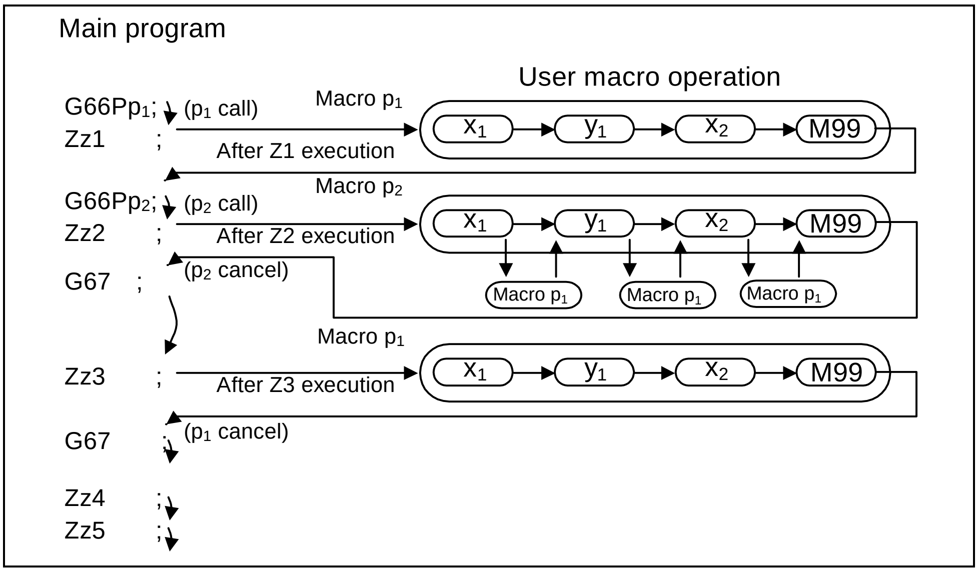
Macro call command nesting depth

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)



13.5.3 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. (fraction rounded up)

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.

13.5.4 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
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

Call command		Argument address	Local variable number
G65 G66	G66.1		
○	○	Q	#17
○	○	R	#18
○	○	S	#19
○	○	T	#20
○	○	U	#21
○	○	V	#22
○	○	W	#23
○	○	X	#24
○	○	Y	#25
○	○	Z	#26
		–	#27
		–	#28
		–	#29
		–	#30
		–	#31
		–	#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.

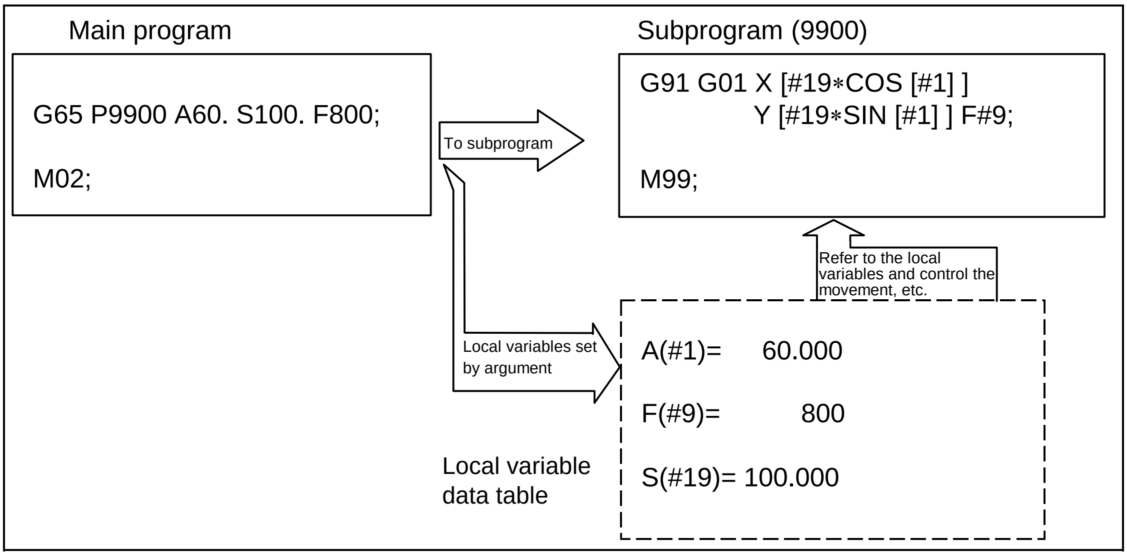
[Argument specification II]

Argument specification II address	Variable in macro
A	# 1
B	# 2
C	# 3
I1	# 4
J1	# 5
K1	# 6
I2	# 7
J2	# 8
K2	# 9
I3	#10
J3	#11
K3	#12
I4	#13
J4	#14
K4	#15
I5	#16
J5	#17
K5	#18

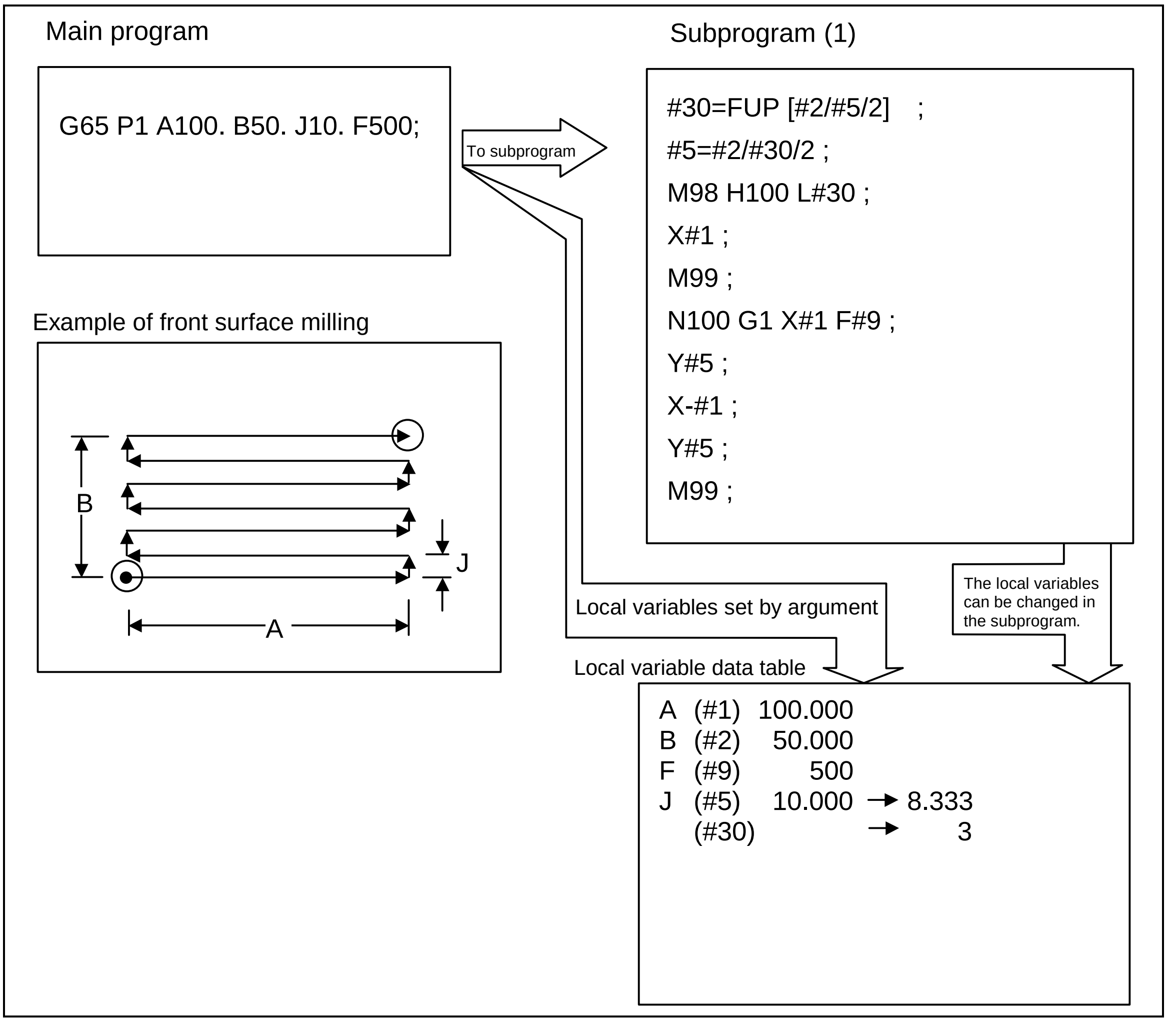
Argument specification II address	Variable in macro
I6	#19
J6	#20
K6	#21
I7	#22
J7	#23
K7	#24
I8	#25
J8	#26
K8	#27
I9	#28
J9	#29
K9	#30
I10	#31
J10	#32
K10	#33

(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.

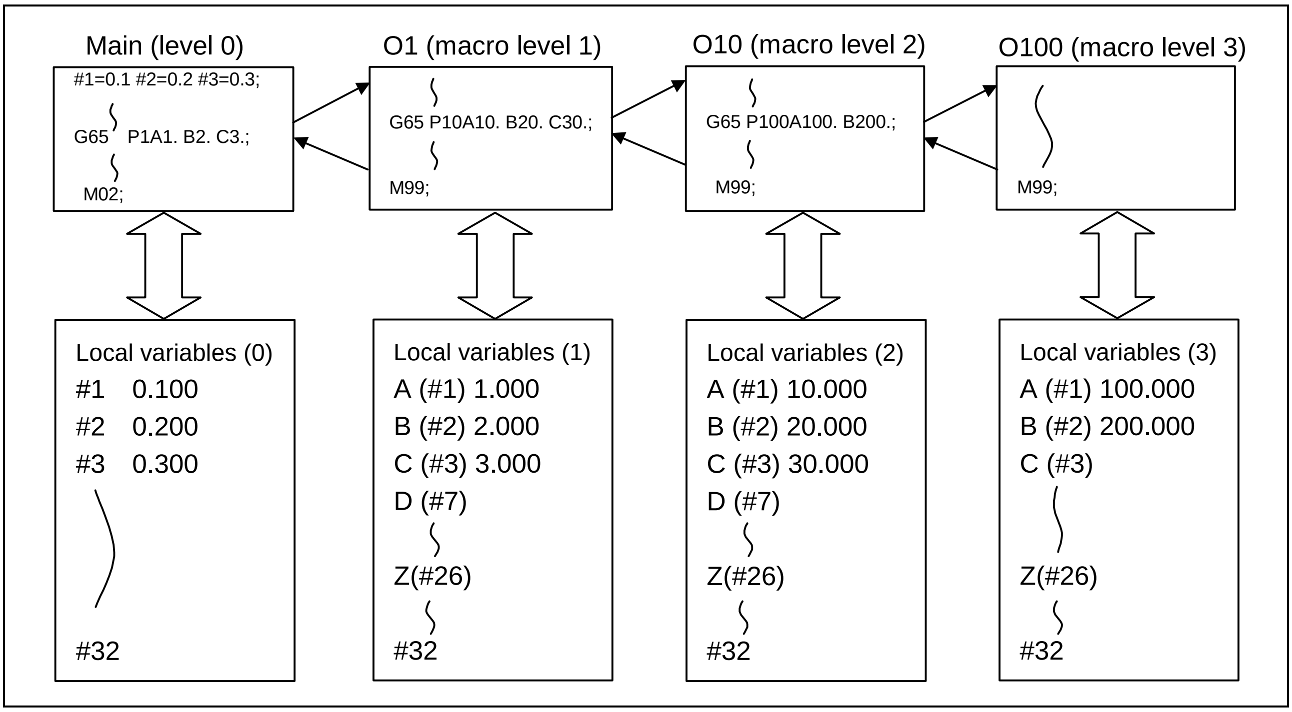


(2) The local variables can be used freely in that subprogram.



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.



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.

To use the macro interface function by part system, set the bit selection parameter "#6454/bit0". Refer to (2) for the signals provided for each part system.

(1) Macro interface common to part systems (input)

System variable	No. of points	Interface input signal	System variable	No. of points	Interface input signal
#1000	1	Register R24 bit 0	#1016	1	Register R25 bit 0
#1001	1	Register R24 bit 1	#1017	1	Register R25 bit 1
#1002	1	Register R24 bit 2	#1018	1	Register R25 bit 2
#1003	1	Register R24 bit 3	#1019	1	Register R25 bit 3
#1004	1	Register R24 bit 4	#1020	1	Register R25 bit 4
#1005	1	Register R24 bit 5	#1021	1	Register R25 bit 5
#1006	1	Register R24 bit 6	#1022	1	Register R25 bit 6
#1007	1	Register R24 bit 7	#1023	1	Register R25 bit 7
#1008	1	Register R24 bit 8	#1024	1	Register R25 bit 8
#1009	1	Register R24 bit 9	#1025	1	Register R25 bit 9
#1010	1	Register R24 bit 10	#1026	1	Register R25 bit 10
#1011	1	Register R24 bit 11	#1027	1	Register R25 bit 11
#1012	1	Register R24 bit 12	#1028	1	Register R25 bit 12
#1013	1	Register R24 bit 13	#1029	1	Register R25 bit 13
#1014	1	Register R24 bit 14	#1030	1	Register R25 bit 14
#1015	1	Register R24 bit 15	#1031	1	Register R25 bit 15

System variable	No. of points	Interface input signal
#1032	32	Register R24, R25
#1033	32	Register R26, R27
#1034	32	Register R28, R29
#1035	32	Register R30, R31

System variable	No. of points	Interface input signal	System variable	No. of points	Interface input signal
#1200	1	Register R26 bit 0	#1216	1	Register R27 bit 0
#1201	1	Register R26 bit 1	#1217	1	Register R27 bit 1
#1202	1	Register R26 bit 2	#1218	1	Register R27 bit 2
#1203	1	Register R26 bit 3	#1219	1	Register R27 bit 3
#1204	1	Register R26 bit 4	#1220	1	Register R27 bit 4
#1205	1	Register R26 bit 5	#1221	1	Register R27 bit 5
#1206	1	Register R26 bit 6	#1222	1	Register R27 bit 6
#1207	1	Register R26 bit 7	#1223	1	Register R27 bit 7
#1208	1	Register R26 bit 8	#1224	1	Register R27 bit 8
#1209	1	Register R26 bit 9	#1225	1	Register R27 bit 9
#1210	1	Register R26 bit 10	#1226	1	Register R27 bit 10
#1211	1	Register R26 bit 11	#1227	1	Register R27 bit 11
#1212	1	Register R26 bit 12	#1228	1	Register R27 bit 12
#1213	1	Register R26 bit 13	#1229	1	Register R27 bit 13
#1214	1	Register R26 bit 14	#1230	1	Register R27 bit 14
#1215	1	Register R26 bit 15	#1231	1	Register R27 bit 15

System variable	No. of points	Interface input signal	System variable	No. of points	Interface input signal
#1232	1	Register R28 bit 0	#1248	1	Register R29 bit 0
#1233	1	Register R28 bit 1	#1249	1	Register R29 bit 1
#1234	1	Register R28 bit 2	#1250	1	Register R29 bit 2
#1235	1	Register R28 bit 3	#1251	1	Register R29 bit 3
#1236	1	Register R28 bit 4	#1252	1	Register R29 bit 4
#1237	1	Register R28 bit 5	#1253	1	Register R29 bit 5
#1238	1	Register R28 bit 6	#1254	1	Register R29 bit 6
#1239	1	Register R28 bit 7	#1255	1	Register R29 bit 7
#1240	1	Register R28 bit 8	#1256	1	Register R29 bit 8
#1241	1	Register R28 bit 9	#1257	1	Register R29 bit 9
#1242	1	Register R28 bit 10	#1258	1	Register R29 bit 10
#1243	1	Register R28 bit 11	#1259	1	Register R29 bit 11
#1244	1	Register R28 bit 12	#1260	1	Register R29 bit 12
#1245	1	Register R28 bit 13	#1261	1	Register R29 bit 13
#1246	1	Register R28 bit 14	#1262	1	Register R29 bit 14
#1247	1	Register R28 bit 15	#1263	1	Register R29 bit 15

System variable	No. of points	Interface input signal	System variable	No. of points	Interface input signal
#1264	1	Register R30 bit 0	#1280	1	Register R31 bit 0
#1265	1	Register R30 bit 1	#1281	1	Register R31 bit 1
#1266	1	Register R30 bit 2	#1282	1	Register R31 bit 2
#1267	1	Register R30 bit 3	#1283	1	Register R31 bit 3
#1268	1	Register R30 bit 4	#1284	1	Register R31 bit 4
#1269	1	Register R30 bit 5	#1285	1	Register R31 bit 5
#1270	1	Register R30 bit 6	#1286	1	Register R31 bit 6
#1271	1	Register R30 bit 7	#1287	1	Register R31 bit 7
#1272	1	Register R30 bit 8	#1288	1	Register R31 bit 8
#1273	1	Register R30 bit 9	#1289	1	Register R31 bit 9
#1274	1	Register R30 bit 10	#1290	1	Register R31 bit 10
#1275	1	Register R30 bit 11	#1291	1	Register R31 bit 11
#1276	1	Register R30 bit 12	#1292	1	Register R31 bit 12
#1277	1	Register R30 bit 13	#1293	1	Register R31 bit 13
#1278	1	Register R30 bit 14	#1294	1	Register R31 bit 14
#1279	1	Register R30 bit 15	#1295	1	Register R31 bit 15

(2) Macro interface by part system (input)

(Note) As for the C64T system, the input/output signals used for this function are valid up to 3rd part system.

System variable	No. of points	Interface input signal						
		$1	**$2**	**$3**	**$4**	**$5**	**$6**	**$7**
		R970	R1070	R1170	R1270	R1370	R1470	R1570
#1000	1	bit0	bit0	bit0	bit0	bit0	bit0	bit0
#1001	1	bit1	bit1	bit1	bit1	bit1	bit1	bit1
#1002	1	bit2	bit2	bit2	bit2	bit2	bit2	bit2
#1003	1	bit3	bit3	bit3	bit3	bit3	bit3	bit3
#1004	1	bit4	bit4	bit4	bit4	bit4	bit4	bit4
#1005	1	bit5	bit5	bit5	bit5	bit5	bit5	bit5
#1006	1	bit6	bit6	bit6	bit6	bit6	bit6	bit6
#1007	1	bit7	bit7	bit7	bit7	bit7	bit7	bit7
#1008	1	bit8	bit8	bit8	bit8	bit8	bit8	bit8
#1009	1	bit9	bit9	bit9	bit9	bit9	bit9	bit9
#1010	1	bit10	bit10	bit10	bit10	bit10	bit10	bit10
#1011	1	bit11	bit11	bit11	bit11	bit11	bit11	bit11
#1012	1	bit12	bit12	bit12	bit12	bit12	bit12	bit12
#1013	1	bit13	bit13	bit13	bit13	bit13	bit13	bit13
#1014	1	bit14	bit14	bit14	bit14	bit14	bit14	bit14
#1015	1	bit15	bit15	bit15	bit15	bit15	bit15	bit15

System variable	No. of points	Interface input signal						
		$1	**$2**	**$3**	**$4**	**$5**	**$6**	**$7**
		R971	R1071	R1171	R1271	R1371	R1471	R1571
#1016	1	bit0	bit0	bit0	bit0	bit0	bit0	bit0
#1017	1	bit1	bit1	bit1	bit1	bit1	bit1	bit1
#1018	1	bit2	bit2	bit2	bit2	bit2	bit2	bit2
#1019	1	bit3	bit3	bit3	bit3	bit3	bit3	bit3
#1020	1	bit4	bit4	bit4	bit4	bit4	bit4	bit4
#1021	1	bit5	bit5	bit5	bit5	bit5	bit5	bit5
#1022	1	bit6	bit6	bit6	bit6	bit6	bit6	bit6
#1023	1	bit7	bit7	bit7	bit7	bit7	bit7	bit7
#1024	1	bit8	bit8	bit8	bit8	bit8	bit8	bit8
#1025	1	bit9	bit9	bit9	bit9	bit9	bit9	bit9
#1026	1	bit10	bit10	bit10	bit10	bit10	bit10	bit10
#1027	1	bit11	bit11	bit11	bit11	bit11	bit11	bit11
#1028	1	bit12	bit12	bit12	bit12	bit12	bit12	bit12
#1029	1	bit13	bit13	bit13	bit13	bit13	bit13	bit13
#1030	1	bit14	bit14	bit14	bit14	bit14	bit14	bit14
#1031	1	bit15	bit15	bit15	bit15	bit15	bit15	bit15

System variable	No. of points	Interface input signal						
		$1	**$2**	**$3**	**$4**	**$5**	**$6**	**$7**
#1032	32	R970, R971	R1070, R1071	R1170, R1171	R1270, R1271	R1370, R1371	R1470, R1471	R1570, R1571
#1033	32	R972, R973	R1072, R1073	R1172, R1173	R1272, R1273	R1372, R1373	R1472, R1473	R1572, R1573
#1034	32	R974, R975	R1074, R1075	R1174, R1175	R1274, R1275	R1374, R1375	R1474, R1475	R1574, R1575
#1035	32	R976, R977	R1076, R1077	R1176, R1177	R1276, R1277	R1376, R1377	R1476, R1477	R1576, R1577

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. (2^0 ~ 2^{31})
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.

To use the macro interface function by part system, set the bit selection parameter "#6454/bit0". Refer to (2) for the signals provided for each part system.

(1) Macro interface common to part systems (output)

System variable	No. of points	Interface output signal	System variable	No. of points	Interface output signal
#1100	1	Register R124 bit 0	#1116	1	Register R125 bit 0
#1101	1	Register R124 bit 1	#1117	1	Register R125 bit 1
#1102	1	Register R124 bit 2	#1118	1	Register R125 bit 2
#1103	1	Register R124 bit 3	#1119	1	Register R125 bit 3
#1104	1	Register R124 bit 4	#1120	1	Register R125 bit 4
#1105	1	Register R124 bit 5	#1121	1	Register R125 bit 5
#1106	1	Register R124 bit 6	#1122	1	Register R125 bit 6
#1107	1	Register R124 bit 7	#1123	1	Register R125 bit 7
#1108	1	Register R124 bit 8	#1124	1	Register R125 bit 8
#1109	1	Register R124 bit 9	#1125	1	Register R125 bit 9
#1110	1	Register R124 bit 10	#1126	1	Register R125 bit 10
#1111	1	Register R124 bit 11	#1127	1	Register R125 bit 11
#1112	1	Register R124 bit 12	#1128	1	Register R125 bit 12
#1113	1	Register R124 bit 13	#1129	1	Register R125 bit 13
#1114	1	Register R124 bit 14	#1130	1	Register R125 bit 14
#1115	1	Register R124 bit 15	#1131	1	Register R125 bit 15

System variable	No. of points	Interface output signal
#1132	32	Register R124, R125
#1133	32	Register R126, R127
#1134	32	Register R128, R129
#1135	32	Register R130, R131

(Note 1) The last values of the system variables #1100 to #1135 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.

<Vacant> is treated as 0.

Any number except 0 and <vacant> is treated as 1.

Any value less than 0.00000001 is indefinite.

System variable	No. of points	Interface output signal	System variable	No. of points	Interface output signal
#1300	1	Register R126 bit 0	#1316	1	Register R127 bit 0
#1301	1	Register R126 bit 1	#1317	1	Register R127 bit 1
#1302	1	Register R126 bit 2	#1318	1	Register R127 bit 2
#1303	1	Register R126 bit 3	#1319	1	Register R127 bit 3
#1304	1	Register R126 bit 4	#1320	1	Register R127 bit 4
#1305	1	Register R126 bit 5	#1321	1	Register R127 bit 5
#1306	1	Register R126 bit 6	#1322	1	Register R127 bit 6
#1307	1	Register R126 bit 7	#1323	1	Register R127 bit 7
#1308	1	Register R126 bit 8	#1324	1	Register R127 bit 8
#1309	1	Register R126 bit 9	#1325	1	Register R127 bit 9
#1310	1	Register R126 bit 10	#1326	1	Register R127 bit 10
#1311	1	Register R126 bit 11	#1327	1	Register R127 bit 11
#1312	1	Register R126 bit 12	#1328	1	Register R127 bit 12
#1313	1	Register R126 bit 13	#1329	1	Register R127 bit 13
#1314	1	Register R126 bit 14	#1330	1	Register R127 bit 14
#1315	1	Register R126 bit 15	#1331	1	Register R127 bit 15

System variable	No. of points	Interface output signal	System variable	No. of points	Interface output signal
#1332	1	Register R128 bit 0	#1348	1	Register R129 bit 0
#1333	1	Register R128 bit 1	#1349	1	Register R129 bit 1
#1334	1	Register R128 bit 2	#1350	1	Register R129 bit 2
#1335	1	Register R128 bit 3	#1351	1	Register R129 bit 3
#1336	1	Register R128 bit 4	#1352	1	Register R129 bit 4
#1337	1	Register R128 bit 5	#1353	1	Register R129 bit 5
#1338	1	Register R128 bit 6	#1354	1	Register R129 bit 6
#1339	1	Register R128 bit 7	#1355	1	Register R129 bit 7
#1340	1	Register R128 bit 8	#1356	1	Register R129 bit 8
#1341	1	Register R128 bit 9	#1357	1	Register R129 bit 9
#1342	1	Register R128 bit 10	#1358	1	Register R129 bit 10
#1343	1	Register R128 bit 11	#1359	1	Register R129 bit 11
#1344	1	Register R128 bit 12	#1360	1	Register R129 bit 12
#1345	1	Register R128 bit 13	#1361	1	Register R129 bit 13
#1346	1	Register R128 bit 14	#1362	1	Register R129 bit 14
#1347	1	Register R128 bit 15	#1363	1	Register R129 bit 15

System variable	No. of points	Interface output signal	System variable	No. of points	Interface output signal
#1364	1	Register R130 bit 0	#1380	1	Register R131 bit 0
#1365	1	Register R130 bit 1	#1381	1	Register R131 bit 1
#1366	1	Register R130 bit 2	#1382	1	Register R131 bit 2
#1367	1	Register R130 bit 3	#1383	1	Register R131 bit 3
#1368	1	Register R130 bit 4	#1384	1	Register R131 bit 4
#1369	1	Register R130 bit 5	#1385	1	Register R131 bit 5
#1370	1	Register R130 bit 6	#1386	1	Register R131 bit 6
#1371	1	Register R130 bit 7	#1387	1	Register R131 bit 7
#1372	1	Register R130 bit 8	#1388	1	Register R131 bit 8
#1373	1	Register R130 bit 9	#1389	1	Register R131 bit 9
#1374	1	Register R130 bit 10	#1390	1	Register R131 bit 10
#1375	1	Register R130 bit 11	#1391	1	Register R131 bit 11
#1376	1	Register R130 bit 12	#1392	1	Register R131 bit 12
#1377	1	Register R130 bit 13	#1393	1	Register R131 bit 13
#1378	1	Register R130 bit 14	#1394	1	Register R131 bit 14
#1379	1	Register R130 bit 15	#1395	1	Register R131 bit 15

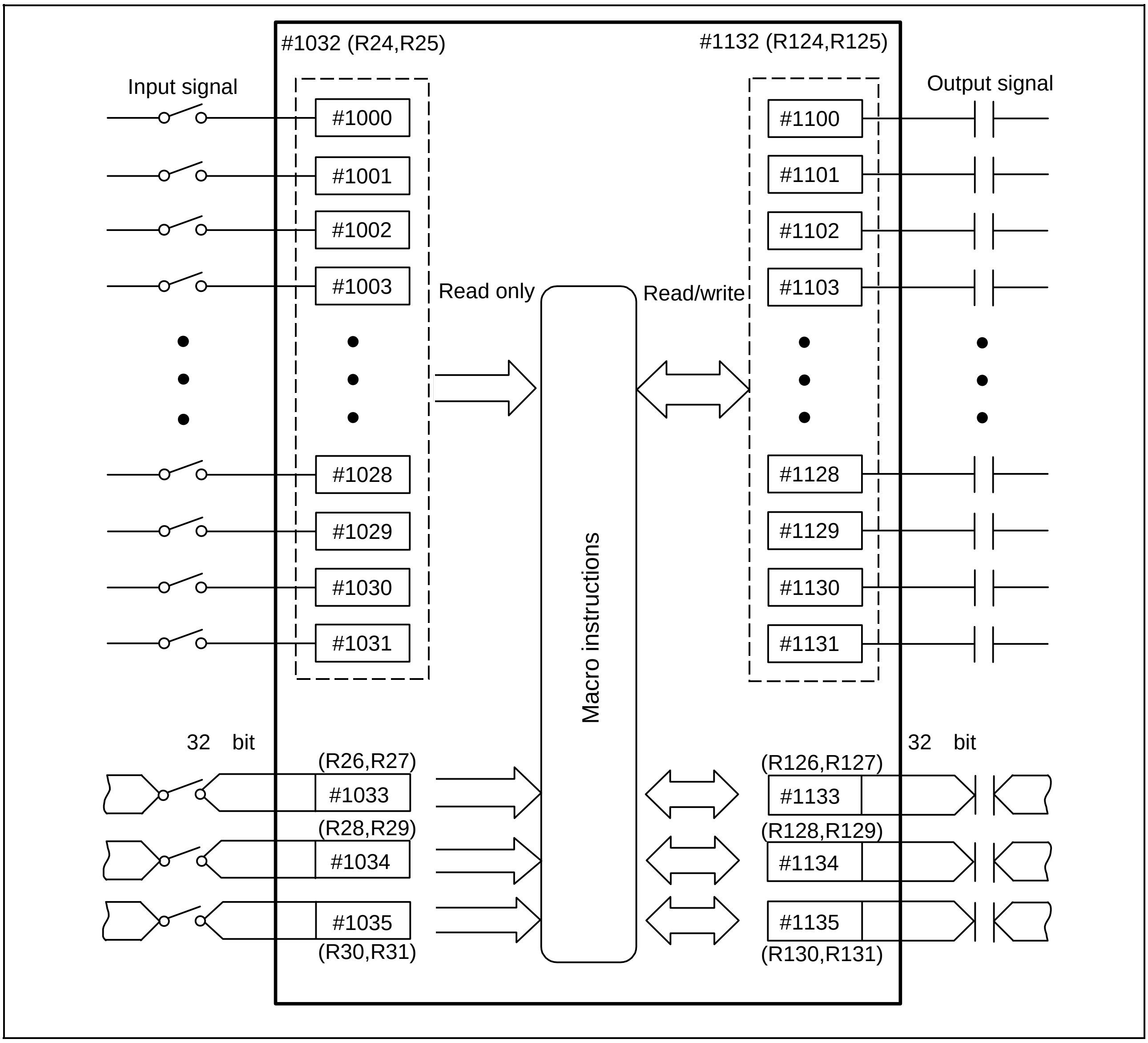
(2) Macro interface by part system (output)

(Note) As for the C64T system, the input/output signals used for this function are valid up to 3rd part system.

System variable	No. of points	Interface output signal						
		$1	$2	$3	$4	$5	$6	$7
		R270	R370	R470	R570	R670	R770	R870
#1100	1	bit0	bit0	bit0	bit0	bit0	bit0	bit0
#1101	1	bit1	bit1	bit1	bit1	bit1	bit1	bit1
#1102	1	bit2	bit2	bit2	bit2	bit2	bit2	bit2
#1103	1	bit3	bit3	bit3	bit3	bit3	bit3	bit3
#1104	1	bit4	bit4	bit4	bit4	bit4	bit4	bit4
#1105	1	bit5	bit5	bit5	bit5	bit5	bit5	bit5
#1106	1	bit6	bit6	bit6	bit6	bit6	bit6	bit6
#1107	1	bit7	bit7	bit7	bit7	bit7	bit7	bit7
#1108	1	bit8	bit8	bit8	bit8	bit8	bit8	bit8
#1109	1	bit9	bit9	bit9	bit9	bit9	bit9	bit9
#1110	1	bit10	bit10	bit10	bit10	bit10	bit10	bit10
#1111	1	bit11	bit11	bit11	bit11	bit11	bit11	bit11
#1112	1	bit12	bit12	bit12	bit12	bit12	bit12	bit12
#1113	1	bit13	bit13	bit13	bit13	bit13	bit13	bit13
#1114	1	bit14	bit14	bit14	bit14	bit14	bit14	bit14
#1115	1	bit15	bit15	bit15	bit15	bit15	bit15	bit15

System variable	No. of points	Interface output signal						
		$1	$2	$3	$4	$5	$6	$7
		R271	R371	R471	R571	R671	R771	R871
#1116	1	bit0	bit0	bit0	bit0	bit0	bit0	bit0
#1117	1	bit1	bit1	bit1	bit1	bit1	bit1	bit1
#1118	1	bit2	bit2	bit2	bit2	bit2	bit2	bit2
#1119	1	bit3	bit3	bit3	bit3	bit3	bit3	bit3
#1120	1	bit4	bit4	bit4	bit4	bit4	bit4	bit4
#1121	1	bit5	bit5	bit5	bit5	bit5	bit5	bit5
#1122	1	bit6	bit6	bit6	bit6	bit6	bit6	bit6
#1123	1	bit7	bit7	bit7	bit7	bit7	bit7	bit7
#1124	1	bit8	bit8	bit8	bit8	bit8	bit8	bit8
#1125	1	bit9	bit9	bit9	bit9	bit9	bit9	bit9
#1126	1	bit10	bit10	bit10	bit10	bit10	bit10	bit10
#1127	1	bit11	bit11	bit11	bit11	bit11	bit11	bit11
#1128	1	bit12	bit12	bit12	bit12	bit12	bit12	bit12
#1129	1	bit13	bit13	bit13	bit13	bit13	bit13	bit13
#1130	1	bit14	bit14	bit14	bit14	bit14	bit14	bit14
#1131	1	bit15	bit15	bit15	bit15	bit15	bit15	bit15

System variable	No. of points	Interface output signal						
		$1	$2	$3	$4	$5	$6	$7
#1132	32	R270, R271	R370, R371	R470, R471	R570, R571	R670, R671	R770, R771	R870, R871
#1133	32	R272, R273	R372, R373	R472, R473	R572, R573	R672, R673	R772, R773	R872, R873
#1134	32	R274, R275	R374, R375	R474, R475	R574, R575	R674, R675	R774, R775	R874, R875
#1135	32	R276, R277	R376, R377	R476, R477	R576, R577	R676, R677	R776, R777	R876, R877


#1032 (R24,R25)
#1132 (R124,R125)
Input signal
Output signal
#1000
#1001
#1002
#1003
#1028
#1029
#1030
#1031
#1100
#1101
#1102
#1103
#1128
#1129
#1130
#1131
Read only
Read/write
Macro instructions
32 bit
32 bit
(R26,R27)
#1033
(R28,R29)
#1034
#1035
(R30,R31)
(R126,R127)
#1133
(R128,R129)
#1134
#1135
(R130,R131)

Tool offset

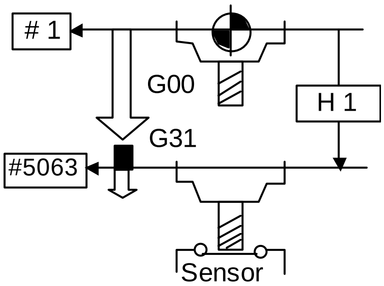
Variable number range		Type 1	Type 2
#10001 to #10000 + n	#2001 to #2000 + n	○	○ (Length dimension)
#11001 to #11000 + n	#2201 to #2200 + n	×	○ (Length wear)
#16001 to #16000 + n	#2401 to #2400 + n	×	○ (Radius dimension)
#17001 to #17000 + n	#2601 to #2600 + n	×	○ (Radius wear)

Tool data can be read and values substituted using the variable numbers.
Either the numbers in the #10000 order or #2000 order can be used.
The last 3 digits of the variable numbers correspond to the tool offset number.
n corresponds to the No. of tool offset sets.
If there are 400 tool offset sets and type 2 is being used, avoid variable Nos. in the #2000 order, and instead use the #10000 order.
The tool offset data are configured as data with a decimal point in the same way as for other variables. Consequently, this decimal point must be commanded when data below the decimal point is to be entered.

