CNC SYSTEM FOR TURNING APPLICATIONS

YASNAC J300L

PROGRAMMING MANUAL

Upon receipt of the product and prior to initial operation, read these instructions thoroughly, and retain for future reference.

REFERENCE

YASNAC J300L OPERATING MANUAL TOE-C843-13.20

YASKAWA

MANUAL NO. TOE-C843-13.21

FOREWORD

This manual gives the information necessary for creating a program using the YASNAC J300L (with basic NC operation panel, 9-inch CRT).

Some information is given in tables in the Appendix so that readers can easily find the necessary information. In the G code table, section numbers are given for each G code to allow quick access to a detailed explanation if necessary.

The YASNAC J300L comes with an operation manual in addition to this programming manual. Use these manuals in conjunction with each other to ensure productive operation.

CAUTIONS

This manual describes all the option functions (identified by the "*" symbol) but some of these may not be available with your YASNAC J300L. To determine the option functions installed in your NC, refer to the specification document or manuals published by the machine tool builder.

Unless otherwise specified, the following conditions apply in programming explanations and programming examples.

- Metric system for input and metric system for output/movement
- ⯐ : Zero point in the base coordinate system
- ⯐ : Reference point

Yaskawa has made every effort to describe individual functions and their relationships to other functions as accurately as possible. However, there are many things that cannot or must not be performed and it is not possible to describe all of these. Accordingly, readers are requested to understand that unless it is specifically stated that something can be performed, it should be assumed that it cannot be performed.

Also bear in mind that the performance and functions of an NC machine tool are not determined solely by the NC unit. The entire control system consists of the mechanical system, the machine operation panel and other machine related equipment in addition to the NC. Therefore, read the manuals published by the machine tool builder for detailed information relating to the machine.

General Precautions

- Some drawings in this manual are shown with the protective cover or shields removed, in order to describe the detail with more clarity. Make sure all covers and shields are replaced before operating this product, and operate it in accordance with the directions in the manual.
- The figures and photographs in this manual show a representative product for reference purposes and may differ from the product actually delivered to you.
- This manual may be modified when necessary because of improvement of the product, modification, or changes in specifications. Such modification is made as a revision by renewing the manual No.
- To order a copy of this manual, if your copy has been damaged or lost, contact your Yaskawa representative listed on the last page stating the manual No. on the front page.
- If any of the nameplates affixed to the product become damaged or illegible, please send these nameplates to your Yaskawa representative.
- Yaskawa is not responsible for any modification of the product made by the user since that will void our guarantee.

NOTES FOR SAFE OPERATION

Read this programming manual thoroughly before installation, operation , maintenance or inspection of the YASNAC J300L.

The functions and performance as NC machine tool are not determined only by an NC unit itself. Before the operation, read thoroughly the machine tool builder's documents relating to the machine tool concerned.

In this manual, the NOTES FOR SAFE OPERATION are classified as "WARNING" or "CAUTION".

⚠ WARNING	Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury to personnel. Symbol ⚡ is used in labels attached to the product.
⚠ CAUTION	Indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury to personnel and damage to equipment. It may also be used to alert against unsafe practice.

Even items described in ⚠ CAUTION may result in a vital accident in some situations. In either case, follow these important items.

Please note that symbol mark used to indicate caution differs between ISO and JIS.

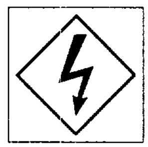
ISO	JIS
⚠	◇!

In this manual, symbol mark stipulated by ISO is used.

On products, caution symbol marks of ISO and JIS are used in labels. Please follow the same safety instructions concerning caution.

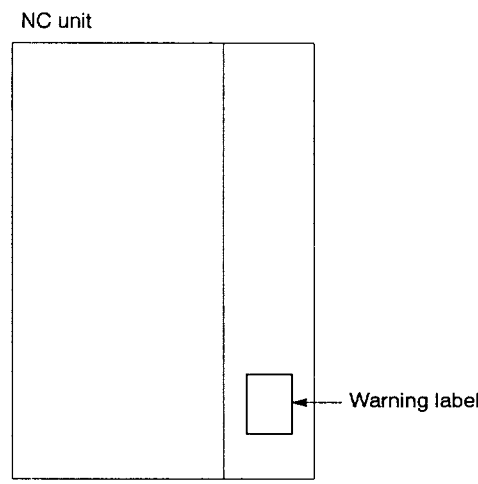
KEY TO WARNING LABELS

The following warning labels are used with the YASNAC J300L.

⚠ WARNING : 

Electric shock hazard
Do not touch the terminals while the power is on, and for 5 minutes after switching off the power supply!

Location of label

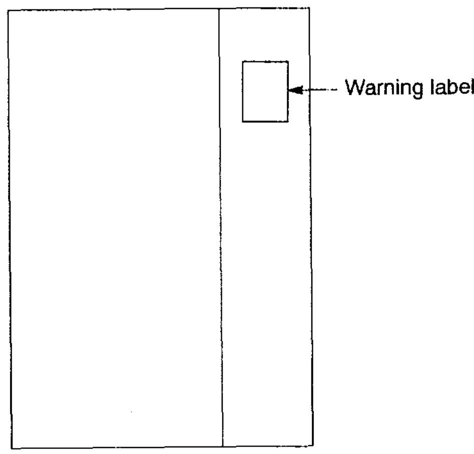


⚠ CAUTION :

Grounding wires must be connected to the unit's grounding terminals.

Location of label

NC unit

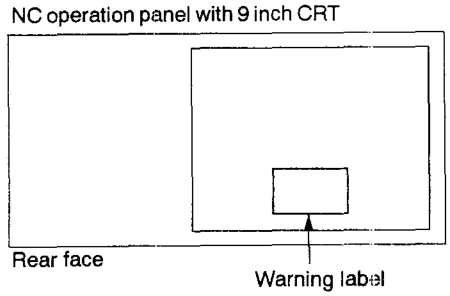


⚠ WARNING :

Electric shock hazard
Do not touch inside.

Location of label

NC operation panel with 9 inch CRT



CONTENTS

3. MOVEMENT CONTROL COMMANDS

4. ENHANCED LEVEL COMMANDS

APPENDIX 1 G CODE TABLE

APPENDIX 2 INDEX

1

1 PROGRAMMING BASICS

Chapter 1 describes the basic terms used in programming and the feed functions.

1.1 FUNDAMENTALS OF PROGRAMMING TERMINOLOGY

This section describes the basic terms used in programming.

1.1.1 Numerically Controlled Axes and the Number of Simultaneously Controllable Axes

The numerically controlled axes and the number of axes that can be controlled simultaneously are indicated in Table 1.1.

Table 1.1 Numerically Controlled Axes and the Number of Simultaneously Controllable Axes

			Description
Controlled axes	Series 1 control	Basic axes	X and Z
		Additional axis control A*	X and Z + C
		Additional axis control B*	Expandable to 5 axes (Y-axis, B-axis, etc.)
Number of simultaneously controllable axes	Positioning (G00)		All axes
	Linear interpolation (G01)		All axes
	Circular interpolation (G02, G03)		2 axes
	Manual operation		All axes

Note 1: For polar coordinate interpolation* and cylindrical interpolation*, circular interpolation is possible on virtual XC or ZC plane. For details, see 2.1.7, "Polar Coordinate Interpolation" and 2.1.6, "Cylindrical Interpolation".

2: With a manual pulse generator, only one axis control is possible.

1.1.2 Least Input Increment and Least Output Increment

1

The least input and output increments vary depending on the type of controlled axis whether it is a rotary axis or a linear axis.

(1) Least Input Increment and 10-time Input Increment

The least input increment to express axis movement distance that is input by using punched tape or manual data input switches is indicated in Table 1.2.

Table 1.2 Least Input Increment (pm1000 D0 = 0)

	Linear Axes (X-, Y-, Z-axis, etc.)	* C-axis
Metric Input	0.001 mm	0.001 deg.
Inch Input	0.0001 inch	0.001 deg.

By setting "1" for parameter pm1000 D0 (pm1000 D0 = 1), the "10-time input increment" specifications indicated in Table 1.3 is selected.

Table 1.3 10-time Input Increment (pm1000 D0 = 1)

	Linear Axes (X-, Y-, Z-axis, etc.)	* C-axis
Metric Input	0.01 mm	0.01 deg.
Inch Input	0.001 inch	0.01 deg.

Note: Selection of "mm-input" and "inch-input" is made by the setting parameter pm0007 D0 or by the specification of G20/G21.

Disregarding of the least input increment mode which has been selected, tool offset data are always written in units of 0.001 mm (or 0.0001 inch, or 0.001 deg.). Offset movement is possible in the specified value. If the offset data are set in units of 0.01 mm, the following operations and the commands for them must be given in units of 0.01 mm.

- Data writing in the MDI mode
- Programming for the memory mode operation
- Program editing

SUPPLEMENT

1. If an NC program written in units of 0.001 mm is executed while the 0.01 mm setting increment is selected, dimension commands are all executed 10 times the specified value.
2. If the program stored in memory is executed in the memory mode after changing the setting for pm1000 D0 (input increment setting parameter), dimension commands in the stored program are executed in either 1/10 or 10 times the specified value.
3. When a program stored in memory is output to a tape, the stored program is output as it is and not influenced by the setting for pm1000 D0 (input increment setting parameter).

(2) Least Output Increment

The least output increment indicates the "minimum unit" of axis movement that is determined by the mechanical system. By selecting the option, it is possible to select the output unit system between "mm" and "inches".

Table 1.4 Least Output Unit (pm1000 D0 = 1)

	Linear Axes (X-, Y-, Z-axis, etc.)	* C-axis
Metric Output	0.001 mm	0.001 deg.
Inch Output	0.0001 inch	0.001 deg.

1.1.3 Maximum Programmable Values for Axis Movement

The maximum programmable values that can be designated for a move command are indicated in Table 1.5. The maximum programmable values indicated in these tables are applicable to addresses I, J, K, R, A, and B which are used for designating "distance" in addition to the move command addresses X, Y, Z, C, U, W, V, and H.

Table 1.5 Maximum Programmable Values for Axis Movement

		Linear Axes (X-, Y-, Z-axis, etc.)	* C-axis
Metric Output	Metric Input	± 999999.999 mm	± 999999.999 deg.
	Inch Input	± 39370.0787 inch	± 999999.999 deg.
Inch Output	Metric Input	± 999999.999 mm	± 999999.999 deg.
	Inch Input	± 99999.9999 inch	± 999999.999 deg.

In incremental programming, the values to be designated must not exceed the maximum programmable values indicated above. In absolute programming, the move distance of each axis must not exceed the maximum programmable values indicated above. In addition to the notes indicated above, it must also be taken into consideration that the cumulative values of move command must not exceed the values indicted in Table 1.6.

Table 1.6 Maximum Cumulative Values

	Linear Axes (X-, Y-, Z-axis, etc.)	* C-axis
Metric Input	± 999999.999 mm	± 999999.999 deg.
Inch Input	± 999999.9999 inch	± 999999.999 deg.

Note: The values indicated above do not depend on the "least output increment".

1.1.4 Tape Format

1

The following describes the important items concerning the tape format.

(1) Label and Label Skip

By entering "label" at the beginning of a punched tape, classification and handling of tape can be facilitated.

The label skip function disregards the data appearing before the first EOB code. With this feature, label can contain address characters and function codes which are not supported by the NC. A code that does not match the selected parity scheme can also be used. The label skip function becomes enabled when the power is turned ON or when the NC is reset. While the label skip function is enabled, "LSK" message is displayed on the screen.

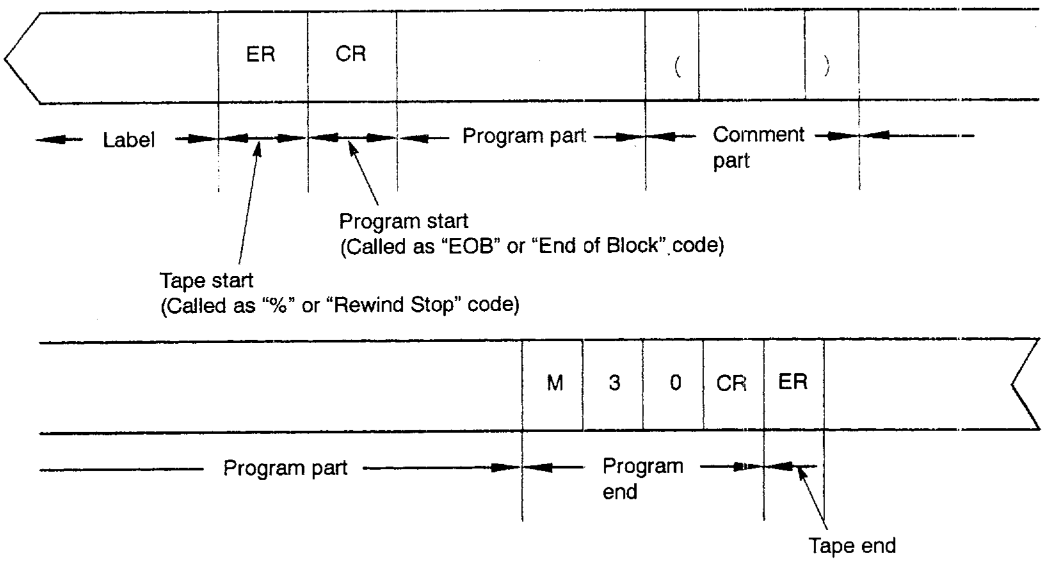
(2) Tape Start and Tape End

At the start and end of a tape, the same code (see Table 1.7) should be punched.

Table 1.7 Tape Start and Tape End

EIA	ISO	Description
ER	%	Tape start/Tape end

- The ER code (rewind stop code) entered following the tape start label indicates the rewind stop when the tape is rewound by the tape rewind command.
- The ER code, expressing the tape end, indicates the stop point when several part programs are stored in NC memory.



Note: As the end of program code, M02 or M99 can be used instead of M30. Whether or not the M codes indicated above are used as the program end M code is determined according to the setting for parameter pm3005 D3.

Fig. 1.1 Single Main Program Punched on Tape (EIA Code)

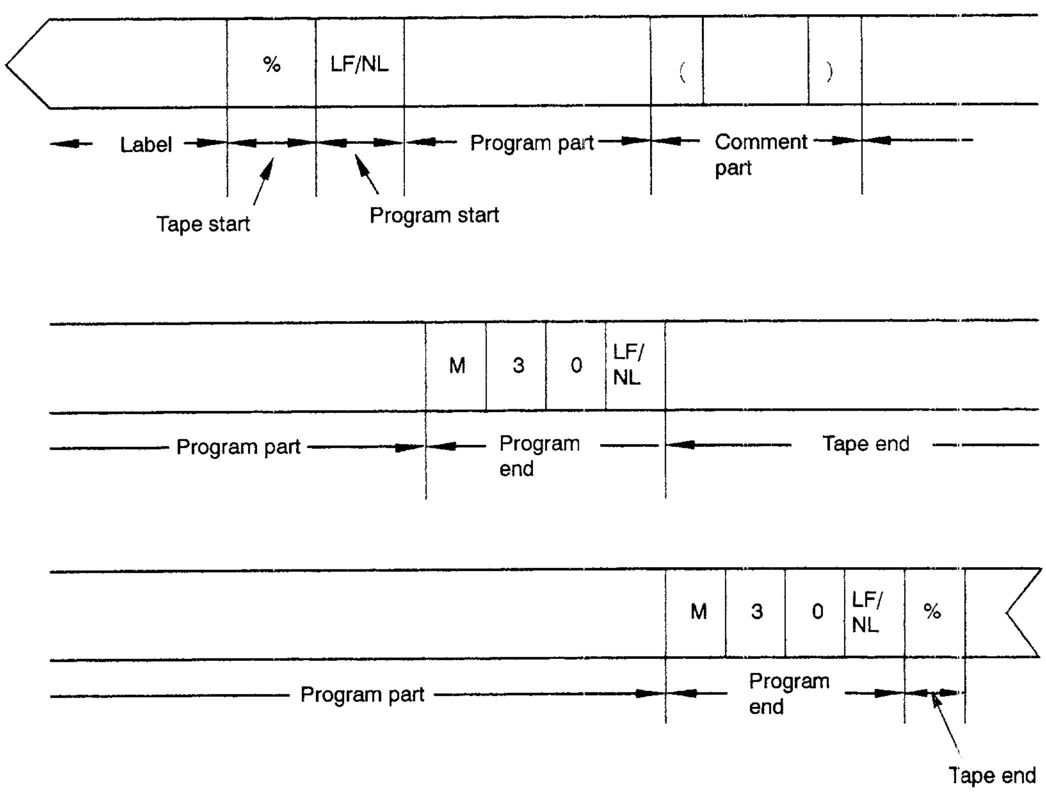


Fig. 1.2 Multiple Programs Punched on Tape (EIA Code)

(3) Program Start and Program End

(a) Program start

When punching a program on a tape, the following code should be punched to declare the beginning of a program. This code cancels the label skip function.

Table 1.8 Program Start

EIA	ISO	Description
CR	LF/NL	Program start

(b) Program end

Any of the following codes indicated in Table 1.9 should be punched at the end of a program to declare the program end.

Table 1.9 Program End

EIA	ISO	Description
M02CR	M02LF/NL	Program end
M30CR	M30LF/NL	Program end and rewind
M99CR	M99LF/NL	Subprogram end

Note 1: When "M02CR" or "M30LF/NL" is executed, the equipment may or may not be reset or rewound depending on equipment specifications.

Refer to the manual published by the machine tool builder.

2: When multiple part programs are started in the NC memory, control may move to the next part program after reading the program end code shown above.

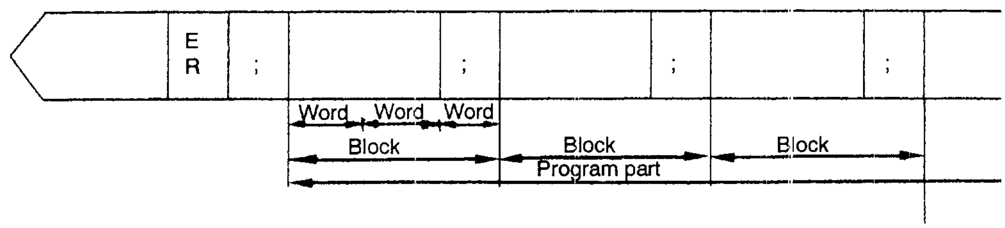
This occurs when part programs are entered by total input.

3: If ER or LF/NL code is executed for a program in which neither M02 nor M30 is entered at the end of the program; the NC is reset.

1.1.5 Program Format

(1) Program Part

The section beginning with the program start code and ending with the program end code is called the program part. The program part consists of blocks, and each block consists of words.



Note: In this manual, the "EOB" code is expressed by a semi-colon (;).

Fig. 1.3 Construction of Program

(a) Program number

By entering a program number immediately after the program start code, it is possible to distinguish a specific program from other programs. A program number consists of address O and a maximum of 5-digit number that follows address O. The NC memory has a capacity to store a maximum of 99 programs; this capacity can be optionally increased to store up to 299 or 999 programs.

(b) Sequence number

A sequence number, consisting of address N and a maximum of 5-digit integer that follows address N, can be entered at the beginning of a block. Sequence numbers are used only for reference numbers of blocks and do not influence the contents and execution order of machining processes. Therefore, sequential or non-sequential numbers may be used for sequence numbers. It is also allowed to leave blocks without assigning sequence numbers. In addition, the same sequence number may be assigned to different blocks. Although there are no restrictions on using sequence numbers, it is recommended to assign sequence numbers in a sequential order. Before executing the sequence number search, it is necessary to execute the program number search to determine the program in which sequence number search should be executed.

SUPPLE-MENT

1. If a sequence number consisting of 6 of more digits is designated, 5 digits from the least insignificant digit are regarded as a sequence number.
2. If address search is executed for a sequence number which is assigned to more than one block, the block searched first is read and search processing is completed at that block.
3. For blocks for which a sequence number is not assigned, search is possible by the address search operation if address data in the block to be searched are designated as the object of address search operation.
4. When designating a sequence number following G25 or M99, designate a 4-digit number.

(c) Word

A word consists of an address character included in the function characters and a numeral of several digits that follow the address character. For example, word "G02" consists of address character "G" and numeral "2".

The function character means a character that can be used in the significant data area. For details of address character and function character codes, refer to Tables 1.10 and 1.11.

1

Table 1.10 Table of Address Characters

Address	Description	Category Note
A	Designation of angle for G01 and G111, Designation of thread angle for G76	O
B	Designation of spindle shift angle for multiple thread cutting operation Designation of angle for multiple chamfering and rounding	O
C	C-coordinate	O
D	Designation of depth and number of cuts for G71 to G76	O
E	Designation of precision feed, Designation of precision lead in thread cutting	B
F	Designation of ordinary feed, Designation of ordinary lead in thread cutting	B
G	Preparatory function	B
H	Incremental command of C-axis	O
I	X-coordinate of center of arc, Canned cycle parameter data, Chamfer size (radius)	B, O
J	Y-coordinate of center of arc	O
K	Z-coordinate of center of arc, Canned cycle parameter data, Chamfer size	B, O
	Increment/decrement amount in variable-lead thread cutting	O
L	Number of repetitions	B, O
M	Miscellaneous function	B
N	Sequence number	B
O	Program number	B
P	Dwell time, Designation of the first sequence number of a canned cycle, program number, and macro program number	B, O
Q	Designation of the first sequence number of a subprogram and the end sequence number of a canned cycle	B, O
	Depth of cut in a hole-machining canned cycle	O
R	Radius of an arc, Amount of rounding, Nose-R amount, Point R coordinate in a hole-machining canned cycle	B, O
S	Spindle function, Clamp spindle speed	B
T	Tool function, Tool coordinate memory number	B, O
U	Incremental command of X-axis, Dwell time, Canned cycle parameter	B, O
V	Incremental command of Y-axis	O
W	Incremental command of Z-axis, Canned cycle parameter	B, O
X	X-coordinate	B
Y	Y-coordinate	O
Z	Z-coordinate	B

Note: B: Basic, O: Option

Table 1.11 Table of Function Characters

EIA code	ISO code	Description	Remarks
Blank	NUL	EIA: Error if designated in the significant information area ISO: Disregarded	
BS	BS	Disregarded	
Tab	HT	Disregarded	
CR	LF/NL	End of block (EOF)	
–	CR	Disregarded	
SP	SP	Space	
ER	%	Rewind stop	
UC	–	Upper case	
LC	–	Lower case	
2-4-5 bits	(	Control out (Comment start)	EIA: Special code
2-4-7 bits	)	Control in (Comment end)	
+	+	Disregarded, User macro operator	
–	–	Minus sign, User macro operator	
0 - 9	0 - 9	Numerals	
A - Z	A - Z	Address characters	
/	/	Optional block skip User macro operator	
Del	DEL	Disregarded (includes all punched holes)	
.	.	Decimal point	
Parameter setting	#	Symbol of sharp (Variable)	EIA: Special code
*	*	Asterisk (Multiplication operator)	
=	=	Equal symbol	
[	[	Left bracket	
]	]	Right bracket	
O	:	For comment in macro program	
$	$	For comment in macro program	
@	@	For comment in macro program	
?	?	For comment in macro program	
,	,	For comment in macro program	

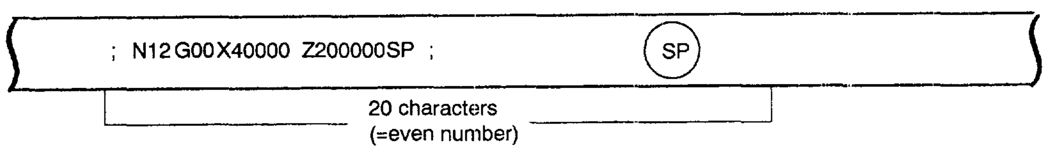
Note 1: If a code not indicated above is designated in the significant information area, it causes an error.

2: Information designated between the control out and control in codes is regarded as insignificant information.

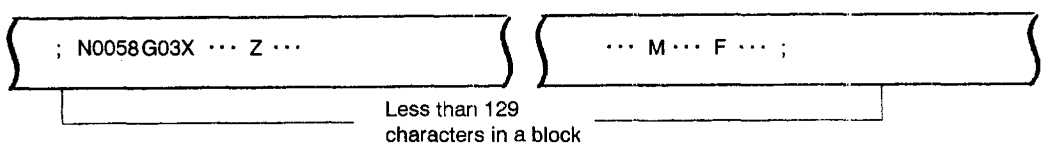
3: Input code (EIA/ISO) is automatically recognized, and output code is determined by the setting for parameter pm0004 D0.

(d) Block

- A block consists of words to define a single step of operation. One block ends with the EOB (end of block) code. The EOB code is expressed by "CR" in the EIA code system and "LF/NL" in the ISO code system.
  In this manual, it is expressed by a semicolon ";" to make the explanation simple.
- Characters not indicated in Tables 1.10 "Table of Address Characters" and 1.11 "Table of Function Characters" must not be used.
- One block can contain up to 128 characters. Note that invalid characters such as "Del" are not counted.



(a) Adding a character for TV check (an error occurs if an even number of characters is contained in a block.)



(b) Number of valid characters allowed in a block

Fig. 1.4 Block

(2) Comment Part

A comment can be displayed by using the control out and control in codes.

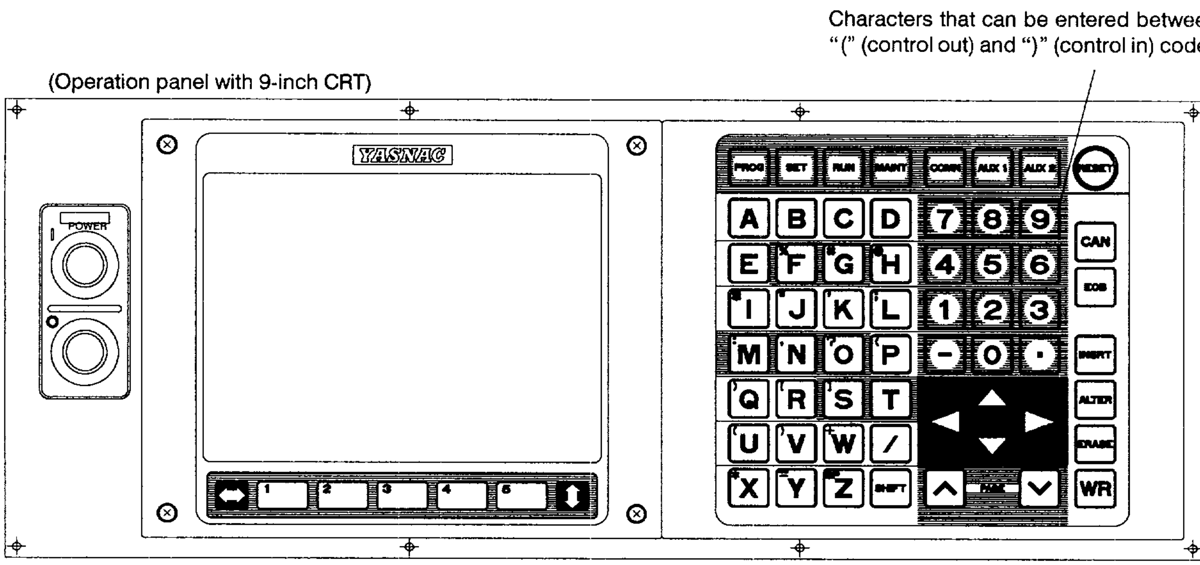
(a) Entering a comment in a program

It is possible to display a required comment on the screen by enclosing it with the control out and control in codes in a part program. The information enclosed by these codes is regarded as insignificant information.

(b) Entering the control out and control in codes

The control out and control in codes can be entered in the same manner as entering ordinary characters.

- "(":Press the [U] key after pressing the [SHIFT] key.
- ")":Press the [V] key after pressing the [SHIFT] key.



Note 1: The characters that can be entered between the control out and control in codes are those that are entered by using the keys enclosed by dark line in Fig. 1.5.

2: It is not allowed to use the control out and control in codes in the area which are already enclosed by the control out and control in codes.

Fig. 1.5 Characters that can be Entered between Control Out and Control In Codes (Keys Enclosed by Dark Line)

<Example of comment display by using the control out and control in codes>

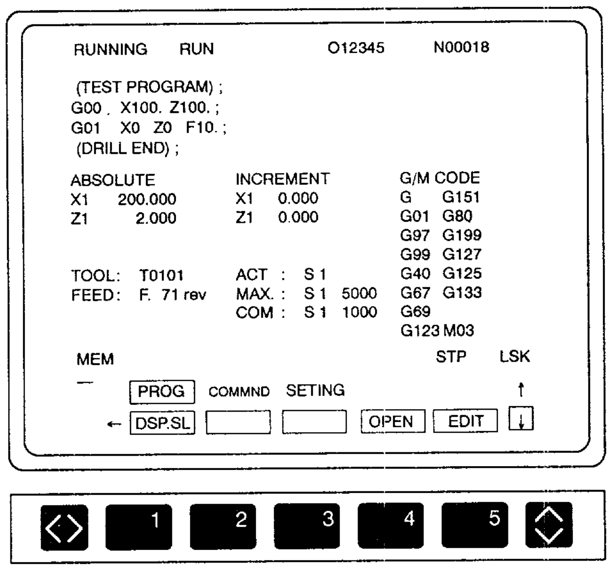


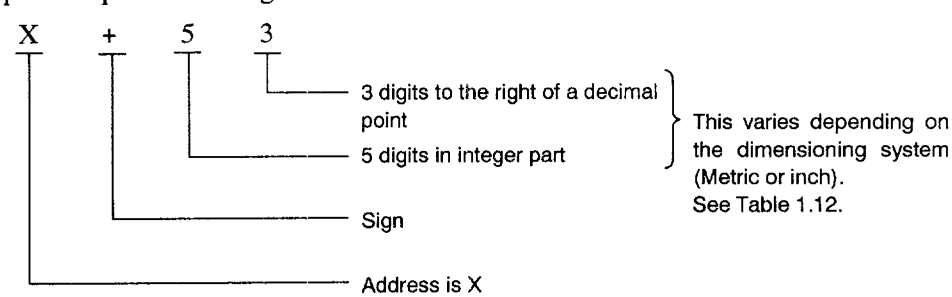
Fig. 1.6 Program Execution Display Screen

(3) Programmable Range (Input Format)

This model of NC adopts the variable block format which complies with JIS B6313.

Programmable range of individual addresses is indicated in Table 1.12. The numbers given in this table indicate the allowable maximum number of digits.

An example of input format is given below.



Input data should be entered without a decimal point. If a decimal point is used, the entered value is treated in a different manner. Leading zeros and the "+" (plus) sign can be omitted for all kinds of address data including sequence numbers. Note that, however, the "–" (minus) sign cannot be omitted.

Table 1.12 Input Format

Address		Metric Output		Inch Output		B: Basic O: Option
		Metric Input	Inch Input	Metric Input	Inch Input	
Program number		O5		O5		B
Sequence number		N5		N5		B
G function		G3		G3		B
Coordinate words	Linear axis (X, Z, I, K, U, W, R, Q, Y, J)	a+63	a+54	a+63	a+54	B, O
	Rotary axis (C, H)	b+63		b+63		O
Feed per minute (mm/min) function		F60 or F63	F52 or F54	F60 or F63	F52 or F54	B
Feed per revolution and thread lead		F33	F24	F33	F24	B
		F34	F26	F34	F26	B
S function		S5		S5		B
T function		T (2 + 2)		T (2 + 2)		B
		T (3 + 3)		T (3 + 3)		O
M function		M3		M3		B
Dwell		U (P) 63		U (P) 63		B
Program number designation		P5		P5		B
Sequence number designation		Q (P) 5		Q (P) 5		B, O
Number of repetitions		L9		L9		B
Designation of angle of line		A (B) 33		A (B) 33		O
Designation of multiple-thread angle		B3		B3		O

Note: The input format for "feed per minute" is set by using parameter pm2004 D0.

pm2004 D0 = 0	F60 mm/min, F52 inch/min
pm2004 D0 = 0	F63 mm/min, F54 inch/min

1.1.6 Optional Block Skip (/1), (/2 to /9) *

If a block containing the slash code "/n (n=1 to 9)" is executed with the external optional block skip switch corresponding to the designated number set ON, the commands in the block following the slash code to the end of block code are disregarded. The slash code "/n" can be designated at any position in a block.

Example:

```
/2  N 1234  G00X100  /3  Z200;
```

If the "/2" switch is ON, the entire block is disregarded, and
if "/3" switch is ON, this block indicates the following.

```
N 1234      G00X100;
```

SUPPLEMENT

1. "1" can be omitted for "/1".
2. The optional block skip function is processed when a part program is read to the buffer register from either the tape or memory. If the switch is set ON after the block containing the optional block skip code is read, the block is not skipped.
3. The optional block skip function is disregarded for program reading (input) and punch out (output) operation.

1.1.7 Buffer Register and Multi-active Register

1

By using the buffer register and multi-active register, the NC ensures smooth control of the machine by reading the blocks of data into the buffer register.

(1) Buffer Register

In normal operation, two blocks of data are buffered to calculate the offset and other data that are necessary for the succeeding operation.

In the nose R offset mode (option), two blocks of data (a maximum of four blocks of data, if necessary) are buffered to calculate the offset data that are necessary for the succeeding operation. In both of the normal operation mode and nose R offset mode, the data capacity of one block is a maximum of 128 characters, including the EOB code.

(2) Multi-active Registers *

With a part program enclosed by M93 and M92, a maximum of seven blocks of data are buffered. If the time required for automatic operation of these seven buffered blocks is longer than the time required for the buffering and calculation of the offset data for the next seven blocks, the program can be executed continuously without a stop between blocks.

Table 1.13 M92 and M93 Codes

M Code	Function
M93	Multi-active registers ON
M92	Multi-active registers OFF

1.2 BASICS OF FEED FUNCTION

This section describes the feed function that specifies feedrate (distance per minute, distance per revolution) of a cutting tool.