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

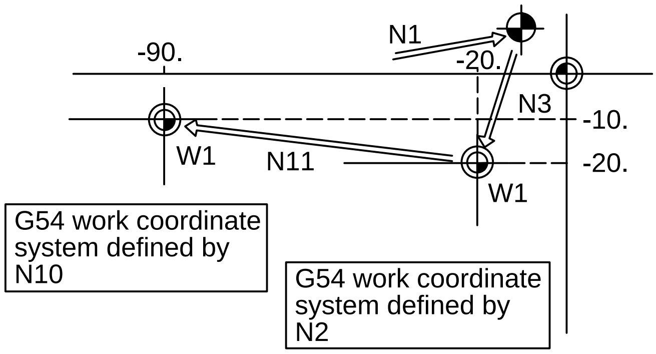
By using variable numbers #5201 to #532n, it is possible to read out the work coordinate system offset data or to substitute values.

(Note) The number of axes which can be controlled differs according to the specifications. The last digit in the variable number corresponds to the control axis number.

Axis No. / Axis name	Axis 1	Axis 2	Axis 3	Axis 4	. .	Axis n	Remarks
External work offset	#5201	#5202	#5203	#5204	. .	#520n	External workpiece offset specifications are required.
G54	#5221	#5222	#5223	#5224	. .	#522n	
G55	#5241	#5242	#5243	#5244	. .	#524n	
G56	#5261	#5262	#5263	#5264	. .	#526n	
G57	#5281	#5282	#5283	#5284	. .	#528n	
G58	#5301	#5302	#5303	#5304	. .	#530n	
G59	#5321	#5322	#5323	#5324	. .	#532n	

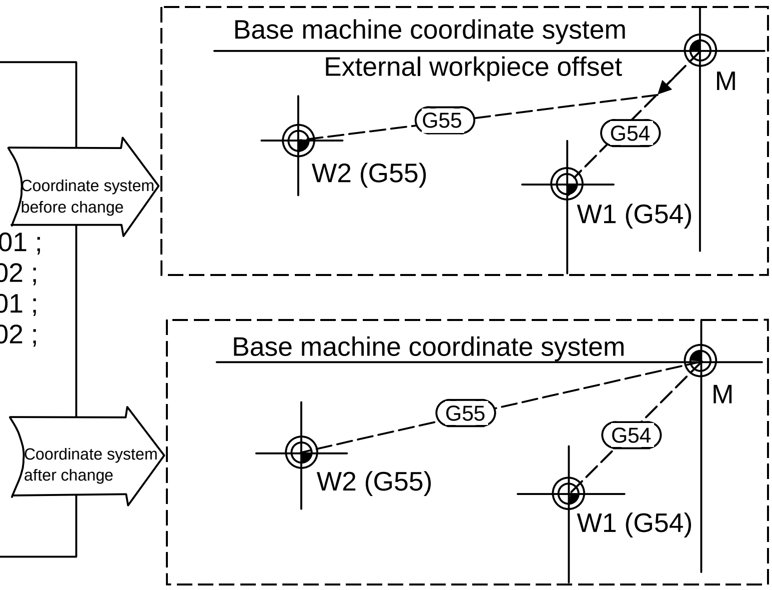
(Example 1)

```
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 ;
```



(Example 2)

```
N100 #5221=#5221+#5201 ;
     #5222=#5222+#5202 ;
     #5241=#5241+#5201 ;
     #5242=#5242+#5202 ;
     #5201=0 #5202=0;
```



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.
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

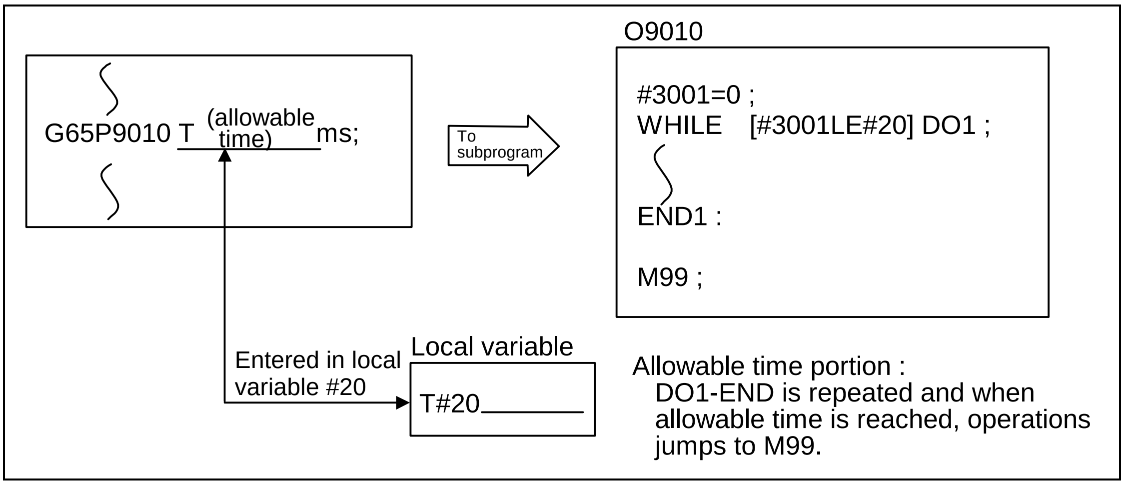
(Note 1) Alarm number 0 is not displayed and any number exceeding 9999 cannot be indicated.
(Note 2) The characters following the first alphabet letter in the right member is treated as the alarm message. Therefore, a number cannot be designated as the first character of an alarm message. It is recommended 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	1ms	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).



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.

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 on the operator message area.

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											6	5	4	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	Canned cycle	: G80:80, G73 to 74, G76:76, G81 to G89:81 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;
%
```

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

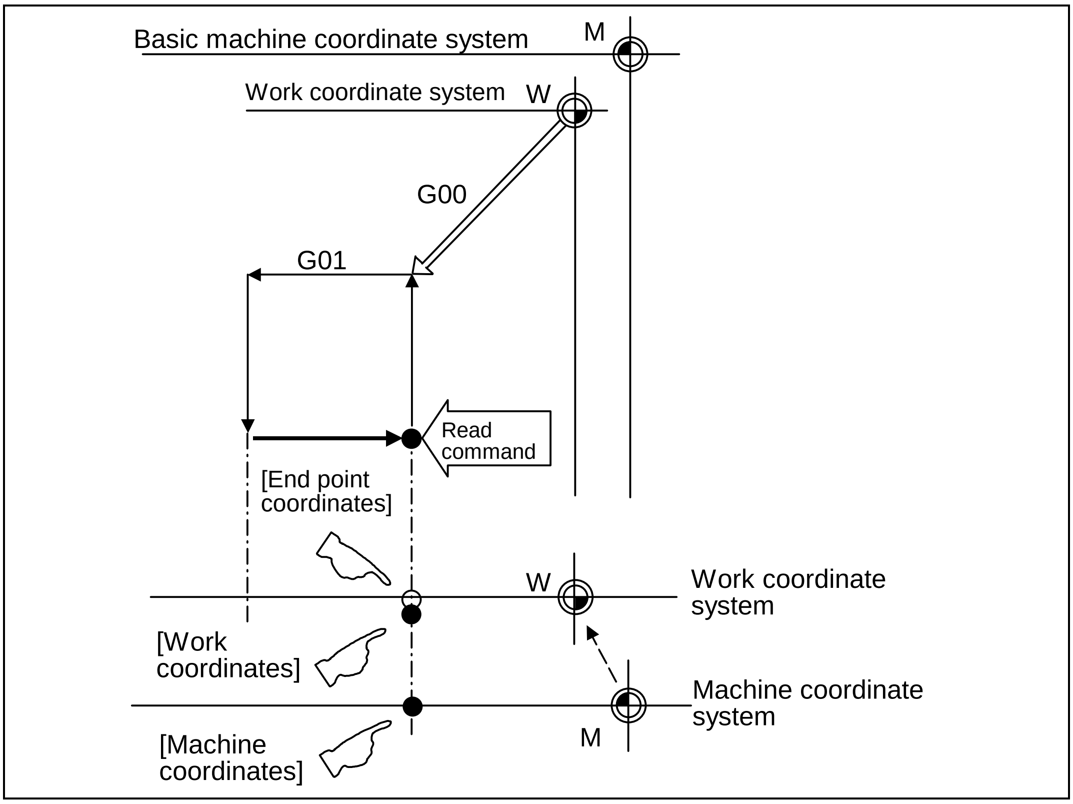
Position information

Using variable numbers #5001 to #5104, it is possible to read the servo deviation amounts, skip coordinates, work coordinates, machine coordinates and end point coordinates in the block immediately before.

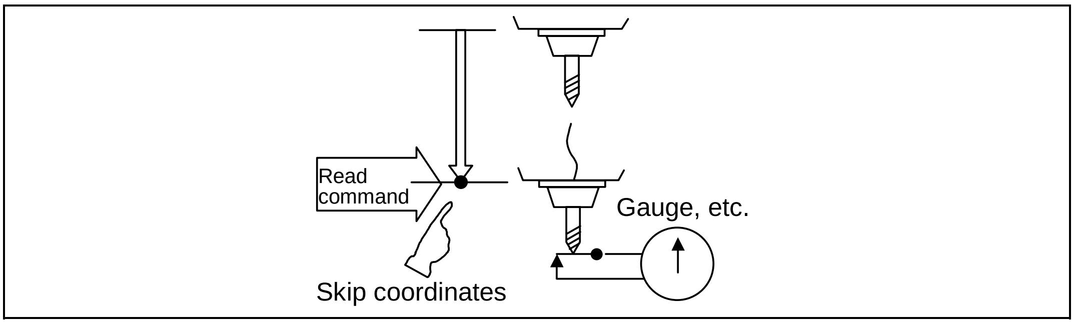
Position information / Axis No.	End point coordinate of block immediately before	Machine coordinate	Work coordinate	Skip coordinate	Servo deviation amount
1	#5001	#5021	#5041	#5061	#5101
2	#5002	#5022	#5042	#5062	#5102
3	#5003	#5023	#5043	#5063	#5103
4	#5004	#5024	#5044	#5064	#5104
:	:	:	:	:	:
n	#5000+n	#5020+n	#5040+n	#5060+n	#5100+n
Remarks (reading during movement)	Yes	No	No	Yes	Yes

(Note1) The number of axes which can be controlled differs according to the specifications.

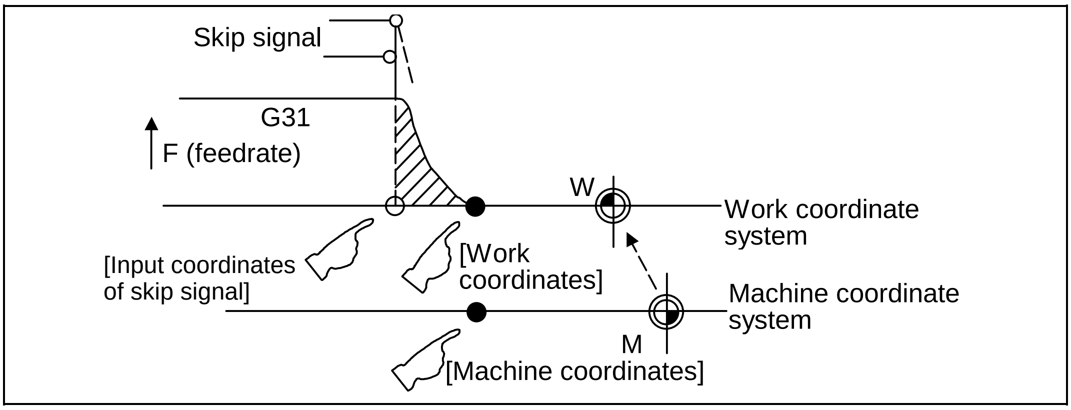
(Note2) The last digit of the variable number corresponds to the control axis number.



(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

An example to measure the distance from the measured reference point to the workpiece edge is shown below.

Main program

```
G65 P9031 X100. Y100. Z-10. F200;
```

To subprogram

Argument
<Local variable>

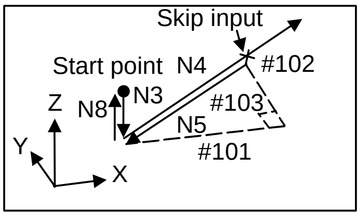
F(#9)	200
X(#24)	100.000
Y(#25)	100.000
Z(#26)	-10.000

<Common variable>

#101	87.245
#102	87.245
#103	123.383

O9031

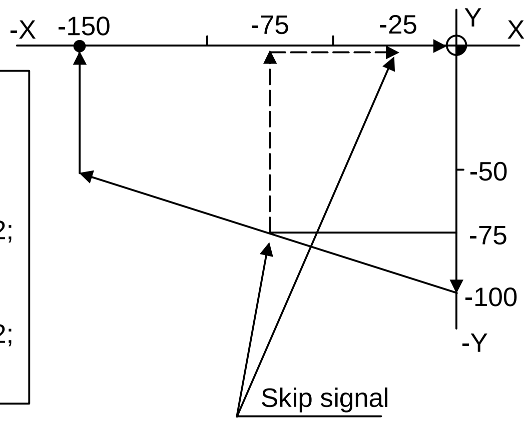
```
N1 #180=#4003;
N2 #30=#5001 #31=#5002;
N3 G91 G01 Z#26 F#9;
N4 G31 X#24 Y#25 F#9;
N5 G90 G00 X#30 Y#31;
N6 #101=#30-#5061 #102=#31-#5062;
N7 #103=SQR [#101*#101+#102*#102] ;
N8 G91 G01Z-#26;
N9 IF [#180 EQ 91] GOTO 11;
N10 G90;
N11 M99;
```



#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	Subprogram return

(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).

Workpiece coordinate shift amount

The workpiece coordinate system shift amount can be read using variables #2501 and #2601.
By substituting a value in these variables, the workpiece coordinate system shift amount can be changed.

Axis No.	Workpiece coordinate system shift amount
1	#2501
2	#2601

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.

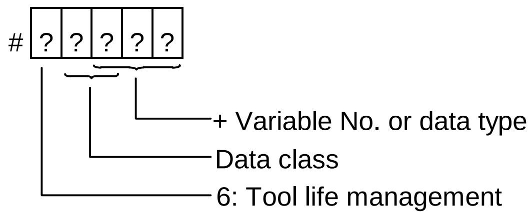
Tool life management

(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



(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.

(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. (Designate the group No. #60000 and registration No. *** .) Note the group No., method and life are common for the groups.	This group's No.	1 to 99999999
61000 +***	Tool No.		Tool No.	1 to 99999999
61500 +***	Tool data flag		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 compen-sation 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

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.	The group 10 life data is registered.
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.	
#111 = #61001 ;	Reads the group 10, #1 tool No.	
G10 L3 ;	Starts the life management data registration.	The life data other than group 10 is registered.
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.	

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.

13.5.5 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 = 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   ;
#10001   = #101   ;
#102     = #10001 ;
```

Common variables after execution

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

(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 arithmetic 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 = #5041	#1 1000.000 #2 1000.000 #3 100.000 } From common #4 200.000 } variables #5 –10.000 From offset amount
(3) Addition and subtraction (+,–)	#11 = #1 + 1000 #12 = #2 – 50. #13 = #101 + #1 #14 = #5041 – 3. #15 = #5041 + #102	#11 2000.000 #12 950.000 #13 1100.000 #14 –13.000 #15 190.000
(4) Logical sum (OR)	#3 = 100 #4 = #3OR14	#3 = 01100100 14 = 00001110 #4 = 01101110 = 110
(5) Exclusive OR (XOR)	#3 = 100 #4 = #3XOR14	#3 = 01100100 14 = 00001110 #4 = 01101010 = 106

(6) Multiplication 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 = #5041∗#101 #30 = #5041/#102	#21 10000.000 #22 10000.000 #23 10000.000 #24 10000.000 #25 1.000 #26 1.000 #27 1.000 #28 1.000 #29 –1000.000 #30 –0.050
(7) Remainder (MOD)	#31 = #19MOD#20	#19/#20 = 48/9 = 5 with 3 over
(8) Logical product (AND)	#9 = 100 #10 = #9AND15	#9 = 01100100 15 = 00001111 #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 0.860 #502 0.860 #503 866.025 #504 866.025 #505 866.025 #506 866.025
(10) Cosine (COS)	#541 = COS [45] #542 = COS [45.] #543 = 1000∗COS [45] #544 = 1000∗COS [45.] #545 = 1000.∗COS [45] #546 = 1000.∗COS [45.] **Note**: COS [45] is equivalent to COS [45.]	#541 0.707 #542 0.707 #543 707.107 #544 707.107 #545 707.107 #546 707.107
(11) Tangent (TAN)	#551 = TAN [60] #552 = TAN [60.] #553 = 1000∗TAN [60] #554 = 1000∗TAN [60.] #555 = 1000.∗TAN [60] #556 = 1000.∗TAN [60.] **Note**: TAN [60] is equivalent to TAN [60.]	#551 1.732 #552 1.732 #553 1732.051 #554 1732.051 #555 1732.051 #556 1732.051
(12) Arctangent (ATAN or ATN)	#561 = ATAN [173205/100000] #562 = ATAN [173205/100.] #563 = ATAN [173.205/100000] #564 = ATAN [173.205/100.] #565 = ATAN [1.732]	#561 60.000 #562 60.000 #563 60.000 #564 60.000 #565 60.000

(13) Arc-cosine (ACOS)	#521 = ACOS [100./141.421] #522 = ACOS [100./141.421] #523 = ACOS [1000./1414.213] #524 = ACOS [10./14.142] #525 = ACOS [0.707]	#521 45.000 #522 45.000 #523 45.000 #524 44.999 #525 45.009
(14) Square root (SQR or SQRT)	#571 = SQRT [1000] #572 = SQRT [1000.] #573 = SQRT [10.∗10.+20.∗20.] #574 = SQRT [14∗#14+#15∗#15] **Note**: In order to increase the accuracy, proceed with the operation inside parentheses.	#571 31.623 #572 31.623 #573 22.361 #574 190.444
(15) Absolute value (ABS)	#576 = –1000 #577 = ABS [#576] #3 = 70.#4 = –50. #580 = ABS [#4 – #3]	#576 –1000.000 #577 1000.00 #580 120.000
(16) BIN, BCD	#1 = 100 #11 = BIN [#1] #12 = BCD [#1]	 #11 64 #12 256
(17) 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 5 #22 5 #23 5 #24 5 #25 –5 #26 –5 #27 –5 #28 –5
(18) 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 4.000 #22 4.000 #23 4.000 #24 4.000 #25 –4.000 #26 –4.000 #27 –4.000 #28 –4.000
(19) 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 5.000 #22 5.000 #23 5.000 #24 5.000 #25 –5.000 #26 –5.000 #27 –5.000 #28 –5.000
(20) Natural logarithms (LN)	#101 = LN [5] #102 = LN [0.5] #103 = LN [–5]	#101 1.609 #102 –0.693 Error "P282"
(21) Exponents (EXP)	#104 = EXP [2] #105 = EXP [1] #106 = EXP [–2]	#104 7.389 #105 2.718 #106 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. |ε/b|, |ε/c|
a = b∗c	1.55×10^{-10}	4.66×10^{-10}	Relative error |ε/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 |ε|°
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.

13.5.6 Control commands

The flow of programs can be controlled by IF-GOTO- and WHILE-DO-.

Branching

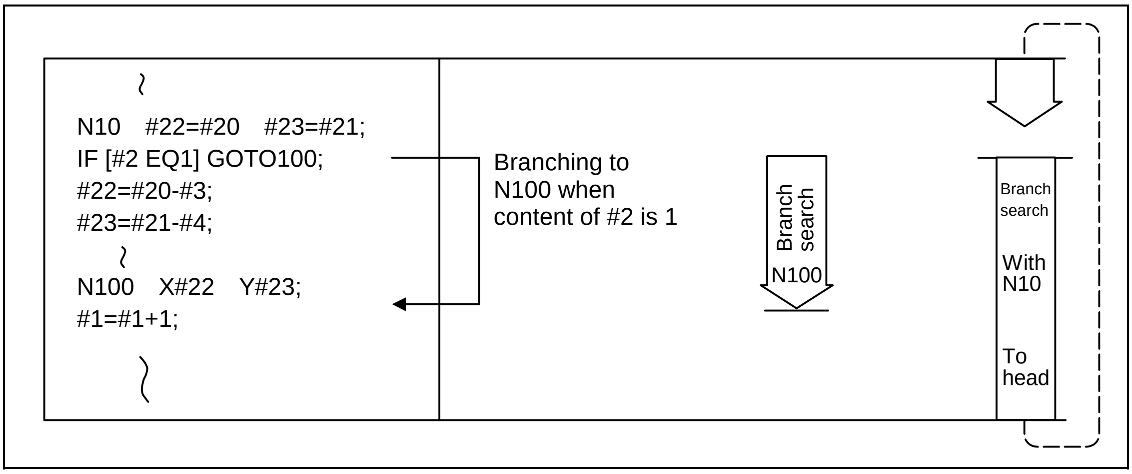
Format

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.

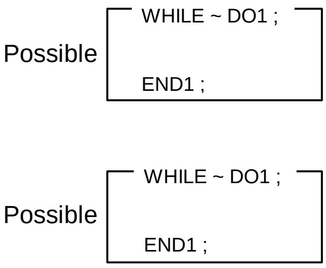
Iteration

Format

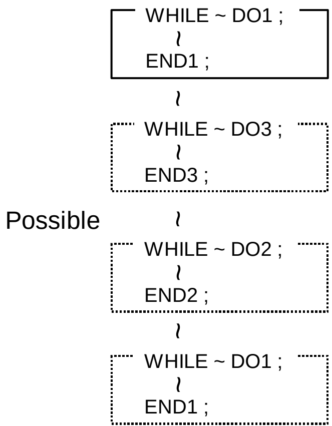
WHILE [conditional expression] DOm ; (m = 1, 2, 3 ..... 127)
~
END m ;

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.

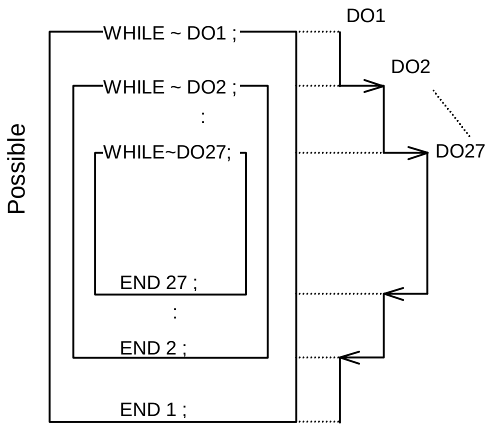
(1) Same identifier number can be used any number of times.



(2) Any number may be used for the WHILE –DOm identifier number.

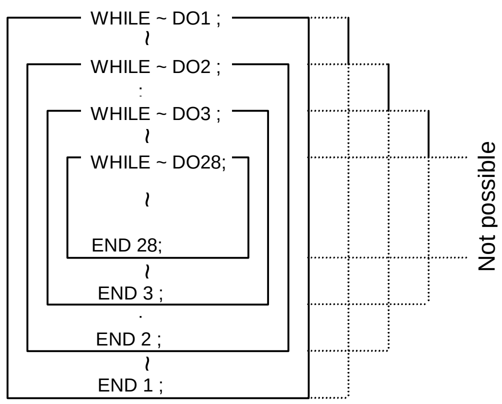


(3) Up to 27 nesting levels for WHILE– DOm. "m" is any number from 1 to 127 for the nesting depth.

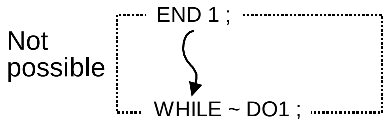


(Note) :With nesting, "m" which has been used once cannot be used.

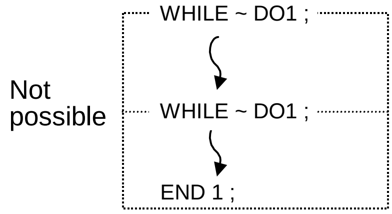
(4) The number of WHILE – DOm nesting levels cannot exceed 27.



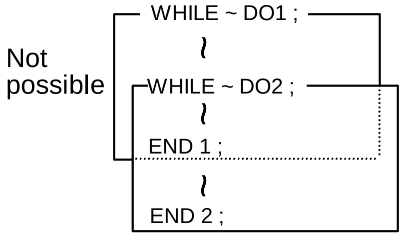
(5) WHILE – DOm must be designated first and ENDm last.



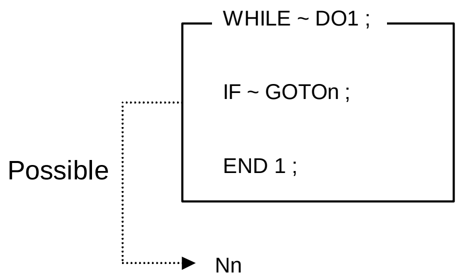
(6) WHILE – DOm and ENDm must correspond on a 1:1 (pairing) basis in the same program.



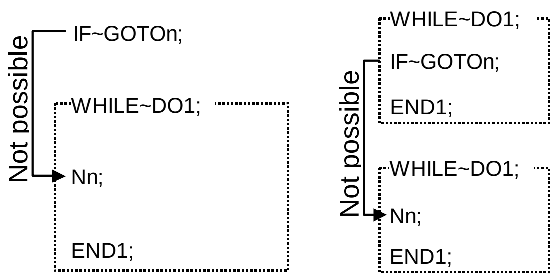
(7) Two WHILE – DOm's must not overlap.



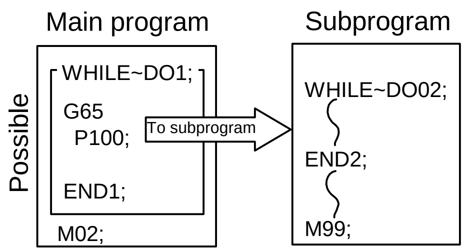
(8) Branching externally is possible from the WHILE – DOm range.



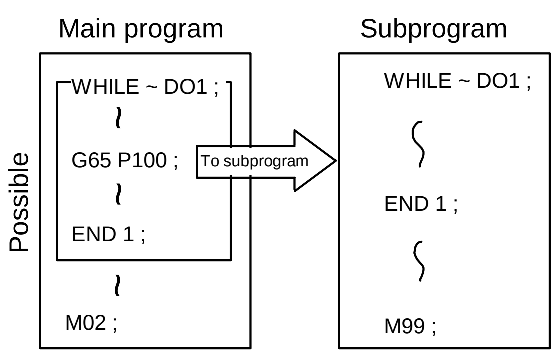
(9) No branching is possible inside WHILE – DOm.



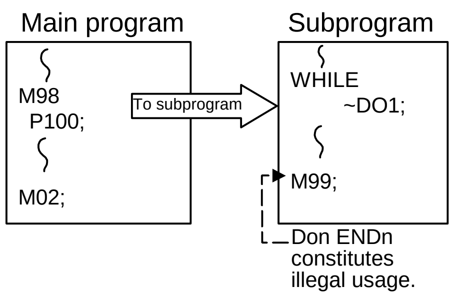
(10) Subprograms can be called by M98, G65 or G66 between WHILE – DOm's.



(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.



(12) A program error will occur at M99 if WHILE and END are not paired in the subprogram (including macro subprogram).



(Note) As the canned cycles G73 and G83 and the special canned cycle G34 use WHILE, these will be added multiple times.

13.5.7 External output commands

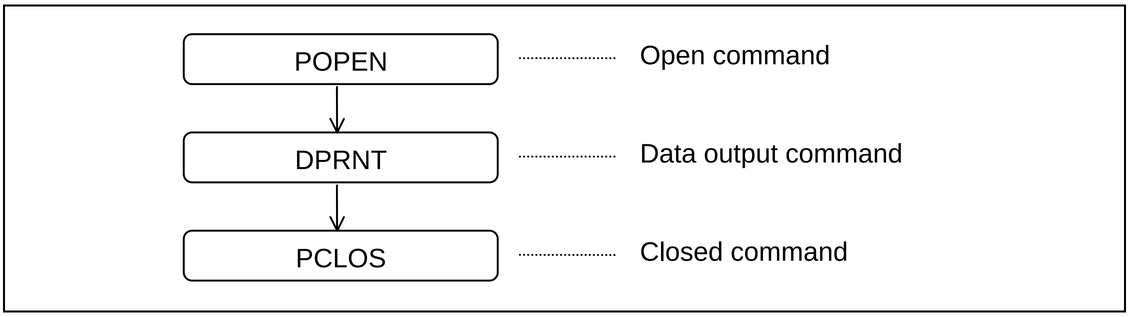
Function and purpose

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



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 [l1 # v1 [d1 c1] l 2 # v2 [d2 c2] • • • • • • • • • • •]		
l1	: Character string	
v1	: Variable number	
d1	: Significant digits above decimal point	c + d ≤ 8
c1	: Significant digits below decimal point	

(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.

(Note) A data output command can be issued even in dual-system mode. In this case, however, note that the output channel is shared for both systems. So, take care not to execute data output in both systems simultaneously.

13.5.8 Precautions

Precautions

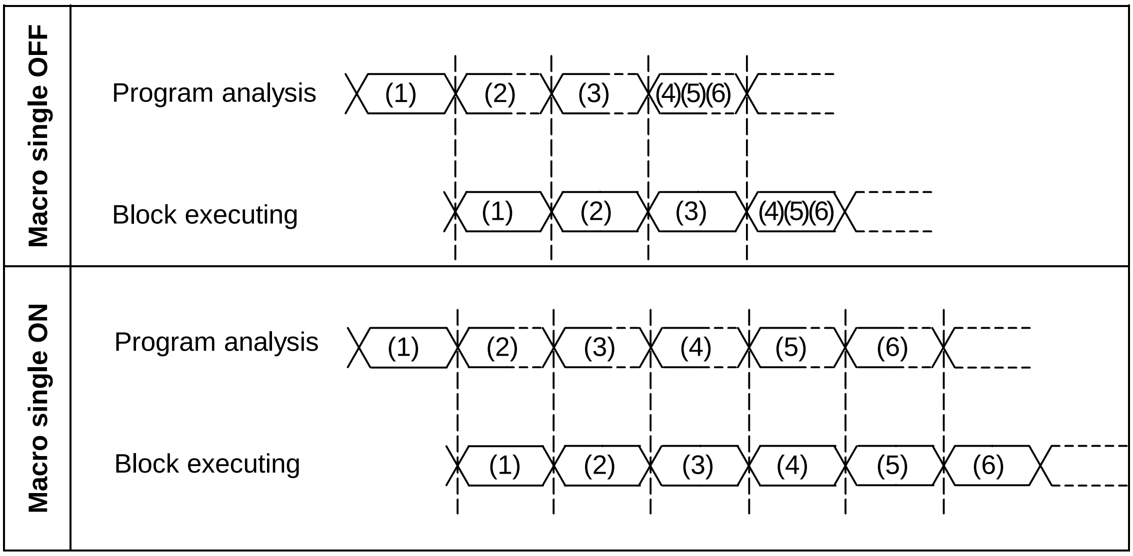
When the user macro commands are employed, it is possible to use the M, S, T and other NC control commands together with the arithmetic, decision, branching and other macro commands for preparing the machining programs. When the former commands are made into executable statements and the latter commands 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)
#102=100.∗ SIN   [210.]   ;       •••••(5)  ← Macro statement
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



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.

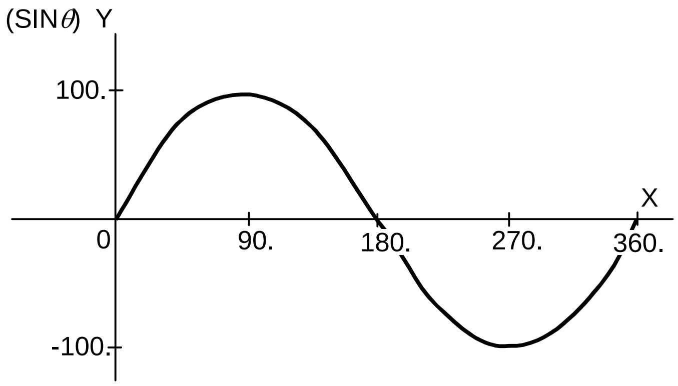
13.5.9 Actual examples of using user macros

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
```



Main program

```
        ~
G65P9910A0B360.C100.F100;
        ~
```

To subprogram

Local variable set by argument

```
#1=0
#2=360.000
#3=100.000
#9=100.000
```

O9910 (Subprogram)

```
WHILE [#1LE#2] DO1;
#101=#3∗SIN [#1] ;      } (Note 1)
G90G01X#1Y#101F#9;      }
#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

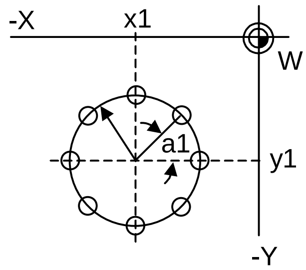
After defining the hole data with canned cycle (G72 to G89), the macro command is issued as the hole position command.