1.2.1 Rapid Traverse

1

Rapid traverse is used for positioning (G00) and manual rapid traverse (RAPID) operation. In the rapid traverse mode, each axis moves at the rapid traverse rate set for the individual axes; the rapid traverse rate is determined by the machine tool builder and set for the individual axes by using parameters. Since the axes move independently of each other, the axes reach the target point at different time. Therefore, the resultant tool paths are not a straight line generally.

The rapid traverse override function can adjust the set rapid traverse rate to F_0, 25%, 50%, and 100% where F_0 indicates a fixed feedrate set for parameter pm2447.

SUPPLEMENT

1. Rapid traverse rate is set in the following units for the individual axes.

   Setting units of rapid traverse rate 0.001 mm/min
   or
   1 deg./min

2. The upper limit of the rapid traverse rate is 240,000 mm/min. Since the most appropriate value is set conforming to the machine capability, refer to the manuals published by the machine tool builder for the rapid traverse rate of your machine.

1.2.2 Cutting Feed (F Command)

1

The feedrate at which a cutting tool should be moved in the linear interpolation (G01) mode or circular interpolation (G02, G03) mode is designated using address characters F and E. The axis feed mode to be used is selected by designating the feed function G code (G98 or G99) as indicated in Table 1.14. Select the required feed mode by designating the feed function G code before specifying an F and E code.

Table 1.14 Cutting Feed Mode G Codes

G Code	Function	Group
G98	Designation of feed per minute (mm/min) mode	10
G99	Designation of feed per revolution (mm/rev) mode	10

See 1.2.3 "Switching between Feed per Minute Mode and Feed per Revolution Mode" for details of these G codes. F and E codes are modal and once designated they remain valid until another F or E code is designated. If feed mode designation G codes are switched between G98 and G99, however, it is necessary to designate the F and E code again. If no new F and E codes are designated, alarm "0370" occurs. Note that it is not allowed to designate an E code in the G98 (feed per minute) mode. If an E code is designated in the G98 mode, alarm "0371" occurs.

(1) Feed per Revolution Mode (G99)

A feedrate of a cutting tool per revolution of the spindle (mm/rev, inch/rev) can be designated by a numeral specified following address character F or E.

Table 1.15 Programmable Range of F and E Commands (Feed per Revolution Mode)

		Format	Programmable Range
mm output	mm input	F33	F0.001 to F500.000 mm/rev
		E34	E0.0001 to E500.0000 mm/rev
	inch input	F24	F0.0001 to F19.6850 inch/rev
		E26	E0.000001 to E19.685000 inch/rev
inch output	mm input	F33	F0.001 to F1270.000 mm/rev
		E34	E0.0001 to E1270.0000 mm/rev
	inch input	F24	F0.0001 to F50.0000 inch/rev
		E26	E0.000001 to E50.00000 inch/rev

Note 1: The allowable maximum value for the X-axis is 1/2 of the value indicated in the table.

2: The upper limit of feedrates could be restricted by the servo system and the mechanical system. For the actual programmable feedrate range, refer to the manuals published by the machine tool builder.

The feedrate per revolution is further restricted as indicated in Table 1.16 due to spindle speed S.

Table 1.16 Restrictions on F and E Commands by Spindle Speed

	Restriction
mm output	$F (E) \times S \leqq 240{,}000$ mm/min
inch output	$F (E) \times S \leqq 24{,}000$ inch/min

An F command specified in the simultaneous 2-axis linear interpolation mode or in the circular interpolation mode represents the feedrate in the tangential direction.

Example of Programming (linear interpolation mode)

With the following program:
G99 S1000 (r/min);
G01 U60. W40. F0.5;

$F \times S = 0.5$ mm/rev $\times$ 1000 r/min
= 500 mm/min
$= \sqrt{300^2 + 400^2}$

(300: X-axis component; 400: Z-axis component)

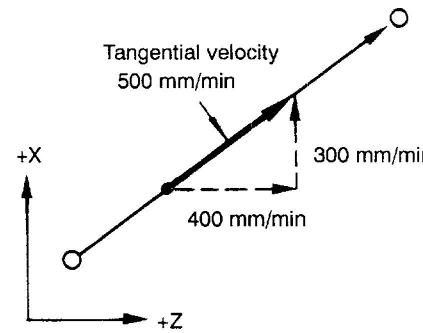
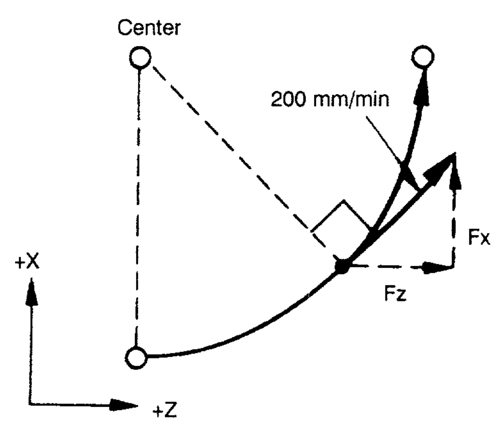


Fig. 1.7 F Command in Simultaneous 2-axis Control Linear Interpolation (Feed per Revolution)

Example of Programming (circular interpolation mode)

With the following program:
G99 S1000 (r/min);
G03 U · · · W · · I · · · F0.2;

$F \times S = 0.2$ mm/rev $\times$ 1000 r/min
= 200 mm/min
$= \sqrt{Fx^2 + Fz^2}$



Note 1: An F0 command causes an input error.

2: A feedrate in the X-axis direction is determined by the radial value.

Fig. 1.8 F Command in the Simultaneous 2-axis Control Circular Interpolation (Feed per Revolution)

SUPPLEMENT

Do not specify a negative value for an F command. An F command with a negative value causes alarm "0102".

(2) Feed per Minute Mode (G98)

A feedrate of a cutting tool per minute (mm/min, inch/min) can be designated by a numeral specified following address character F. It is possible to set the F60 format and F63 format (mm input) by the setting for parameter pm2004 D0. The programmable range is indicated in Table 1.17.

Table 1.17 Programmable Range of F Commands (Feed per Minute Mode)

			Format	Programmable Range (Linear Axis)	Programmable Range (Rotary Axis)
pm2004 D0 = 0	mm output	mm input	F60	F1 to F240000 mm/min	F1 to F240000 deg/min
		inch input	F52	F0.01 to F94488.18 inch/min	F0.01 to F240000.00 deg/min
	inch output	mm input	F60	F1 to F609600 mm/min	F1 to F240000 deg/min
		inch input	F52	F0.01 to F24000.00 inch/min	F0.01 to F240000.00 deg/min
pm2004 D0 = 1	mm output	mm input	F63	F0.001 to F240000.000 min/min	F0.001 to F240000.000 deg/min
		inch input	F54	F0.0001 to F94488.1890 inch/min	F0.001 to F240000.0000 deg/min
	inch output	mm input	F63	F0.001 to F609600.000 mm/min	F0.001 to F240000.000 deg/min
		inch input	F54	F0.0001 to F24000.0000 inch/min	F0.0001 to F24000.0000 deg/min

Note 1: The allowable maximum value for the X-axis is 1/2 of the value indicated in the table.

2: The upper limit of feedrates could be restricted by the servo system and the mechanical system. For the actual programmable feedrate range, refer to the manuals published by the machine tool builder.

(3) Simultaneous 2-axis Control

An F command specified in the simultaneous 2-axis linear interpolation mode or in the circular interpolation mode represents the feedrate in the tangential direction.

Example of Programming (linear interpolation mode)

With the following program:
G98;
G01 U60. W40. F500.;

$$\underset{(mm/min)}{F} = 500 = \sqrt{\underbrace{300^2}_{\text{X-axis component}} + \underbrace{400^2}_{\text{Z-axis component}}}$$

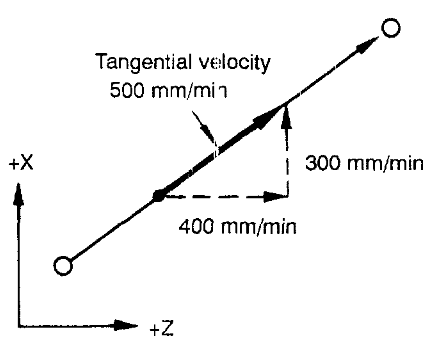
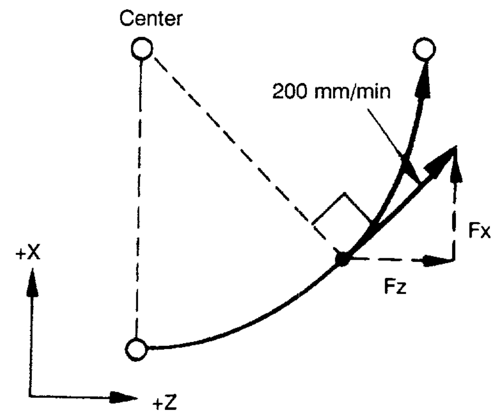


Fig. 1.9 F Command in Simultaneous 2-axis Control Linear Interpolation (Feed per Minute)

Example of Programming (circular interpolation mode)

With the following program:
G98;
G03 X · · · Z · ·   I · · · F200.;

$F = 200 = \sqrt{Fx^2 + Fz^2}$
(mm/min)



Note 1: An F0 command causes an input error.

2: A feedrate in the X-axis direction is determined by the radial value.

Fig. 1.10 F Command in the Simultaneous 2-axis Control Circular Interpolation (Feed per Minute)

SUPPLEMENT

Do not specify a negative value for an F command. An F command with a negative value causes alarm “0102”.

(4) Rotary Axis and Linear Axis

An F command specified in the interpolation mode between a rotary axis and a linear axis represents the feedrate in the tangential direction.

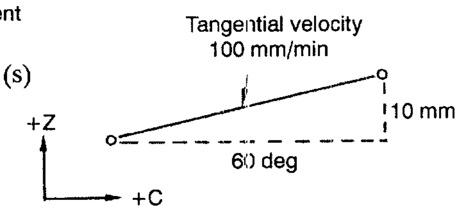
Example of Programming

G98;
G01 W10. H60. F100.;

- mm input (F60)

$$\text{Distance} = \sqrt{\underbrace{10000^2}_{\text{Z-axis component}} + \underbrace{60000^2}_{\text{C-axis component}}} = 60827.625$$

$$\text{Time} = \frac{60827.625}{1000000} = 0.6082\ (\text{min}) = 36.5\ (\text{s})$$



- inch input (F52)

$$\text{Distance} = \sqrt{\underbrace{100000^2}_{\text{Z-axis component}} + \underbrace{60000^2}_{\text{C-axis component}}} = 1166190.0379$$

$$\text{Time} = \frac{1166190.0379}{1000000} = 0.1166\ (\text{min}) = 6.9\ (\text{s})$$

Fig. 1.11 F Command in Interpolation between Rotary Axis and Linear Axis (Feed per Minute)

(5) Independent Rotary Axis Command

If a rotary axis command is specified independently, feedrate is determined according to the selected input increment system. In the case of inch input system, the unit of feedrate is determined by the setting for parameter.

Table 1.18

		Format	pm2004 D7 = 0	pm2004 D7 = 1
pm2004 D0 = 0	mm input	F60	1 = 1 deg/min	1 = 1 deg/min
	inch input	F52	1 = 0.1 deg/min	1 = 0.01 deg/min
pm2004 D0 = 1	mm input	F63	1 = 0.001 deg/min	1 = 0.001 deg/min
	inch input	F54	1 = 0.001 deg/min	1 = 0.0001 deg/min

1.2.3 Switching between Feed per Minute Mode and Feed per Revolution Mode (G98/G99)

Before specifying a feedrate command (E, F), a G code that determines whether the specified feedrate command is interpreted as feed per minute value or feed per revolution value should be specified. These G codes (G98, G99) are modal and once they are specified they remain valid until the other G code is specified. When the feed mode designation G code is specified, the presently valid E and F codes are canceled. Therefore, an E and F code must be specified newly after switching the feed mode by designating G98 or G99 command. The initial status that is established when the power is turned on is set by parameter pm4000.

Table 1.19 Parameter pm4000 and Initial Status

Parameter	Initial G Code
pm4000 D2 = 0	G98
pm4000 D2 = 1	G99

(1) Feed per Minute Mode (G98)

By specifying “G98;”, the F codes specified thereafter are all executed in the feed per minute mode.

Table 1.20 Meaning of G98 Command

G98	Meaning
mm input	mm/min
inch input	inch/min

(2) Feed per Revolution Mode (G99)

By specifying “G99;”, the F codes specified thereafter are all executed in the feed per revolution mode.

Table 1.21 Meaning of G99 Command

G99	Meaning
mm input	mm/rev
inch input	inch/rev

1.2.4 Automatic Acceleration and Deceleration

Automatic acceleration/deceleration control is provided for rapid traverse and cutting feed operation, respectively.

1

(1) Acceleration and Deceleration for Rapid Traverse and Manual Axis Feed Operation

For positioning (G00), manual rapid traverse (RAPID), manual continuous feed (JOG), and manual handle feed (HANDLE), linear pattern automatic acceleration/deceleration is applied. Rapid traverse rate and acceleration/deceleration time constant for rapid traverse are set for following parameters.

Table 1.22 Parameters Used for Setting Rapid Traverse Rate and Acceleration/Deceleration Time Constant

	X-axis	Z-axis	3rd-axis	4th-axis	5th-axis
Rapid traverse rate	pm2801	pm2802	pm2803	pm2804	pm2805
Acceleration/deceleration time constant	pm2461	pm2462	pm2463	pm2464	pm2465

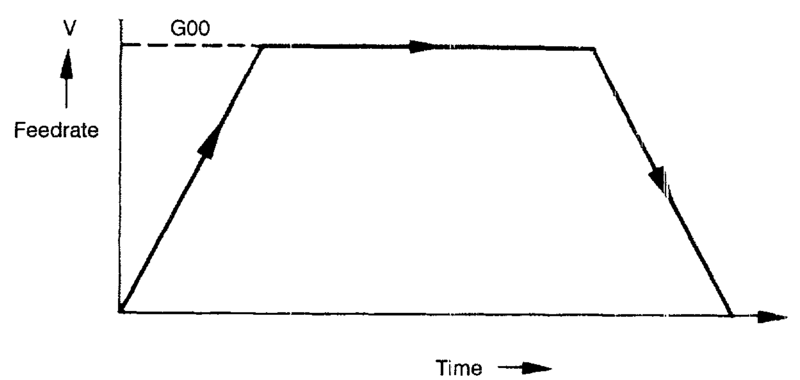


Fig. 1.12 Automatic Acceleration/Deceleration in Linear Pattern

1

(2) Acceleration and Deceleration for Cutting Feed

For cutting feed (G01 to G03 mode), feedrate is controlled by the automatic acceleration/deceleration in the exponential pattern.

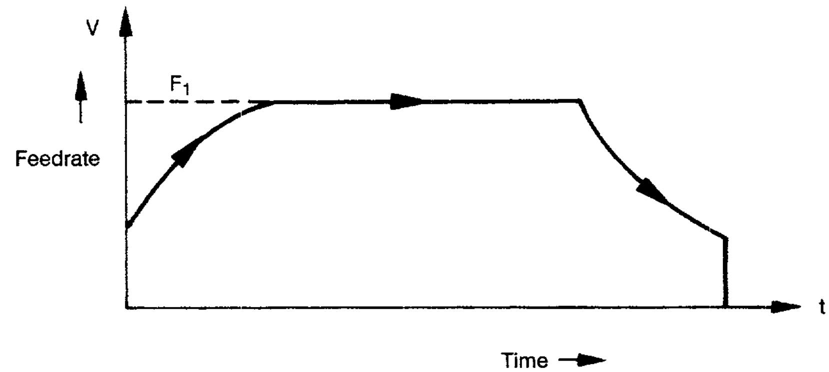


Fig. 1.13 Acceleration/Deceleration in Exponential Pattern

Time constant for cutting feed and feedrate bias are set for parameters. For tapping, time constant and feedrate bias can be set independently.

Table 1.23 Parameters for Tapping

	X-axis	Z-axis	3rd-axis	4th-axis	5th-axis
Feedrate time constant	pm2501	pm2502	pm2503	pm2504	pm2505
Feedrate bias	pm2821	pm2822	pm2823	pm2824	pm2825
Tapping time constant	pm2511	pm2512	pm2513	pm2514	pm2515
Tapping feedrate bias	pm2831	pm2832	pm2833	pm2834	pm2835

SUPPLEMENT

For the parameters indicated above, the most optimum values are set for respective machines. Do not attempt to change the setting unless necessary.

2

COMMANDS CALLING AXIS MOVEMENTS

2

Chapter 2 describes the interpolation commands, thread cutting function, and reference point return function.

2

2.1 INTERPOLATION COMMANDS

This section describes the positioning commands and the interpolation commands that control the tool path along the specified functions such as straight line and arc.

2.1.1 Positioning (G00, G06)

In the absolute programming mode, the axes are moved to the specified point in a workpiece coordinate system, and in the incremental programming mode, the axes move by the specified distance from the present position at a rapid traverse rate.

For calling the positioning, the following G codes can be used.

Table 2.1 G Codes for Positioning

G Code	Function	Group
G00	Positioning in the error detect ON mode	01
G06	Positioning in the error detect OFF mode	*

(1) Positioning in the Error Detect ON Mode (G00)

When "G00 X(U) · · · Z(W) · · · (*C(H) · · · *Y(V) · · ·);" is designated, positioning is executed in the "error detect ON" mode, in which the program advances to the next block only when the number of lag pulses due to servo lag are checked after the completion of pulse distribution has reduced to the permissible value.

In the G00 mode, positioning is made at a rapid traverse rate in the simultaneous 2-axis (*5-axis) control mode. The axes not designated in the G00 block do not move. In positioning operation, the individual axes move independently of each other at a rapid traverse rate that is set for each axis. The rapid traverse rates set for the individual axes differ depending on the machine. For the rapid traverse rates of your machine, refer to the manuals published by the machine tool builder.

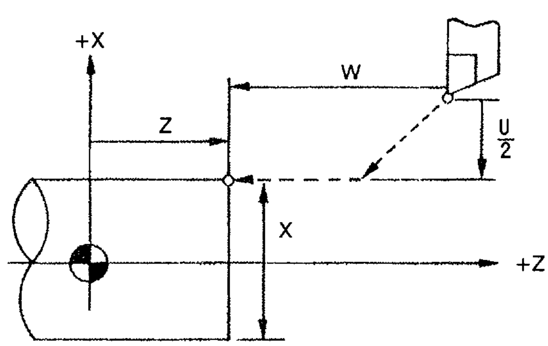


Fig. 2.1 Positioning in Simultaneous 2-axis Control Mode

SUPPLEMENT

1. In the G00 positioning mode, since the axes move at a rapid traverse rate set for the individual axes independently, the tool paths are not always a straight line. Therefore, positioning must be programmed carefully so that a cutting tool will not interfere with a workpiece or fixture during positioning.

2. The block where a T command is specified must contain the G00 command. Designation of the G00 command is necessary to determine the speed for offset movement which is called by the T command.

Example of Programming

```
G50 X150. Z100. ;
G00 T0101 S1000 M03 ;    ← ① G00 determines the speed
                             for offset movement.
(G00) X30. Z5. ;         ← ② Designation of G00 can be omitted
                             since it is a modal command.
```

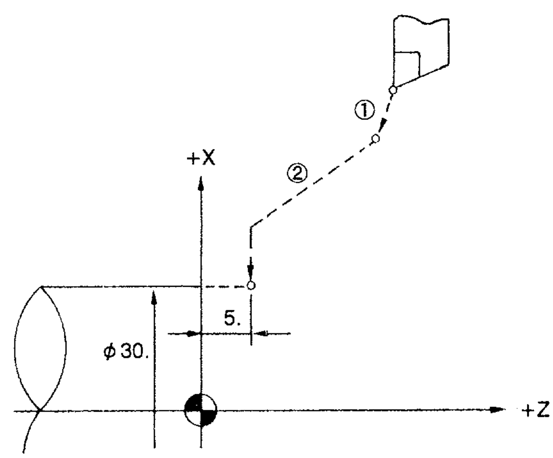


Fig. 2.2

(2) Positioning in the Error Detect OFF Mode (G06)

When "G06 X(U) · · · Z(W) · · · (* C(H)· · · Y(V) · · ·);" is specified, positioning is executed in the "error detect OFF" mode.

In the G06 mode, positioning is executed in the simultaneous 2 axis (*up to 5 axis) control mode. Note that the G06 command is not modal and valid only in the designated block. In this mode, program advances to the next block immediately after the completion of pulse distribution.

2.1.2 Linear Interpolation (G01)

With the commands of "G01 X(U) · · · Z(W) · · · (*C(H) · · · Y(V) · · ·) F(E) · · · ;", linear interpolation is executed in the simultaneous 2-axis (*5-axis) control mode. The axes not designated in the G01 block do not move. For the execution of the linear interpolation, the following commands must be specified.

2

(1) Command Format

To execute the linear interpolation, the commands indicated below must be specified.

(a) Feedrate

Feedrate is designated by an F or E code. The axes are controlled so that vector sum (tangential velocity in reference to the tool moving direction) of feedrate of the designated axes will be the specified feedrate.

$$F\ (\text{mm/min}) = \sqrt{Fx^2 + Fz^2 + (Fc^2)}$$

(Fx: feedrate in the X-axis direction)

SUPPLE-MENT

If no F or E code is designated in the block containing G01 or in the preceding blocks, execution of a G01 block causes alarm "0370".

- With an F code, axis feedrate is specified in either feed per spindle revolution (mm/rev or inch/rev) or feed per minute (mm/min or inch/min).
- If the optional C-axis is selected, the feedrate of X- and Z-axis and that of C-axis differ from each other. Feedrates of these axes obtained by the same F code are indicated in Table 2.2 below.

Table 2.2 Feedrates of X-/Z-axis and C-axis (F Command)

F Function (Feed per Minute)				Minimum F Command Unit	
				Feedrate of X-/Z-axis	Feedrate of C-axis
pm2004 D0 = 0	mm output	mm input	F60	1 mm/min	1 deg/min
		inch input	F51	0.1 inch/min	2.54 deg/min
	inch output	mm input	F60	1 mm/min	0.3937 deg/min
		inch input	F51	0.1 inch/min	0.1 deg/min
pm2004 D0 = 1	mm output	mm input	F63	0.001 mm/min	0.001 deg/min
		inch input	F54	0.0001 inch/min	0.00254 deg/min
	inch output	mm input	F63	0.001 mm/min	0.0003937 deg/min
		inch input	F54	0.0001 inch/min	0.0001 deg/min

POINT

For the C-axis, a feedrate cannot be specified in the feed per minute mode.

(b) End Point

The end point can be specified in either incremental or absolute values corresponding to the designation of an address character or G90/G91. For details, see 3.2.1, "Absolute/Incremental Programming".

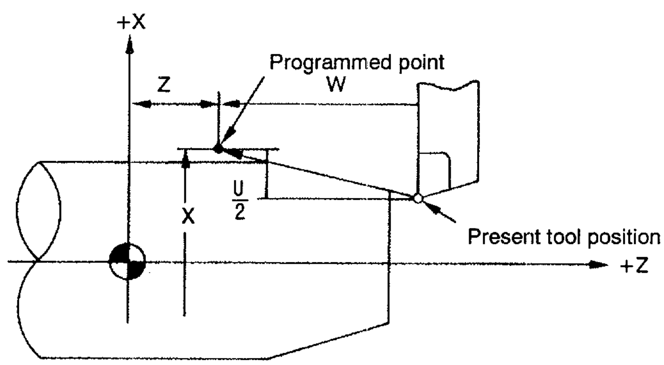


Fig. 2.3 Linear Interpolation

Example of Programming

```
G50 X100. Z60.;
G00 T0202 S600 M03;
    X35. Z5.;
G01 Z0 F1.;       } Axes are moved in the G01 linear interpolation mode.
    X60. F0.2;
```

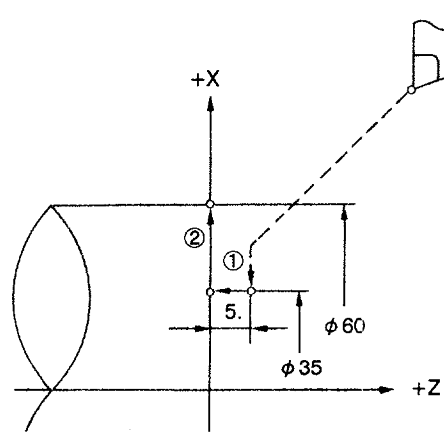


Fig. 2.4 Example of Programming

(2) Angle-designated Linear Interpolation*

By selecting the optional angle-designated linear interpolation function, it is possible to execute linear interpolation by designating an angle.

With the commands of "G01 X(U) · · · A · · · F(E) · · · ;" or "G01 Z(W) · · · A · · · F(E) · · · ;", linear interpolation is executed at an angle A which is measured from the +Z-axis to the end point specified by either X or Z coordinate as shown in Fig. 2.5.

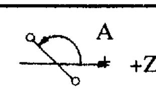
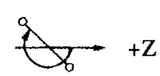
Feedrate is specified by an F or E code along the tangential direction. Programmable range of an angle A is indicated in Table 2.3.

Table 2.3 Programmable Range of Angle (A)

	Programmable Range of Angle (A)
mm input	0 to ±360.000°
inch input	

How the angle designated by command A is measured is determined by the sign which precedes the specified value as indicated in Table 2.4.

Table 2.4 Definition of Angle

Sign	Definition	
A +	Angle measured in the countercloc wise direction from the +Z-axis	
A -	Angle measured in the clockwise direction from the +Z-axis	

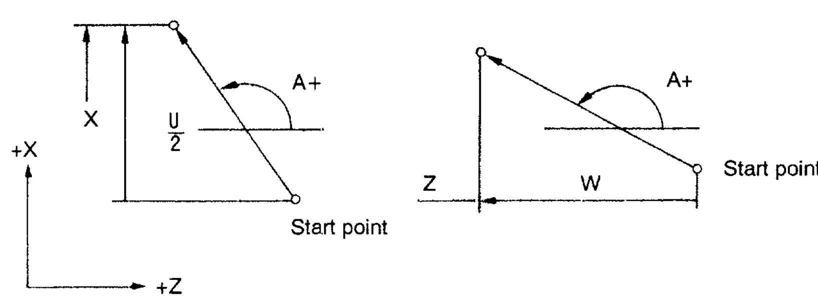


Fig. 2.5 Angle-designated Linear Interpolation

Example of Programming

.

.

.

G01 X50. A150. F0.3; ← ①
G01 Z0. A-180.; ← ②

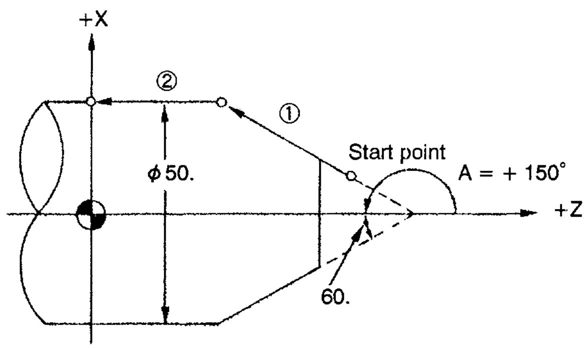


Fig. 2.6 Angle-designated Linear Interpolation

2.1.3 Circular Interpolation (G02, G03, G22, G23)

By specifying the following commands in a program, the cutting tool moves along the specified arc in the ZX plane so that tangential velocity is equal to the feedrate specified by the F or E code.

G02(G03) X(U) · · · Z(W) · · · I · · · K · · · (R · · ·) F(E) · · · ;

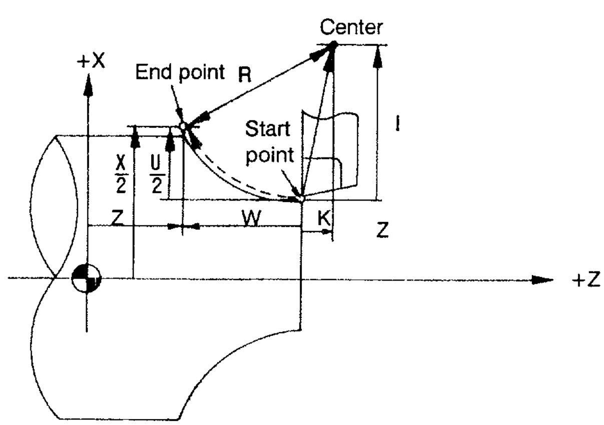


Fig. 2.7 Circular Interpolation

2

(1) Command Format

To execute the circular interpolation, the commands indicated in Table 2.5 must be specified.

Table 2.5 Commands Necessary for Circular Interpolation

Item	Address	Description
Direction of Rotation	G02	Clockwise (CW)
	G03	Counterclockwise (CCW)
End Point Position	X (U)	X coordinate of arc end point (diametric value)
	Z (W)	Z coordinate of arc end point
	* Y (V)	Y coordinate of arc end point
Distance from the Start Point to the Center	I	Distance along the X-axis from the start point to the center of arc (radial value)
	K	Distance along the Z-axis from the start point to the center of arc
	* J	Distance along the Y-axis from the start point to the center of arc
Radius of Circular Arc	R	Distance to the center of arc from the start point

(a) Rotation direction

The direction of arc rotation should be specified in the manner indicated in Fig. 2.8.

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

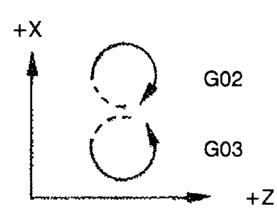


Fig. 2.8 Rotation Direction of Circular Arc

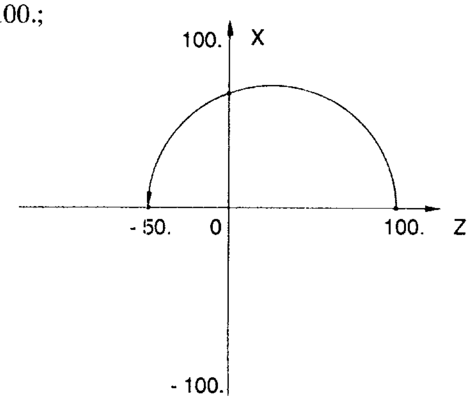
(b) End point

The end point can be specified in either incremental or absolute values corresponding to the designation of G90 or G91.

If the specified end point is not on the specified arc, the arc radius is gradually changed from the start point to the end point to generate a spiral so that the end point lies on the specified arc.

Example of Programming

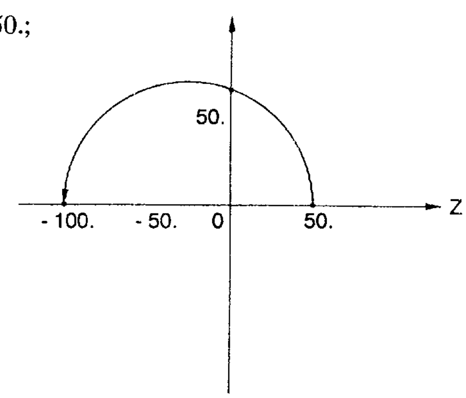
G01 Z100. X0 F10.;
G03 Z-50. K-100.;



(a) End point positioned inside the circumference

Example of Programming

G01 Z50. X0;
G03 Z-100. K-50.;



(b) End point lying outside the circumference

Fig. 2.9 Interpolation with End Point off the Specified Arc

(c) Center of arc

The center of arc can be specified in two methods - designation of the distance from the start point to the center of the arc and designation of the radius of the arc.

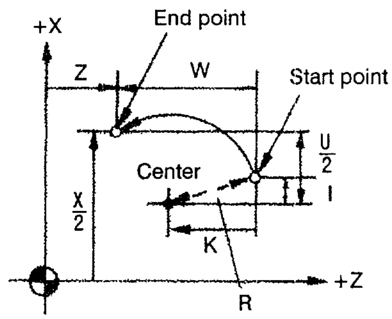


Fig. 2.10

- Specifying the distance from the start point to the center

  Independent of the designated dimensioning mode (G90 or G91), the center of an arc must be specified in incremental values referenced from the start point.

- Specifying the radius

  When defining an arc, it is possible to specify the radius by using address R instead of specifying the center of the arc by addresses I or K. This is called "circular interpolation with R designation" mode.

  · For the circular arc with the central angle of 180 deg. or smaller, use an R value of "R > 0".

  · For the circular arc with the central angle of 180 deg. or larger, use an R value of "R < 0".

  Example of Programming

  G02 X(U) · · · Z(W) · · · R ± · · · F(E) · · ·;

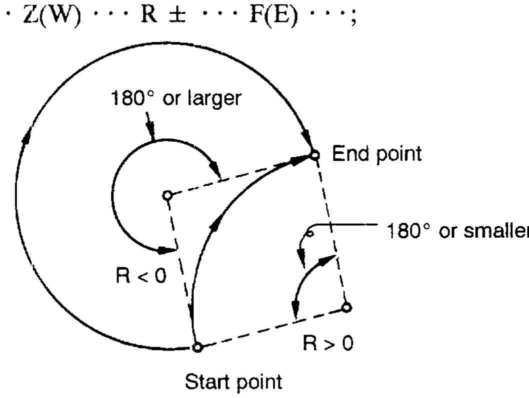


Fig. 2.11 Circular Interpolation with Radius R Designation

SUPPLEMENT

If an R command is used to specify the radius of an arc, G22 and G23 can be used instead of G02 and G03. When G22 or G23 is used, the programming format is the same as used when G02 or G03 is specified with an exception of a G code. If G22 or G23 is used, however, it is not allowed to define the center of the arc by I and K commands. If these commands are used with G22 or G23, alarm "0162" occurs.

2

(2) Supplements to Circular Interpolation

A circular arc extending to multiple quadrants can be defined by the commands in a single block.