Main program

```
      ≀
G81Z–100.R50.F300L0
G65P9920Aa1Bb1Rr1Xx1Yy1 ;
      ≀
```

To subprogram

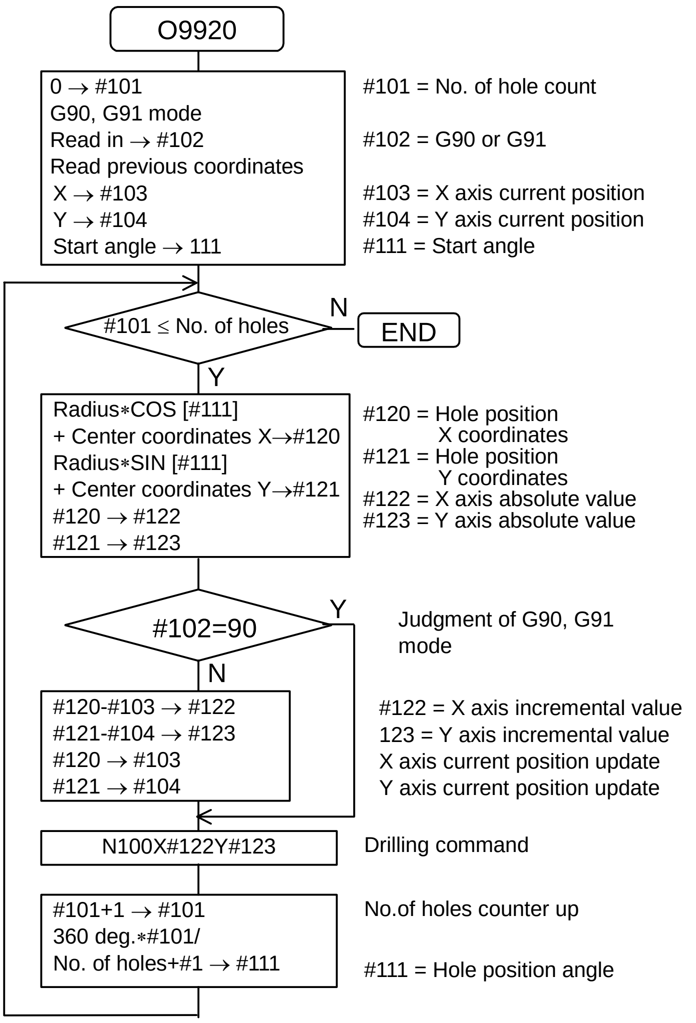
a1 ; Start angle
b1 ; No. of holes
r1 ; Radius
x1 ; X axis center position
y1 ; Y axis center position



O9920 (Subprogram)

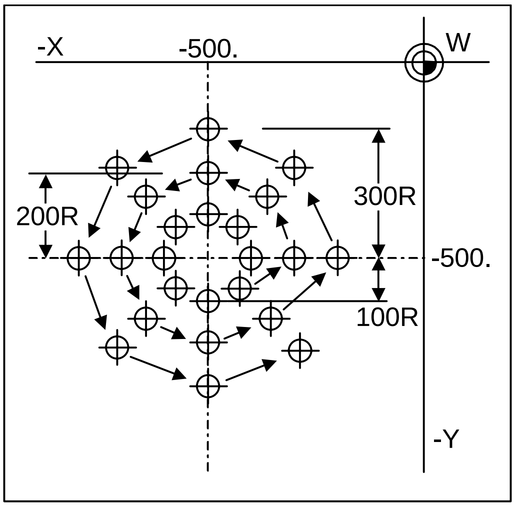
```
#101=0 ;
#102=#4003 ;
#103=#5001 ;        (Note 1)
#104=#5002 ;
#111=#1 ;

WHILE [#101LT#2] DO1 ;

#120=#24+#18∗COS [#111] ;   (Note 1)
#121=#25+#18∗SIN [#111] ;

#122=#120 #123=#121 ;
IF [#102EQ90] GOTO100 ;

#122=#120 – #103 ;   (Note 1)
#123=#121 – #104 ;

#103=#120   (Note 1)
#104=#121

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.



```
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.;
G65 P9920 X-500. Y-500. A0 B8 R200.;
G65 P9920 X-500. Y-500. A0 B8 R300.;
        •
        •
```

To subprogram
To subprogram
To subprogram



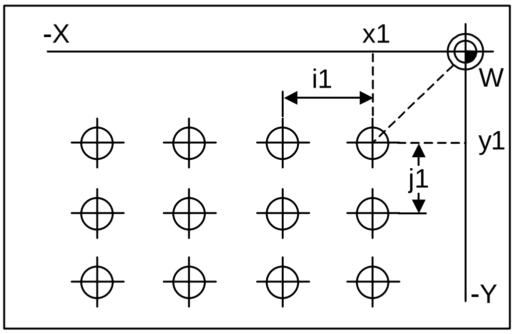
(Example 3) Grid

After defining the hole data with the canned cycle (G72 to G89), macro call is commanded as a hole position command.

```
G81 Zz1 Rr1 Ff1;
G65Pp1 Xx1 Yy1 Ii1 Jj1 Aa1 Bb1;
```

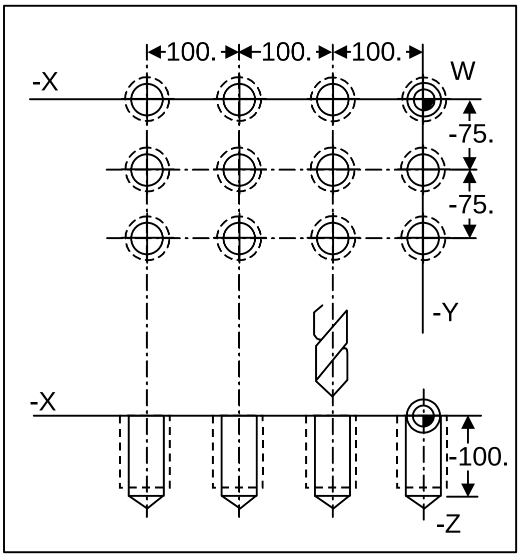
x1 ; X axis hole position
y1 ; Y axis hole position
i1 ; X axis interval
j1 ; Y axis interval
a1 ; No. of holes in X direction
b1 ; No. of holes in Y direction

Subprogram is on next page



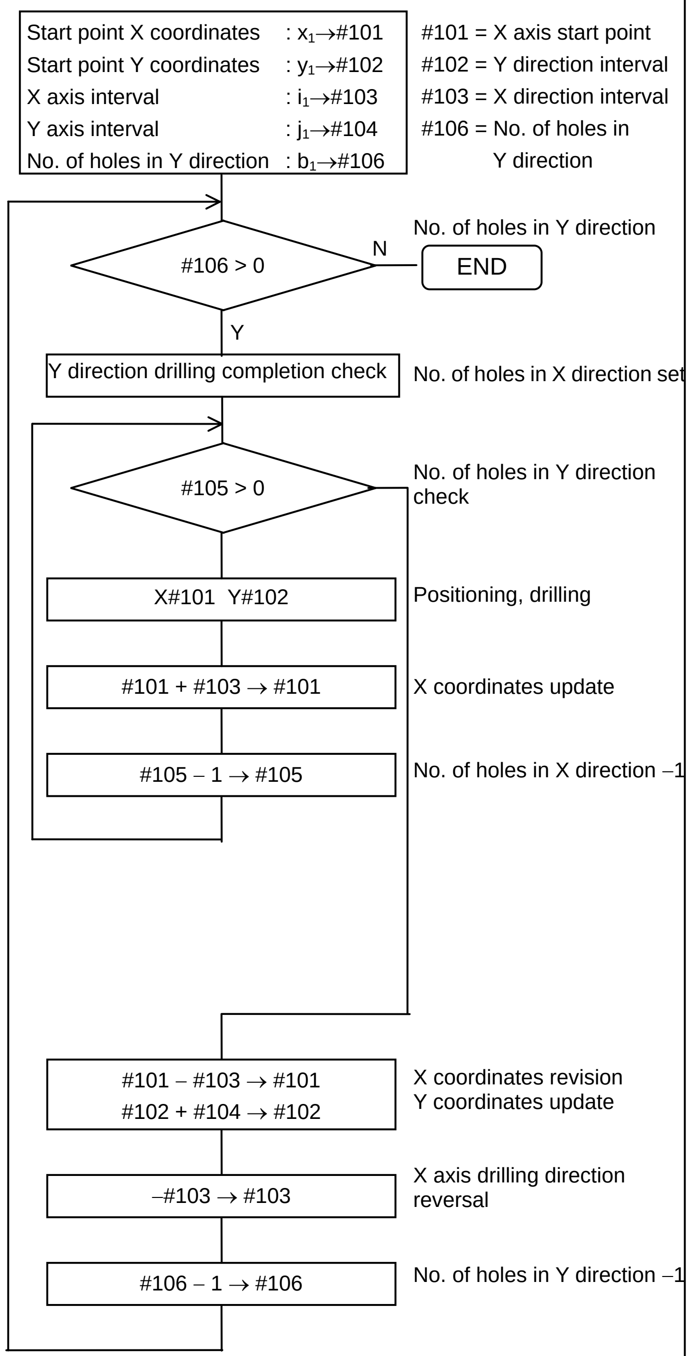

```
G28 X0 Y0 Z0;
T1 M06;
G90 G43 Z100.H01;
G54 G00 X0 Y0;
G81 Z-100. R3.F100 L0 M03;
G65 P9930 X0 Y0 I-100. J-75. A5B3;
        ~
G84 Z-90. R3. F250 M03;
G65 P9930 X0 I-100. J-75. A5B3;
        ~
```

To subprogram
To subprogram



O9930 (Subprogram) O9930

```
#101=#24 ;
#102=#25 ;
#103=#4 ;
#104=#5 ;          } (Note 1)
#106=#2 ;

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 ;
```

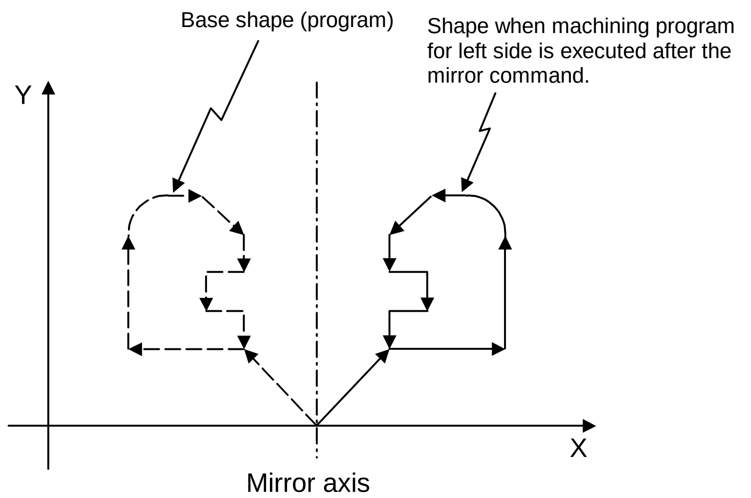


(Note 1) The processing time can be shortened by programming in one block.

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 Xx_1 Yy_1 Zz_1 ; (Mirror image ON)
G50.1 Xx_2 Yy_2 Zz_2 ; (Mirror image OFF)
Xx/Yy/Zz : Mirror image command axis

Detailed description

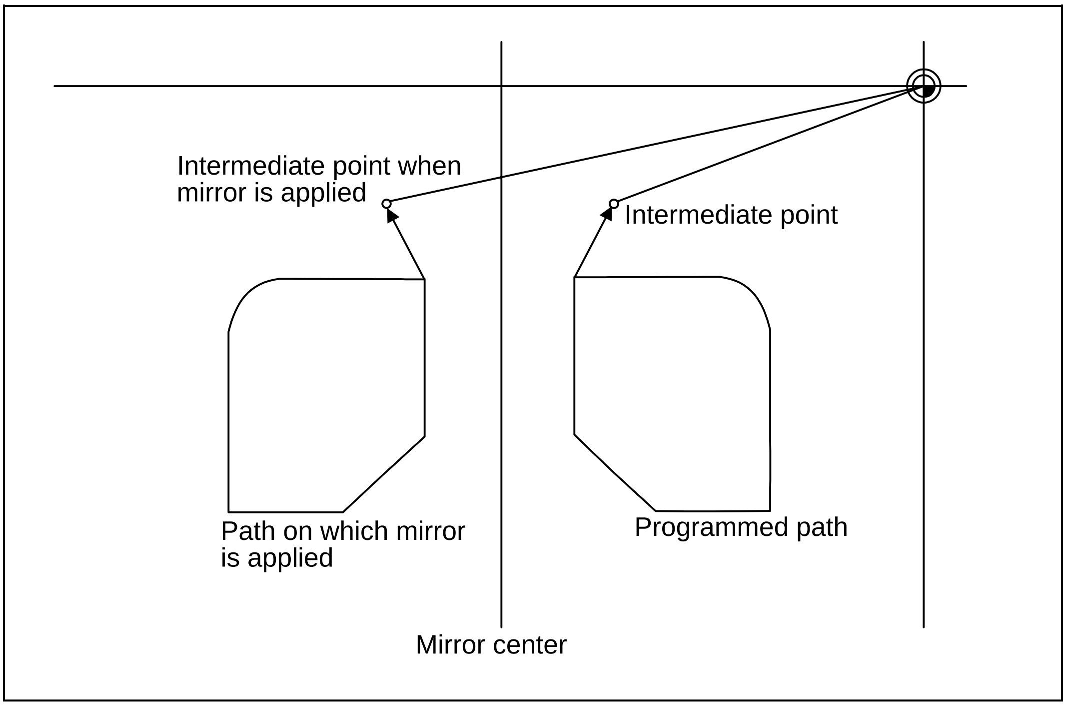
(1) The coordinate word for G51.1 is commanded with the mirror image command axis, and the coordinate value commands the mirror image center coordinate with an absolute value or incremental value.

(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.



(6) Return from zero point during mirror image
If the return command (G29) from the zero point is commanded during the mirror image, the mirror will be applied on the intermediate point.

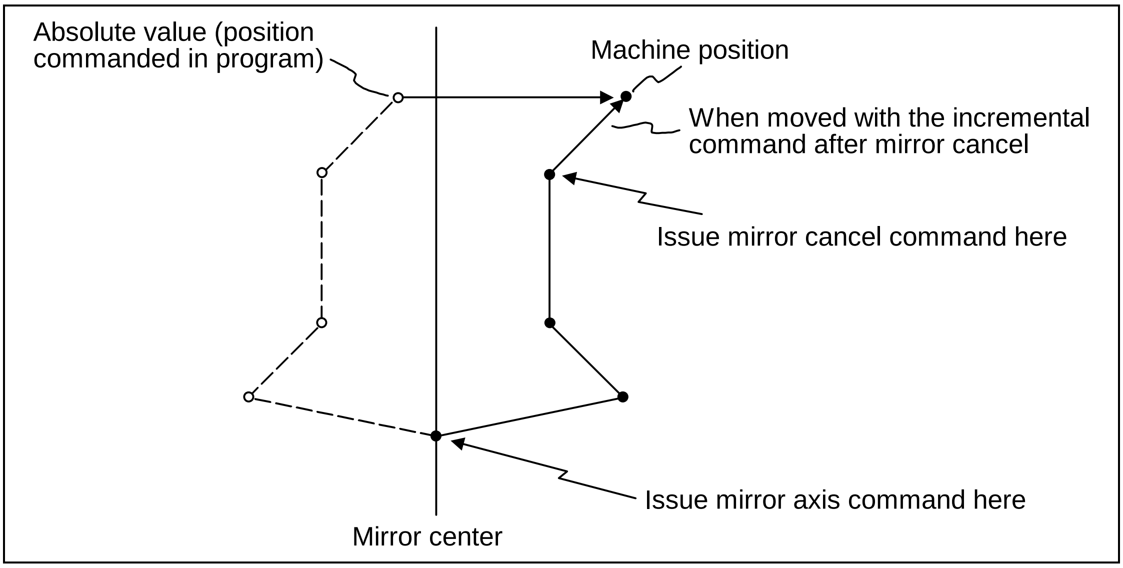
(7) The mirror image will not be applied on the G53 command.

Precautions

CAUTION

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

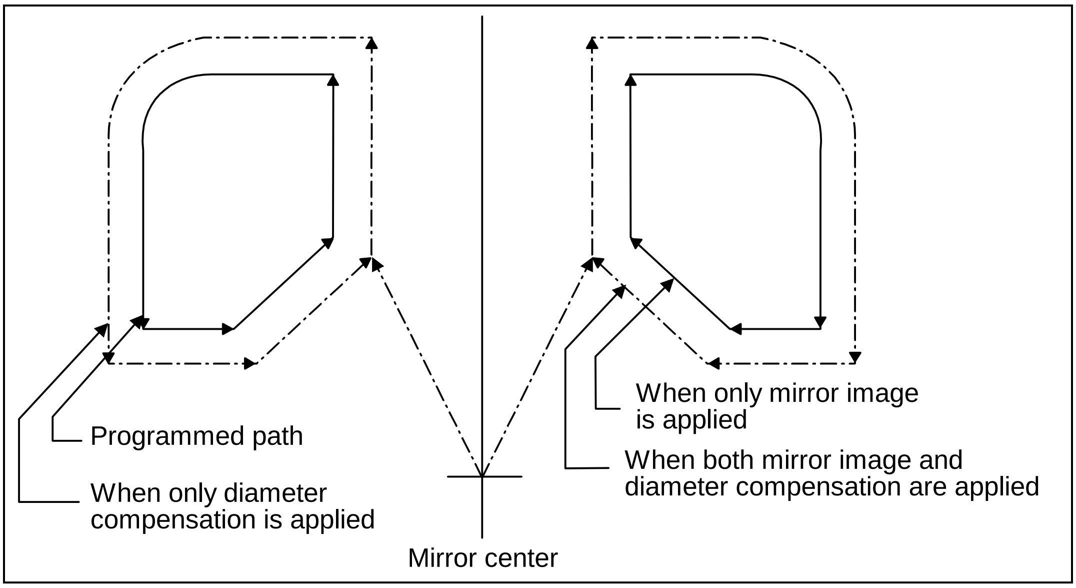
If mirror image is canceled at a point other than the mirror center, the absolute value and machine position will deviate as shown below. (In this state, execute the absolute value command (positioning with G90 mode), or execute reference point return with G28 or G30 to continue the operation.) The mirror 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.



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.



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.

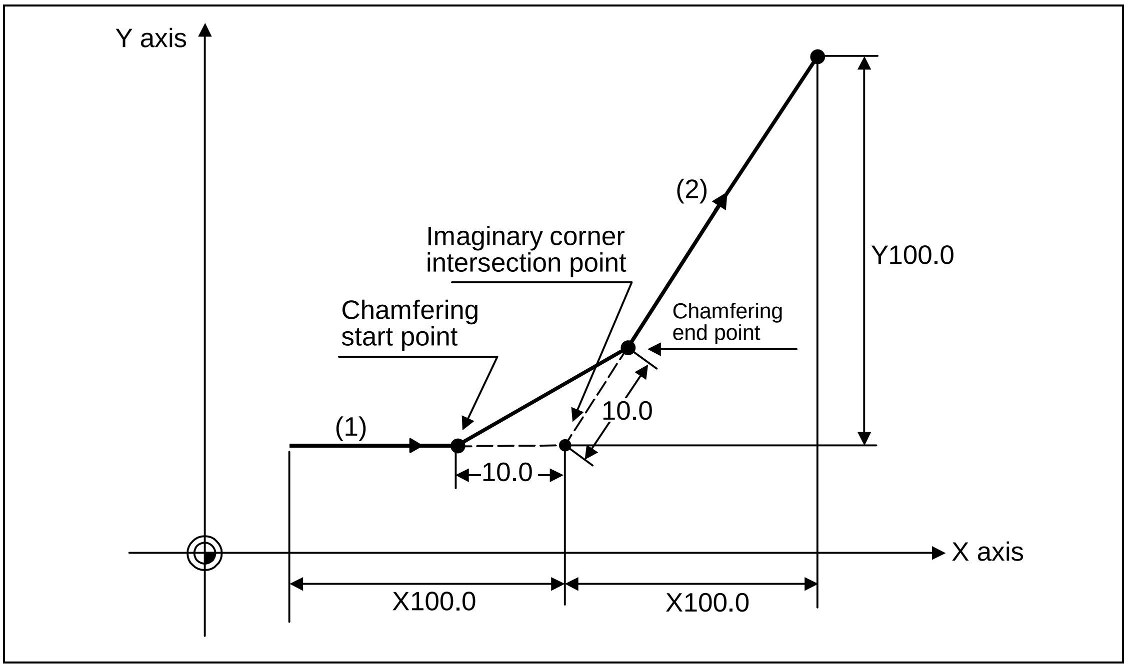
Command format

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. ;



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) Program error "P381" results when there is an arc command in the block following the corner chamfering block.
(6) Program error "P382" results when the block following the corner chamfering block does not have a linear command.
(7) Program error "P383" results when the movement amount in the corner chamfering block is less than the chamfering amount.
(8) 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.

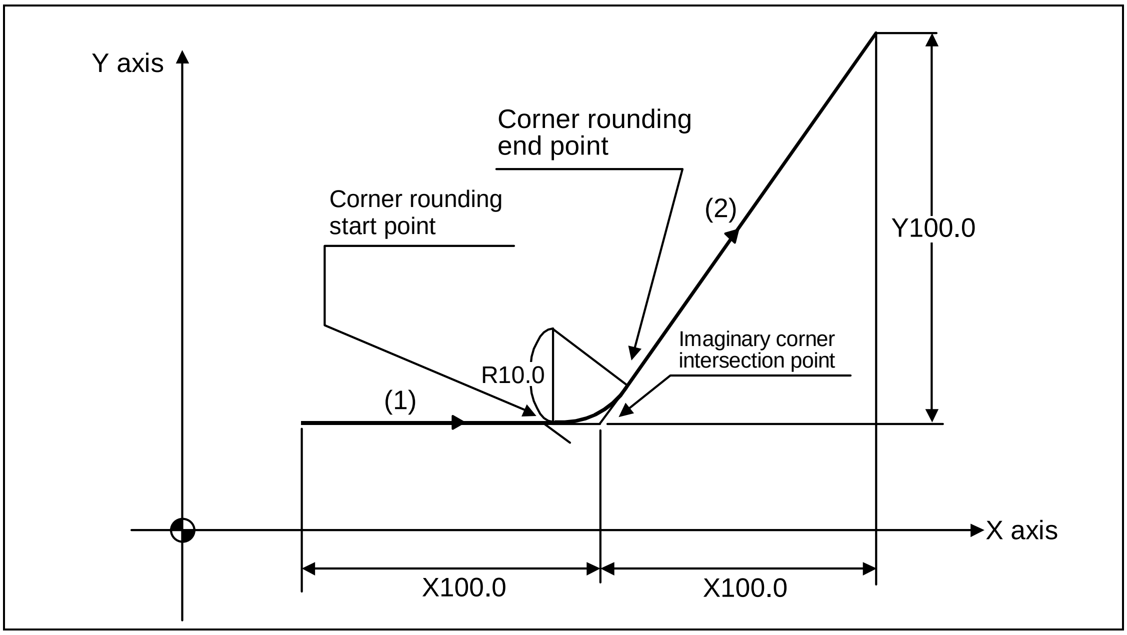
Command format

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 Circle cutting; G12, G13

Function and purpose

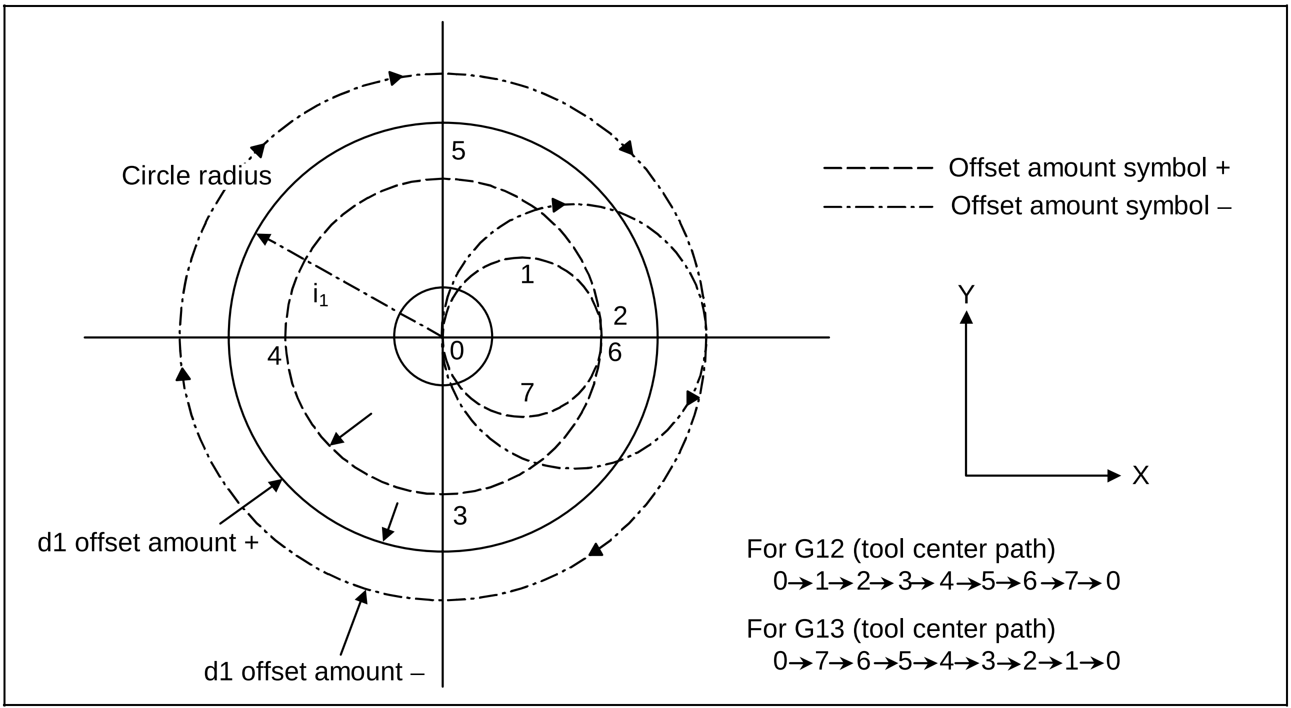
Circle cutting starts the tool from the center of the circle, and cuts the inner circumference of the circle. The tool continues cutting while drawing a circle and returns to the center position.

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

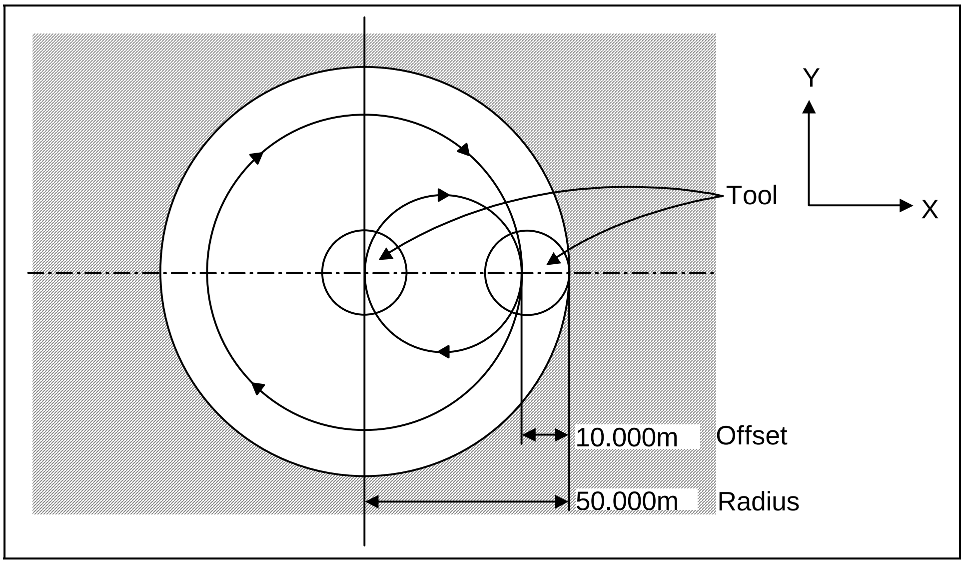
Detailed description

(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.



Example of program

(Example 1) G12 I5000 D01 F100 ; (Input setting unit 0.01)
When offset amount is +10.00mm



Cautions

(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.

13.9 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 <u>major classification number</u> N <u>data number</u> H <u>bit type data</u> ;
P <u>major classification number</u> A <u>axis number</u> N <u>data number</u>
D <u>byte type data</u> ;
P <u>major classification number</u> A <u>axis number</u> N <u>data number</u> S <u>word type data</u> ;
P <u>major classification number</u> A <u>axis number</u> N <u>data number</u> L <u>2 word type data</u> ;

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) Whether the parameter value is replaced or added depends on the modal state of G90/G91 when G10 is commanded.

(Note 3) Refer to Appendix Table 1 for the P, N number correspondence table.

(Note 4) For a bit type parameter, the data type will be H□ (□ is a value between 0 and 7).

(Note 5) The axis number is set in the following manner: 1st axis is 1, 2nd axis is 2, and so forth. When using multiple part system, the 1st axis in each part system is set as 1, the second axis is set as 2, and so forth.

(Note 6) Command G10L50, L11 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  ;
```

13.10 Macro interrupt ; M96, M97

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__ H__ ;	**User macro interrupt enable**
M97 ;	**User macro interrupt disable**
P	:Interrupt program number
H	:Interrupt sequence 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.

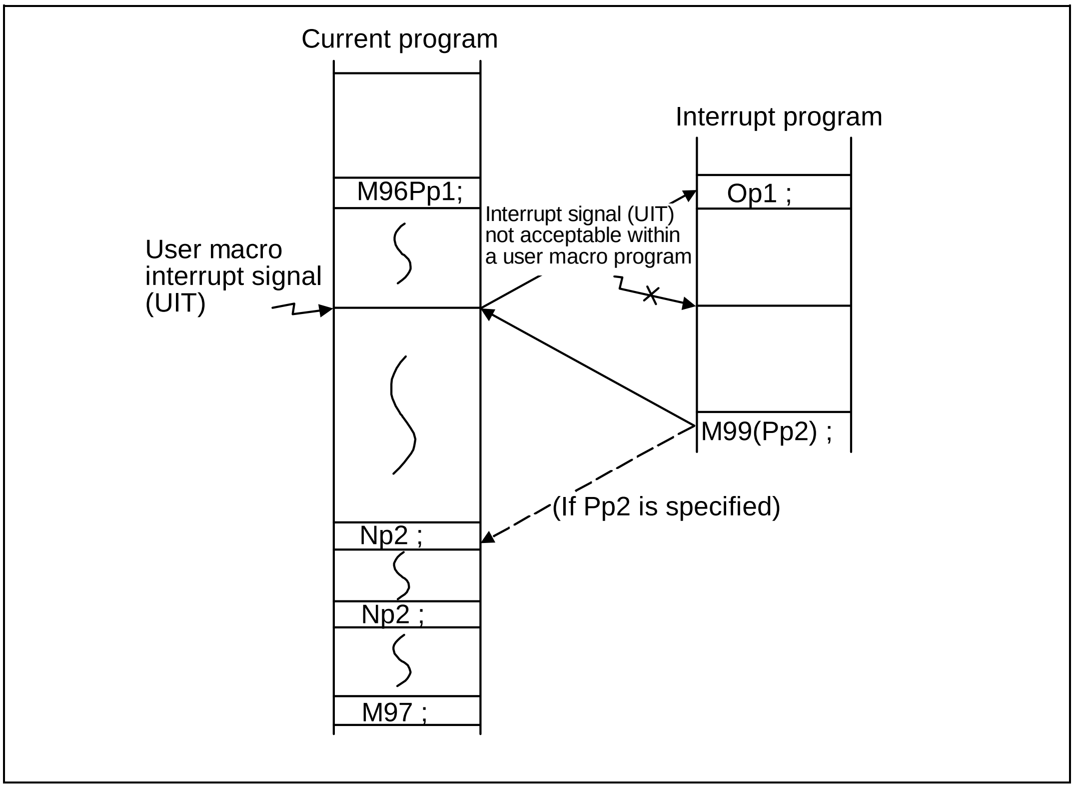
Interrupt enable conditions

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 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

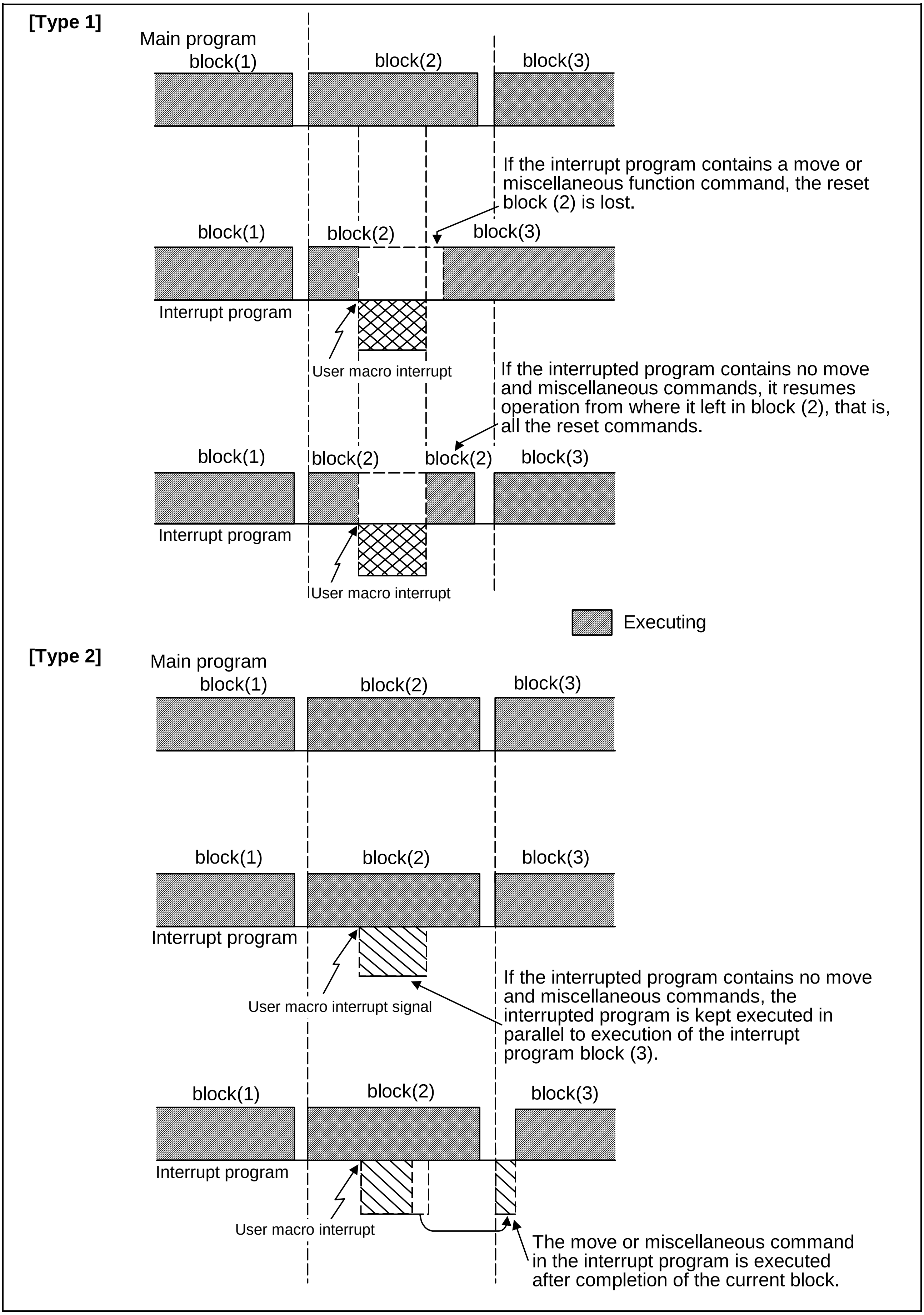
Interrupt types 1 and 2 can be selected by the parameter "#1113 INT_2".