Example of Programming

```
G01 Z · · · F · · ·;
G02 X60. Z-46.6 I20. K-19.596 F · · ·;
```

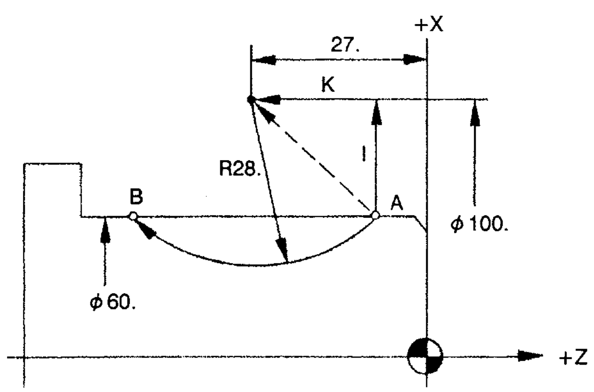


Fig. 2.12 Circular Interpolation over Multiple Quadrants

Table 2.6

Center of Arc	(10000, 2700)
I value	$\frac{100-60}{2} = 20$ mm
K value	$-\sqrt{28^2-20^2} = -\sqrt{384} = -19.596$ mm

2.1.4 Chamfering (G11)

With the commands of "G11 X(U) · · · K · · · {or Z(W) · · · I · · · } F(E) · · · ;", chamfering at corners is specified. In the designation of chamfering, single axis command of either X-axis or Z-axis should be used.

G11 is a modal G code of 01 group. Once designated, it remains valid until other G code in the 01 group is specified next.

(1) X-axis Chamfering

G11 X (U) · · · K ± · · · F (E) · · · ;

- ± — Designation of chamfering direction
- · · · — Chamfer size

With the commands indicated above, X-axis chamfering is executed.

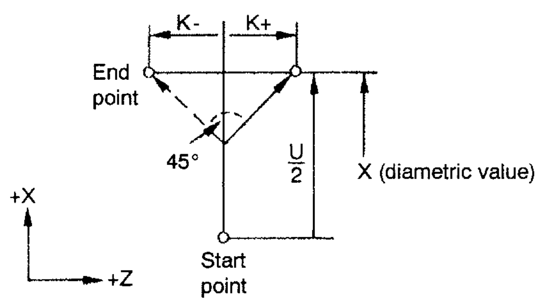


Fig. 2.13 X-axis Chamfering

(2) Z-axis Chamfering

G11 Z (W) · · · I ± · · · F (E) · · · ;

- ± — Designation of chamfering direction
- · · · — Chamfer size (radial value)

With the commands indicated above, Z-axis chamfering is executed.

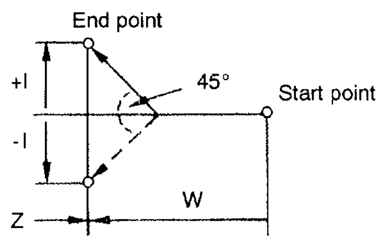


Fig. 2.14 Z-axis Chamfering

Example of Programming

```
G00  X30.  Z0 ;
G11  Z-20.  I8.  F30 ;      ←— ①
(G11)  X80.  K-7. ;         ←— ②
```

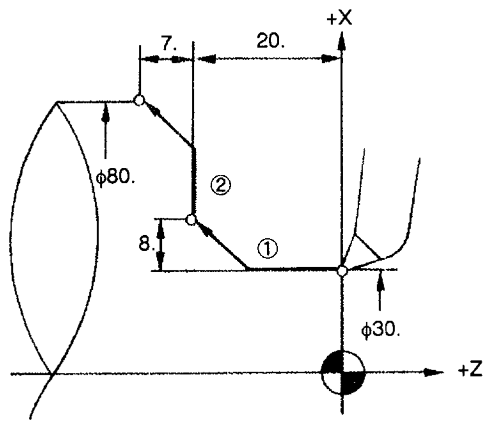


Fig. 2.15 Example of Programming

SUPPLEMENT

1. The following restrictions apply to the chamfer size K and I.

   $|K| < |U/2|, |I| < |W|$

   The K and I values must be smaller than the total move distance in the direction of the designated axis. A format error occurs if a value exceeding this limit is specified.

2. Alarm "0445" occurs if both addresses X and Z are specified in the same block, a block not including I or K is specified in the G11 mode, or I or K value is "0".

3. The nose R offset offset function* is valid for the block where G11 is specified.

4. It is possible to specify the G11 block in the commands of blocks that define finishing shape for a multiple repetitive cycle (G70 to G73).

5. It is possible to specify chamfering by specifying G01 instead of G11.

   G01 X(U) · · · K · · · {or Z(W) · · · I · · ·} F(E) · · · ;

2.1.5 Rounding (G12)

With the commands of "G12 X(U) · · · K · · · {or Z(W) · · · I · · ·} F(E) · · ·;", corner rounding is executed. In the designation, single axis command of either X-axis or Z-axis should be used. Rounding is executed in a quarter circle.

G12 is a modal G code of 01 group. Once designated, it remains valid until other G code in the 01 group is specified next.

(1) X-axis Rounding

G12 X (U) · · · K ± · · · F (E) · · · ;

- ± — Designation of rounding direction
- · · · — Rounding size

With the commands indicated above, X-axis rounding is executed.

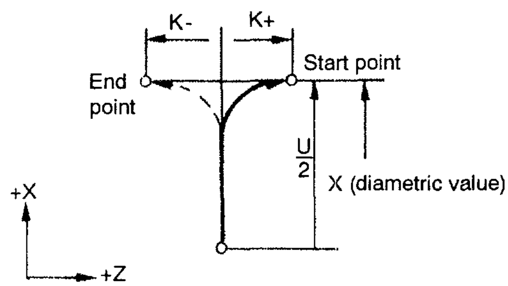


Fig. 2.16 X-axis Rounding

(2) Z-axis Rounding

G12 Z (W) · · · I ± · · · F (E) · · · ;

- ± — Designating of rounding direction
- · · · — Rounding size (radial value)

With the commands indicated above, Z-axis rounding is executed.

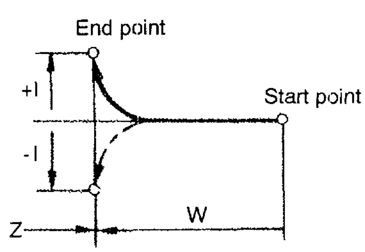


Fig. 2.17 Z-axis Rounding

2

Example of Programming

G00 X20. Z0 ;
G12 Z-25. I9. F30 ; ←— ①
(G12) X70. K-6. F20 ; ← ②

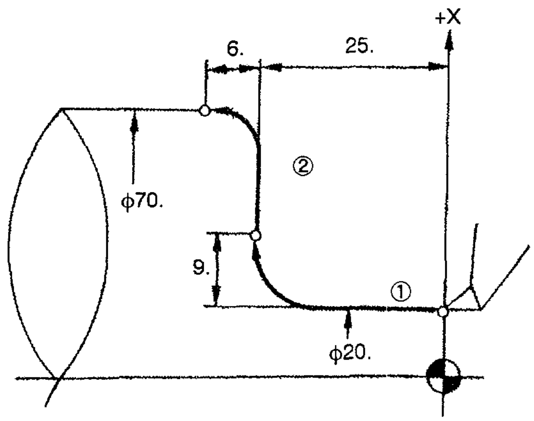


Fig. 2.18 Example of Programming

SUPPLE-MENT

1. The following restrictions apply to the rounding size K and I.

   $|K| < |U/2|, |I| < |W|$

   The K and I values must be smaller than the total move distance in the direction of the designated axis. A format error occurs if a value exceeding this limit is specified.

2. Alarm "0445" occurs if both addresses X and Z are specified in the same block, a block not including I or K is specified in the G12 mode, or I or K value is "0".

3. The nose R offset offset function* is valid for the block where G12 is specified.

4. It is possible to specify the G12 block in the commands of blocks that define finishing shape for a multiple repetitive cycle (G70 to G73).

5. It is possible to specify chamfering by specifying G01 instead of G12.

   G01 X(U) · · · R · · · {or Z(W) · · · R · · ·} F(E) · · · ;

2.1.6 Cylindrical Interpolation (G124, G125) *

The cylindrical interpolation function allows programming of machining on a cylindrical workpiece (grooving on a cylindrical workpiece) in the manner like writing a program in a plane using the cylinder developed coordinate system. This functions allows programming both in absolute commands (C, Z) and incremental commands (H, W).

2

(1) Programming Format

(a) Features of G124, G125

The following G codes are used for cylindrical interpolation.

Table 2.7 G Codes Used for Cylindrical Interpolation

G Code	Function	Group
G124	Cylindrical interpolation mode ON	22
G125	Cylindrical interpolation mode OFF	22

These G codes are buffering prohibiting G codes.

Specify G124 and G125 in a block without other commands. If other G code is specified with G124 or G125 in the same block, alarm "0161" (UNMATCH G CODE) occurs.

G124 and G125 are modal G codes of 20 group. Once G124 is specified, the cylindrical interpolation mode ON state remains until G125 is specified. When the power is turned ON or the NC is reset, the G125 (cylindrical interpolation mode OFF) state is set.

(b) Programming format

```
G124 C・・・ ;    ← Cylindrical interpolation mode ON
   ・            ← Machining program in the cylindrical interpolation mode
   ・
G125 ;          ← Cylindrical interpolation mode OFF
                   where, C = Radius of cylindrical workpiece
                   (1 = 0.001 mm or 0.0001 inch)
```

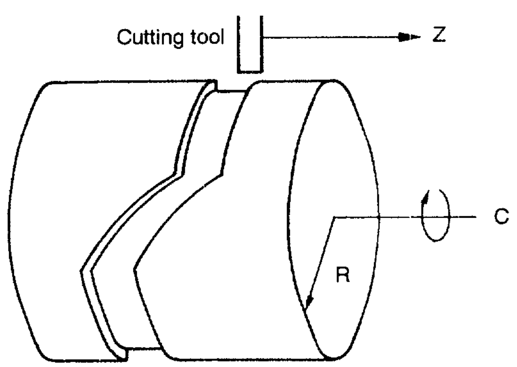
The radius of a cylindrical workpiece must always be specified. If a C command is not specified, alarm "0162" (LACK OF ADDRESS) occurs.

(c) Feedrate

In the cylindrical interpolation mode, interpolation is executed in the virtual C-Z plane. Therefore, after the entry to the cylindrical interpolation mode, it is necessary to specify feedrates in the C-Z plane. Use address F to specify feedrates. F value represents feedrates (mm/min, inch/min) in the C-Z plane.

- For cylindrical interpolation, use the G98 (feed per minute) mode. Cylindrical interpolation is not possible in the G00 mode. To execute positioning, cancel the cylindrical interpolation mode. Note that G00 mode may be specified in a plane other than the C-Z plane.

(2) Example of Programming



Example of programming

```
0100 ;
G98 ;
T0101 ;
G00 X44.0 C0 ;            <---- Positioning at the start point of cutting
G124 C45.0 ;              <---- Cylindrical interpolation mode ON
G01 G42 Z47.5 F100 ;      \
C60.0 ;                   |
Z32.5 C120.0 ;            |
C240.0 ;                  |
G03 Z40.0 C249.549 R7.5 ; > Machining program
G02 Z47.5 C259.099 R7.5 ; |
G01 C360.0 ;              |
G40 Z44.0 ;               /
G125 ;
M30 ;                     <---- Cylindrical interpolation mode OFF
```

Fig. 2.19 Coordinate System for Cylindrical Interpolation

(3) Relationships between Cylindrical Interpolation and Operations

In the cylindrical interpolation mode, the following G codes may be specified: (G00), G01, G02, G03, G04, G10, G22, G23, G40, G41, G42, G65, G66, G67, (G90, G91), G98, and G134. Alarm "0161" (UNMATCH G CODE) occurs if a G code other than those indicated above is specified in the cylindrical interpolation mode.

2

SUPPLE-MENT

1. In the G00 mode, only X-axis can be specified.
2. G90 and G91 are valid only when special G code specification is selected.
3. In the G134 mode, only M commands may be specified.

- In the cylindrical interpolation mode, the tool radius offset function can be used. Turning ON/OFF of the tool radius offset function must be made in the cylindrical interpolation mode. The tool radius offset function is valid only in the cylindrical interpolation mode and the polar coordinate interpolation mode.
- In the cylindrical interpolation mode, cutting in the linear interpolation (G01) mode and circular interpolation (G02/G03) mode is available. Circular interpolation is permitted only in the C-Z plane. If circular interpolation commands are specified in other plane, an alarm occurs. For the definition of an arc, use either addresses I and K to specify the center of arc or address R to directly specify the radius of the arc. Note that designation of address R is optional.
- The nose R offset function must be canceled before specifying G124.
- It is not allowed to specify G124 with the mirror image function ON. Similarly, it is not allowed to turn ON the mirror image function in the G124 mode. If the mirror image function is turned ON in the G124 mode, an alarm occurs.
- T and S commands must not be specified in the cylindrical interpolation mode. Designation of M commands is possible in the cylindrical interpolation mode.
- The spindle function is invalid in the cylindrical interpolation mode.
- In the cylindrical interpolation mode, the manual absolute function is fixed to OFF.

- In the cylindrical interpolation mode, program restart is not possible. If program restart is attempted from a block in the cylindrical interpolation mode, alarm "0481" (PROG, ERROR IN G124 MODE) occurs. However, program restart is allowed for blocks in which the cylindrical interpolation mode blocks are included.

2.1.7 Polar Coordinate Interpolation (G126, G127) *

The polar coordinate interpolation function allows programming of machining that is executed by the combination of tool movement and workpiece rotation in a virtual rectangular coordinate system.

In the machining accomplished by the combination of a linear axis (X-axis) and a rotary axis (C-axis), the C-axis is assumed to be a linear axis that is perpendicular to the X-axis. By assuming a rotary axis as a linear axis, machining an arbitrary shape that is defined by the X- and C-axis can be programmed easily in the X-C rectangular coordinate system. In this programming, both of absolute commands (X, C) and incremental commands (U, H) can be used.

(1) Programming Format

When G126 is specified, the polar coordinate interpolation mode is established and the virtual coordinate system is set in the X-C plane with the origin of the absolute coordinate system taken as the origin of this coordinate system. Polar coordinate interpolation is executed in this plane. Note that polar coordinate interpolation starts when G126 is specified assuming the present position of the C-axis to be "0".

POINT

Return the C-axis to the origin of the absolute coordinate system before specifying G126.

(a) Features of G126 and G127

The following G codes are used to turn ON/OFF the polar coordinate interpolation mode.

Table 2.8 G Codes Used for Turning ON/OFF the Polar Coordinate Interpolation

G Code	Function	Group
G126	Polar coordinate interpolation mode ON	19
G127	Polar coordinate interpolation mode OFF	19

Specify G126 and G127 in a block without other commands. If other G code is specified with G126 or G127 in the same block, alarm "0161" (UNMATCH G CODE) occurs.

G126 and G127 are modal G codes of 19 group. Once G126 is specified, the polar coordinate interpolation mode ON state remains until G127 is specified. When the power is turned ON or the NC is reset, the G127 (polar coordinate interpolation mode OFF) state is set.

(b) Feedrates

In the polar coordinate interpolation mode, interpolation is executed in the X-C plane. It is necessary to specify feedrates after entering the polar coordinate interpolation mode. For the designation of feedrates, use address F. Feedrate F expresses feedrates (mm/min, inch/min) in the X-C plane. In the polar coordinate interpolation mode, specify feedrates in the G98 (feed per minute) mode. It is not possible to specify G00 (G codes that include rapid traverse cycle). To execute positioning, cancel the polar coordinate interpolation mode. It is allowed to specify G00 in a plane other than the X-C plane.

- Restrictions on feedrates

  The following must be satisfied so that the actual speed of the rotary axis does not exceed rapid traverse rate:

  $F/D \leqq (\pi/360) \times$ (Rapid traverse rate of rotary axis)

  F (mm / min) : F command $\times$ Feed override

  D (mm) : Diametric value when a cutting tool approaches closest to the workpiece center
  (tool paths after offset if the tool radius offset function is used.)

- Example of calculation
  - To find the maximum value of F command (override: 100%)

    Conditions: To carry out grooving of 15 mm wide along the center-line

    To use 12 mm diameter end mill

    C-axis rapid traverse rate is 12000 deg/min.

    $F/(15 - 12) \leqq (\pi/360) \times 12000$
    $F \leqq 314$
    From the result indicated above, it is possible to specify F300 in the program.

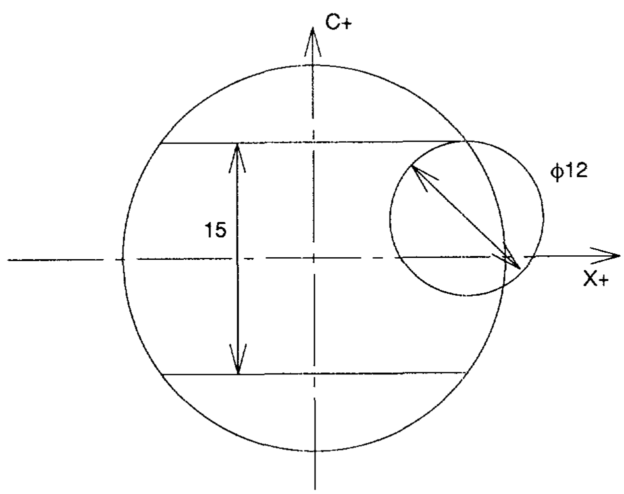


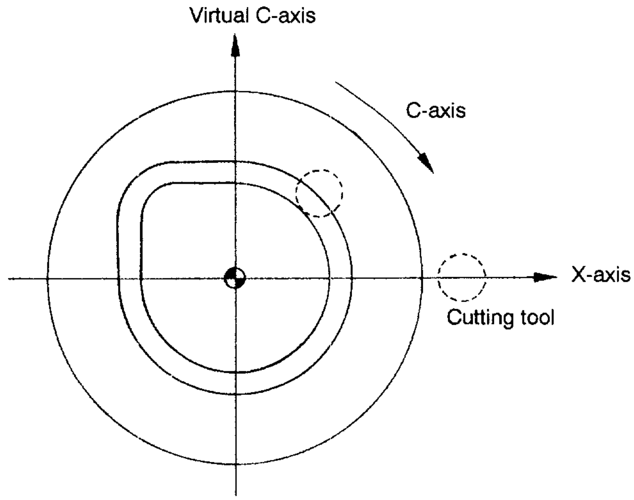
Fig. 2.20

  - To find the minimum machining diameter with F command of 80 mm/min.

    Conditions: Rapid traverse rate of the rotary axis is 12000 deg/min.

    $80/D \leqq (\pi/360) \times 12000$
    $D \leqq 0.764$
    From the result indicated above, machining is not possible if the tool path after offset comes closer to the center of the workpiece than the calculated value.

(2) Example of Programming



Example of programming

00001 ;	
G98 ;	
T0101 ;	
G00 X120.0 C0 ;	Positioning at the cutting start point
G126 ;	Polar coordinate interpolation mode ON
G01 G42 X40.0 F100.0 ;	Machining program using the polar coordinate interpolation function
G03 X0 C40.0 I-20.0 ;	
G01 X-25.0 ;	
G03 X-40.0 C25.0 K-15.0 ;	
G01 C0 ;	
G03 X20.0 I20.0 ;	
G01 G40 X120.0 ;	
G127 ;	Polar coordinate interpolation mode OFF
M30 ;	

Fig. 2.21 Coordinate System for Polar Coordinate Interpolation

2

(3) Negative Polar Coordinate Specification

Fo the machines in which the positive/negative designation of the X-axis is reversed, it is possible to select the negative X-axis specification. In this specification, the plus and minus sign of the X-axis in the virtual X-C plane is reversed. The coordinate system used for programming is shown in Fig. 2.22. Whether or not the negative X-axis specification is selected is specified by using parameter pm4019 D1.

pm4019 D1 = 1	Negative X-axis specification
pm4019 D1 = 0	Normal specification

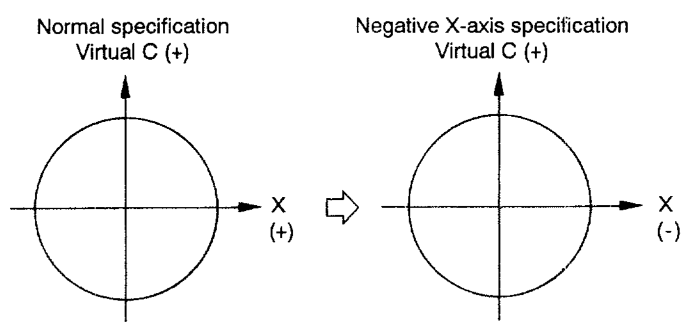


Fig. 2.22 Coordinate System of Negative X-axis Specification Polar Coordinate

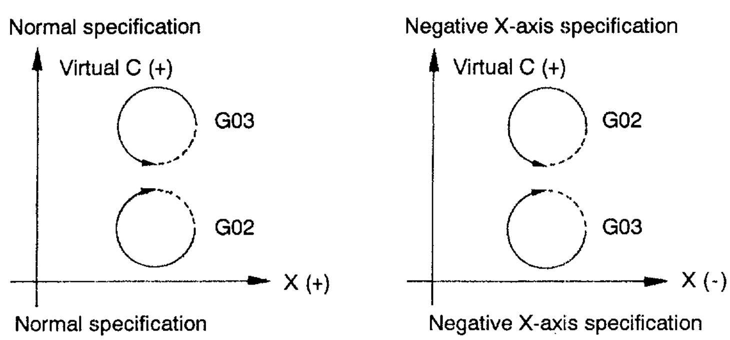


Fig. 2.23 Direction of Rotation for Arc Commands

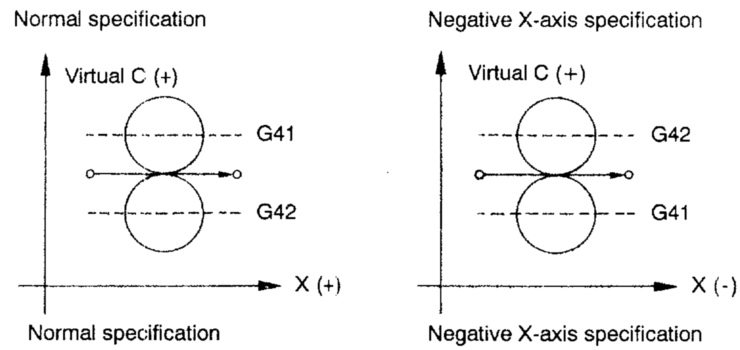


Fig. 2.24 Offset Direction of Tool Radius Offset Function

SUPPLE-MENT

1. If the negative X-axis specification is selected, the direction of rotation for the arc commands (G02, G03) and the direction of offset for the tool radius offset (G41, G42) are reversed from those in the normal specification. This must be taken into consideration when writing a program.
2. Turn ON the polar coordinate interpolation mode when the X-coordinate is plus for the normal specification and when it is minus for the negative X-axis specification.

(4) Relationships between Polar Coordinate Interpolation Function and Operations

In the cylindrical interpolation mode, the following G codes may be specified: (G00), G01, G02, G03, G04, G10, G22, G23, G40, G41, G42, G65, G66, G67, (G90, G91), G98, and G134. Alarm "0161" (UNMATCH G CODE) occurs if a G code other than those indicated above is specified in the polar coordinate interpolation mode.

SUPPLE-MENT

1. In the G00 mode, only X-axis can be specified.
2. G90 and G91 are valid only when special G code specification is selected.
3. In the G134 mode, only M commands may be specified.

- In the polar coordinate interpolation mode, the tool radius offset function can be used. Turning ON/OFF of the tool radius offset function must be made in the polar coordinate interpolation mode. The tool radius offset function is valid only in the cylindrical interpolation mode and the polar coordinate interpolation mode.

- In the polar coordinate interpolation mode, cutting in the linear interpolation (G01) mode and circular interpolation (G02/G03) mode. Circular interpolation is permitted only in the X-C plane. If circular interpolation commands are specified in other plane, an alarm occurs. For the definition of an arc, use either addresses I and K to specify the center of arc or address R to directly specify the radius of the arc. Note that designation of address R is optional.
- The nose R offset function must be canceled before specifying G126.
- It is not allowed to specify G126 with the mirror image function ON. Similarly, it is not allowed to turn ON the mirror image function in the G124 mode. If the mirror image function is turned ON in the G126 mode, an alarm occurs.
- T and S commands must not be specified in the polar coordinate interpolation mode. Designation of M commands is possible in the polar coordinate interpolation mode.
- The spindle function is invalid in the polar coordinate interpolation mode.
- In the polar coordinate interpolation mode, the manual absolute function is fixed to OFF.
- In the polar coordinate interpolation mode, program restart is not possible. If program restart is attempted from a block in the polar coordinate interpolation mode, alarm "0483" (PROG, ERROR IN G126 MODE) occurs. However, program restart is allowed for blocks in which the cylindrical interpolation mode blocks are included.
- If a command that causes the tool paths to pass the center of the polar coordinate in the polar coordinate interpolation mode, alarm "0483" occurs since the C-axis feedrate becomes infinite.
- In the polar coordinate interpolation mode, selection is possible for the X- and C-axis commands whether they are specified in diametric values or radial values.

pm1000 D1 = 0	X- and C-axis commands are specified in diameter.
pm1000 D1 = 1	X- and C-axis commands are specified in radius.

2.2 USING THE THREAD CUTTING FUNCTION

2.2.1 Thread Cutting and Continuous Thread Cutting (G32)

2

With the commands of "G32X (U) · · · Z (W) · · · F (E) · · · ;", it is possible to cut straight thread, tapered thread, or scroll thread in the lead specified by an F (normal thread cutting) or E (precise thread cutting) command to the point specified by absolute coordinate values (X, Z) or incremental coordinate values (U, W). Note that chamfering of thread is not possible in the G32 mode. Use the G92 or G76* mode to include chamfering in thread cutting.

(1) Programmable Range of F and E Codes

Table 2.9 indicates the programmable range of thread lead F and E.

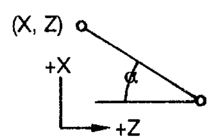
Table 2.9 Programmable Range of F and E Commands

		Format	Programmable Range of F and E
mm output	mm input	F33	F0.001 - F500.000 mm
		E34	E0.0001 - E500.0000 mm
	inch input	F24	F0.0001 - F19.6850 inch
		E26	E0.000004 - E19.685000 inch
inch output	mm input	F33	F0.001 - F1270.000 mm
		E34	E0.0003 - E1270.0000 mm
	inch input	F24	F0.001 - F50.0000 inch
		E26	E0.000010 - E50.000000 inch

(2) Direction of Thread Lead

The direction of thread lead specified by the F and E commands is indicated in Table 2.10.

Table 2.10 Direction of Thread Lead

Taper Angle α		Direction of Thread Lead
	a ≦ 45°	Lead in the Z-axis direction should be specified.
	a > 45°	Lead in the X-axis direction should be specified.

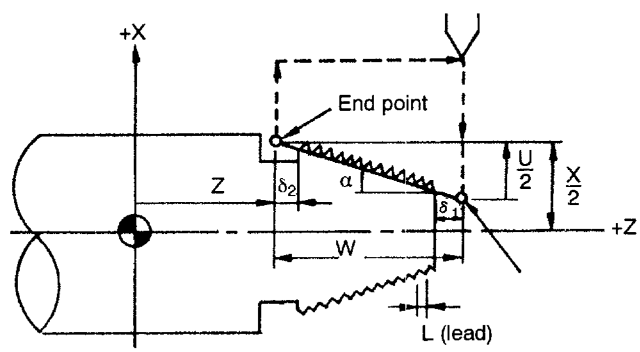


Fig. 2.25 Thread Cutting

(3) Restrictions on F and E by Spindle Speed S

As indicated in Table 2.11, there are restrictions on the designation of F and E commands by Table 2.11spindle speed S. Concerning the X-axis feedrate component, its upper limit is 1/2 of the values indicated in Table 2.11.

Table 2.11 Restrictions on F and E Commands by Spindle Speed S

	Limit Value
mm output	F (E) × S ≦ 240,000 mm/min
inch output	F (E) × S ≦ 24,000 inch/min

(4) Programming Formats

Programming formats of thread cutting are indicated in Table 2.12.

Table 2.12 Programming Formats of Thread Cutting

Thread Type		Command Format
Straight thread	Normal	G31 Z (W) · · · F · · · ;
	Precision	G32 Z (W) · · · E · · · ;
Tapered thread	Normal	G32 X (U) · · · Z (W) · · · F · · · ;
	Precision	G32 X (U) · · · Z (W) · · · E · · · ;
Scroll thread	Normal	G32 X (U) · · · F · · · ;
	Precision	G32 X (U) · · · E · · · ;

- Example of programming for cutting straight thread

Thread lead L = 5.0 mm
δ_1 = 5.0 mm
δ_2 = 3.0 mm
Depth of cut per pass = 1.0 mm

```
G00 U-42. ;      ← ①
G32 W-68. F5.0 ; ← ②
G00 U 42. ;
    W 68. ;
    U-44. ;
G32 W-68. ;
G00 U 44. ;
 .
 .
```

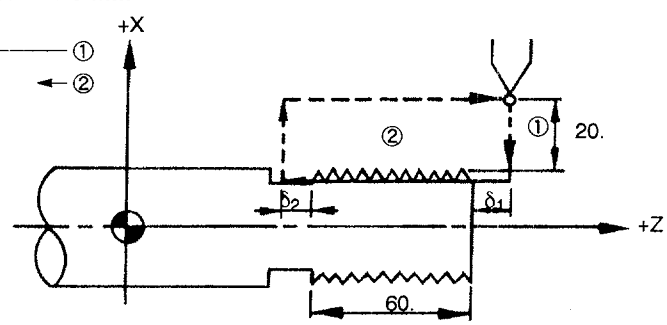


Fig. 2.26 Example of Programming for Cutting Straight Thread

- Example of programming for cutting tapered thread

Thread lead L = 4.0 mm
δ_1 = 3.0 mm
δ_2 = 2.0 mm
Depth of cut per pass = 1.0 mm

```
G00 X13.  ←———— ①
G32 X38. W-35. F4.0 ;  ←②
G00 X60. ;
    W35. ;
    X11. ;
G32 X36. W-35. ;
G00 X60. ;
 .
 .
```

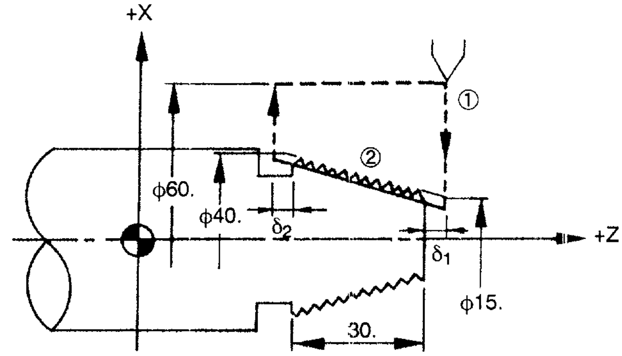


Fig. 2.27 Example of Programming for Cutting Tapered Thread

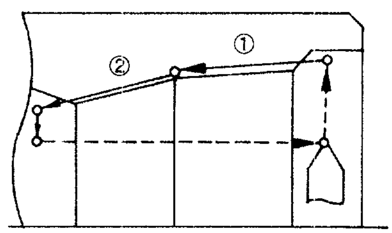
2

(5) Continuous Thread Cutting

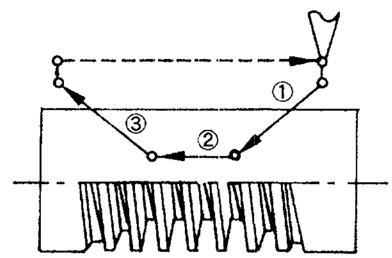
Since the NC has buffer register, designation for continuous thread cutting is possible. In addition, continuous threads can be cut smoothly because the block-to-block pause time is "0" for thread cutting command blocks.

Example of Programming

```
G32 X(U) ··· Z(W) ··· F(E) ··· ;   ←— ①
G32 X(U) ··· Z(W) ··· ;   ←———— ②
G32 X(U) ··· Z(W) ··· ;   ←———— ③
 ·
 ·
 ·
```



(a) Reinforced pipe coupling



(b) Worm screw

Fig. 2.28 Continuous Thread Cutting

1. If designation of thread lead (F, E) is changed during thread cutting cycle, lead accuracy is lost at joints of blocks. Therefore, thread lead designation must not be changed during thread cutting cycle.

2. If continuous thread cutting is specified, M codes must not be specified. If an M code is specified, the cycle is suspended at the specified block and continuous thread cannot be cut.

(6) Margin for Incomplete Thread Portions (δ1, δ2)

At the start and end of thread cutting, lead error is generated. Therefore, margins δ_1 and δ_2 should be given at the start and end portions in thread cutting.

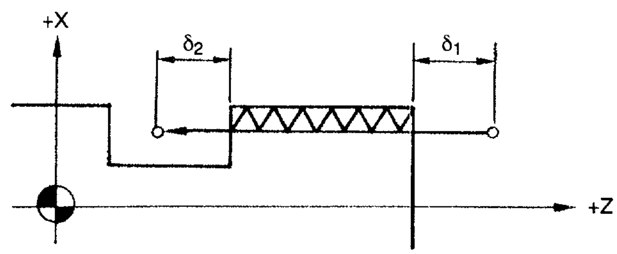


Fig. 2.29 Margins for Incomplete Threads

These margins δ_1 and δ_2 can be calculated as indicated in Table 2.13.