[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

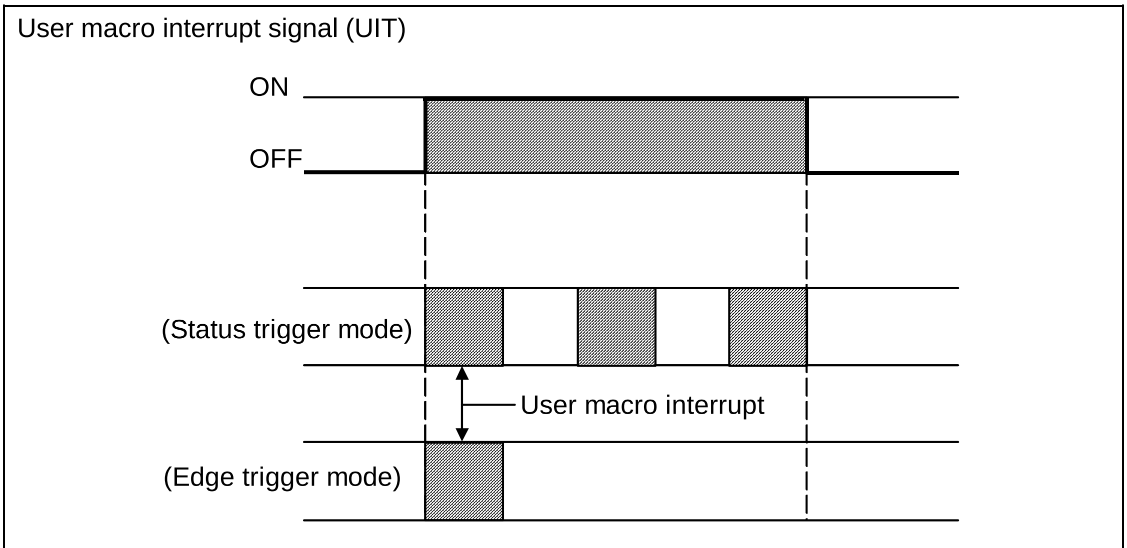
Subprogram type interrupt	The user macro interrupt program is called as a subprogram. As with calling by M98, the local variable level remains unchanged before and after an interrupt.
Macro type interrupt	The user macro interrupt program is called as a user macro. As with calling by G65, the local variable 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.



Accepting user macro interrupt signal (UIT)

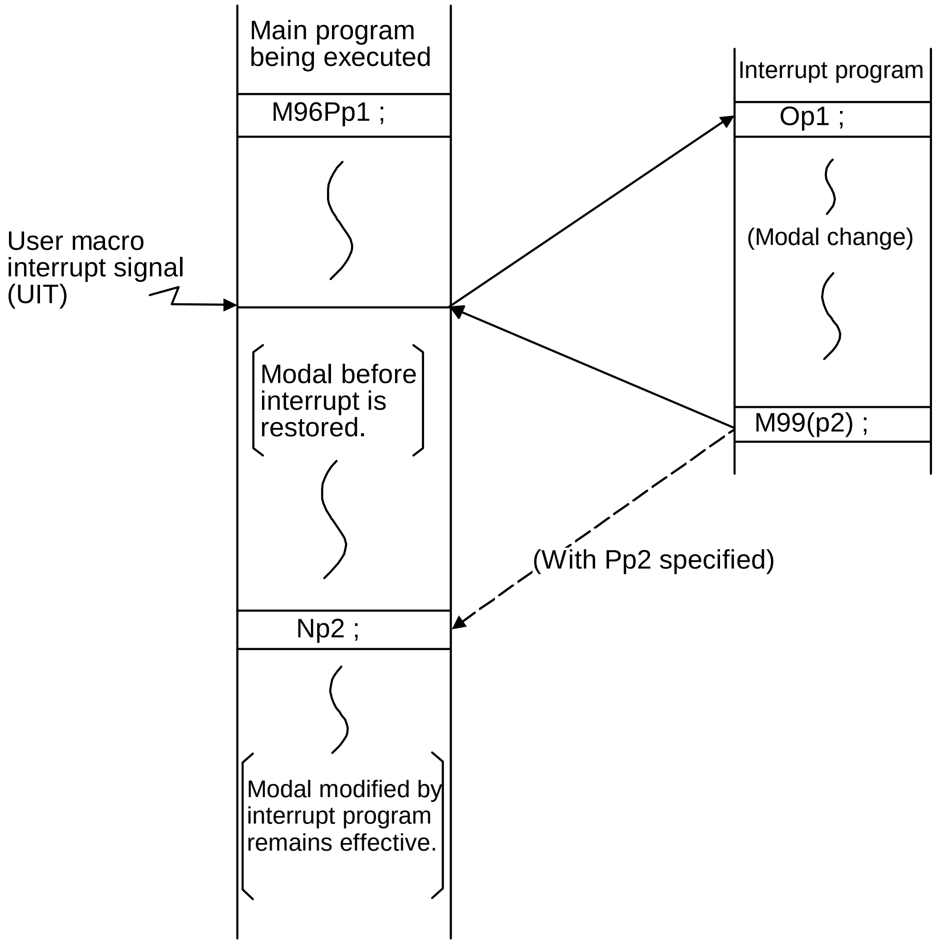
Returning from user macro interrupt

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 Np2; 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.



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 Instruction 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.

13.11 Tool change position return ; G30.1 to G30.6

Function and purpose

By specifying the tool change position in a parameter "#8206 TOOL CHG. P" and also specifying a tool change 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 change position and the order in which the axes begin to return can be changed by commands.

Command format

(1) The format of tool change position return commands is as follows.

G30. n;
n = 1 to 6 : Specify the axes that return to the tool change position and the order in which they return.

For the commands and return order, see next table.

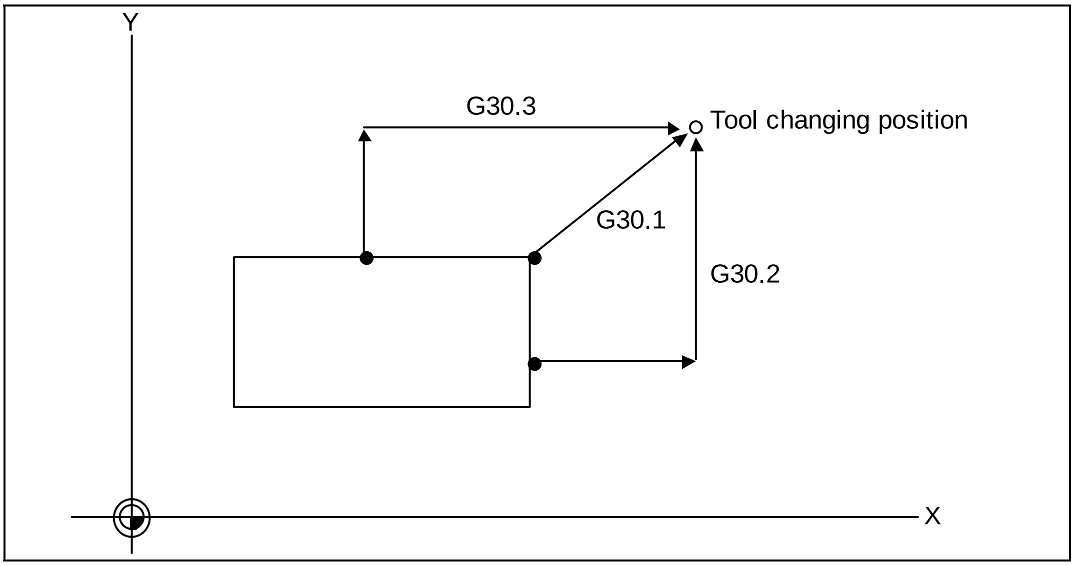
Command	Return order	
G30.1	Z axis → X axis • Y axis	(→ added axis)
G30.2	Z axis → X axis → Y axis	(→ added axis)
G30.3	Z axis → Y axis → X axis	(→ added axis)
G30.4	X axis → Y axis • Z axis	(→ added axis)
G30.5	Y axis → X axis • Z axis	(→ added axis)
G30.6	X axis • Y axis • Z axis	(→ added axis)

(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 change position, then the X and Y axes does.)

(2) The tool change 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 change position only after the standard axes complete returning (see the above table).
The added axis alone cannot return to the tool change position.

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.)



1) G30.1 command: The Z axis returns to the tool change position, then the X and Y axes simultaneously do the same thing. (If tool change position return is on for an added axis, the added axis also returns to the tool change position after the X, Y and Z axes reach the tool change position.)

2) G30.2 command: The Z axis returns to the tool change position, then the X axis does the same thing. After that, the Y axis returns to the tool change position.
(If tool change position return is on for an added axis, the added axis also returns to the tool change position after the X, Y and Z axes reach the tool change position.)

3) G30.3 command: The Z axis returns to the tool change position, then the X axis does the same thing. After that, the X axis returns to the tool change position.
(If tool change position return is on for an added axis, the added axis also returns to the tool change position after the X and Z axes reach the tool change position.)

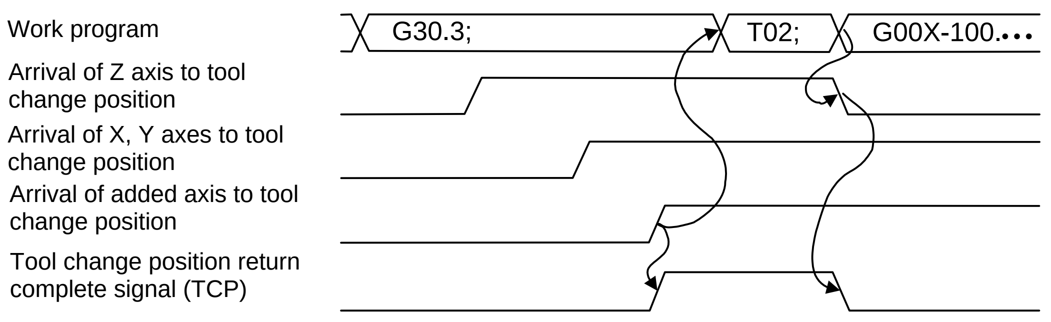
4) G30.4 command: The X axis returns to the tool change position, then the Y axis and Z axis simultaneously do the same thing.
(If tool change position return is on for an added axis, the added axis also return to the tool change position after the X, Y and X axes reach the tool change position.)

5) G30.5 command: The Y axis returns to the tool change position, then the X and Z axes return to the tool change position simultaneously.
(If tool change position return is on for an added axis, the added axis also returns to the tool change position after the X, Y and Z axes reach the tool change position.)

6) G30.6 command: The X, Y and Z axes return to the tool change position simultaneously.
(If tool change position return is on for an added axis, the added axis also returns to the tool change position after the X, Y and Z axes reach the tool change position.)

(2) After all necessary tool change position return is completed by a G30.n command, tool change position return complete signal TCP (X64B) is turned on. When an axis out of those having returned to the tool change position by a G30.n command leaves the tool change 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 change position after the X and Y axes did (after the additional axis did if additional axis tool change position return is valid). The TCP signal is then turned off when the X or Y axis leaves the position. If tool change 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 change 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] (G30.3 command with tool change position return for added axes set on)



(3) When a tool change position return command is issued, tool offset data such as for tool length offset and tool radias 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 change position. To make the next axis return to the tool change position, therefore, a cycle start needs to be specified.

13.12 High-accuracy control; G61.1

Function and purpose

Until now, trouble such as the following occurred when using control:

(1) Corner rounding occurred at the corners that linear and linear are connected 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)

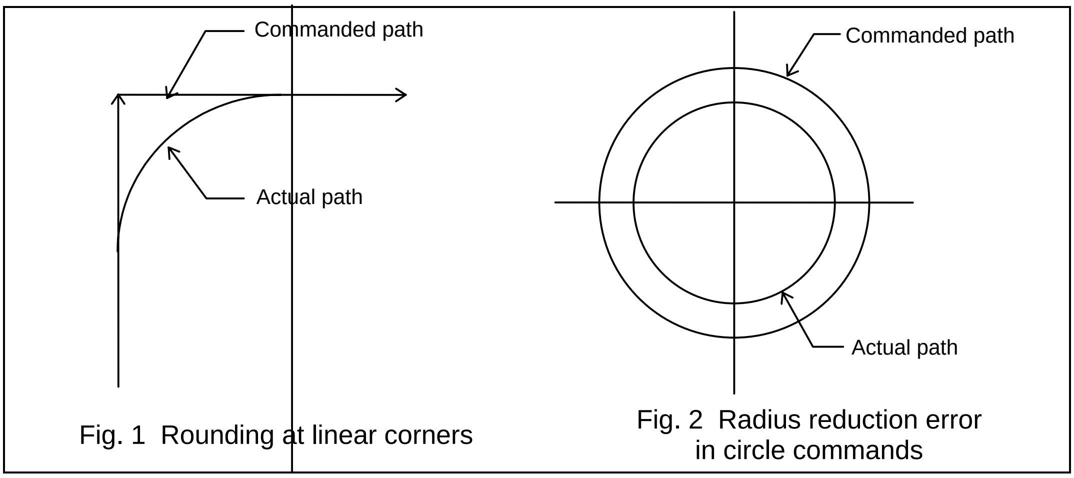


Fig. 1 Rounding at linear corners

Fig. 2 Radius reduction error in circle commands

This function controls the operation so the lag is eliminated in control systems and servo systems. With this function, machining accuracy can be improved, especially during high-speed machining, and machining time can be reduced.
The high-accuracy control function is configured of the following functions.

(1) Pre-interpolation acceleration/deceleration (linear acceleration/deceleration)
(2) Optimum speed control
(3) Vector accuracy interpolation
(4) Active feed forward
(5) Arc entrance/exit speed control

Command format

G61.1 Ff1 ;	
G61.1	: High-accuracy control mode
f1	: Feedrate

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)

Detailed description

(1) The "high-accuracy control" specifications are required to use this function.
If G61.1 is commanded when the specifications are not available, program error (P123) will occur.

(2) The feedrate command F is clamped by the rapid traverse rate or maximum cutting feedrate set with the parameters.

(3) Refer to the "Optimum speed control" mentioned later for details on the speed clamp during an arc command.

(4) The own system waits for the other system to move and reach the designated start point, and then starts.

(5) The modal holding state of the high-accuracy control mode depends on the conditions of the base specification parameter "#1151 rstint" (reset initial) and "#1148 I_G611" (initial high-accuracy).

Parameter		Default state	Reset			Block interruption	Block stop	Emergency stop	NC alarm	OT	Emergency stop cancel
Reset initial (#1151)	Initial high accuracy (#1148)	Power ON	Reset 1	Reset 2	Reset & rewind	Mode changeover (automatic/manual) / Feed hold	Single block	Emergency stop switch / External emergency stop	Servo alarm	H/W OT	Emergency stop switch / External emergency stop
OFF	OFF	C	H	C	C	H	H	H	H	H	H
OFF	ON	H	* H	* H	H	H	H	H	H	H	H
ON	OFF	C	C	C	C	H	H	H	H	H	C
ON	ON	H	* H	* H	H	H	H	H	H	H	* H

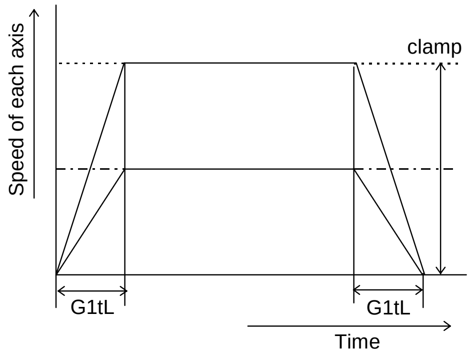
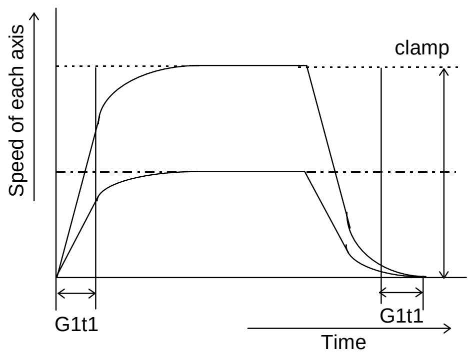
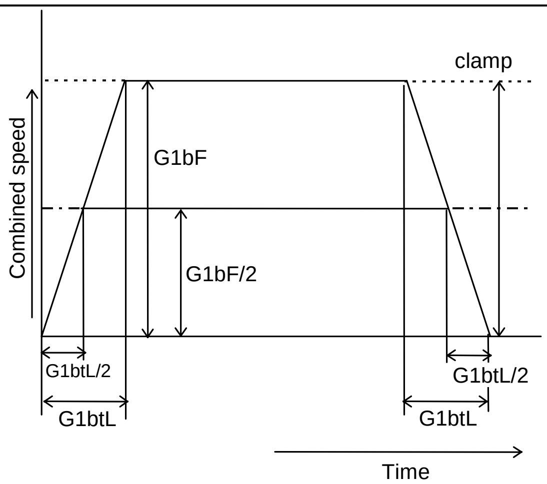
H (hold) : Modal hold (G61.1 → G61.1)
C (cancel) : Modal cancel (G61.1 → G64)

(Note) The cases marked with an asterisk (*) in the above table indicate that the modal will shift to the high-accuracy control mode (G61.1) even in modes other than the high-accuracy control mode (modes G61 to G64).

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 commanded 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

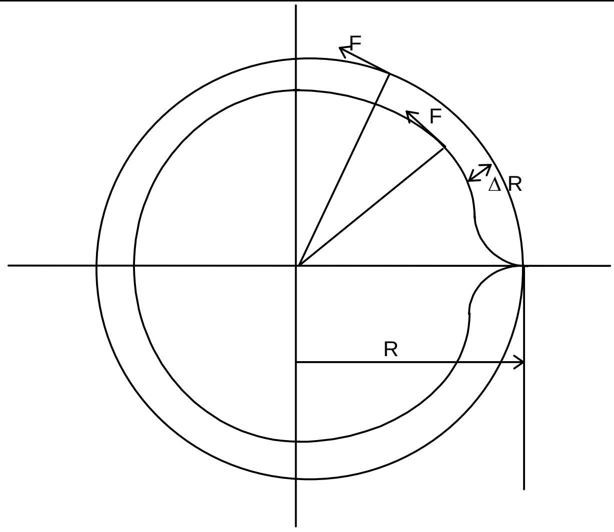
	Acceleration/deceleration pattern	
Normal mode	  	(a) Because of the constant time constant acceleration/deceleration, the rising edge/falling edge becomes more gentle as the command speed becomes slower. (b) The acceleration/deceleration time constant can be independently set for each axis. Linear type, 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 function 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 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.

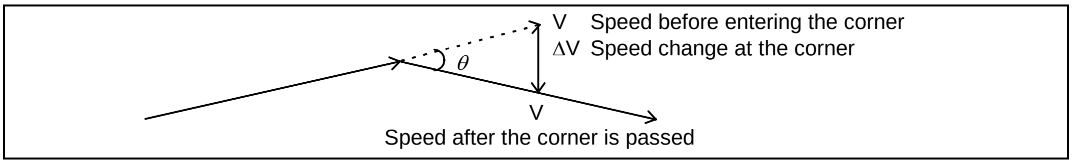
Ts : Acceleration/deceleration time constant in the CNC (s)
Tp : Servo system position loop time constant (s)
Kf : Feed forward coefficient

Optimum speed control

(1) Optimum corner deceleration

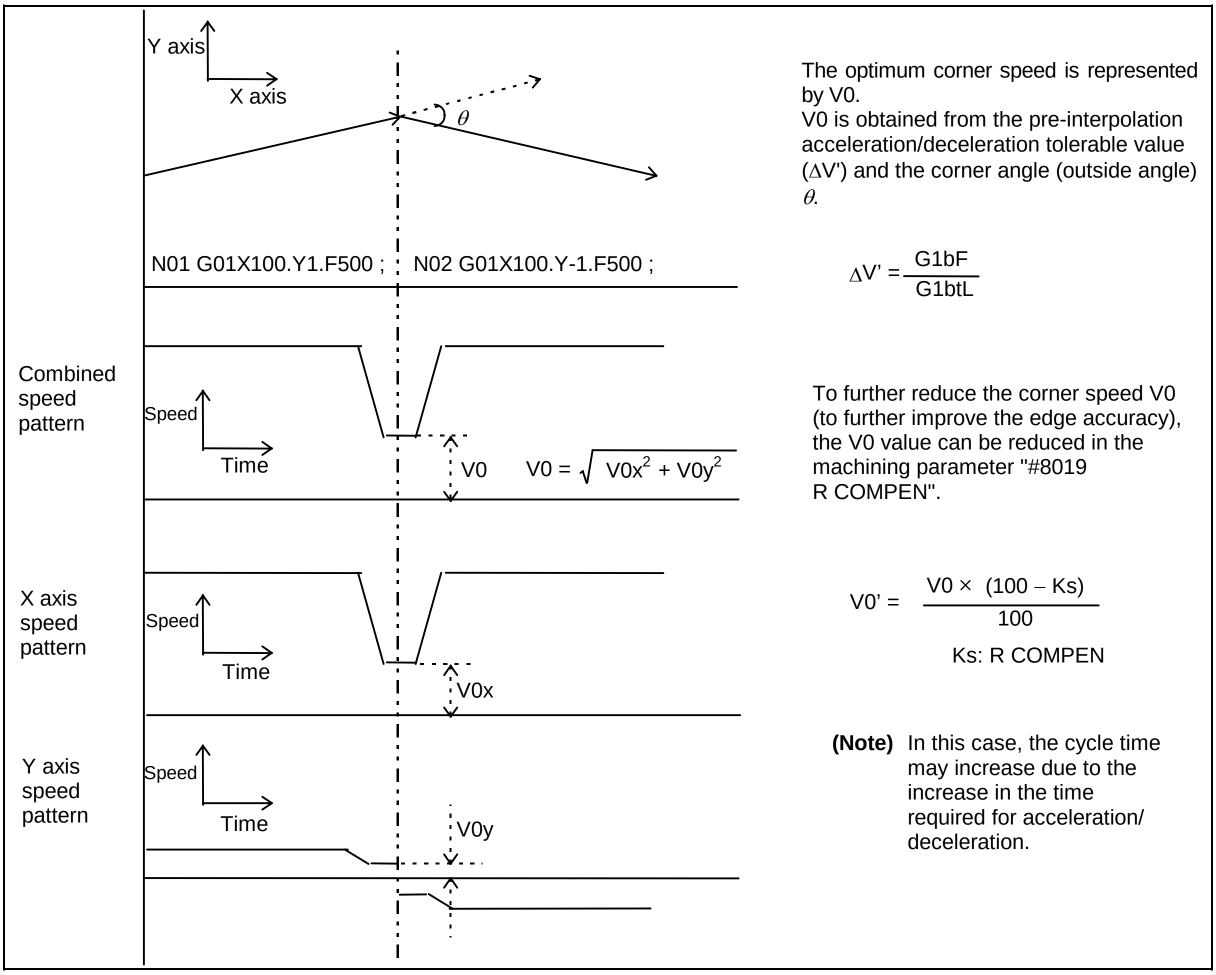
By calculating the angle of the seam between blocks, and carrying out acceleration/ deceleration control in which the corner is passed at the optimum speed, highly accurate edge machining can be realized.
When the corner is entered, that corners optimum speed (optimum corner speed) is calculated from 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, and the corner is passed at a speed V, the acceleration ΔV occurs due to the change in the direction of progress.



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").

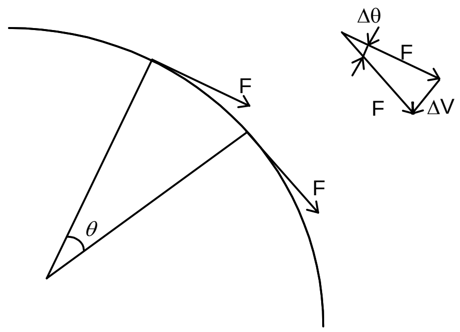
In this case the speed pattern is as follows.



(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 ΔV.

$$F \leq \sqrt{R \times \Delta V \times 60 \times 1000}\ \text{(mm/min)}$$

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

When the above F' expression is substituted in the expression expressing the maximum logical arc radius reduction error amount ΔR explained in the section "a) 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 ΔR by the set percentage.

$$\Delta R' \leq \frac{\Delta R \times (100 - Ks)}{100}\ \text{(mm)}$$

ΔR' : Maximum arc radius reduction error amount
Ks : R COMPEN (%)

After setting the "R COMPEN", the above ΔR' will appear on the parameter screen.

R COMPEN (0.078) 50

Accuracy coefficient setting value

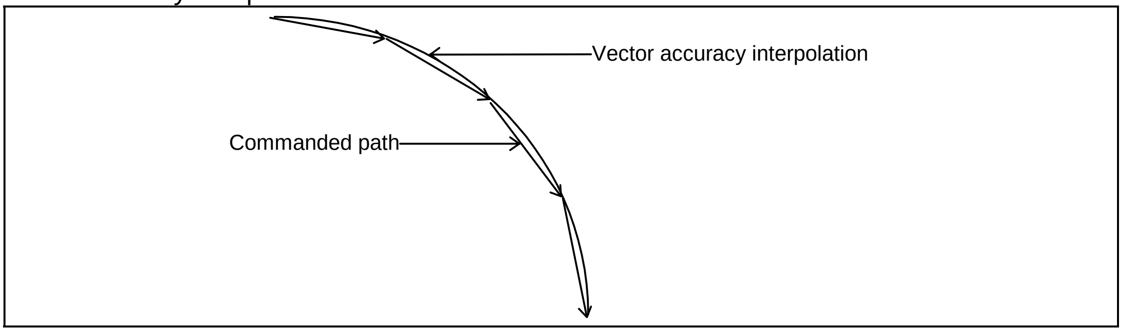
ΔR'

(Note 1) When the "R COMPEN" is set, the arc clamp speed will drop, so in a machining program with many arc commands, the machining time will take longer.

(Note 2) The "R COMPEN" is valid only when the arc speed clamp is applied. To reduce the radius reduction error when not using the arc speed clamp, the commanded speed F must be lowered.

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.

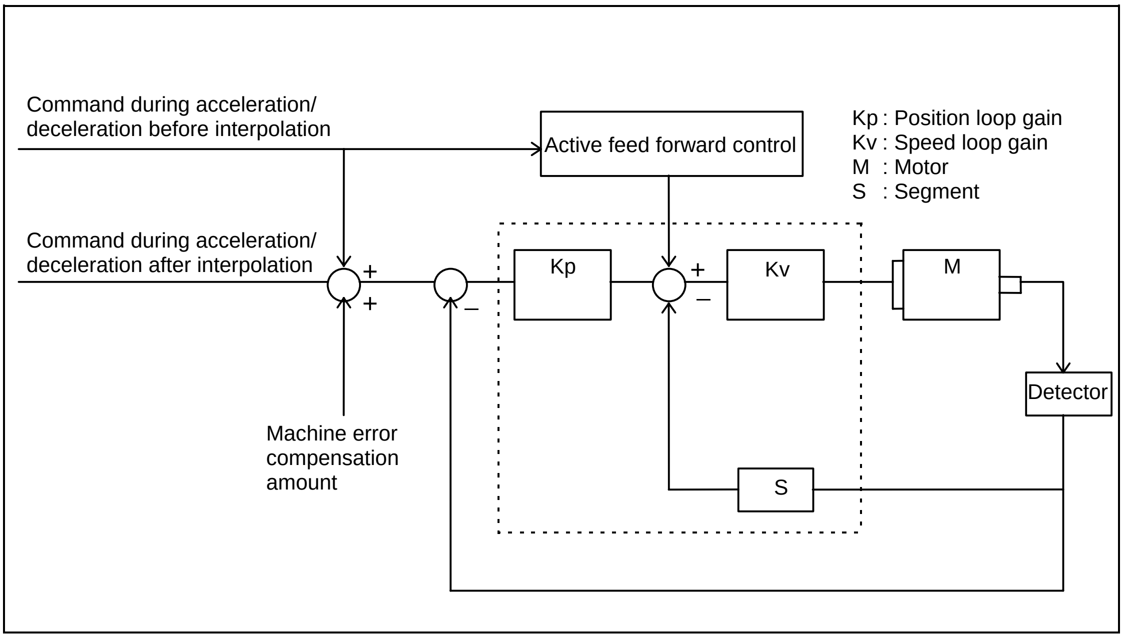


Feed forward control

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) Active feed forward control



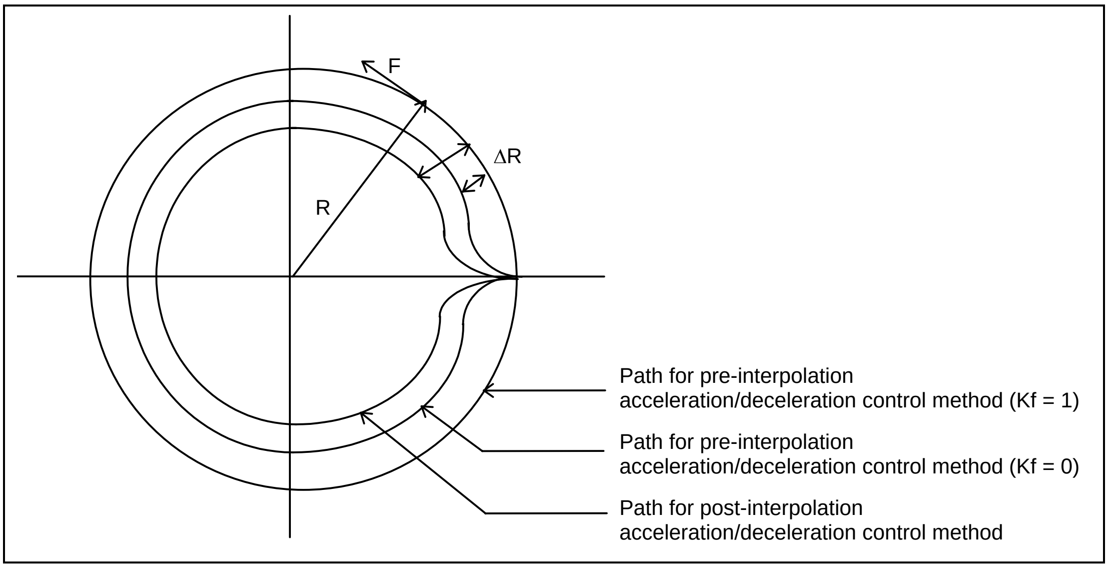
(2) Reduction of arc radius reduction error amount using feed forward control

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 active feed forward control/SHG control.

The logical radius reduction error amount ΔR in the high-accuracy control mode is obtained with the following expression.

<table>
<tr><th>Active feed forward control</th><th>SHG control + active feed forward control</th></tr>
<tr><td colspan="2">$$\Delta R \leq \frac{1}{2R} \{Tp^2 (1 - Kf^2)\} \left(\frac{F}{60}\right)^2$$
R : Arc radius (mm)
F : Cutting feedrate (mm/min)
Tp : Position loop time constant (s)
Kf : Feed forward coefficient</td></tr>
<tr><td colspan="2">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.</td></tr>
<tr><td>Kf = 1 (Feed forward gain 100%)</td><td>The equivalent feed forward gain to set Kf to 1 can be obtained with the following expression.
$$100\sqrt{1-\left\{1-\left(\frac{fwd\_g}{50}\right)^2\right\}\left(\frac{\text{PGN1 for conventional control}}{2 \times \text{PGN1 for SHG control}}\right)^2}$$</td></tr>
</table>

The feed forward gain can be set independently for G00 and G01.



(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.

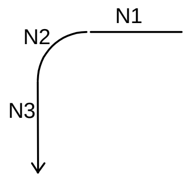
The validity of this control can be changed with the base specification parameter "#1149 cireft". The deceleration speed is designated with the base specification parameter "#1209 cirdcc".

(Example 1) When not using corner deceleration

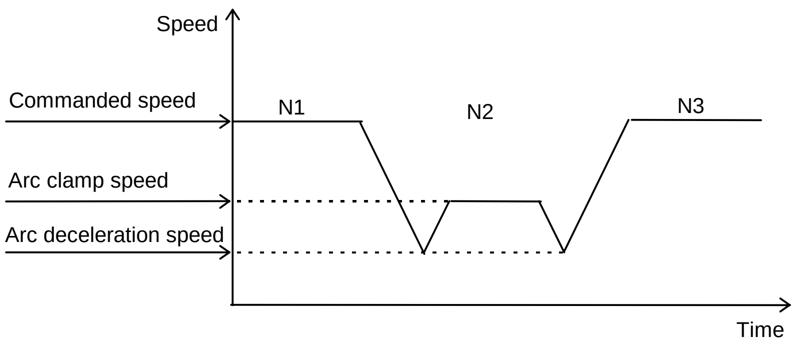
<Program>

```
G61.1 ;
 •
 •
N1 G01 X-10. F3000 ;
N2 G02 X-5. Y-5. J-2.5 ;
N3 G01 Y-10. ;
 •
 •
```

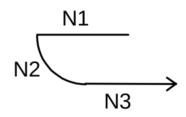
<Operation>



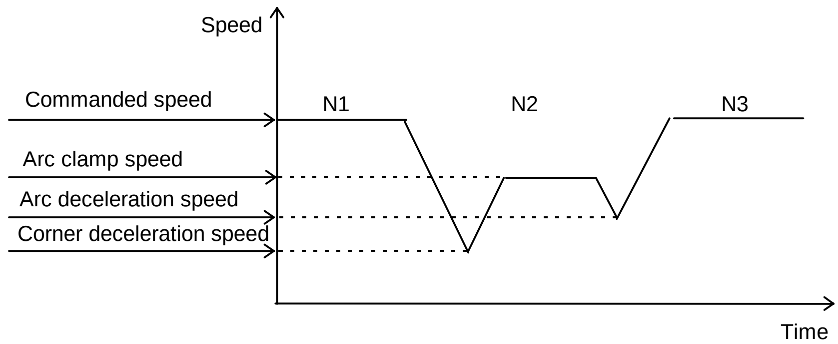
<Deceleration pattern>



(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>



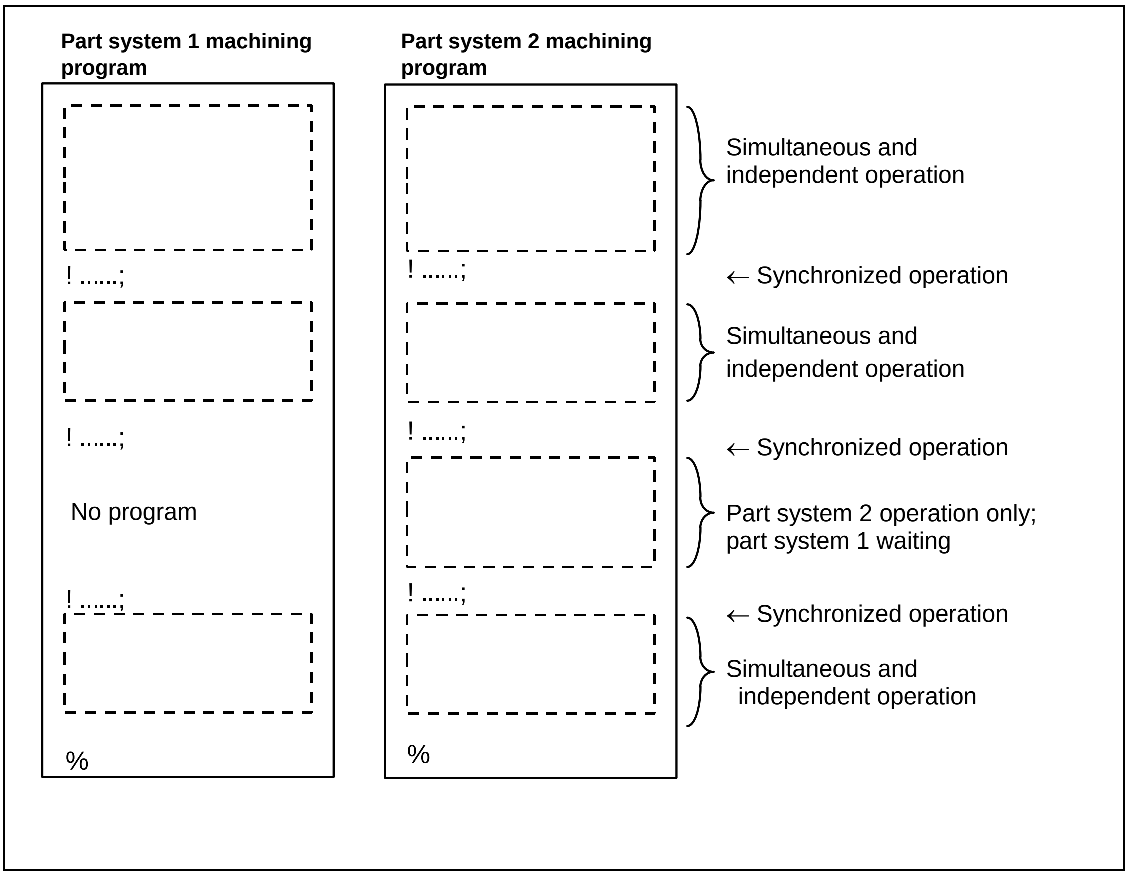
13.13 Synchronizing operation between part systems

⚠CAUTION

⚠ When programming a multi-part system, carefully observe the movements caused by other part systems' programs.

Function and purpose

The multi-axis, multi-part system complex control NC system can simultaneously run multiple machining programs independently. The synchronizing-between-part systems function is used in cases when, at some particular point during operation, the operations of part systems 1 and 2 are to be synchronized or in cases when the operation of only one part system is required.

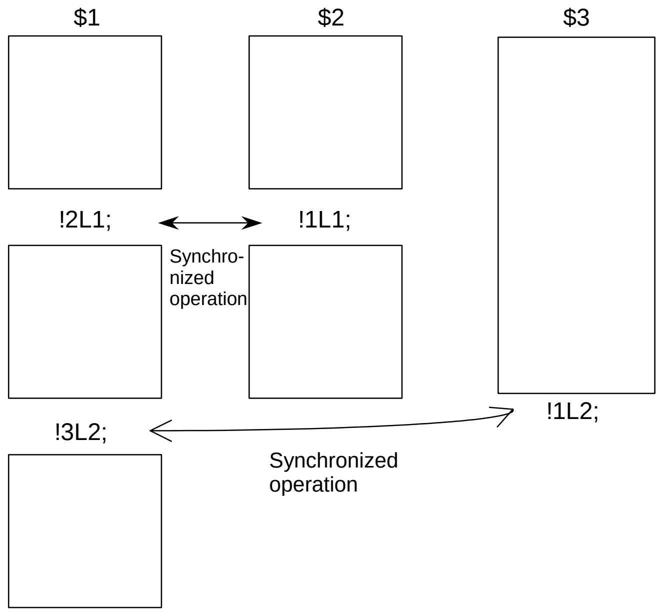


Command format

(1) Command for synchronizing with nth part system

!nLl;

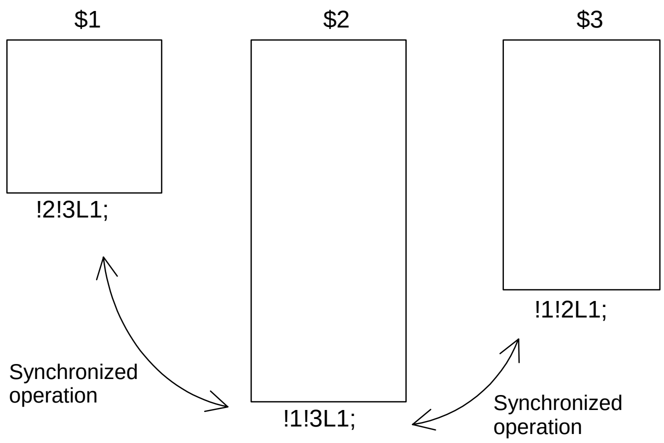
n : Part system number
l : Synchronizing number 01 to 9999



(2) Command for synchronizing among three part systems

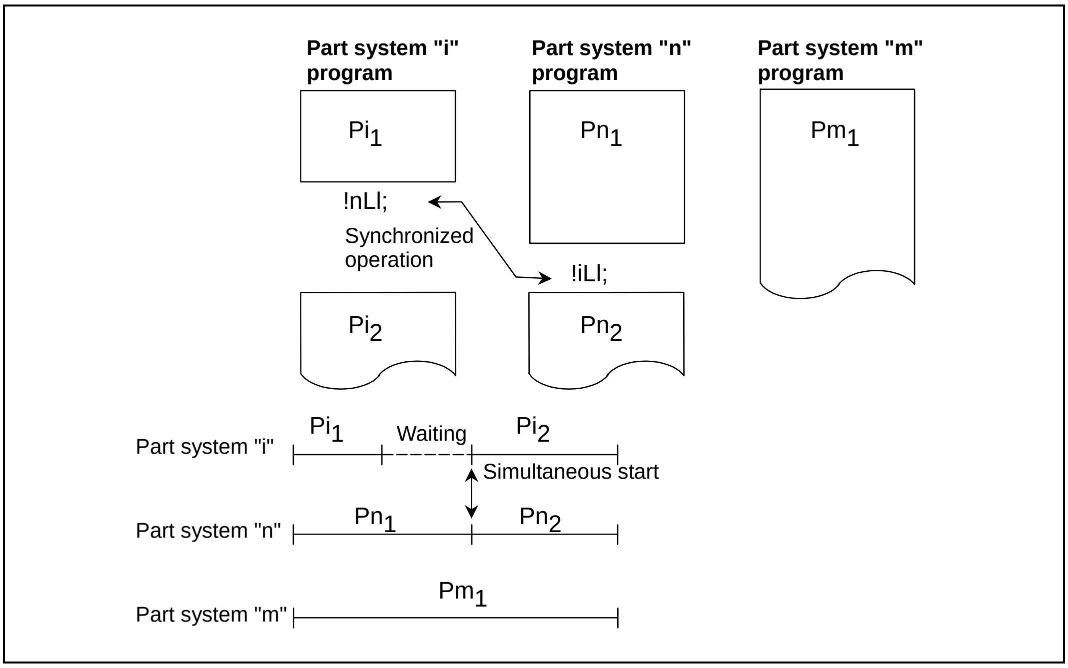
!n!m・・・Ll;

n, m : Part system number n = m
l : Synchronizing number 01 to 9999

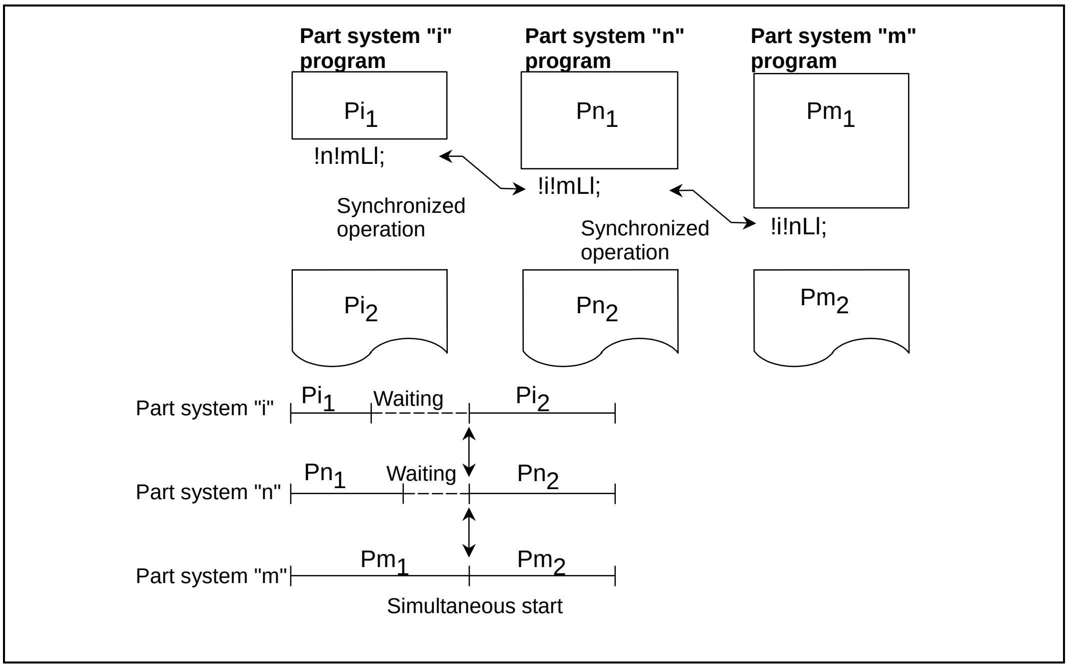


Detailed description

(1) When the "!nLl" code is issued from the part system "i", the operation of that program will wait until the "!iLl" code is issued from the part system "n".
When the "!iLl" code is issued, the programs of both part systems "i" and "n" will start running simultaneously.



(2) Synchronizing among three part systems is as follows. When the "!n!mLl" command is issued from the part system "i", the program of part system "i" operation will wait until the "!i!mLl" command is issued from the part system "n" and the "!i!nLl" command is issued from the part system "m".
When the synchronizing commands are issued, programs of part systems "i", "n" and "m" will start operating simultaneously.



(3) Program error (P35) occurs when an illegal system number has been issued.

(4) The synchronizing command is normally issued in a single block. However, if a movement command or M, S or T command is issued in the same block, whether to synchronize after the movement command or M, S or T command or to execute the movement command or M, S or T command after synchronization will depend on the parameter (#1093 Wmvfin).
#1093 Wmvfin 0: Synchronize before movement command execution.
1: Synchronize after executing movement command.

(5) If there is no movement command in the same block as the synchronizing command, when the next block movement starts, synchronization may not be secured between the part systems. To synchronize the part systems when movement starts after synchronization, issue the movement command in the same block as the synchronizing command.

(6) Synchronizing is done only while the part system to be synchronized is operating automatically. If this is not possible, the synchronizing command will be ignored and operation will advance to the next block.

(7) The L command is the synchronizing identification number. The same numbers are synchronized but when they are omitted, the numbers are handled as L0.

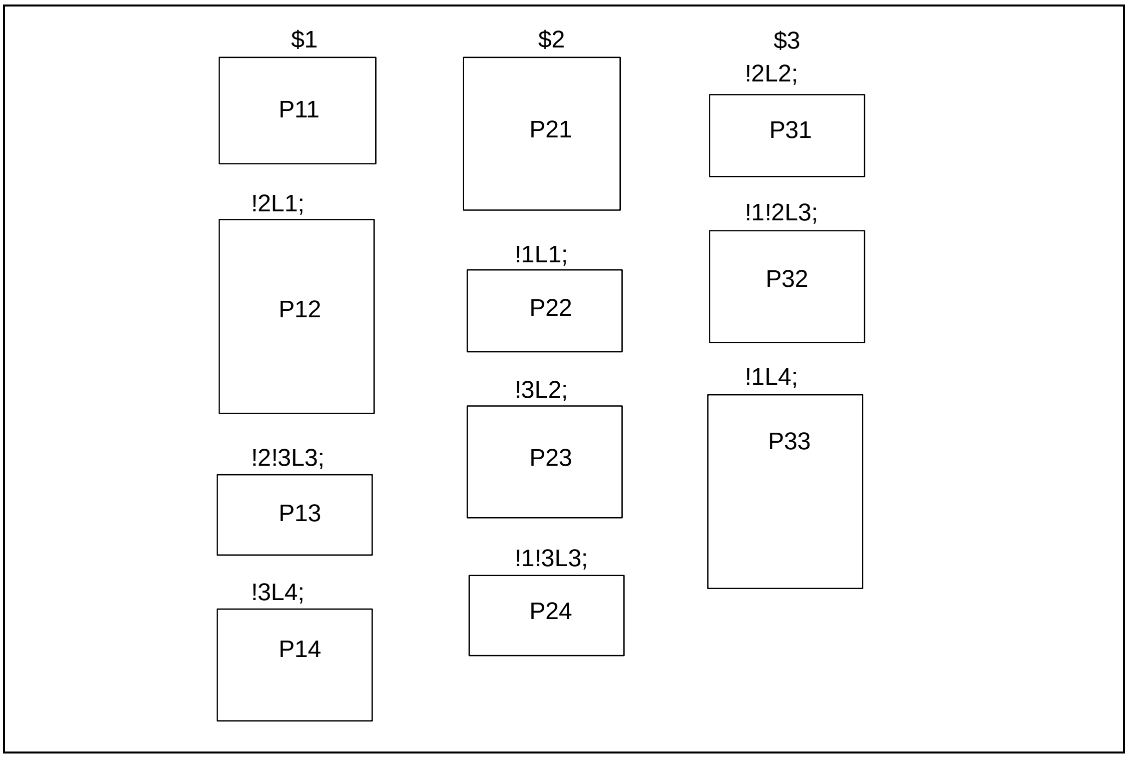
(8) The synchronizing command designates the number of the other part system number to be synchronized, and can also be issued along with its own part system number.

(Example) Part system "i" command: !i!n!mLl;

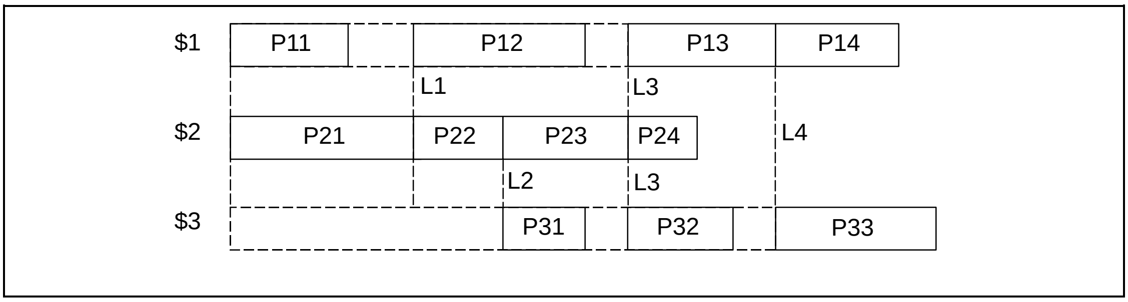
(9) When the part system No. is omitted (when only "!" is issued), part system 1 will be handled as "!2" and part system 2 as "!1". The command using only "!" cannot be used for synchronizing with part system 3 and following.
If the command using only "!" is used for part system 3 or following, the program error (P33) will occur.

(10) "SYN" will appear in the operation status section during synchronization. The synchronizing signal will be output to the PLC I/F. ($1: X63C, $2: X6BC, $3: X73C, $4: X7BC, $5: X83C, $6: X8BC, $7: X93C)

Example of synchronizing between part systems



The above programs are executed as follows:



13.14 Start Point Designation Synchronizing (Type 1); G115

Function and purpose

The part system can wait for the other part system to reach the start point before starting itself.

The synchronization point can be set in the middle of a block.

Command format

!nL1 G115 X_ Z_ C_ ;

!nL1	Synchronizing command
G115	G command
X_ Z_ C_	Start point (Command axis and workpiece coordinate values for checking synchronization of other part system.)

Detailed description

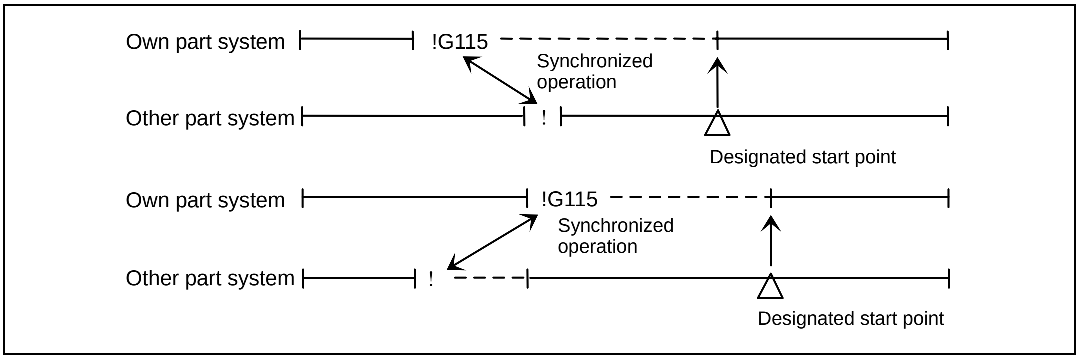
(1) Designate the start point using the workpiece coordinates of the other part system.
(2) The start point check is executed only for the axis designated by G115.

(Example) !L2 G115 X100.;

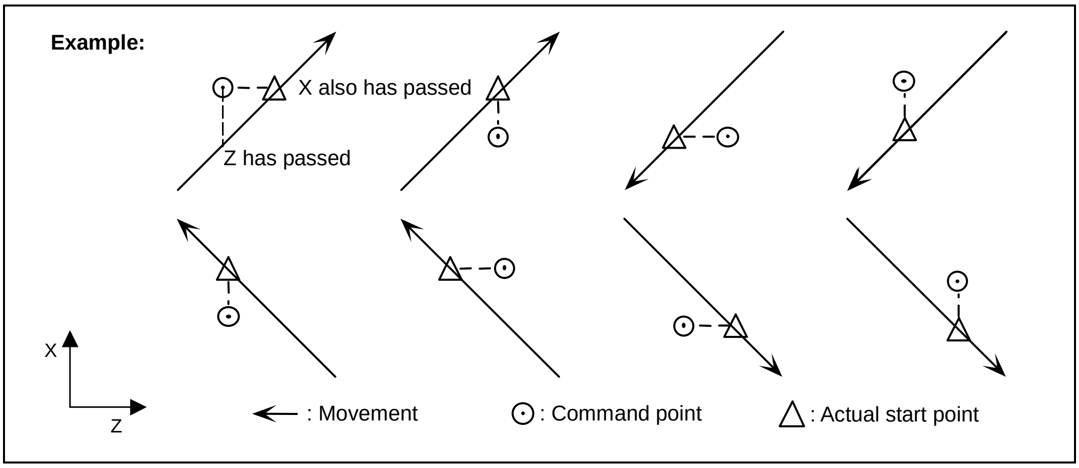
Once the other part system reaches X100., the own part system will start. The other axes are not checked.

(3) The other part system starts first when synchronizing is executed.

(4) The own part system waits for the other part system to move and reach the designated start point, and then starts.

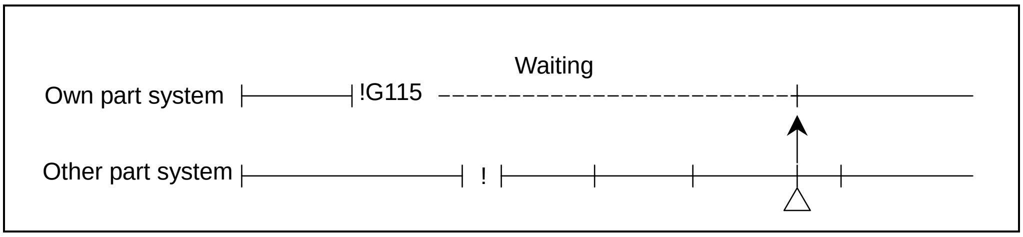


(5) When the start point designated by G115 is not on the next block movement path of the other part system, the own system starts once the other part system has reached all of the start point axis coordinates.

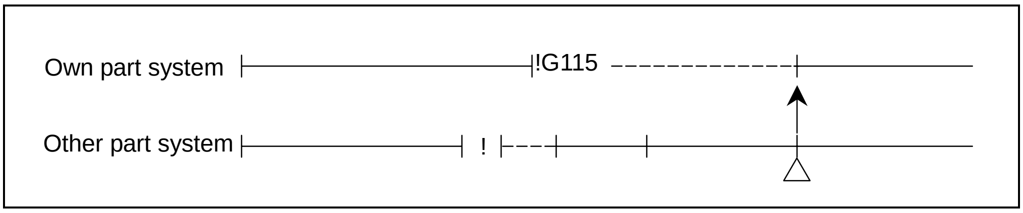


(6) The following operation is executed by parameters (base specification parameter #1229 set01/bit5) when the start point cannot be determined by the next block movement of the other system.

(a) When the parameter is ON
Operation waits until the start point is reached by the movement in the next and subsequent blocks.



(b) When the parameter is OFF
The own part system starts upon completion of the next block movement.



(7) The waiting status continues when the G115 command has been duplicated between part systems.

(8) Designate the start point using the workpiece coordinates of the other part system.

(9) Program error "P33" occurs when the G115 command is issued for 3 part systems.

(10) The single block stop function does not apply for the G115 block.

(11) When the G115 command is issued continuously in 2 or more blocks, the block in which it was issued last will be valid.

(12) A program error (P32) will occur if an address other than an axis is designated in G115 command block.

13.15 Start Point Designation Synchronizing (Type 2); G116

Function and purpose

Starting of the other part system can be delayed until the own part system reaches the designated start point.
The synchronization point can be set in the middle of a block.

Command format

!nL1 G116 X_ Z_ C_ ;

!nL1	Synchronizing command
G116	G command
X_ Z_ C_	Start point (Command axis and workpiece coordinate values for checking synchronization of own part system.)

Detailed description

(1) Designate the start point using the workpiece coordinates of the own part system.

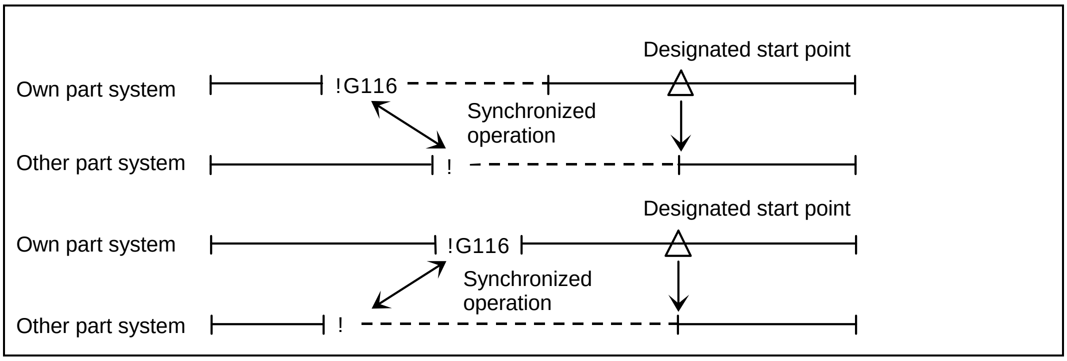
(2) The start point check is executed only for the axis designated by G116.

(Example) !L1 G116 X100.;

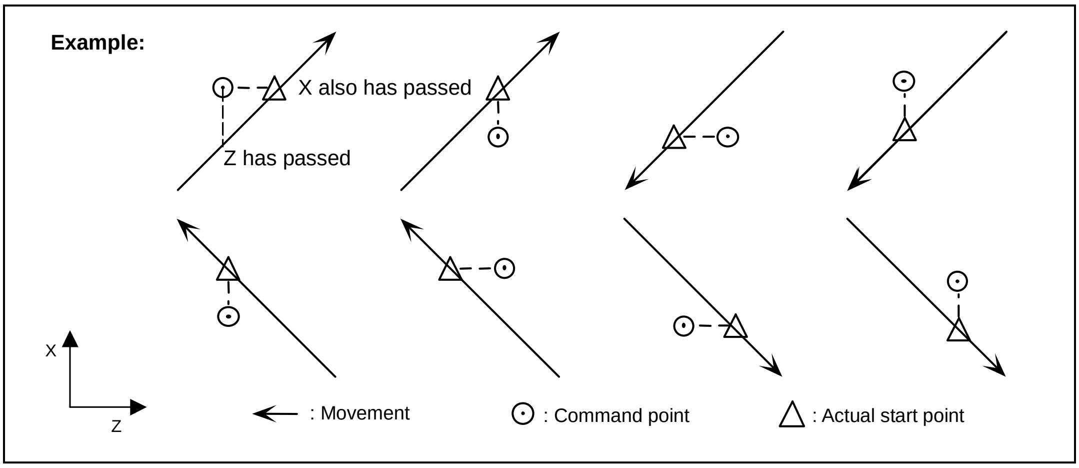
Once the own part system reaches X100., the other part system will start. The other axes are not checked.

(3) The own part system starts first when synchronizing is performed.

(4) The other part system waits for the own part system to move and reach the designated start point, and then starts.

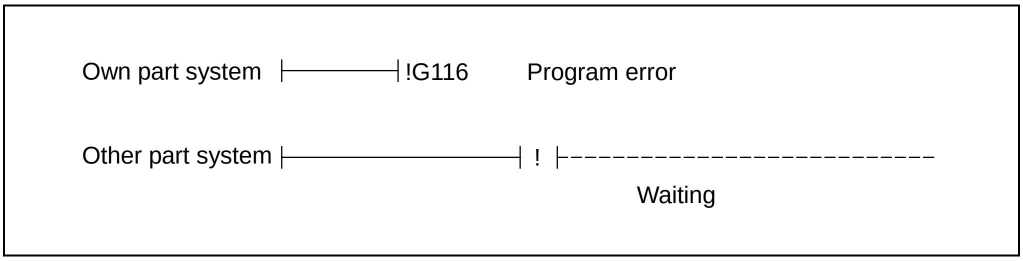


(5) When the start point designated by G116 is not on the next block movement path of the own system, the other system starts once the own system has reached all of the start point axis coordinates.

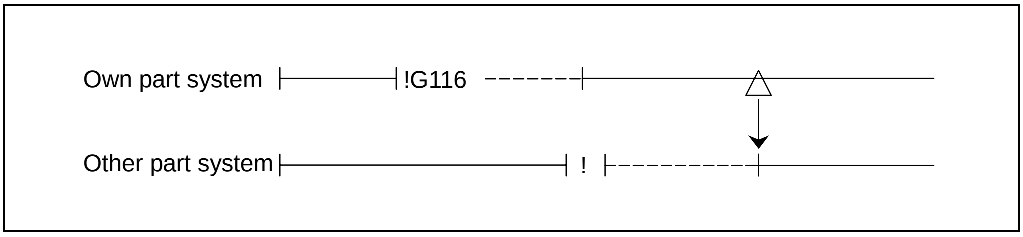


(6) The next operation is executed by parameters (base specification parameter #1229 set01/bit5) when the start point cannot be determined by the next block movement of the own part system.

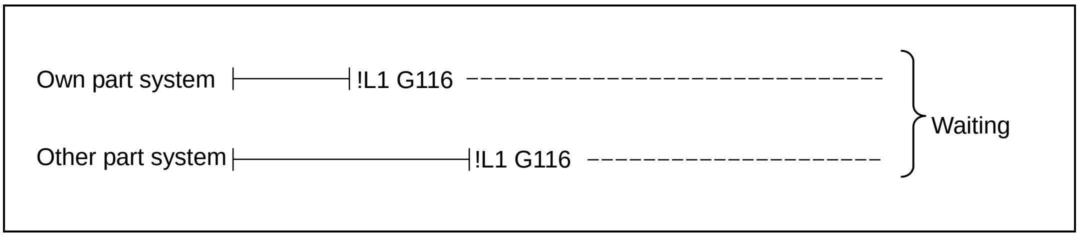
(a) When the parameter is ON
Program error "P33" occurs before the own part system moves.



(b) When the parameter is OFF
The other part system starts upon completion of the next block movement.

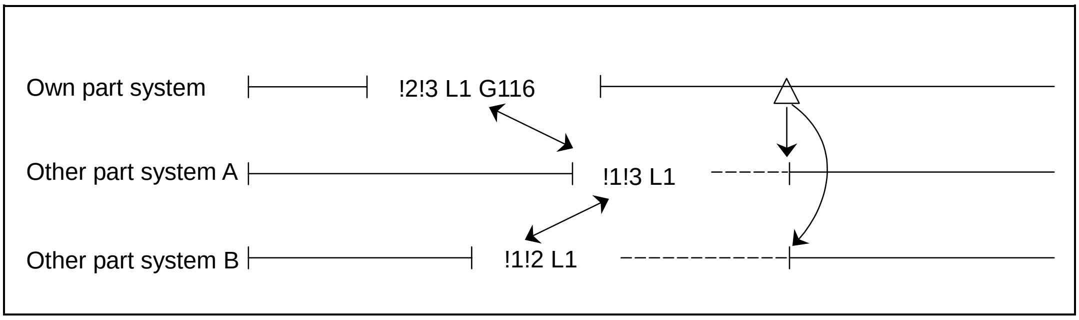


(7) If the G116 command overlaps between part systems, the waiting state will continue.



(8) Designate the start point using the workpiece coordinates of each part system.

(9) The two other part systems start when the G116 command is issued for 3 part systems.



(10) The single block stop function does not apply for the G116 block.

(11) When the G116 command is issued continuously in 2 or more blocks, the block in which it was issued last will be valid.

(12) A program error (P32) will occur if an address other than an axis is designated in G116 command block.

13.16 Miscellaneous function output during axis movement; G117

Function and purpose

This function controls the timing of the miscellaneous function to be output. The miscellaneous function is output when the position designated in axis movement is reached.

Command format

G117 X_ Z_ M_ S_ T_ (2nd M)_ ;	
X Z	Start point of operation
M_ S_ T_ (2nd M)_	Miscellaneous function

Detailed description

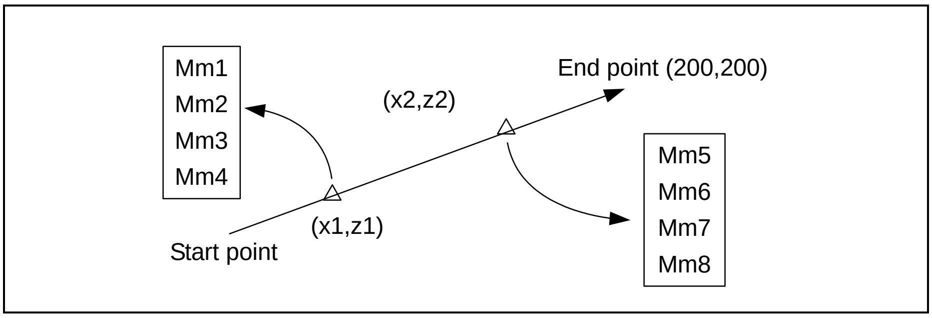
(1) This command is issued independently immediately before the block with the movement command that activates the miscellaneous function.

(2) Single block stop does not apply to this command.

(3) The maximum number of groups to which the miscellaneous functions in the G117 block can be issued is as follows:
M commands : 4 sets
S commands : 2 sets
T commands : 1 set
2nd miscellaneous function : 1 set

(4) This command can be issued in up to two consecutive blocks.
When issued in three or more consecutive blocks, the last two blocks will be valid.

(Example) G117 Xx1 Zz1 Mm1 Mm2 Mm3 Mm4;
G117 Xx2 Zz2 Mm5 Mm6 Mm7 Mm8;
G01 X200 Z200;



(5) When the operating start point commanded by G117 is not on the movement path, the miscellaneous function will be output once the movement has reached all the coordinate values of the operating start point. In addition, only the commanded axis is checked.

(Example) G117 X100. M××; M×× is output when X100. is reached.

(Note) The other axes are not subject to the check.

(6) The completion of the miscellaneous function in the previous group is checked at the operating start point, and the miscellaneous function of the next group is output. Thus, normal PLC interfacing is possible.

(7) A miscellaneous function issued in the same block as the block with the movement command is output before the movement and starts the movement. During movement, operation will not stop at the operating start point. However, at the end point of the block, the completion of all the miscellaneous functions is checked first, and then the execution of the next block is started.

(8) G117 should be issued in the sequence of operating start points. Program error (P33) occurs if the sequence of the operating start point is the reverse of the movements.
When operating start points coincide, the miscellaneous functions are output in the sequence in which they were issued.

(9) When an operating start point cannot be determined by the next block movement, the next operation is performed by the parameter.

Basic specification parameter "#1229 set01/bit5"	Operation
ON	Program error P33 occurs before movement
OFF	The functions are output when the next block movement is completed.

(10) The following tables show the combinations of (8) and (9).