Table 2.13 Calculation of Margins for Incomplete Thread Portions

	Approximate Value	Meaning
δ_1	$\delta_1 > \frac{L \cdot S}{60 \cdot K}(\ln\frac{1}{a}-1)$	L (mm) : Thread lead S (r / min) : Spindle speed K : Constant (normally 30) a (-) : Thread accuracy $= \frac{\Delta L}{L}$ .... (Lead error) ln : Natural logarithm (log)
δ_2	$\delta_2 > \frac{L \cdot S}{60 \cdot K}$	

a	1/50	1/100	1/150	1/200	1/250	1/300
$(\ln\frac{1}{a}-1)$	2.91	3.61	4.01	4.29	4.52	4.70

Example of calculation

Thread lead L = 3.0 mm
Spindle speed S = 5.0 r/min
Thread accuracy =1/100

} δ_1 and δ_2 for this case

$$\delta_1 > \frac{L \cdot S}{60 \cdot K}(\ln\frac{1}{a}-1) = \frac{3.0 \times 500}{60 \cdot K} \times 3.61 = 3.0 \text{ mm}$$

$$\delta_2 > \frac{L \cdot S}{60 \cdot K} = \frac{3.0 \times 500}{60 \cdot K} = 0.83 \text{ mm}$$

Keep the spindle speed at the same value until one thread is cut. If the spindle speed is not maintained constant, accuracy could be lost due to servo lag.

1. During thread cutting, override operation and feed hold operation are disregarded.
2. If G32 is specified in the G98 (feed per minute) mode, alarm "0452" occurs.
3. If a thread cutting command is executed in the dry run mode, axes move at the jog feedrate.

2.2.2 Multiple-thread Cutting (G32) *

Multiple-thread cutting (multiple threads in a lead) is possible without shifting the thread cutting start point. In thread cutting operation, axis feed starts in synchronization with the start-point pulse (1 pulse/turn) output from the spindle pulse generator attached to the spindle. Therefore, the thread cutting start point is always at the same point on the workpiece circumference. In multiple-thread cutting operation, axis feed starts when the spindle rotates by a certain angle after the output of the start-point pulse from the spindle pulse generator.

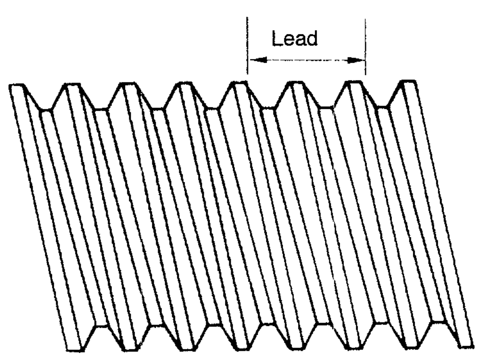


Fig. 2.30 Double-start Thread

With the commands of "G32 X (U) · · · Z (W) · · · F (E) · · · B · · · ;", the spindle rotates by the angle specified by address B after the output of the start-point pulse of the spindle pulse generator. After that thread cutting starts toward the point specified by X (U) and Z (W) at the lead specified by an F or E command.

(1) Address B Specified in Multi-thread Cutting

Least input increment : 0.001°

Programmable range : $0 \leqq B < 360.000$

If decimal point input is used, "B1." is equal to 1° (B1. = 1°). B commands are non-modal and valid only in the specified block.

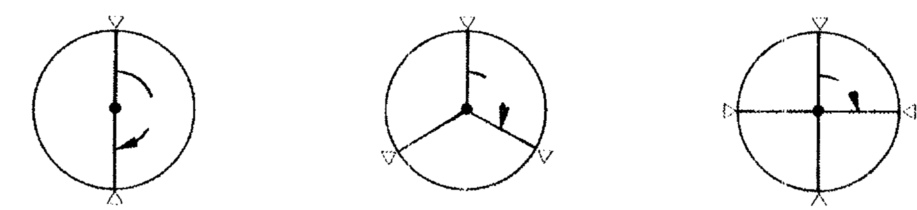
(2) Number of Threads and B Command

In general, the thread cutting start points lie on the workpiece circumference; the intervals of these points are calculated by dividing 360° by the number of threads. Examples of multiple threads (double-start, triple-start, and quadra-start threads) are shown in Fig. 2.31.

Thread cutting start point - double-start thread

Thread cutting start point - triple-start thread

Thread cutting start point - quadra-start thread



1st thread : No B command
2nd thread : B180.

1st thread : No B command
2nd thread : B120.
3rd thread : B240.

1st thread : No B command
2nd thread : B90.
3rd thread : B180.
4th thread : B270.

Fig. 2.31 Number of Threads and B Commands

(3) Spindle Rotating Angle from Start-point Pulse Specified by B Command

For the designation of spindle rotating angle measured from the start-point pulse, the least detectable increment is 360°/4096 pulses ≒ 0.0879°/pulse since the pulses output from the spindle pulse generator (4096 pulses/rotation) are used. For a B command, an error of ±1 pulse of the spindle rotation detection pulses could be generated. An example of programming for double-start thread is indicated below.

Example of Programming

```
G00 U · · · ;
G32 W · · · F · · · ;
G00 U · · · ;
W · · · ;
U · · · ;
G32 W · · · ;
  ·
  ·
  ·
```

Thread cutting of thread A

```
G00 U · · · ;
G32 W · · · B180. ;
G00 U · · · ;
W · · · ;
U · · · ;
G32 W · · · B180. ;
  ·
  ·
  ·
```

Thread cutting of thread B

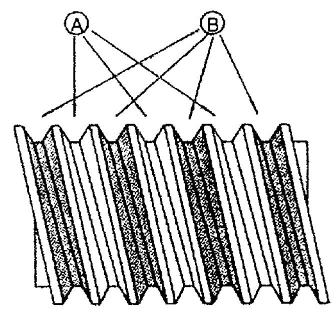


Fig. 2.32 Spindle Rotation Angle from Start-point Pulse by B Command

SUPPLE-MENT

1. If a B command value is outside the programmable range (0 to 360.000), alarm "0453" occurs.
2. If a B command is specified for multiple-thread cutting, continuous thread cutting is not possible.

   G32W B90

   G32W . . . . ← Since the operation is suspended at this block to wait for the start-point pulse, continuous thread cannot be cut.
3. The spindle rotation angle from the start-point pulse is specified using a B command (0 to 360°) disregarding of the spindle rotating direction.

2.2.3 Variable Lead Thread Cutting (G34) *

With the commands of "G34 X (U) Z (W) · · · K · · · F (E) · · · ;", variable lead thread can be cut; thread lead variation per one spindle rotation is specified by address K. The least input increment of a K command is 0.0001 mm/rev or 0.00001 inch/rev. If the setting for parameter pm1000 D0 = 1, the least input increment of a K command is 0.001 mm/rev or 0.0001 inch/rev.

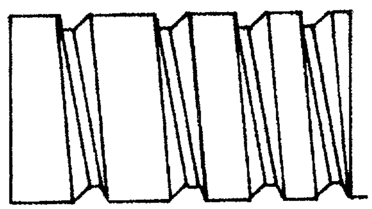


Fig. 2.33 Variable Lead Thread

(1) Restrictions on Programmable Range of K Commands

The programmable range of K commands is restricted by the formula indicated below.

F : Fixed lead command (mm/rev or inch/rev)

K : Variable lead command (mm/rev or inch/rev)

W : Distance along the Z-axis from the start point to the end point (mm or inch) <"U" along the X-axis in the case of face thread cutting.>

S : Spindle speed (rev/mm)

N : Number of spindle revolutions from the start point to the end point (rev)

$$N = \frac{-(F + K/2) + \sqrt{(F + \frac{K}{2})^2 + 2 \cdot K \cdot W}}{K}$$

2

(2) Feedrate at End Point

Specify the commands so that the feedrate at the end point will not exceed the upper limit indicated in Table 2.14.

Table 2.14 Upper Limit of Feedrate at End Point

	Upper Limit
mm output	500 mm/rev
inch output	50 inch/rev

$S \times (F + \frac{K}{2} + KN) \leqq pm2800$ (Max. cutting feedrate)

(3) Feedrate at End Point

Specify the commands so that the feedrate at the end point will not be a negative value.

$(F + \frac{K}{2})^2 + 2KW > 0$

SUPPLE-MENT

1. In the continuous block thread cutting for variable lead thread cutting, distribution of command pulses is interrupted at joints between blocks.
2. If a K command is outside the programmable range, alarm "0450" occurs.
3. If G34 is executed in the dry run mode, the axes move at a feedrate designated for the jog feedrate if "parameter pm2000 D1 = 1".
4. If address B is designated in the G34 block, alarm "0450" occurs.

2.3 REFERENCE POINT RETURN

2.3.1 Automatic Return to Reference Point (G28)

With the commands of "G28 X(U) · · · Z(W) · · · (* C(H) · · · * Y(V) · · ·);", the numerically controlled axes are returned to the reference point. The axes are first moved to the specified position at a rapid traverse rate and then to the reference point automatically. This reference point return operation is possible in up to simultaneous 2-axis (* 5-axis) control. The axes not designated in the G28 block are not returned to the reference point.

Example of Programming

(G90/G91) G28 X(U) · · · Z(W) · · · (C(H) · · ·);

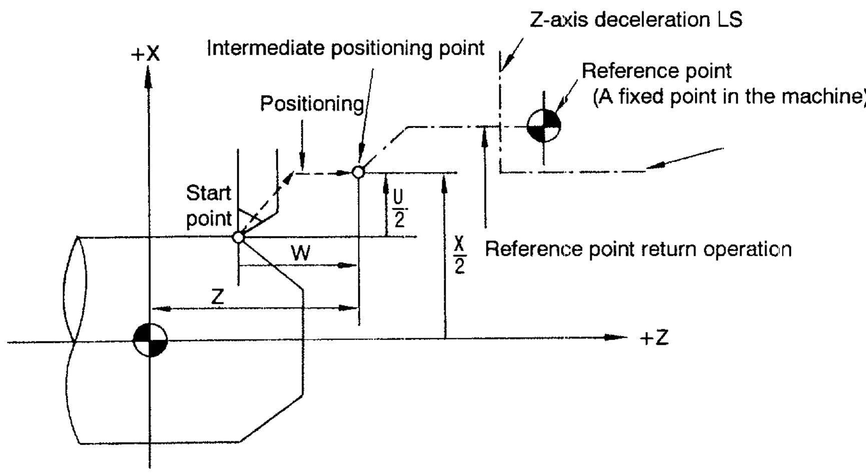


Fig. 2.34 Reference Point Return

(1) Reference Point Return Operation

Reference point return operation is the series of operations in which the axes return to the reference point after the reference point return operation has been started manually.

Reference point return is accomplished in two ways:

(a) Low-speed reference point return

In low-speed reference point return operation, a deceleration limit switch is used. In high-speed reference point return operation, the first return operation is executed in the low-speed type using a deceleration limit switch; the reference point data are stored after the completion of the first reference point return and in subsequent reference point return operations is executed without using a deceleration limit switch.

Alarm "2061" to "2065" occurs if an attempt is made to start the reference point return operation from a position where return operation is impossible.

2

(b) High-speed reference point return

See parameter pm4003 D6 and D7.

It is possible to use the "high-speed reference point return" in place of the "automatic reference point return". In this case, the reference point return is executed in the following manner.

- After the positioning at the intermediate positioning point B, the axes return directly to the reference point at a rapid traverse rate. The axes can be returned to the reference point in a shorter time compared to the normal reference point return operation that uses a deceleration limit switch for the individual axes.
- Even if point B is located outside the area in which reference point return is allowed, the high-speed reference point return specification allows the axes to return to the reference point.
- High-speed reference point return is enabled only for the axes for which normal reference point return has been completed either manually (manual reference point return) or by executing the G28 command after turning ON the power.
- If low-speed reference point return has not been completed for the X- and Z-axis either manually or by executing the G28 command after power-ON, low-speed reference point return is executed for the axis (X- and/or Z-axis) which is specified in the G28 block.
- High-speed automatic reference point return is valid only when reference point return is called by G28, and it does not influence manual reference point return operation.

2

(2) C-axis* Control Integrated with Spindle Control

Reference point return is executed for the C-axis each time the control mode is changed over from the spindle control to the C-axis control.

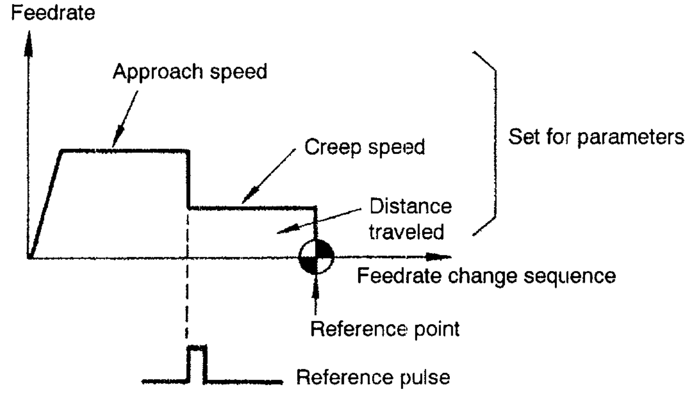


Fig. 2.35 Reference Point Return Pattern of C-axis Integrated with Spindle

SUPPLE-MENT

For the C-axis integrated with the spindle, a deceleration limit switch is not used even in low-speed reference point return operation.

(3) Supplements to the Automatic Reference Point Return Commands

- Concerning machine lock intervention, there are two types of operation: turning ON the machine lock after suspending axis movement by using the feed hold function, and turning OFF the machine lock after suspending axis movement again by using the feed hold function. Table 2.15 shows how the machine operates according to the machine lock intervention.

Table 2.15 Machine Operation according to Machine Lock Intervention

<table>
<tr><th colspan="2"></th><th>Machine Lock Intervention during Positioning to Intermediate Positioning Point</th><th>Machine Lock Intervention during Positioning to Reference Point</th></tr>
<tr><td rowspan="2">Machine Lock OFF → ON</td><td>Low-speed type</td><td rowspan="2">Although positioning is continued to the intermediate positioning point (position data display only), movement to the reference point is not executed.
Display data are not updated, either.</td><td>Display data are infinitely updated. Although positioning is made at the reference point after the detection of the actuation of the deceleration limit switch, this cannot be detected due to machine lock and, therefore, the display data are infinitely updated.</td></tr>
<tr><td>High-speed type</td><td>In response to the machine lock intervention, the axes stops moving. After that, the display data (position data in the workpiece coordinate system) are updated until the reference point return is completed. (without axis movement)</td></tr>
<tr><td rowspan="2">Machine Lock OFF → ON → OFF</td><td>Low-speed type</td><td rowspan="2">Although positioning is continued to the intermediate positioning point, the position is displaced by the machine lock intervention amount.</td><td>The axes move to the reference point (position data display is offset by the machine lock intervention amount).</td></tr>
<tr><td>High-speed type</td><td>Actual axis position is displayed due to the intervention of machine lock. Accordingly, although the display data (position data in the workpiece coordinate system) agree with the reference point, the axes are not located at the reference point.</td></tr>
</table>

- Before specifying the G28 command, the tool position offset mode and nose R offset mode should be canceled. If the G28 command is specified without canceling these modes, they are canceled automatically.

- It is possible to select valid/invalid of reference point return for each axis. If the axis for which "reference point return invalid" has been set is specified in the G28 block, alarm "0241" occurs. Refer to parameter pm4002 D0 to D4.

- It is possible to display alarm "0411" (X-axis) to "0415" (5th-axis) when an axis move command other than G28 is executed without completing reference point return after turning ON the power. Whether or not such alarm display should be given is determined by the setting for parameter pm4022. The direction of reference point return is set for pm4002 D0 to D4 for the individual axes.
- The absolute coordinate values of the axes specified in the G28 block are saved to memory as the intermediate positioning point. For the axes not specified in the G28 block, the intermediate positioning point saved in the previous reference point return operation remains valid.
- If M and/or T command is specified with G28 in the same block, the axes continue moving to the reference point disregarding whether or not the FIN processing is completed before the positioning of an axis at the intermediate positioning point. Therefore, DEN is output at the reference point.
- The deceleration limit switch position must be carefully attended to when executing the reference point return for the first time after turning ON the power. For details, refer to 2.4.2, "Manual Reference Point Return" of the Operating Manual.

2

2

2.3.2 Reference Point Return Check (G27)

This function checks whether the axes are correctly returned to the reference point at the completion of the part program which is created so that the program starts and ends at the reference point in the machine by specifying the commands of "G27 X(U) · · · Z(W) · · · (* C(H) · · · * Y(V) · · ·);".

In the G27 mode, the function checks whether or not the axes positioned by the execution of these commands in the simultaneous 2-axis (* 3-axis) control mode are located at the reference point. For the axes not specified in this block, positioning and check are not executed.

(1) Operation after the Check

When the position reached after the execution of the commands in the G27 block agrees with the reference point, the reference point return complete lamp lights. The automatic operation is continuously executed when all of the specified axes are positioned at the reference point. If there is an axis that has not been returned to the reference point, reference point return check error (alarm "0421" (X-axis) to "0425" (5th-axis)) occurs and the automatic operation is interrupted. In this case, the cycle start lamp goes OFF.

(2) Supplements to the Reference Point Return Check Command and Other Operations

- If G27 is specified in the tool position offset mode, positioning is made at the position displaced by the offset amount and the positioning point does not agree with the reference point. It is necessary to cancel the tool offset mode before specifying G27. Note that the tool position offset function is not canceled by the G27 command.
- The reference point return check is not executed if G27 is executed in the machine lock ON state.
- The mirror image function is valid to the direction of axis movement in the reference point return operation called by G27. To avoid a position unmatch error, the mirror image function should be canceled by specifying G69 (mirror image OFF) before executing G27.