G17 First block / Second block	During intermediate point movement	Not during intermediate point movement
During intermediate point movement	Refer to (8).	Program error (P33) due to (8).
Not during intermediate point movement	Refer to (9) for second block.	Refer to (9). With output, the sequence of first block, second block is followed regardless of the sequence of the designated points.

Precautions

(1) Command G117 in order of the operation start points. If the operation start point order is the opposite of the movement, a program error (P33) will occur.

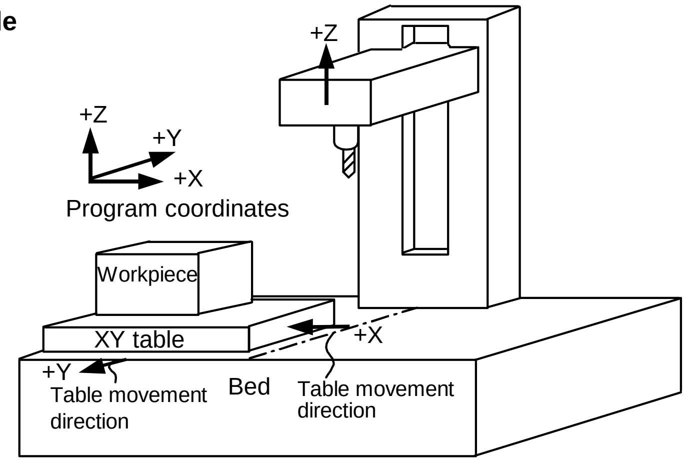
14. Coordinates System Setting Functions

14.1 Coordinate words and control axes

Function and purpose

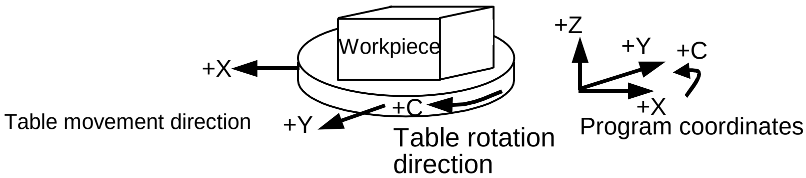
There are three controlled axis for the basic specifications, but when an additional axis is added, up to 14 axes can be controlled. Pre-determined alphabetic coordinate words that correspond to the axes are used to designate each machining direction.

For XY table



For XY table

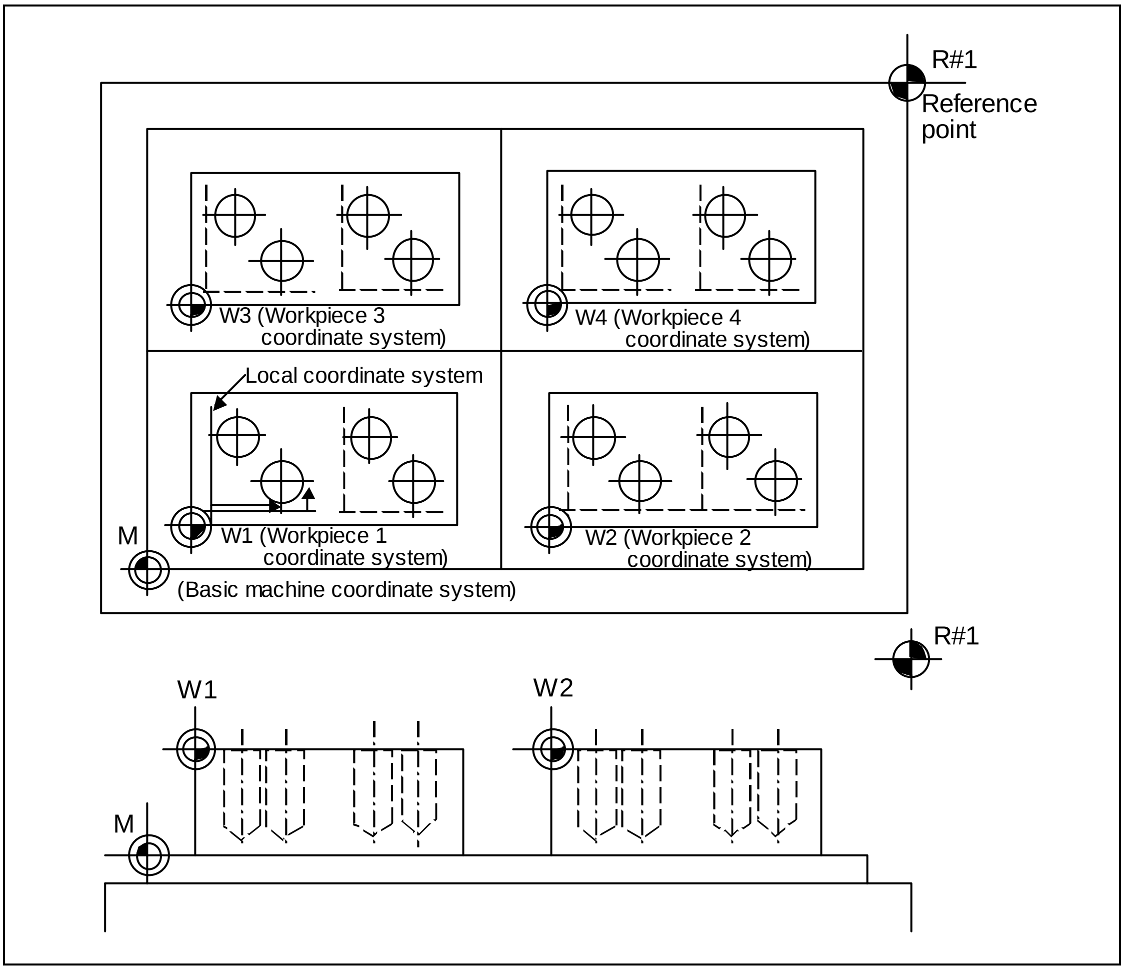
For XY and rotary table



14.2 Basic machine, work and local coordinate systems

Function and purpose

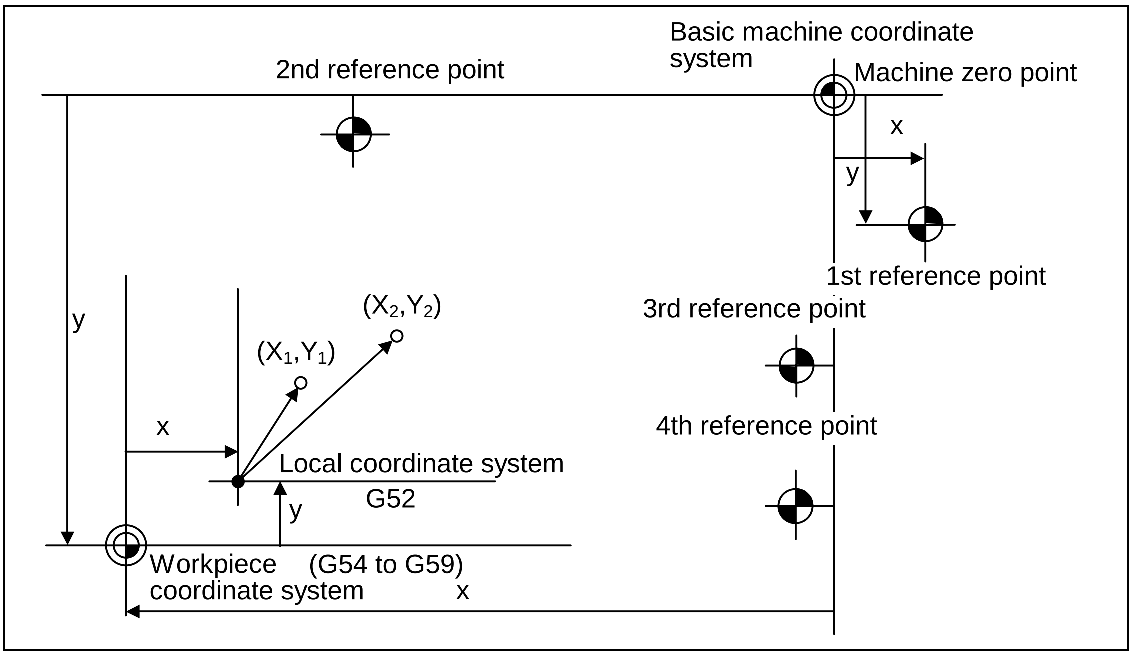
The basic machine coordinate system is fixed in the machine and it denotes that position which is determined inherently by the machine.
The work coordinate systems are used for programming and in these systems the reference point on the workpiece is set as the coordinate zero point.
the local coordinate 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 selection ; G53

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.

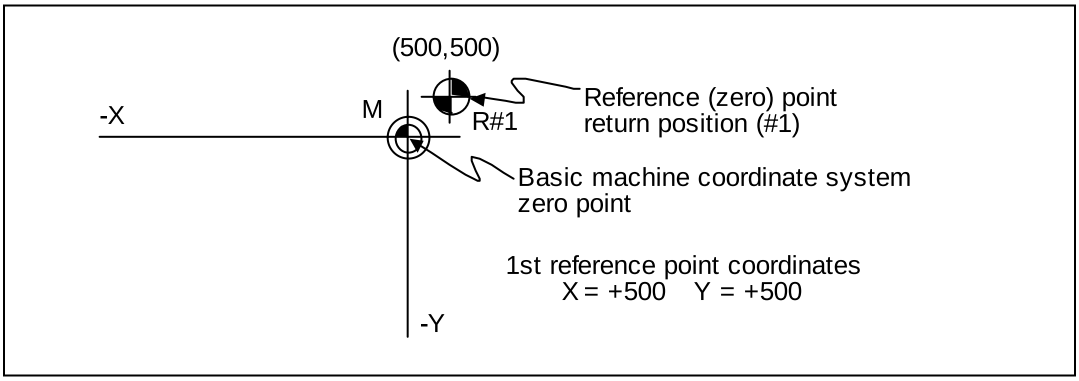
Command format

Basic machine coordinate system selection

(G90) G53 Xx Yy Zz αα ;	
αα	: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

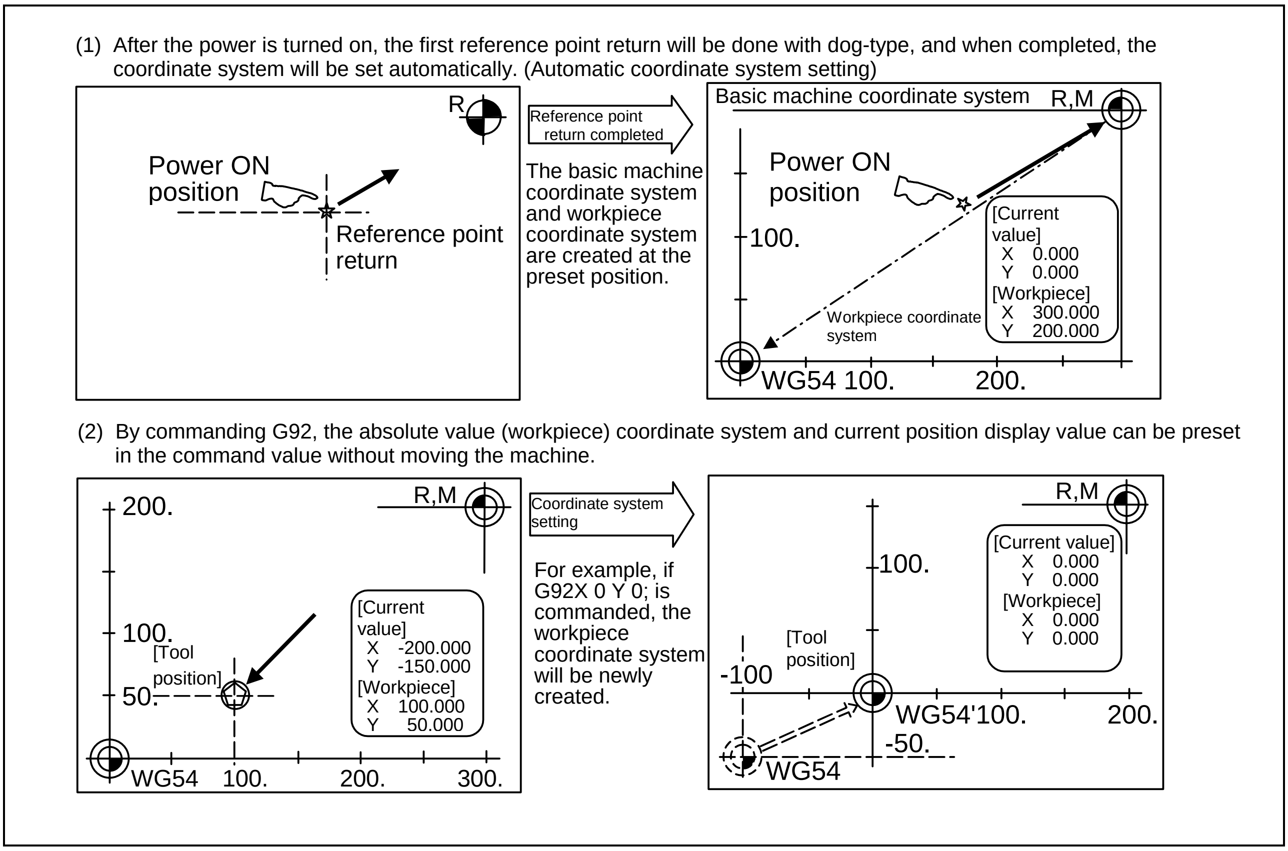
Function and purpose

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

G92 Xx_1 Yy_1 Zz_1 $\alpha\alpha_1$;	
$\alpha\alpha$	:Additional axis

Detailed description



(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.

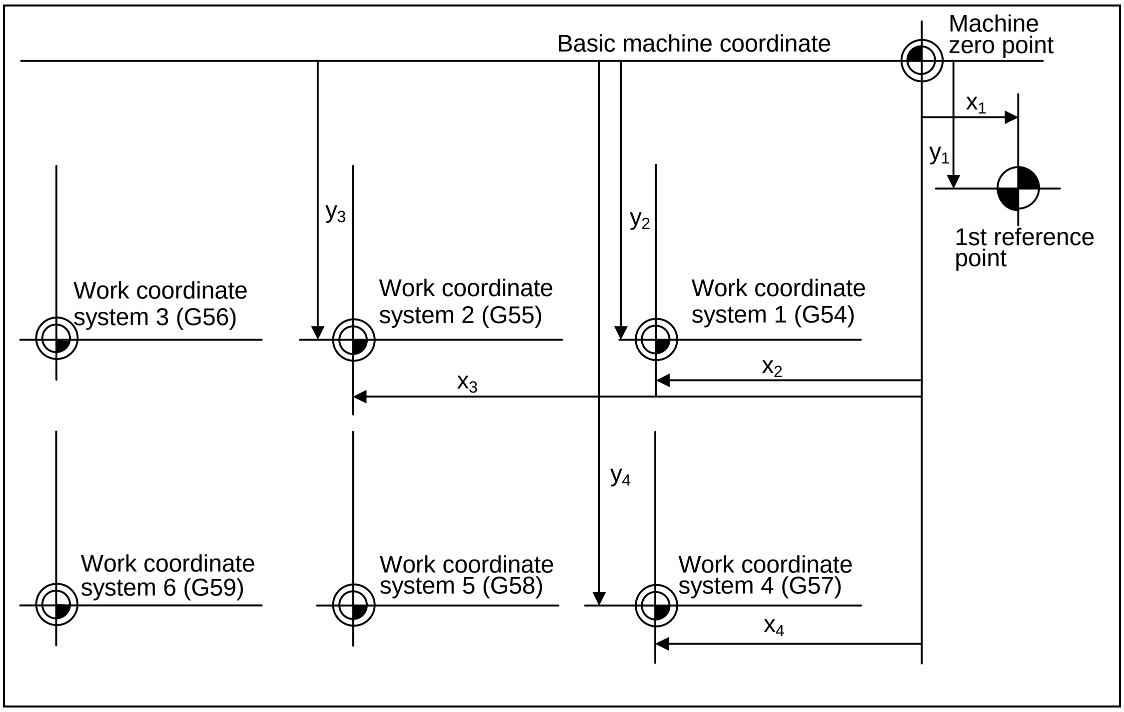
(1) Execute reference point return while the coordinate system is deviated.

(2) After that, command G92G53X0Y0Z0;. With this command, the workpiece coordinate value and current value 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, the following functions are canceled: workpiece coordinate system shift based on G92, local coordinate system setting based on G52, workpiece coordinate system shift based on origin setting and workpiece coordinate system shift based on manual interrupt.

(4) When a parameter has been used to select the dog-type of first manual reference point return or automatic reference point return after the power has been turned ON, the dog-type reference point return will be executed for the 2nd and subsequent manual reference point returns or automatic reference point returns.

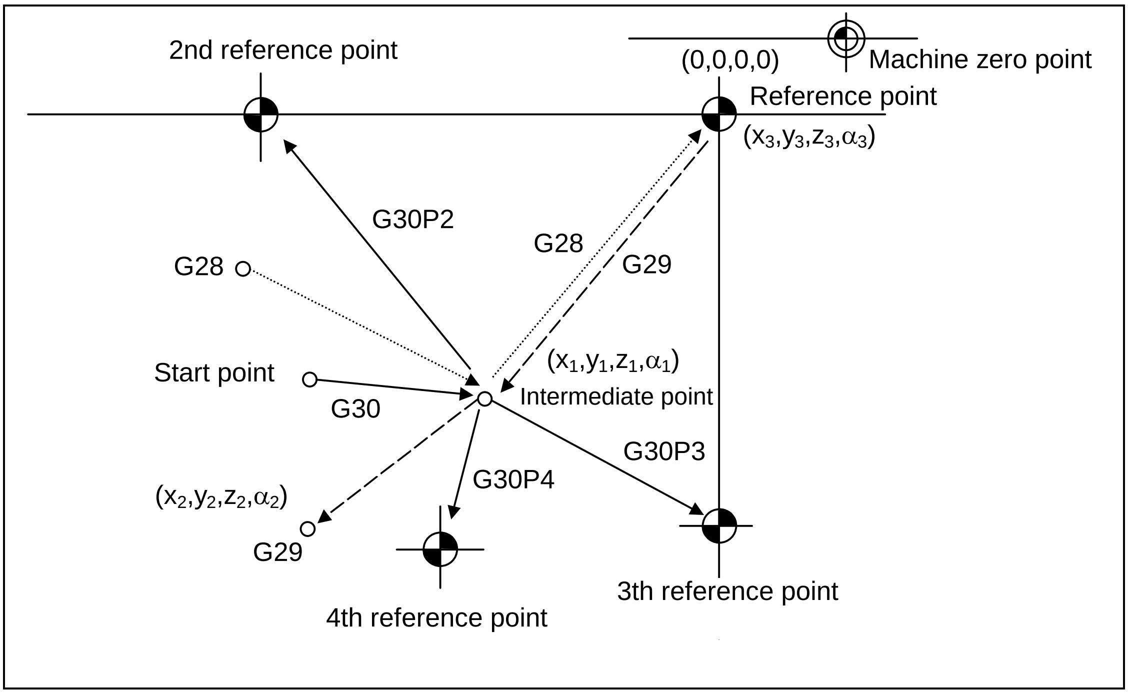
⚠CAUTION

⚠ **If the workpiece coordinate offset amount is changed during automatic operation (including single block operation), the changes will be valid from the next block of the command several blocks later.**

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 $\alpha\alpha_1$;
G00 Xx_3 Yy_3 Zz_3 $\alpha\alpha_3$;
In this case, x_3, y_3, z_3 and α_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 $\alpha\alpha_1$;
G00 Xx_2 Yy_2 Zz_2 $\alpha\alpha_2$;
Rapid traverse (non-interpolation type) applies independently for each axis for the positioning from the reference point to the intermediate point.
In this case, x_1, y_1, z_1 and α_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, α_1) of the positioning point are assigned by the position command modal. (G90, G91).

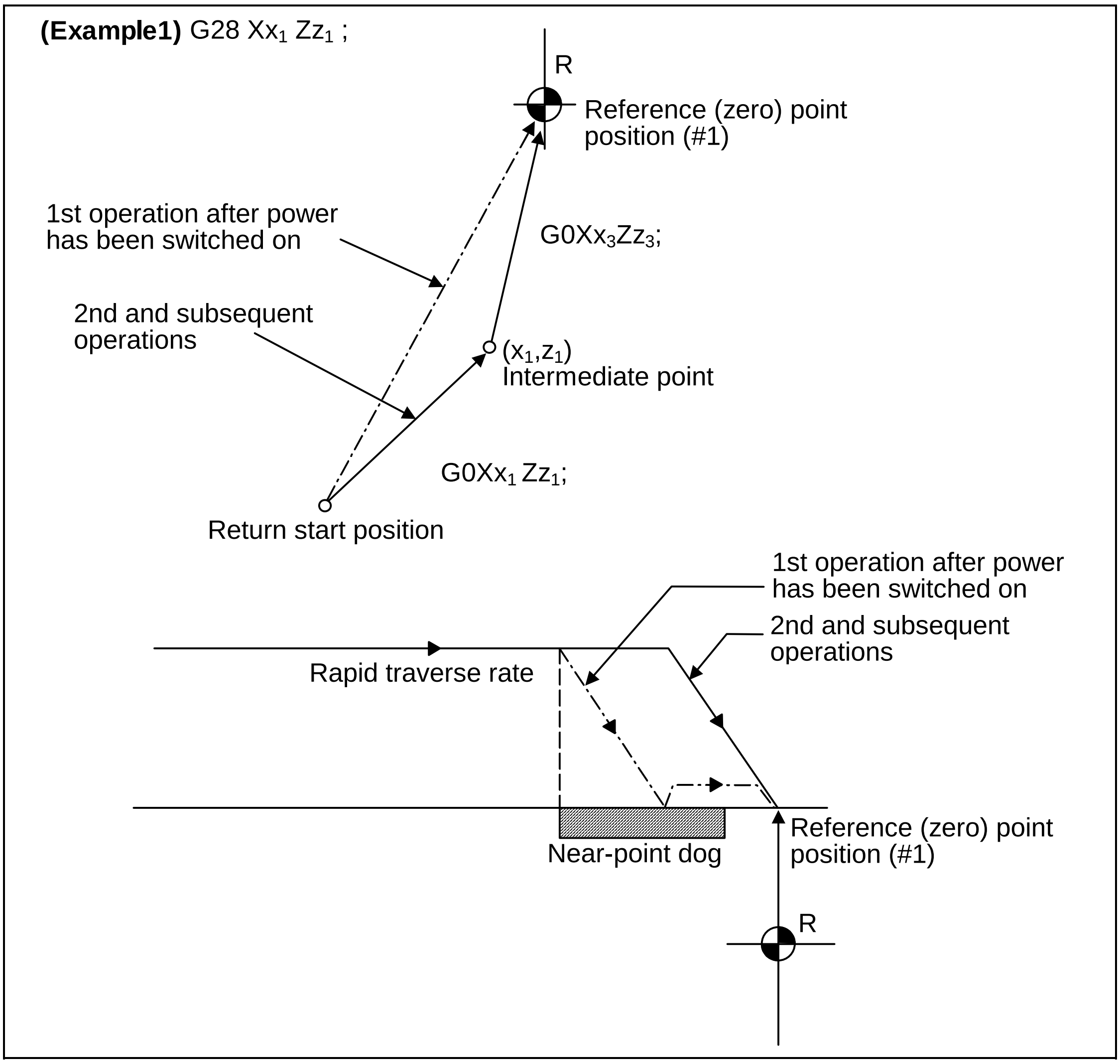
(8) G29 is valid for either G28 or G30 but the commanded axes are positioned after a return has been made to the latest intermediate point.

(9) The tool offset will be canceled during reference point return unless it is already canceled, and the offset amount will be cleared.

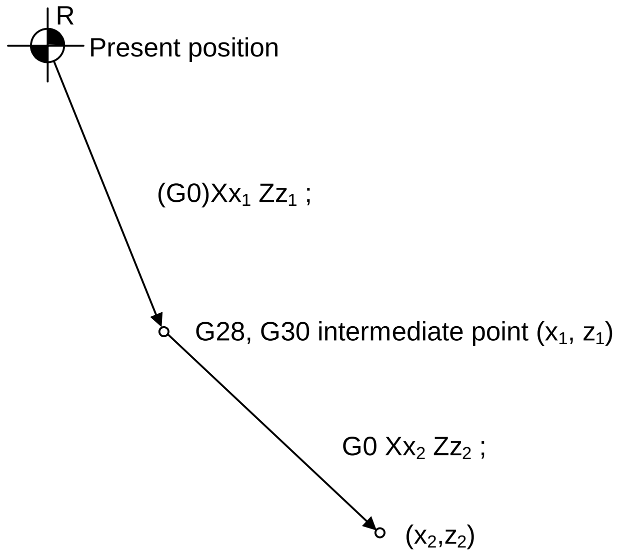
(10) Control from the intermediate point to the reference (zero) point is ignored for reference (zero) point return in the machine lock status. The next block is executed when the commanded axis survives as far as the intermediate point.

(11) Mirror image is valid from the start point to the intermediate point during reference (zero) point return in the mirror image mode 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 will move to the reference (zero) point.

Example of program

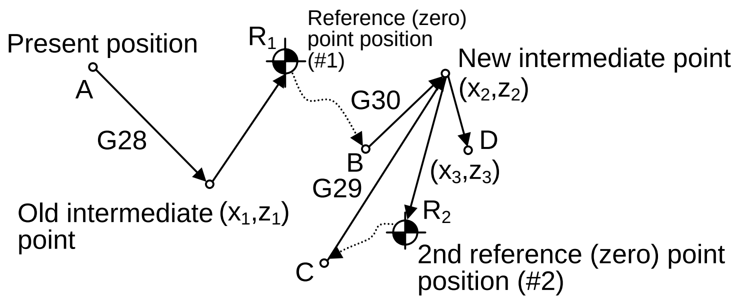


(Example2) G29 Xx_2 Zz_2 ;



(Example 3)

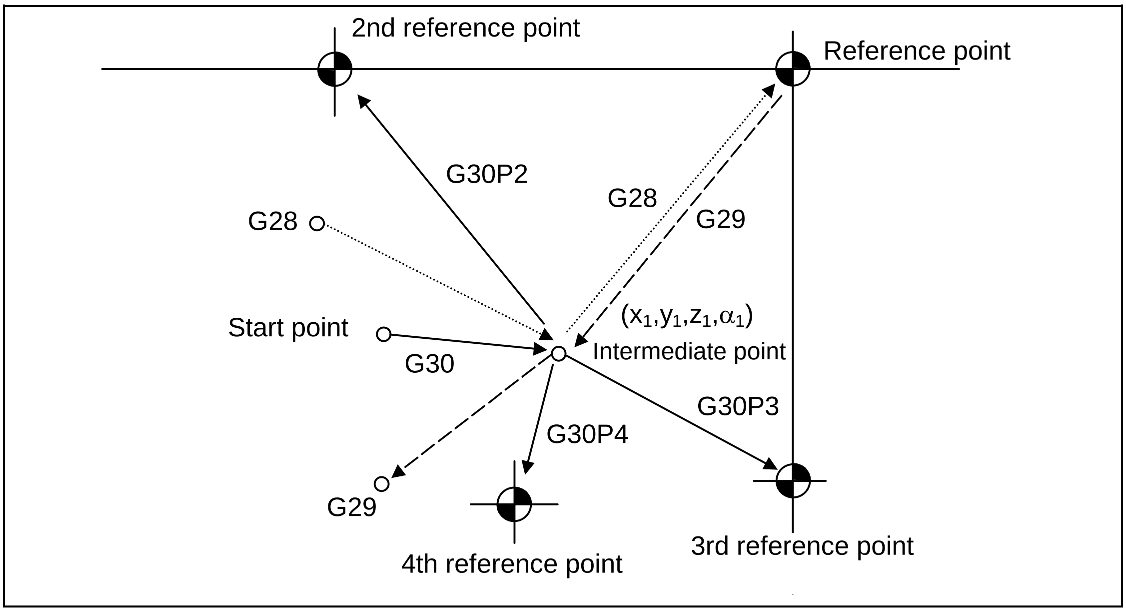
G28 Xx_1 Zz_1 ; (From point A to reference (zero) point)
G30 Xx_2 Zz_2 ; (From point B to 2nd reference (zero) point)
G29 Xx_3 Zz_3 ; (From point C to point D)



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).

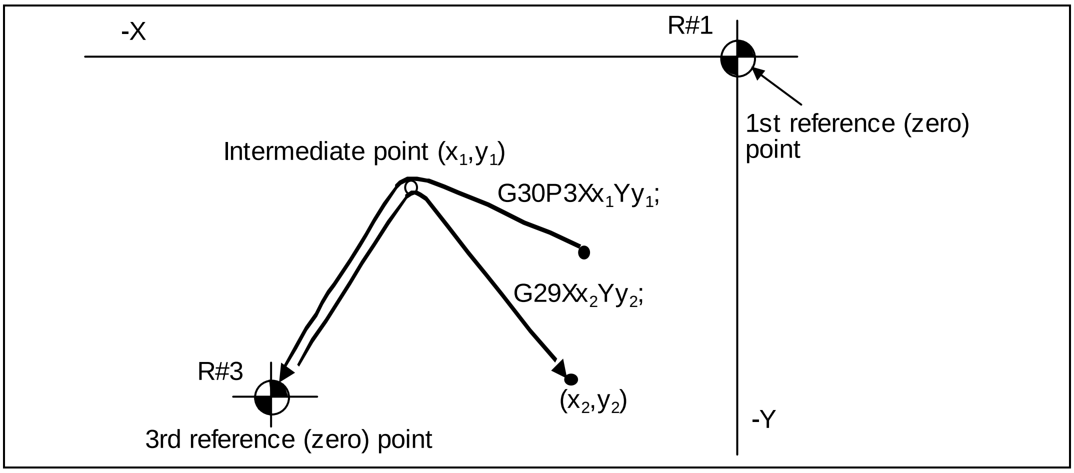


Command format

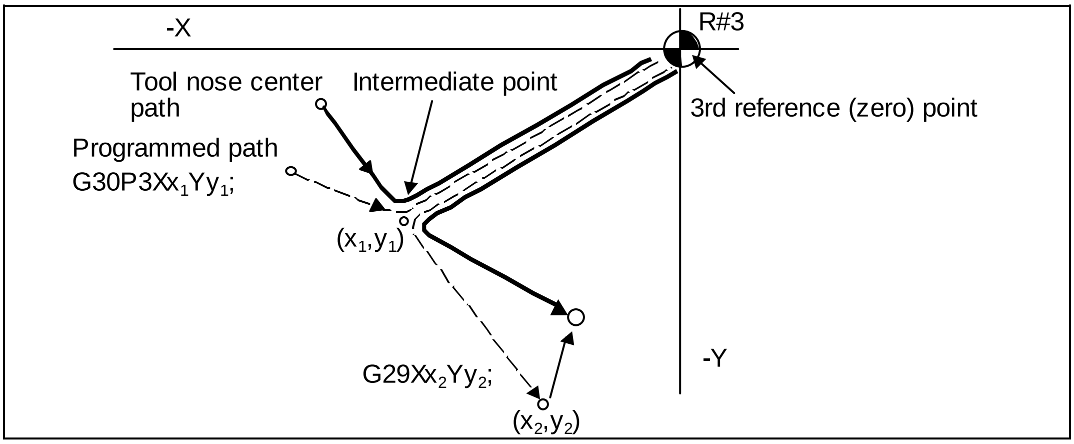
G30 P2 (P3, P4) Xx_1 Yy_1 Zz_1 aa_1;	
$\alpha\alpha_1$	:Additional axis

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.



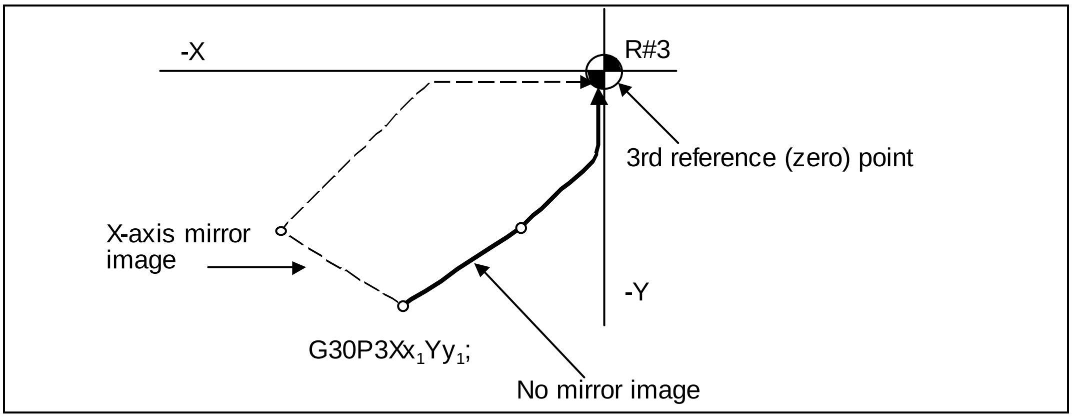
(5) With reference (zero) point return on a plane during compensation, the tool moves without tool diameter compensation (zero compensation) from the intermediate point. with a subsequent G29 command, the tool moves with tool diameter compensation until the G29 command from the intermediate point.



(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.



14.9 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.

14.10 Workpiece coordinate system setting and offset ; G54 to G59 (G54.1)

Function and purpose

(1) The workpiece 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 to 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 imaginary 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 tool diameter, tool length and tool position offset.) (G54, G92)

Command format

(1) Workpiece coordinate system selection (G54 to G59)

(G90) G54 Xx_1 Yy_1 Zz_1 $\alpha\alpha_1$;
$\alpha\alpha_1$:Additional axis

(2) Workpiece coordinate system setting (G54 to G59)

(G54) G92 Xx_1 Yy_1 Zz_1 $\alpha\alpha_1$;
$\alpha\alpha_1$:Additional axis

(3) Workpiece coordinate system selection (expanded : P1 to P48)

G54.1 Pn ;

(4) Workpiece coordinate system setting (expanded : P1 to P48)

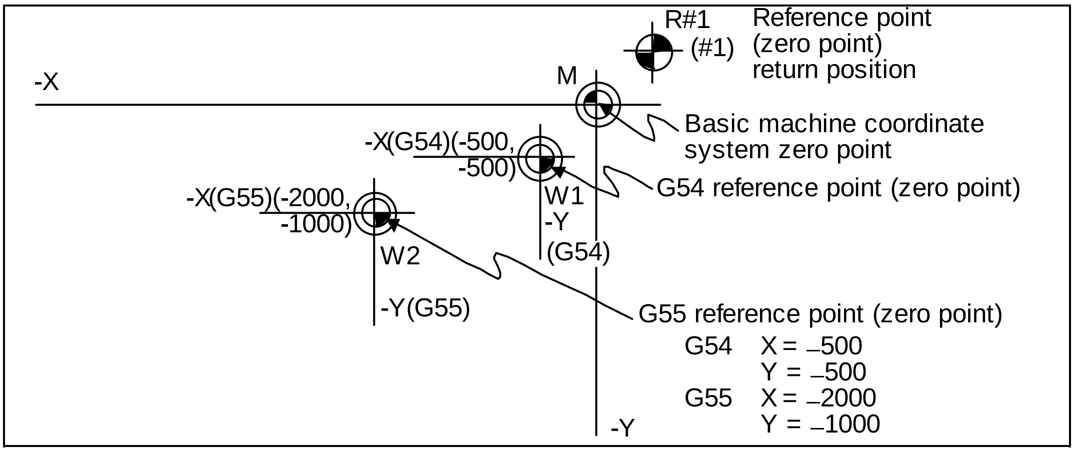
G54.1 Pn ;
G92 Xx Yy Zz ;

(5) Workpiece coordinate system offset amount setting (expanded : P1 to P48)

G10 L20 Pn Xx Yy Zz ;

Detailed description

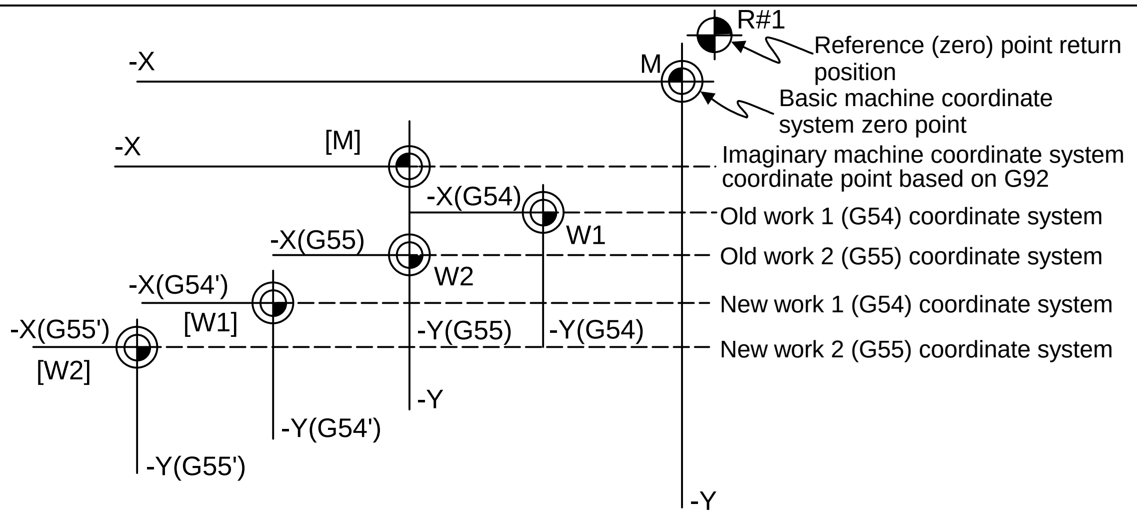
(1) With any of the G54 through G59 commands, the tool diameter offset amounts for the commanded axes will not be canceled even if workpiece coordinate system selection is commanded.
(2) The G54 workpiece coordinate system is selected when the power is switched on.
(3) Commands G54 through G59 are modal commands (group 12).
(4) The coordinate system will move with G92 in a workpiece coordinate system.
(5) The offset setting in a workpiece coordinate system denotes the distance from the zero point of the basic machine coordinate system.



(6) The offset settings of workpiece 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 Yy 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 Yy 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 workpiece coordinate system 1 is set by issuing the G92 command in the G54 (workpiece coordinate system 1) mode. At the same time, the other workpiece coordinate systems 2 through 6 (G55 to G59) will move in parallel and new workpiece 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 workpiece 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 workpiece coordinate system will be set at a position which deviates from that imaginary basic machine coordinate system by an amount equivalent to the workpiece 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 workpiece 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 (P39) will occur when the G54.1 command is executed.

(14) When number of workpiece offset sets additional specifications is not added, the program error (P172) will occur when the G10 L20 command is executed.
(15) 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.
(16)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.
(17) 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 6th axis
P 1	#7001 to #7006
P 2	#7021 to #7026
P 3	#7041 to #7046
P 4	#7061 to #7066
P 5	#7081 to #7086
P 6	#7101 to #7106
P 7	#7121 to #7126
P 8	#7141 to #7146
P 9	#7161 to #7166
P10	#7181 to #7186
P11	#7201 to #7206
P12	#7221 to #7226
P13	#7241 to #7246
P14	#7261 to #7266
P15	#7281 to #7286
P16	#7301 to #7306
P17	#7321 to #7326
P18	#7341 to #7346
P19	#7361 to #7366
P20	#7381 to #7386
P21	#7401 to #7406
P22	#7421 to #7426
P23	#7441 to #7446
P24	#7461 to #7466

	1st axis to 6th axis
P25	#7481 to #7486
P26	#7501 to #7506
P27	#7521 to #7526
P28	#7541 to #7546
P29	#7561 to #7566
P30	#7581 to #7586
P31	#7601 to #7606
P32	#7621 to #7626
P33	#7641 to #7646
P34	#7661 to #7666
P35	#7681 to #7686
P36	#7701 to #7706
P37	#7721 to #7726
P38	#7741 to #7746
P39	#7761 to #7766
P40	#7781 to #7786
P41	#7801 to #7806
P42	#7821 to #7826
P43	#7841 to #7846
P44	#7861 to #7866
P45	#7881 to #7886
P46	#7901 to #7906
P47	#7921 to #7926
P48	#7941 to #7946

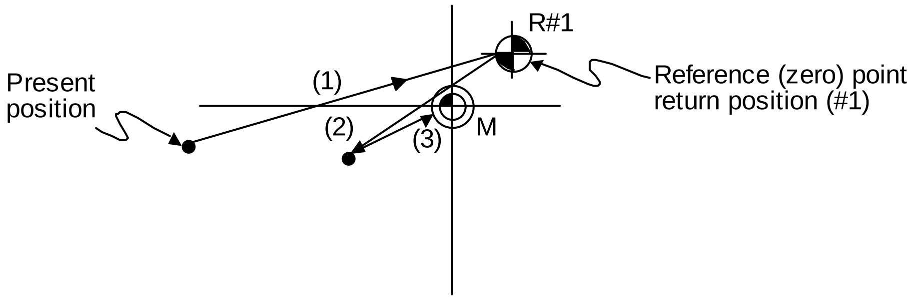
⚠ CAUTION

⚠ 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)

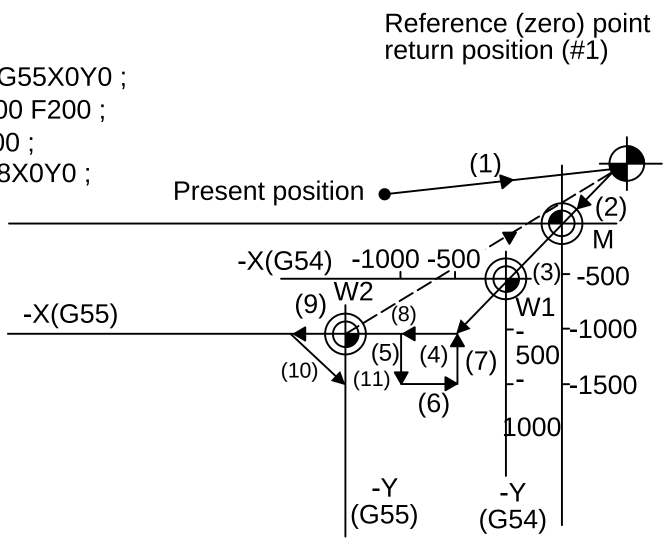
(1) G28 X0Y0 ;
(2) G53 X–1000 Y–500 ;
(3) G53 X0Y0 ;



When the first reference point coordinate is zero, the basic machine coordinate system zero point and reference (zero) point return position (#1) will coincide.

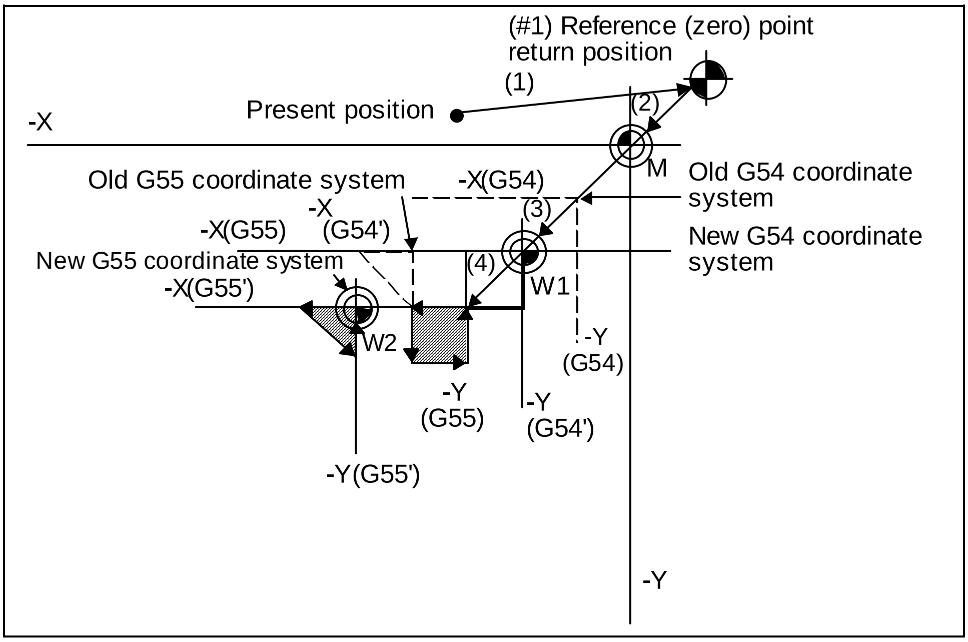
(Example 2)

(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 workpiece 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 workpiece coordinate system deviates
(4) G92 X0 Y0 ;	New workpiece coordinate system is set.
(5) M98 P1111 ;	



(Note) The workpiece 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

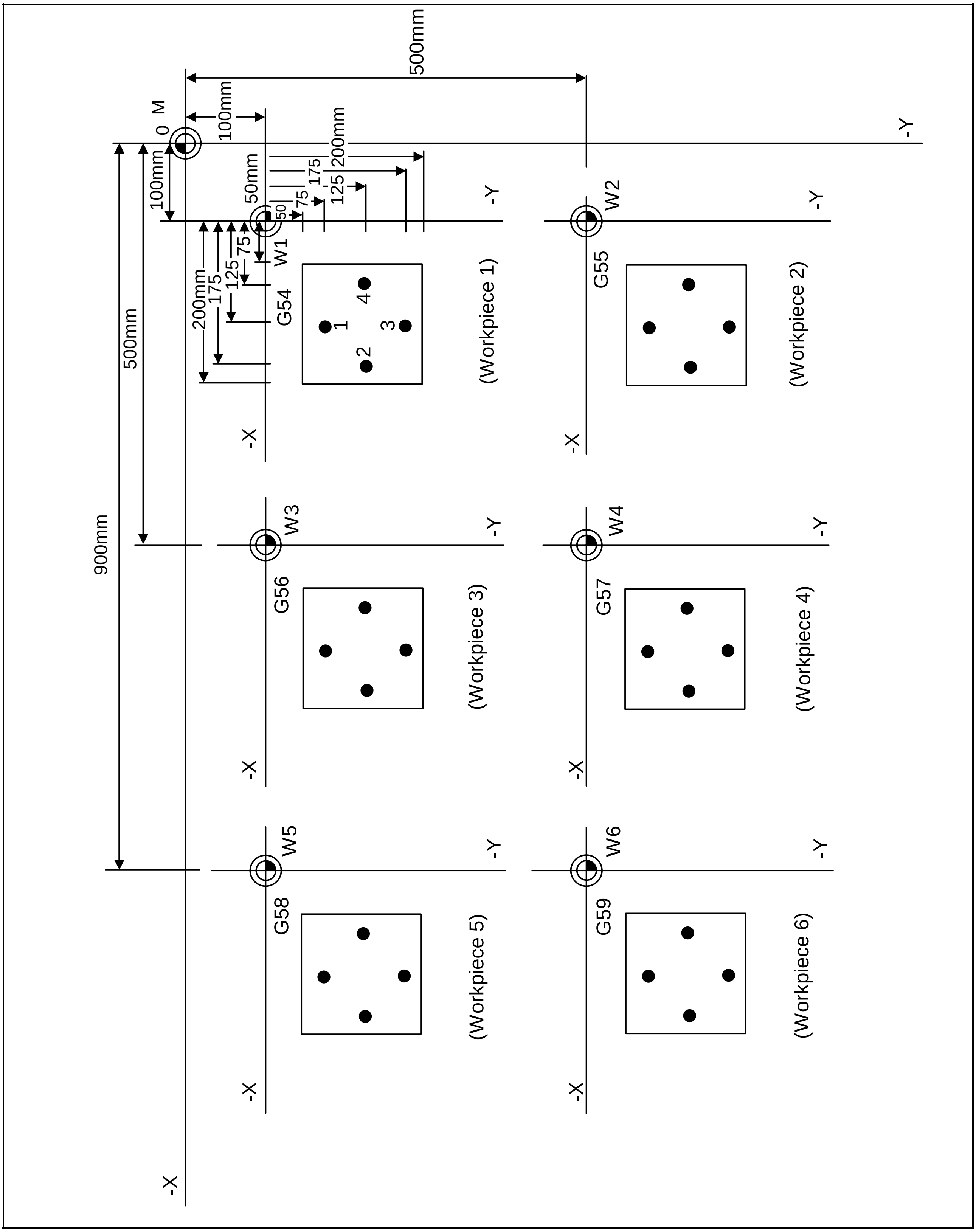
```
Workpiece1  X = –100.000  Y = –100.000 ......................................  G54
         2  X = –100.000  Y = –500.000 ......................................  G55
         3  X = –500.000  Y = –100.000 ......................................  G56
         4  X = –500.000  Y = –500.000 ......................................  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 G01 X-200. F50;
       Y-200.  ;
       X- 50.  ;                           Face 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 ;
%
```


500mm
100mm
0 M
-Y
200mm
175
125
75
50
50mm
100mm
W1
W2
G54
G55
1
2
3
4
(Workpiece 1)
(Workpiece 2)
200mm
175
125
75
500mm
900mm
-X
W3
W4
G56
G57
(Workpiece 3)
(Workpiece 4)
W5
W6
G58
G59
(Workpiece 5)
(Workpiece 6)

14.11 Local coordinate system setting; G52

Function and purpose

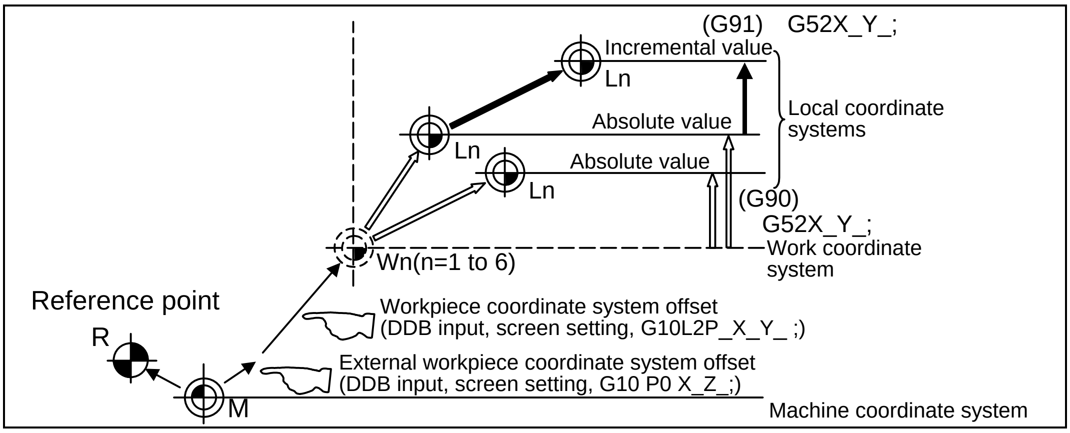
The local coordinate systems can be set independently on the G54 through G59 workpiece coordinate systems using the G52 command so that the commanded position serves as the programmed zero point.
The G52 command can also be used instead of the G92 command to change the deviation between the zero point in the machining program and the machining workpiece zero point.

Command format

G54 (54 to G59) G52Xx_1 Yy_1 Zz_1 $\alpha\alpha_1$;	
$\alpha\alpha_1$	:Additional axis

Detailed description

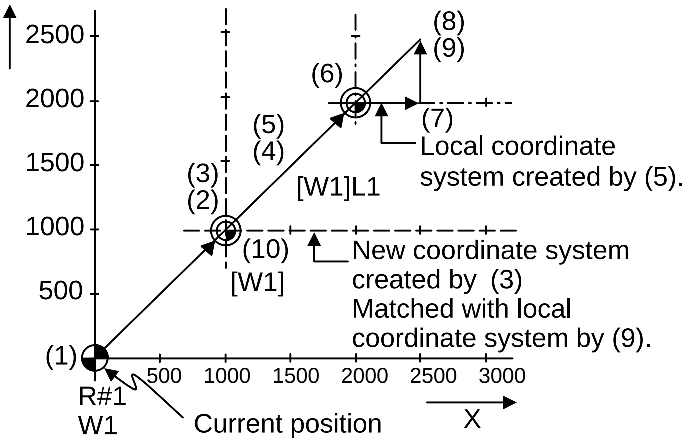
(1) The G52 command is valid until a new G52 command is issued, and the tool does not move. This command comes in handy for employing another coordinate system without changing the zero point positions of the workpiece coordinate systems (G54 to G59).
(2) The local coordinate system offset will be cleared 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 (G90) cause the tool to move to the local coordinate system position.



(Note) If the machining program is executed many times repeatedly, the workpiece coordinate system may deviate slightly per execution.
Command to execute reference point return at the program end.

(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 ;

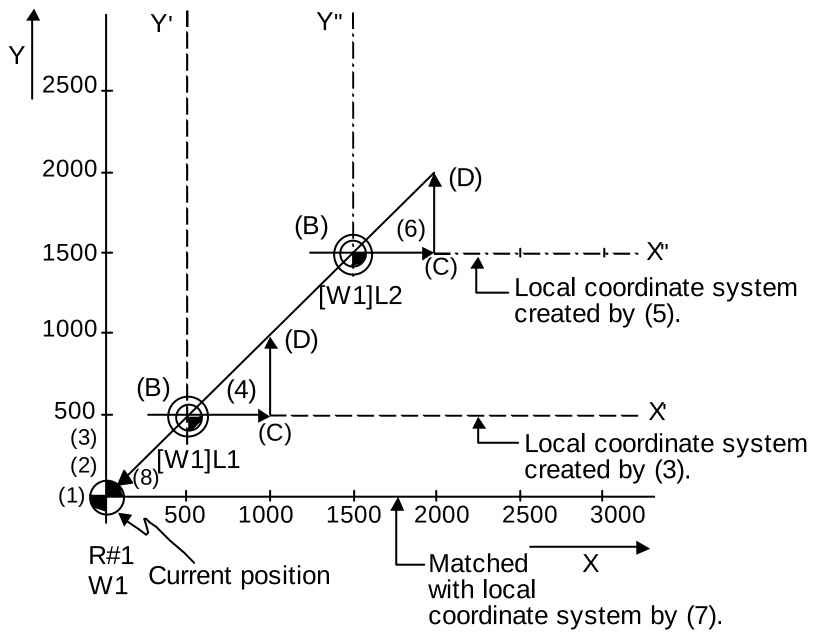


The local coordinate system is created by (5), canceled (9) and matched with the coordinate system

(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 ;



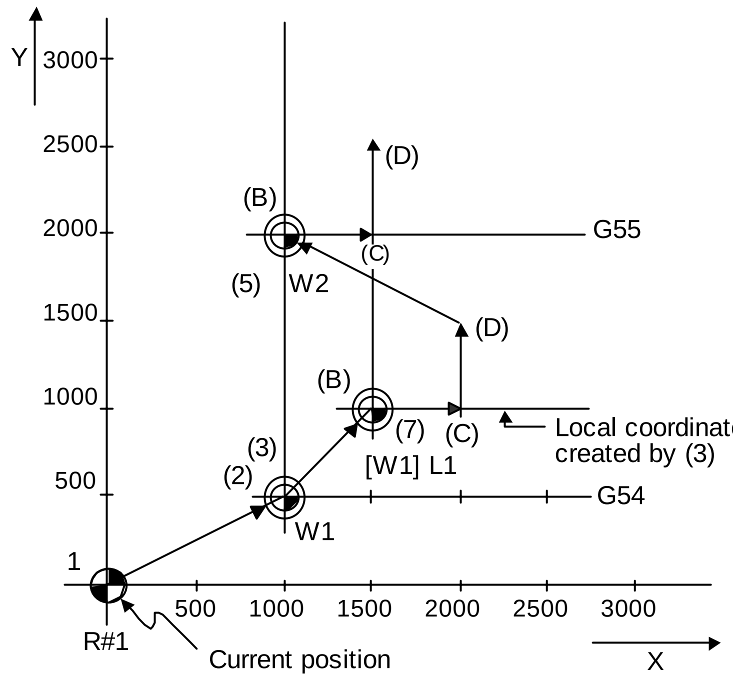
(Explanation)

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 (parameter setting value)
Y	500	2000	

```
(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 ;
     %
```



(Explanation)

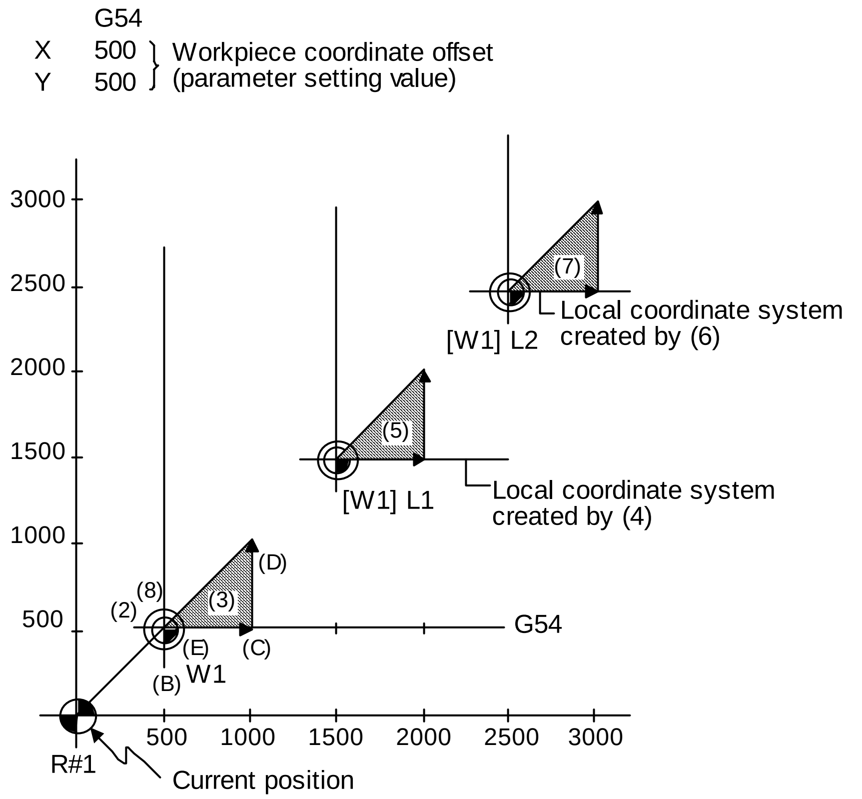
The local coordinate system is created at the G54 coordinate system (500,500) position by (3), but the local coordinate system is not created for the G55 coordinate system.

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 ;
%
```



(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).

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.

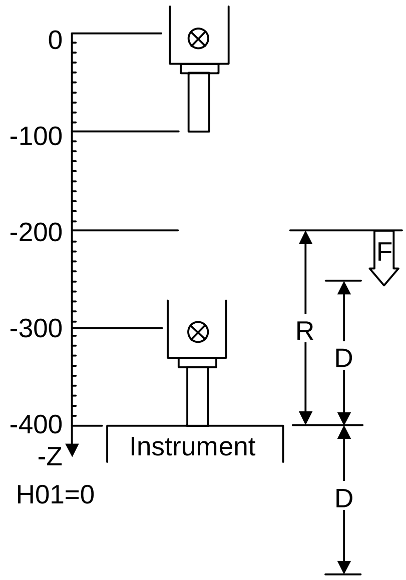
When R_, D_ of F_ is omitted, the value set in the parameter is used instead.

<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

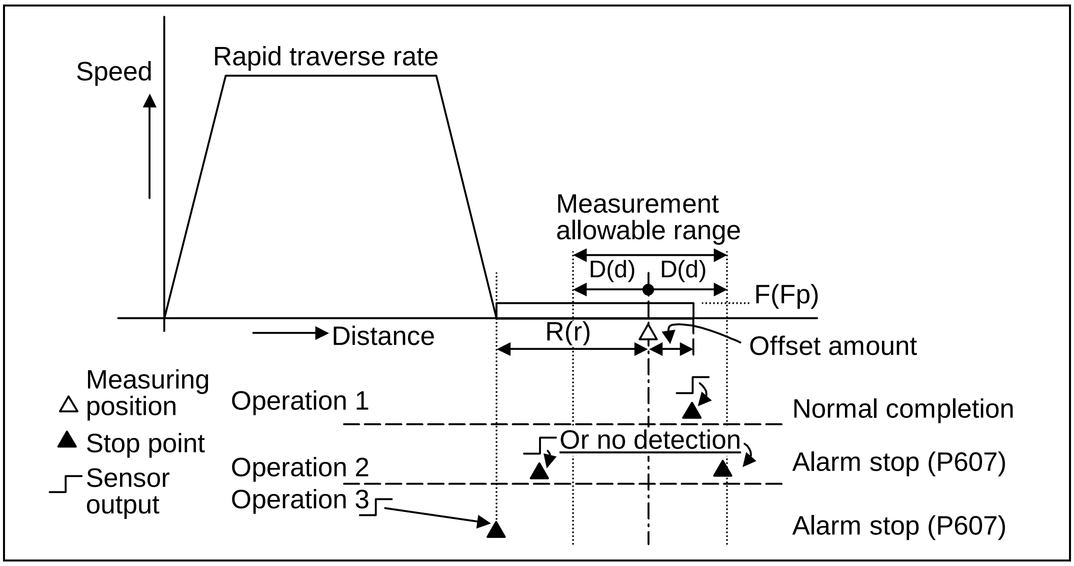
For new measurement



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}).

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:

|Measurement point – start point| > R command or parameter r
> 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.

```
G43H_;
G37 Z_ R_ D_ F_;
```

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 αα 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 commanded as the feedrate in the same block as G31 command, commanded speed "f" will apply; if it not commanded, 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) Normally, the machine will not automatically accelerate or decelerate with the G31 block. However, setting the base specification parameter "#21101 add01/bit3" to "1" allows the automatic acceleration/deceleration valid.
In such case, the acceleration/deceleration will apply following to the cutting feed acceleration/deceleration pattern set with the axis specification parameter "#2003 smgst". Since the deceleration at skip signal input follows the cutting feed acceleration/deceleration pattern mentioned above, the coasting amount from the skip signal input to stop may be larger than the normal specifications (when automatic acceleration/deceleration is invalid)

(3) The stop condition (such as feed hold, stroke end) is also valid for the G31 block.

(4) With the normal specifications, override and dry run are invalid during execution of G31 block. However, setting the base specification parameter "#21101 add01/bit3" to "1" allows the override and dry run.

(5) The G31 command is unmodal and so it needs to be commanded each time.

(6) If the skip signal 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.

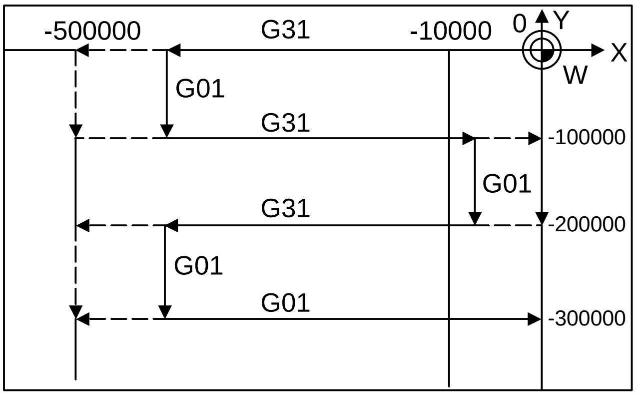
(7) When the G31 command is issued during nose R compensation, program error (P608) will result.

(8) When there is no F command in the G31 command and the parameter speed is also zero, program error (P603) will result.

(9) If only the Z axis is commanded when the machine lock is ON or the Z axis cancel switch is ON, the skip signal will be ignored and execution will continue as far as the end of the block.

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 #506n (nth 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.
~	

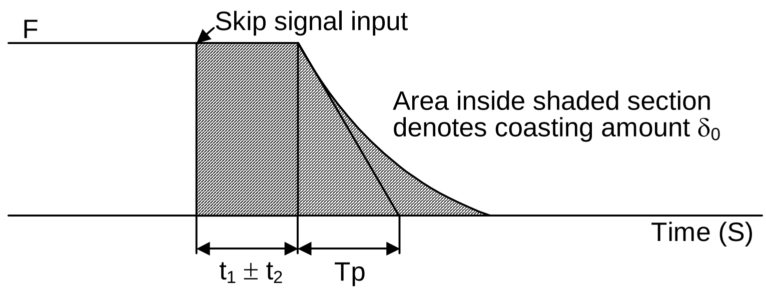
Detailed description (G31 coasting)

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

$$\delta_0 = \frac{F}{60} \times Tp + \frac{F}{60} \times (t_1 \pm t_2) = \underbrace{\frac{F}{60} \times (Tp + t_1)}_{\delta_1} \pm \underbrace{\frac{F}{60} \times t_2}_{\delta_2}$$

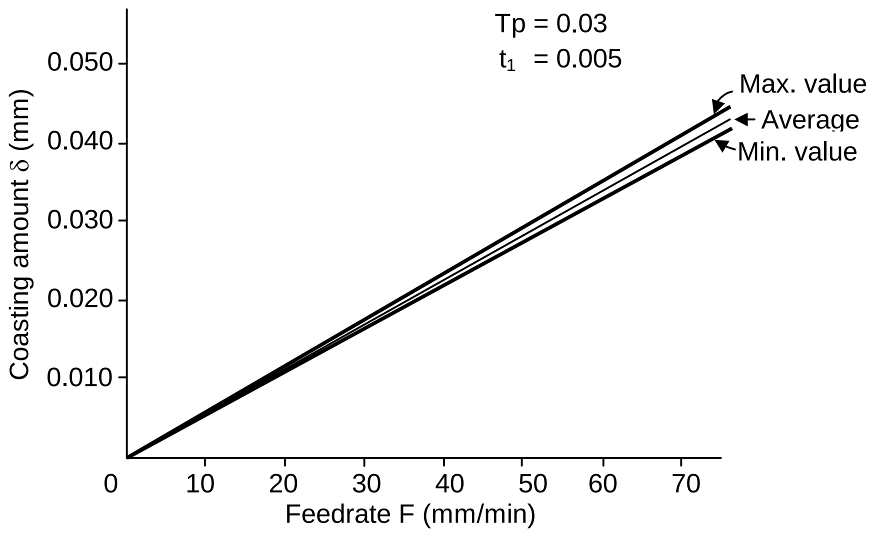
δ_0 : Coasting amount (mm)
F : G31 skip speed (mm/min.)
Tp : Position loop time constant (s) = (position loop gain)$^{-1}$
t_1 : Response delay time (s) = (time taken from the detection to the arrival of the skip signal at the controller via PC)
t_2 : Response error time (0.001 s)

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)

(Note) When the base specification parameter "#21101 add01/bit3" is set to "1", the automatic acceleration/deceleration becomes valid for the deceleration at skip signal input.
Thus, the coasting amount from the skip signal input to stop may be larger than when the automatic acceleration/deceleration is invalid.

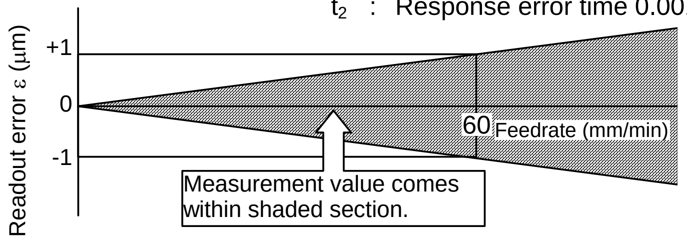
Detailed description (Skip coordinate readout error)

(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 signal input coordinates

Readout error of skip input coordinates
Readout error with a 60mm/min feedrate is:

$\varepsilon = \pm \frac{60}{60} \times 0.001 = \pm 0.001$ (mm)

Measurement value is within readout error range of ± 1μm.

(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 t_1 ;

```
        ≀
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
        ≀
```

(2) Compensating for work coordinates

#110 = Skip feedrate ;
#111 = Response delay time t_1 ;
#112 = Position loop time constant Tp ;

```
        ≀
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
        ≀
```

15.3 Multi-step skip function1; 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**			
	4	**3**	**2**	**1**
1	×	×	×	○
2	×	×	○	×
3	×	×	○	○
4	×	○	×	×
5	×	○	×	○
6	×	○	○	×
7	×	○	○	○
8	○	×	×	×
9	○	×	×	○
10	○	×	○	×
11	○	×	○	○
12	○	○	×	×
13	○	○	×	○
14	○	○	○	×
15	○	○	○	○

(Skip when "○" signal is input.)

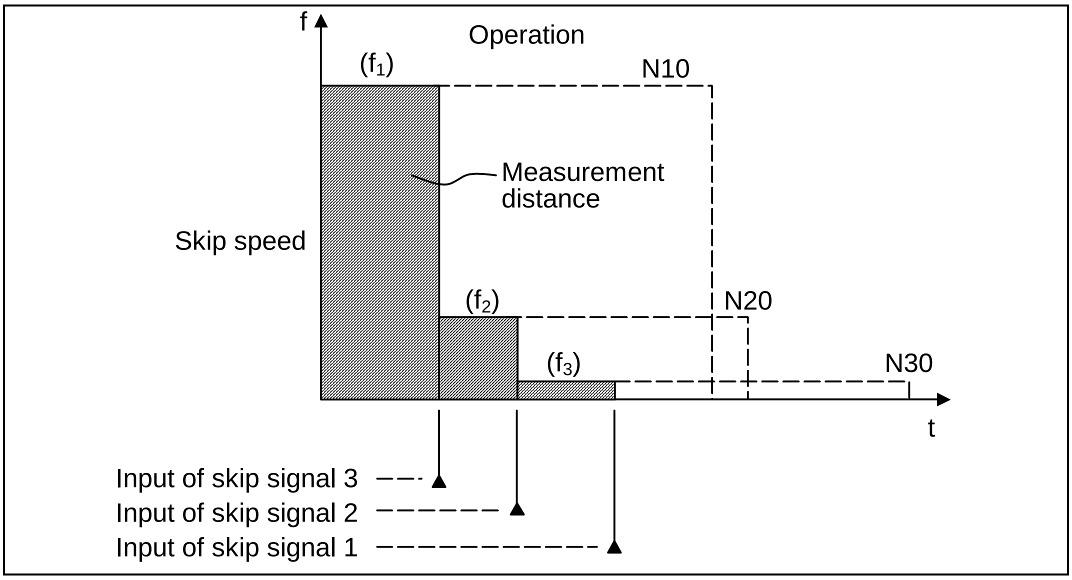
Example of operation

(1) The multi-step 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 ;
```



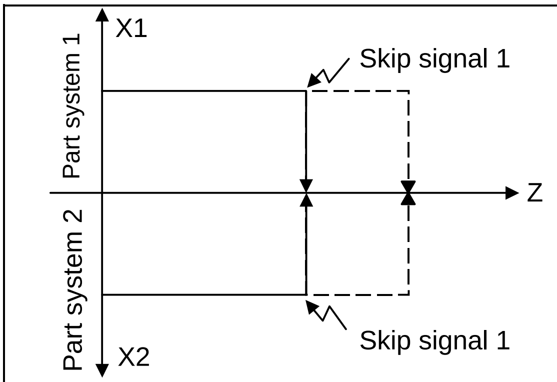
(Note 1) If skip signal 1 is input before skip signal 2 in the above operation, N20 is skipped at that point and N30 is also ignored.

(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.

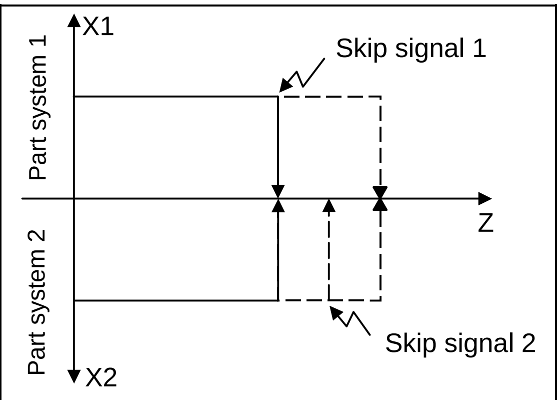
15.4 Multi-step skip function 2; G31

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 multi-step skip commands are issued simultaneously in different part systems, both part systems perform skip operation simultaneously if the input skip signals are the same, or they perform skip operation separately if the input skip signals are different. The skip operation is the same as with a normal skip command (G31 without P parameter).



Same skip signals input in both part systems 1 and 2



Different skip signals input in part systems 1 and 2

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. Similarly, if the skip condition is met during revolution dwelling, the remaining revolution is canceled and the following block is executed.

Command format

G31 Xx Zz αα Pp Ff ;	
Xx Zz αα	: 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 command P	Valid skip signal			
	4	3	2	1
1	×	×	×	○
2	×	×	○	×
3	×	×	○	○
4	×	○	×	×
5	×	○	×	○
6	×	○	○	×
7	×	○	○	○
8	○	×	×	×
9	○	×	×	○
10	○	×	○	×
11	○	×	○	○
12	○	○	×	×
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-step 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).

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. However, the parameters with "●" in "etc" column (ZERO-RTN PARAM 2027, 2028, 2029) must be excluded.

(Example 1) To set 30mm in a parameter when the input setup unit is B (0.001mm) and metric system.
L60000

(Example 2) To set 5 inch in a parameter when the input setup unit is B (0.0001 inch) and inch system.
L100000

(Note 3) The binary type parameters must be converted into byte-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.)

P No. 2 (Axis independent parameter)

Parameter No.	Details	N No.	Data type	Setting range	(Unit)	Remarks
#8201	Axis bit parameter 2	896	Bit	Same as above		bit0 : bit1 : bit2 : bit3 : bit4 : bit5 : bit6 : Axis removal bit7 :
#8202	Axis bit parameter 1	897	Bit	d0 : No. d bit OFF or d1 : No. d bit ON (d : 0 ~ 7)		bit0 : bit1 : bit2 : Soft limit invalid bit3 : bit4 : bit5 : bit6 : bit7 :
#8204	Soft limit (–) (User stroke end lower limit)	916	2-word	$\pm$ 99999999 $\times$ 2	Interpolation unit	
#8205	Soft limit (+) (User stroke end upper limit)	912	2-word	$\pm$ 99999999 $\times$ 2	Interpolation unit	
#8206	Tool change	924	2-word	$\pm$ 99999999 $\times$ 2	Interpolation unit	

P No. 2 (Axis independent parameter)

Parameter No.	Details	N No.	Data type	Setting range	(Unit)	Remarks
#2013	OT-	292	2-word	$\pm$ 99999999 $\times$ 2	Interpolation unit	Axis specifications parameter
#2014	OT+	288	2-word	$\pm$ 99999999 $\times$ 2	Interpolation unit	Axis specifications parameter
#2015	tlml-	300	2-word	$\pm$ 99999999 $\times$ 2	Interpolation unit	Axis specifications parameter
#2016	tlml+	296	2-word	$\pm$ 99999999 $\times$ 2	Interpolation unit	Axis specifications parameter
#2017	tap_g	58	Word	0.25 ~ 200.00	(rad/s)	Axis specifications parameter
#2025	G28rap	260	2-word	1 ~ 999999	(min)	Zero point return parameter
#2026	G28crp	38	Word	1 ~ 60000	(min)	Zero point return parameter
#2027	G28sft	44	Word	0 ~ 65535	(μm)	Zero point return parameter ●
#2029	grspc	42	Word	-32767 ~ 999	(mm)	Zero point return parameter ●
#2028	grmask	40	Word	0 ~ 65535	(μm)	Zero point return parameter ●
#2030	dir(-)	20	Bit2	0 ~ 1		Zero point return parameter
#2031	noref	21	Bit2	0 ~ 1		Zero point return parameter
#2032	nochk	54	Bit0	0 ~ 1		Zero point return parameter
#2037	G53ofs	272	2-word	$\pm$ 99999999 $\times$ 2	Interpolation unit	Zero point return parameter
#2038	#2_rfp	276	2-word	$\pm$ 99999999 $\times$ 2	Interpolation unit	Zero point return parameter
#2039	#3_rfp	280	2-word	$\pm$ 99999999 $\times$ 2	Interpolation unit	Zero point return parameter
#2040	#4_rfp	284	2-word	$\pm$ 99999999 $\times$ 2	Interpolation unit	Zero point return parameter
#2061	OT-1B-	324	2-word	$\pm$ 99999999 $\times$ 2	Interpolation unit	Axis specifications parameter 2
#2062	OT-1B+	320	2-word	$\pm$ 99999999 $\times$ 2	Interpolation unit	Axis specifications parameter 2

Appendix 1. Program Parameter Input N No. Correspondence Table

P No. 5 (PLC constant)

Parameter No.	Details	N No.	Data type	Setting range	(Unit)	Remarks
#6301 ~ #6348	PLC constant	1 ~ 48	2-word	0 ~ 99999999		• N No. corresponds to the constant No. (# No.) on the PLC constant screen.

P No. 6 (PLC timer)

Parameter No.	Details	N No.	Data type	Setting range	(Unit)	Remarks
#6000 ~ #6015	10ms addition timer (T0 ~ T15)	0 ~ 15	Word	0 ~ 32767	0.01 s	• Each N No. corresponds to the # No. on the PLC timer screen.
#6016 ~ #6095	10ms addition timer (T16 ~ T95)	16 ~ 95	Word	0 ~ 32767	0.1 s	
#6096 ~ #6103	10ms addition timer (T96 ~ T103)	96 ~ 103	Word	0 ~ 32767	0.1 s	

P No. 7 (PLC counter)

Parameter No.	Details	N No.	Data type	Setting range	(Unit)	Remarks
#6200 ~ #6223	Counter (C0 ~ C23)	0 ~ 23	Word	0 ~ 32767		• N No. corresponds to the # No. on the PLC counter screen.

P No. 8 (Bit selection parameter)

Parameter No.	Details	N No.	Data type	Setting range	(Unit)	Remarks
#6401 ~ #6496	Bit selection parameter	0 ~ 96	Word	8-digit designation (Reading abbreviation not possible) Each bit 0 or 1 d0 : No. d bit OFF or d1 : No. d bit ON (d : 0 ~ 7)		• N No. corresponds to the # No. on the bit selection screen. • N Nos. 49 to 96 are used by the machine maker and Mitsubishi. These must not be used by the user.

Appendix 1. Program Parameter Input N No. Correspondence Table

P No. 11 (Axis common parameters (per part system))

Parameter No.	Details	N No.	Data type	Setting range	(Unit)	Remarks
#8004	Automatic tool length measurement instrument speed	844	2-word	1 ~ 60000	(mm/min)	Machining parameter
#8005	Automatic tool length measurement deceleration range r	836	2-word	0 ~ 99999999 × 2	Interpolation unit	Machining parameter
#8006	Automatic tool length measurement deceleration range d	840	2-word	0 ~ 99999999 × 2	Interpolation unit	Machining parameter
#8008	Automatic corner override max. angle	756	2-word	0 ~ 180	Degree (°)	Machining parameter
#8009	Automatic corner override precorner length	760	2-word	0 ~ 99999999	Interpolation unit	Machining parameter
#8010	Wear data input max. value	776	2-word	0 ~ 99999	Interpolation unit	Machining parameter
#8011	Wear data input max. addition	780	2-word	0 ~ 99999	Interpolation unit	Machining parameter
#8013	G83 return amount	832	2-word	0 ~ 99999999 × 2	Interpolation unit	Machining parameter
#8014	Thread cutting cycle cutoff angle	1011	Byte	0 ~ 89	Degree (°)	Machining parameter
#8015	Thread cutting cycle chamfering amount	1012	Byte	1 ~ 127	0.1 lead	Machining parameter
#8016	G71 cut amount	788	2-word	0 ~ 99999 × 2	Interpolation unit	Machining parameter
#8017	G71 cut amount change amount	792	2-word	0 ~ 99999 × 2	Interpolation unit	Machining parameter
#8301 X	X axis chuck barrier range 1	1136	2-word	± 99999999 × 2	Interpolation unit	Barrier
#8302 X	X axis chuck barrier range 2	1140	2-word	± 99999999 × 2	Interpolation unit	Barrier
#8303 X	X axis chuck barrier range 3	1144	2-word	± 99999999 × 2	Interpolation unit	Barrier
#8304 X	X axis chuck barrier range 4	1148	2-word	± 99999999 × 2	Interpolation unit	Barrier
#8305 X	X axis chuck barrier range 5	1152	2-word	± 99999999 × 2	Interpolation unit	Barrier
#8306 X	X axis chuck barrier range 6	1156	2-word	± 99999999 × 2	Interpolation unit	Barrier
#8301 Z	Z axis chuck barrier range 1	1160	2-word	± 99999999 × 2	Interpolation unit	Barrier
#8302 Z	Z axis chuck barrier range 2	1164	2-word	± 99999999 × 2	Interpolation unit	Barrier
#8303 Z	Z axis chuck barrier range 3	1168	2-word	± 99999999 × 2	Interpolation unit	Barrier
#8304 Z	Z axis chuck barrier range 4	1172	2-word	± 99999999 × 2	Interpolation unit	Barrier

P No. 11 (Axis common parameters (per system))

Parameter No.	Details	N No.	Data type	Setting range	(Unit)	Remarks
#8305 Z	Z axis chuck barrier range 5	1176	2-word	± 99999999 × 2	Interpolation unit	Barrier
#8306 Z	Z axis chuck barrier range 6	1180	2-word	± 99999999 × 2	Interpolation unit	Barrier

Appendix 2. Program Error

(The message in bold characters appears on 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
P10	**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
P11	**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.
P20	**DIVISN ERROR** An axis command which cannot be divided by the command unit has been issued.	• Check the program.
P30	**PARITY H** The number of holes per character on the paper tape is an even number for EIA codes and an odd number for ISO codes.	• Check the paper tape. • Check the tape puncher and tape reader.
P31	**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.
P32	**ADDRESS ERROR** An address not listed in the specifications has been used.	• Check and revise the program address. • Check the specifications.
P33	**FORMAT ERROR** The command format in the program is not correct.	• Check the program.
P34	**G-CODE ERROR** A G code not listed in the specifications has been used.	• Check and correct the G code address in the program.
P35	**CMD-VALUE OVER** The setting range for the addresses has been exceeded.	• Check the program.
P36	**PRGRAM END ERR** "EOR" has been read during tape and memory operation.	• Enter the M02 and M30 commands at the end of the program. • Enter the M99 command at the end of the subprogram.
P37	**PROG NO. ZERO** A zero has been designated for a program number or sequence number.	• The program numbers are designated across a range from 1 to 99999999. • The sequence numbers are designated across a range from 1 to 99999.
P39	**NO SPEC ERR** A command not found in the specifications was issued.	• Check the specifications
P40	**PREREAD BL. ERR** When executing tool radius compensation, there was an error in the pre-read block, so the interference could not be checked.	• Review the program.

Error No.	Details	Remedy
P60	**OVER CMP. LENG.** The commanded movement distance is too long. (2^{31} was exceeded.)	• Review the axis address command range.
P62	**F-CMD NOTHING** No feedrate command has been issued.	• 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.
P70	**ARC ERROR** There is an error in the arc start and end points as well as in the arc center.	• Check the numerical values of the addresses that specify the start and end points as well as the arc center in the program. • Check the "+" and "–" directions of the address numerical values.
P71	**ARC CENTER** The arc center is not sought during R-specified circular interpolation.	• Check the numerical values of the addresses in the program.
P72	**NO HELICAL SPC** A helical command has been issued though it is not included in the specifications.	• Check the helical specifications. • An Axis 3 command was issued in the circular interpolation command. • If the command is not a helical command, the linear command axis will be moved to the next block.
P90	**NO THREAD SPEC** A thread cutting command has been issued though it is not included in the specifications.	• Check the specifications.
P93	**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.
P111	**PLANE CHG (CR)** A plane selection command (G17, G18, G19) was issued during the coordinate rotation command (G68).	• After the G68 command, always command G69 (coordinate rotation cancel), and then issue the plane selection command.
P112	**PLANE CHG (CC)** • A plane selection command (G17, G18, G19) has been issued when the tool radius compensation command (G41, G42) or nose radius compensation command (G41, G42, G46) is issued. • After nose R compensation was completed, there was no axis movement command after G40, and the plane selection command was issued before the compensation was canceled.	• Issue the plane selection command after completing the tool radius compensation and nose R compensation commands (issue the axis movement command after issuing the G40 cancel command).
P113	**ILLEGAL PLANE** The arc command axis is not on the selected plane.	• Issue arc command on the correctly selected plane.

Error No.	Details	Remedy
P122	**NO AUTO C-OVER** 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.
P130	**2ND AUX. ADDR** The second miscellaneous function address specified in the program does not match that set by the parameter.	• Check and correct the second miscellaneous function address in the program.
P131	**NO G96 SPEC** (No constant surface speed) The constant surface 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 surface speed command (G96) to the speed command (G97).
P132	**SPINDLE S = 0** No spindle speed command has been input.	• Review the program.
P133	**CONTROL AXIS NO. ERR** An invalid constant surface speed control axis has been specified.	• Review the parameter specified for the constant surface speed control axis.
P150	**NO C-CMP SPEC** • A tool radius compensation command (G41, G42) has been issued though there are no tool radius compensation specifications. • A nose R compensation command (G41, G42 G46) has been issued though there are no nose R compensation specifications.	• 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 radius compensation (G41, G42, or G46) command, the intersection point after one block is skipped cannot be determined.	• Review the program.
P153	**I.F ERROR** An interference error has arisen while the tool radius compensation command (G41, G42) or nose radius compensation command (G41, G42, G46) was being executed.	• Review the program.
P155	**F-CYC ERR (CC)** A fixed cycle command has been issued in the tool radius compensation mode.	• The tool radius compensation mode is established when a fixed cycle command is executed and so the tool radius compensation cancel command (G40) should be issued.
P156	**BOUND DIRECT** At the start of G46 nose radius 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.

<table>
<tr><th>Error No.</th><th>Details</th><th>Remedy</th></tr>
<tr><td>P157</td><td>SIDE REVERSED
During G46 nose radius compensation, the compensation direction is inverted.</td><td>• 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 G46 inversion error avoidance parameter.</td></tr>
<tr><td>P158</td><td>ILLEGAL TIP P
During G46 nose radius compensation, the tip point is illegal (other than 1 to 8).</td><td>• Change the tip point number to a legal one.</td></tr>
<tr><td>P170</td><td>NO CORR. NO.
The compensation number (D○○, T○○, H○○) command was not given when the tool radius compensation (G41, G42, G43, G46) command was issued. Alternatively, the compensation number is larger than the number of sets in the specifications.</td><td>• Add the compensation number command to the compensation command block.
• Check the number of compensation number sets and correct it to a compensation number command within the permitted number of compensation sets.</td></tr>
<tr><td>P172</td><td>P10 L-NO. ERR (G10 L-number error)
The L address command is not correct when the G10 command is issued.</td><td>• Check the address L-Number of the G10 command and correct the number.</td></tr>
<tr><td>P173</td><td>G10 P-NO. ERR (G10 compensation error)
When the G10 command is issued, a compensation number not within the permitted number of sets in the specifications has been commanded for the compensation number command.</td><td>• First check the number of compensation sets and then set the address P designation to within the permitted number of sets.</td></tr>
<tr><td>P177</td><td>COUNTING LIFE
Registration of tool life management data with G10 was attempted when the used data count valid signal was ON.</td><td>• The tool life management data cannot be registered when counting the used data. Turn the used data count valid signal OFF.</td></tr>
<tr><td>P178</td><td>LIFE REGISTRATION OVER
The No. of registration groups, total No. of registered tools or the No. of registrations per group exceeded the specifications range.</td><td>Review the No. of registrations.
The maximum No. of registrations is shown below.
<table>
<tr><th>System</th><th>System 1</th><th>System 2</th></tr>
<tr><td>No. of groups</td><td>80</td><td>40/40</td></tr>
<tr><td>No. of tools</td><td>80</td><td>40/40</td></tr>
<tr><td>Per group</td><td colspan="2">16</td></tr>
</table></td></tr>
<tr><td>P179</td><td>Group No. Illegal
• 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, which must be commanded independently, was issued in the same block as other M code commands.
• One or more M code commands set in the same group were found in the same block.</td><td>• The group No. cannot be commanded in duplicate. When registering the group data, register it in group units.
• Correct to the correct group No.</td></tr>
</table>

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 speed command has not been issued when the tapping fixed cycle command is given.	• Issue the spindle speed command (S) when the tapping 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 unit. • 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 boring fixed cycle command.	• Specify the pitch data and the number of threads by F or E command.
P184	**NO PTC/THD CND** The pitch or the number of threads per inch is illegal in the tap cycle of the drilling fixed cycle command	• 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 command value in the lathe cutting 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** A compound type fixed cycle I command (G70 to G73) was issued although this cycle was undefined in the specification.	• Check the specification.
P201	**PROG. ERR (MRC)** When called with a compound type fixed cycle I command, the subprogram contained at least one of the following commands: • Reference point return command (G27, G28, G30) • Thread cutting (G33) • Fixed-cycle skip-function (G31) • The first move block of the finish shape program in compound type fixed cycle I contains an arc command.	• Delete the following G codes from this subprogram that is called with the compound type fixed cycle I commands (G70 to G73): G27, G28, G30, G31, G33, fixed-cycle G-code. • Remove G02 and G03 from the first move block of the finish shape program in multiple fixed cycle I.
P202	**BLOCK OVR (MRC)** The number of blocks in the shape program of the compound type fixed cycle I is over 50.	• The number of blocks in the shape program called by the compound type fixed cycle I commands (G70 to G73) must be decreased below 50.

Error No.	Details	Remedy
P203	**CONF. ERR (MRC)** The compound type fixed cycle I (G70 to G73) shape program could not cut the work normally because it defined an abnormal shape.	• Check the compound type fixed cycle I (G70 to G73) shape program.
P204	**C-FORMAT ERR** A command value of the compound type fixed cycle (G70 to G76) is illegal.	• Check the compound type fixed cycle (G70 to G76) command value.
P210	**NO PAT CYC SPC** A compound type fixed cycle 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.	• Review the program.
P222	**G36 ANGLE ERR** A G36 command specifies 0 for angle intervals.	• Review the program.
P223	**G12, G13 R ERR** The radius value specified with a G12 or G13 command is below the compensation amount.	• Review the program.
P224	**NO G12, G13 SPEC** There are no circular cutting specifications.	• Check the specifications.
P230	**NESTING OVER** A subprogram has been called 8 or more times in succession from the subprogram.	• Check the number of subprogram calls and correct the program so that it does not exceed 8 times.
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 the IC card, check the program in the IC card and the number of IC card program calls.
P232	**NO PROGRAM NO.** The subprogram has not been set when the subprogram is called.	• 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.
P260	**NO COOD-RT SPC** The coordinate rotation command was issued when the coordinate rotation specifications were not available.	• Check the specifications.

Error No.	Details	Remedy
P270	**NO MACRO SPEC** A macro specification was commanded though there are no such command specifications.	• Check the specifications.
P271	**NO MACRO INT.** A macro interrupt command has been issued though it is not included in the specifications.	• Check the specifications.
P272	**NC/MACRO ILL.** An NC statement and a macro statement exist together in the same block.	• Review the program and place the executable statement and macro statement in separate blocks.
P273	**MACRO OVERCALL** The frequency of the macro call has exceeded the limit imposed by the specification.	• Review 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.	• Review the program.
P276	**CALL CANCEL** A G67 command was issued though it was not during the G66 command modal.	• Review 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 [,] which can be commanded in a single block has exceeded five.	• Review 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.	• Review the program and correct it so that " [" and "] " parentheses are paired up properly.
P282	**CALC. IMPOSS** The arithmetic formula is incorrect.	• Review the program and correct the formula.
P283	**DIVIDE BY ZERO** The denominator of the division is zero.	• Review the program and correct it so that the denominator for division in the formula is not zero.
P284	**INTEGER OVER** In the process of the calculation the integral number has exceeded -2^{31} $(2^{31}-1)$.	• Check the arithmetic formula in the program and correct it so that the value of the integral number after calculation does not exceed -2^{31}.
P285	**OVERFLOW VALUE** The variable data has overflowed.	• Check the variable data in the program.
P290	**IF SNT. ERR** There is an error in the IF conditional GOTO☐ statement.	• Review the program.
P291	**WHILE SNT. ERR** There is an error in the WHILE conditional DO☐~END☐ statement.	• Review the program.

Error No.	Details	Remedy
P292	**SETVN SNT. ERR** There is an error in the SETVN☐ statement when the variable name setting was made.	• Review the program. • The number of characters in the variable name of the SETVN statement must be 7 or less.
P293	**DO-END EXCESS** The number of ☐s for DO-END☐ in the WHILE conditional DO☐-END☐ statement has exceeded 27.	• Review the program and correct it so that the number of the DO-END statement does not exceed 27.
P294	**DO-END MMC.** The DO's and END's are not paired off properly.	• Review the program and correct it so that the DO and END 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 MACRO ADDR.** 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	**VAR. NAME ERROR** The variable names have not been commanded properly.	• Review 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.
P380	**NO CORNER R/C** A command was issued for corner rounding or corner chamfering though there are no such specifications.	• Check the specifications. • Remove the corner rounding or chamfering command from the program.
P381	**NO ARC R/C SPC** Corner rounding or chamfering 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 rounding/ chamfering is not a movement command.	• Replace the block succeeding the corner rounding/chamfering command by movement command block.

Error No.	Details	Remedy
P383	**CORNER SHORT** In the corner rounding or chamfering command, the movement distance was shorter than the value in the corresponding command.	• Make the corner rounding or chamfering less than the movement distance since this distance is shorter than the corner rounding or chamfering.
P384	**CORNER SHORT** When the corner rounding or chamfering command was input, the movement distance in the following block was shorter than the length of the corner rounding or chamfering.	• Make the corner rounding or chamfering less than the movement distance since this distance in the following block is shorter than the corner rounding or chamfering.
P385	**G0 G33 IN CORN** A block with corner rounding/chamfering 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 specification.	• 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.
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 END EPR (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 GEOMT IB** Although the geometric IB specifications are not included, a geometric command is given.	• Check the specifications.
P420	**NO PARAM** Although the programmable parameter input specifications are not provided, the command was given.	• Check the specifications.

Error No.	Details	Remedy
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.** 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.
P450	**NO CHUCK BARR.** The chuck barrier on command (G22) was specified although the chuck barrier was undefined in the specification.	• Check the specifications.
P460	**TAPE I/O ERROR** An error has arisen in the tape reader or, alternatively, in the printer during macro printing.	• Check the power and cable for the connected device. • Check the input/output unit parameters.
P461	**FILEI/O ERROR** A file of the machining program cannot be read.	• During memory operation, the program saved in the memory may be corrupted. Output all of the programs and tool data, etc., once, and format the memory.
P600	**NO AUTO TLM** An automatic tool length measurement command (G37) was executed 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.

Error No.	Details	Remedy
P602	**NOMULTI 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 F0** The skip speed is 0.	• Specify the skip speed.
P604	**TLM ILL. AXIS command** No axis or more than one axis was specified in the automatic tool length measurement block.	• Specify 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 automatic tool length measurement 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 tool radius compensation processing.	• Specify a diameter cancel (G40) command, or remove the skip command.
P610	**ILLEGAL PARA.** • G114.1 was commanded when the spindle synchronization with PLC I/F command was selected. • Spindle synchronization was commanded to a spindle that is not connected serially.	• Check the program. • Check the argument of G114.1 command. • Check the state of spindle connection.
P701	**REGARD A POINT** A decimal point was added to a decimal point invalid address.	• Do not add a decimal point to the decimal point invalid address.
P990	**PRE-CALCULATION ERROR** combining commands that required pre-reading (nose R offset, corner chamfering corner R, geometric I, geometric IB, and compound type fixed cycle commands) 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 code Commanded G code	01 G00 ~ G03	02 G17 ~ G19	03 G90, G91		05 G94, G95	06 G20, G21	07 G40 ~ G42	08 G43, G44, G49
G00~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 ○ Radius is compen-sated, and then moves	Arc and G43~ 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~G42 are ignored ○	G04 is executed G43~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, G11 are executed G40~G42 are ignored ○	G10, G11 are executed G43~G49 are ignored ○
G17 ~ G19 Plane selection	○ ○	○ G command commanded last is valid. ○	○ ○		○ ○	○ ○	○ Plane axis changeover during radius compensa-tion causes error P112	○ ○

Appendix 3. Order of G Function Command Priority

G code Commanded G code	01 G00 ~ G03	02 G17 ~ G19	03 G90, G91		05 G94, G95	06 G20, G21	07 G40 ~ G42	08 G43, G44, G49
G20, G21 Inch/metric changeover	○ ○	○ ○	○ ○		○ ○	Possible in same block ○	○ ○	○ ○
G27 ~ G30 Reference point compare/ return	○ G00~G03.1 modals are updated G27~G30 are executed ○	○ ○	○ ○		○ ○	○ ○	G27~G30 are executed G40~G42 are ignored ○	G27~G30 are executed G43~G49 are ignored ○
G31 ~ 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~G33 are ignored ○	○ ○	○ ○		○ ○	○ ○	G37 is executed G40~G42 are ignored	G37 is executed G43~G49 are ignored ○
G40 ~ G42 Tool radius compensation	Arc and G41, G42 cause error P151 G41 and G42 in arc modal cause error P151	○ Plane axis changeover during radius compensa-tion causes error P112	○ ○		○ ○	○ ○	G command commanded last is valid. ○	○ ○

Appendix 3. Order of G Function Command Priority

G code Commanded G code	09 G73 ~ G89	10 G98, G99	12 G54 ~ G59	13 G61 ~ G64	14 G66 ~ G67	17 G96, G97		19 G50.1 G51.1
G00~G03.1 Positioning/ interpolation	Group 1 command is executed Group 9 is canceled	○ ○	○ ○	○ ○	G66 ~ G67 are executed G00~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~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, G11 are executed G73~G89 are ignored ○	○ ○	○ G10 is executed G54~G59 modals are updated ○	○ ○	G66 ~ G67 are executed G10 is ignored ○	○ ○		G10, G11 are executed G50.1 and G51.1 are ignored ○
G17 ~ G19 Plane selection	○ ○	○ ○	○ ○	○ ○	○ ○	○ ○		○ ○

Appendix 3. Order of G Function Command Priority

G code Commanded G code	09 G73 ~ G89	10 G98, G99	12 G54 ~ G59	13 G61 ~ G64	14 G66 ~ G67	17 G96, G97		19 G50.1 G51.1
G20, G21 Inch/metric changeover	○ ○	○ ○	○ ○	○ ○	○ ○	○ ○		○ ○
G27 ~ G30 Reference point compare/ return	○ ○	○ ○	○ ○	○ ○	G66 ~ G67 are executed G27~G30 are ignored ○	○ ○		G27~G30 are executed G50.1 and G51.1 are ignored ○
G31 ~ G31.3 Skip	○ ○	○ ○	○ ○	○ ○	○ ○	○ ○		○ ○
G33 Thread cutting	Group 1 command is executed Group 9 is canceled ○	○ ○	○ ○	○ ○	G66 ~ G67 are executed G33 modals is updated ○	○ ○		○ ○
G37 Automatic tool length measurement		○ ○	○ ○	○ ○	G66 ~ G67 are executed G37 modals is ignored ○	○ ○		G37 is executed G50.1 and G51.1 are ignored ○
G40 ~ G42 Tool radius compensation	Error:P155 Error:P155	○ ○	○ ○	○ ○	○ ○	○ ○		○ ○

Appendix 3. Order of G Function Command Priority

G code Commanded G code	01 G00~G03.1 G33	02 G17 ~ G19	03 G90, G91		05 G94, G95	06 G20, G21	07 G40 ~ G42	08 G43, G44, G49
G43, G44, G49 Length compensation	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~G42 are ignored ○	G52 is executed G43~G49 are ignored ○
G53 Machine coordinate system	○ ○	○	○ ○		○	○ ○	G53 is executed G40~G42 are ignored ○	G53 is executed G40~G42 are ignored ○
G54 ~ G59 Workpiece coordinate system	○ ○	○ ○	○ ○		○ ○	○ ○	○ ○	○ ○
G61 ~ G64 Mode selection	○ ○	○ ○	○ ○		○ ○	○ ○	○ ○	○ ○
G65 Macro call	G65 is executed G00~G03.1 modals are updated ○	○ ○	○ ○		○ ○	○ ○	○ ○	G65 is executed G43~G49 modals are updated ○

Appendix 3. Order of G Function Command Priority

G code Commanded G code	01 G00~G03.1 G33	02 G17, G19	03 G90, G92		05 G94, G95	06 G20, G21	07 G40 ~ G42	08 G43, G44 G49
G66 ~ G67 Macro call	G66 ~ G67 are executed G00~G03.1 modals are updated ○	○ ○	○ ○		○ ○	○ ○	○ ○	○ G66 ~ G67 are executed G43~G49 modals are updated ○
G73 ~ G89 Canned cycle	G73~G89 are canceled G01~G33 modals are updated ○	○ ○	○ ○		○ ○	○ ○	Error:P155 Canned cycle during compensa-tion 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 code Commanded G code	09 G73 ~ G89	10 G98, G99	12 G54 ~ G59	13 G61 ~ G65	14 G66 ~ G67	17 G96, G97		19 G50.1 G51.1
G43, G44, G49 Length compensation	○ ○	○ ○	○ ○	○ ○	G66 ~ G67 are executed G43~G49 modals are updated ○	○ ○		○ ○
G50.1 G51.1 Program mirror image	○ ○	○ ○	○ ○	○ ○	G66 ~ G67 are executed G50.1 G51.1 is ignored ○	○ ○		G command commanded last is valid. ○
G52 Local coordinate system	G52 is executed G73~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 ~ G59 Workpiece coordinate system	○ ○	○ ○	G command commanded last is valid. ○	○ ○	G66 ~ G67 are executed G54~G59 modals are updated ○	○ ○		○ ○
G61 ~ G64 Mode selection	○ ○	○ ○	○ ○	G command commanded last is valid. ○	○ ○	○ ○		○ ○
G65 Macro call	G65 is executed G73~G89 are ignored ○	○ ○	○ ○	○ ○	Error ○	○ ○		G65 is executed G50.1 G51.1 is ignored ○

Appendix 3. Order of G Function Command Priority

G code Commanded G code	09 G73 ~ G89	10 G98, G99	12 G54 ~ G59	13 G61 ~ G67	14 G66 ~ G67	17 G96, G97		19 G50.1 G51.1
G66 ~ G67 Macro call	G66 ~ G67 are executed G73~G89 are ignored ○	○ ○	G66 ~ G67 are executed G54~G59 modals are updated ○	○ ○	G command commanded last is valid.	○ ○		G66 ~ G67 are executed G50.1 G51.1 is ignored ○
G73 ~ G89 Canned cycle	G command commanded last is valid. ○	○ ○	○ ○	○ ○	G66 ~ G67 are executed G73~G89 are ignored ○	○ ○		○ All axes become mirror center ○
G90, G91 Absolute value/ incremental value	○ ○	○ ○	○ ○	○ ○	○ ○	○ ○		○ ○
G92 Coordinate system setting	G92 is executed G73~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
December 2000	BNP-B2260*	First edition created.
May 2004	BNP-B2260B	• The contents revised following to the software Ver.C and Ver.D. • Mistakes, etc. were corrected.

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 instruction manual may not be reproduced in any form, in part or in whole, without written permission from Mitsubishi Electric Corporation.

MITSUBISHI ELECTRIC CORPORATION
HEAD OFFICE : MITSUBISHI DENKI BLDG., 2-2-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN

MODEL	MC6/C64/C64T(M/T)
MODEL CODE	008-047
Manual No.	BNP-B2260B（ENG)

Specifications subject to change without notice.
(0405) MEE
Printed in Japan on recycled paper.