2.3.3 Return from Reference Point Return (G29)

The commands of "G29 X · · · Z · · ·;" the axes, having been returned to the reference point by the execution of the automatic reference point return function (G28, G30), to the intermediate positioning point by back tracing the paths along which the reference point return has been executed.

Example of Programming

G28 $\underbrace{\text{X} \cdots \text{Z} \cdots;}_{\text{Point B}}$ Point A → B → C (Reference point)

·

·

·

G29 $\underbrace{\text{X} \cdots \text{Z} \cdots;}_{\text{Point D}}$ Point C → B → D

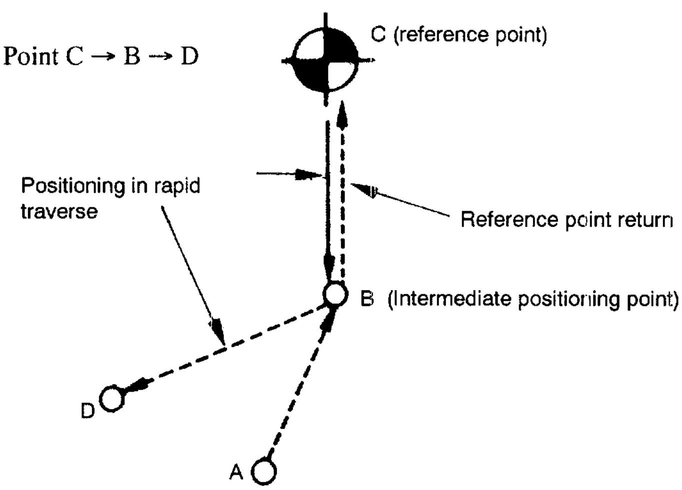


Fig. 2.36 Return from Reference Point

2

(1) Intermediate Positioning Point

- It is not possible to specify the intermediate positioning point in the G29 block. The axes return to the previous point at a rapid traverse rate along the paths taken in the return to the reference point. Note that the axes not specified in the G29 block do not move.
- If G28 or G30 (see 2.2.4, "Second to Fourth Reference Point Return (G30)*") has been executed several times before the execution of G29, point B to be set for the execution of B29 is established at the intermediate positioning point set in the last G28 or G30 operation. The following program written in absolute commands explains how point B is set for the return operation from the reference point.

				Coordinate values of intermediate positioning point X Z
N20	G28	Z10.	Z20.	(10., 20.)
N23	G28	X30. ;		(30., 20.)
N24	G29	X-40.	Z-50. ;	

X-40. Z-50. — End point

G00 X30. Z20. ; — Intermediate positioning point

G00 X-40. Z-50. ; — End point

Example of Programming

N31 T0300;
N32 G28 U80. W20.;
N33 T0400;
N34 G29 U-80. W40.;

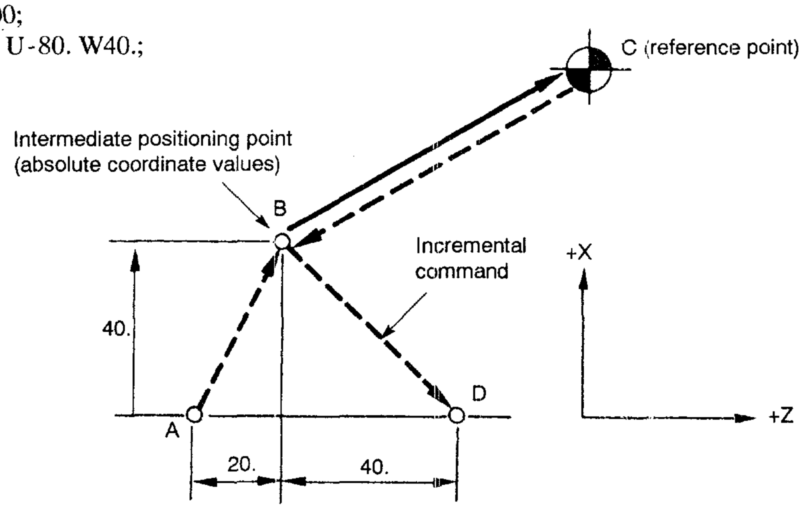


Fig. 2.37 Coordinate Values of Point B for G29 Operation

- In the following cases, the intermediate positioning point used for the execution of G29 does not agree with the intermediate positioning point specified for the execution of G28 or G30. Therefore, do not specify such commands or attempt such operation.
  - Execution of the following before the execution of G29 after the completion of G28:

    Coordinate system setting (G50 or coordinate system setting operation in POS. job)
    Intervention of machine lock
    Intervention of manual operation with manual absolute OFF
  - Execution of G28, or G30 or G29 in a block specified after the cancellation of the mirror image at a position different from the position where the mirror image was started.
  - Execution of G28, or M30 or M29 after the intervention of manual operation with the manual absolute OFF.

(2) Supplements to the Return Command from the Reference Point Return

(a) Automatic reference point return

If G29 is specified without the execution of G28 or G30 after turning ON the power, alarm "0240" occurs.

(b) Nose R offset and canned cycle

If G29 is specified in the nose R offset mode (G41, G42) or in a canned cycle (G70 to G76, G90, G92, G94, G81 to G89), alarm "0170" or "0182" occurs.

(c) Tool position offset

It is necessary to cancel the tool position offset function before specifying G28, G30, or G29. If these G codes are executed in the offset mode, the intermediate positioning point B' is also offset, causing the tool to move to point B. Note that the tool position offset function is not canceled by G29.

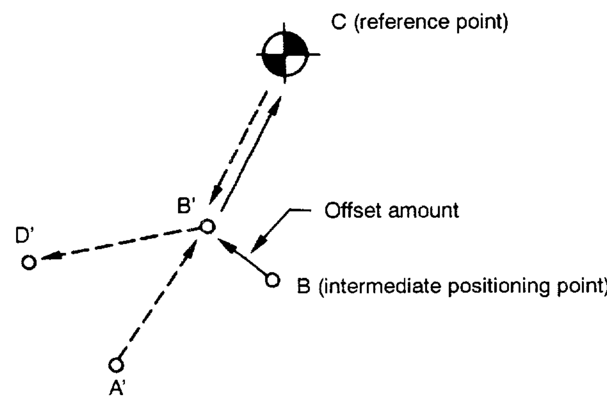


Fig. 2.38 G29 Operation Executed in the Tool Position Offset Mode

2.3.4 Second to Fourth Reference Point Return (G30) *

With the commands of "G30 Pn X(U) · · · Z(W) · · · (* C(H) · · · Y(V) · · ·);", the axes are moved to P2 (second reference point), P3 (third reference point*), or P4 (fourth reference point*) in the simultaneous 3-axis (* 5-axis) control mode after the positioning at the specified intermediate positioning point. If "G30 P3 U-40. W30.;" is specified, the X- and Z-axis return to the third reference point. If "Pn" is omitted, the second reference point is selected. The axes not specified in the G30 block do not move.

2

(1) Reference Point Positions

The position of each reference point is determined in reference to the first reference point. The distance from the first reference point to each of the reference points is set for the following parameters.

Table 2.16 Reference Points

	X-axis	Z-axis	3rd-axis	4th-axis	5th-axis
2nd reference point	pm6811	pm6812	pm6813	pm6814	pm6815
3rd reference point	pm6821	pm6822	pm6823	pm6824	pm6825
4th reference point	pm6831	pm6832	pm6833	pm6834	pm6835

(2) Supplements to the 2nd to 4th Reference Point Return Commands

- For the points to be considered to for the execution of G30, refer to the supplements in 2.2.1, "Automatic Return to Reference Point (G28)".
- If G29 is specified after G30, positioning is made at the point specified with G29 after passing the intermediate positioning point specified with G30. Only the coordinate value of intermediate positioning point of the axis specified with G30 is updated.
- For the execution of G30, reference point return must have been completed after power-ON either manually or by the execution of G28. If an axis for which reference point return has not been completed is included in the axes specified in the G30 block, alarm "0240" occurs.

3

MOVEMENT CONTROL COMMANDS

Chapter 3 describes the procedure used for setting and selecting the coordinate system and the programming for controlling the movement of a cutting tool.

3

3

3.1 SETTING THE COORDINATE SYSTEM

3.1.1 Base Coordinate System (G50)

Before programming axis movement, a coordinate system must be set. When a coordinate system is set, a single absolute coordinate system is determined and absolute move commands specified after the setting of a coordinate system are all executed in it. The G50 command sets the position of the origin of a coordinate system used for programming.

G50 is a non-modal G code that is valid only in the specified block. The block in which the G50 command is specified must not contain other G codes, M codes, S codes, and T codes. Especially, if an S or T code is specified in a block with G50 like "G50 S · · ·;" or "G50T · · ·;", such designation calls specific functions and does not set a coordinate system.

3

(1) Commands

For setting a coordinate system, both absolute and incremental commands may be used.

(a) Coordinate system setting in absolute commands

With the commands of "G50 X · · · Z · · · (*C · · · *Y · · ·) ;", a coordinate system is set so that the present tool nose position has the absolute coordinate values specified in the G50 block (X, Z, C, B *, Y *). In other words, the addresses in the G50 block specify the distance from the point that should be set as the origin (0, 0) of the coordinate system used for programming to the present tool nose position. Axis movement commands can be specified for up to 2 axes (* 5 axes max.) simultaneously. Note that the axes not specified the G50 block do not move.

An example of coordinate system setting is shown in Fig. 3.1. In this example, the coordinate system is set at the position where reference point return has been executed. A coordinate system can be set at any position.

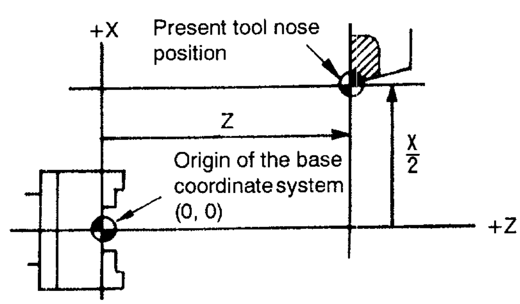


Fig. 3.1 Setting of Base Coordinate System (G50) at Reference Return Position

3

(b) Coordinate system setting in incremental commands

If addresses U, W, and H are specified with the commands of "G50 X · · · Z · · · (* H · · · * V · · ·) ;", a new coordinate system is set in reference to the present coordinate system by shifting it the distance specified in incremental values of U (X-axis direction), W (Z-axis direction), and H (C-axis direction).

This feature is effectively used in several applications - an operation that uses cutting tools having considerable differences, for example. In this case, the cutting tools should first be divided into two groups and the difference between the length of standard tool in one group and that in the other group should be entered in a program. Then, a new base coordinate system can be set for the second tool group.

Example of programming

G50 U100. W-100.;

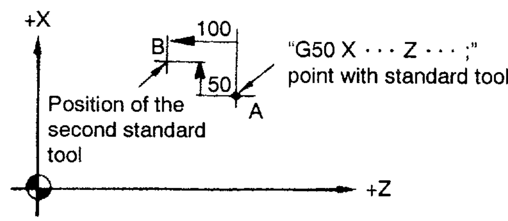


Fig. 3.2 Setting of Coordinate System with Incremental Values

(2) Coordinate System and Tool Position Offset

After setting the coordinate system by executing the commands of "G50 X80. Z62.;" taking the cutting tool No. 01, if the cutting tool No. 02 which has the tool position offset amount as shown in Fig. 3.3 is selected and offset is executed, the cutting tool No. 02 moves to point A.

Example of Programming

N3 G50 X80. Z62.;
N4 G00 T0101;
•
•
•
N10 G00 T0202;
•
•
•

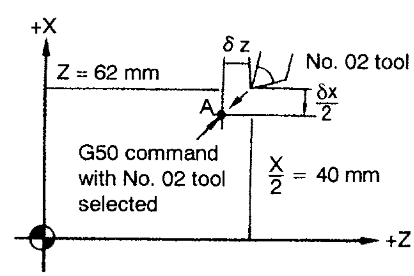


Fig. 3.3 Coordinate System and Tool Position Offset

As shown in Fig. 3.3, if the coordinate system is set in reference to the standard tool and offset data are set for other tools, it is possible to program all tool movements in a single coordinate system.

(3) Automatic Coordinate System Setting

It is possible to set a coordinate system automatically at the completion of manual reference point return. To set the coordinate system in this manner, the setting values should be set for parameters for each of mm and inch input operation as indicated below.

Table 3.1 Parameters for "mm" Input and "inch" Input

	X-axis	Z-axis	C-axis	4th-axis	5th-axis
mm input	pm4801	pm4802	pm4803	pm4804	pm4805
inch input	pm4811	pm4812	pm4813	pm4814	pm4815

POINT

3. Whether or not the automatic coordinate system setting function is made valid or not should be set for parameters pm4006 D0 to D4 for the individual axes.
4. To use the workpiece coordinate system shift function, set the coordinate system by adding the workpiece coordinate system shift values to the coordinate system setting values when setting the coordinate system using the automatic coordinate system setting function.
5. The coordinate system that has been set using the automatic coordinate system setting function becomes invalid when other coordinate system setting function such as G50 is executed.

3

(4) Supplements to the Base Coordinate System Commands

- If a T code is specified in the block next to the one in which the G50 command is specified, it is necessary to enter G00 in the block where the T code is specified to define the offset movement feedrate.

  G50 X · · · Z · · · ;
  G00 S500 M03 T0101;

- Cancel the tool position offset and nose R offset function before specifying G50.

- When the power is turned ON, coordinate values (0, 0, 0) is set for the present tool position. Therefore, the coordinate system must always be set before starting an operation. Concerning the C-axis integral with the spindle, use the automatic coordinate system setting function - change the mode to the C-axis control and execute the reference point return, and the coordinate system is set for the C-axis at the position where the reference point return has been completed.

- Once the coordinate system is set, it is not influenced by the reset operation. To reset the coordinate system, use either of the following operation.

  · To set "0" on the [ABS] function screen.

  · To set "0" for the coordinate values by executing "G50 X0 Z0 (C0);" in the MDI mode.

  · To turn OFF the power once and turn it ON again.

- The present tool position in the base coordinate system is displayed on the [ABS] function name which is called in the [POSIT.] job.

- Whether or not the workpiece coordinate shift is valid when G50 is specified is determined by the setting for parameter pm4012 D0.

3.1.2 Workpiece Coordinate System (G50T, G51) *

The function to set workpiece coordinate systems is provided to set the coordinate system for the individual cutting tools so that the program can be executed at the same program origin even if the cutting tool to be used is changed by the tool selection operation.

(1) Tool Coordinate Data Memory (Number)

Before specifying "G50T" command, it is necessary to write the coordinate data to the tool coordinate data memory for each of the cutting tools.

3

(a) Tool coordinate data memory

The number of tool coordinate data memory areas corresponds to the number of tool offset data memory area pairs. See Table 3.2.

Table 3.2 Tool Coordinate Data Memory

	Number of Tool Offset Data Memory Area Pairs	Tool Coordinate Data Memory Areas (Number)
1	When 0 to 16	51 to 66 (16 areas)
2	When 0 t 50	51 to 99 (49 areas)

(b) Tool coordinate data memory numbers and tool numbers

Tool coordinate data memory number "51" corresponds to tool number "01". Similarly, tool coordinate data memory number "52" corresponds to tool number "02", and so on.

Table 3.3 Correspondence between Tool Coordinate Data Memory Numbers and Tool Numbers

Tool Coordinate Data Memory No.	Tool No.
51	01
52	02
•	•
•	•
80	30

3

(c) Coordinate data: Xtn, Ztn

The coordinate data (Xtn, Ztn) as shown in Fig. 3.4 are written to the tool coordinate data memory for the individual tools Tn.

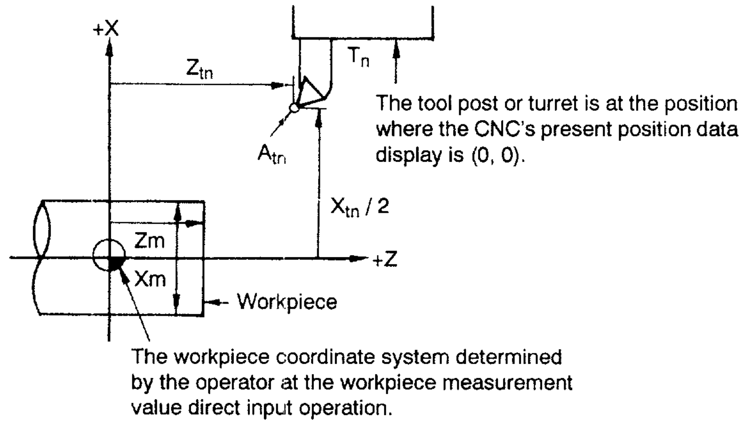


Fig. 3.4 Tool Coordinate Data Memory

(2) Commands

(a) Features of G50T and G51

Table 3.4 G Codes Related to Workpiece Coordinate System

G Code	Function	Group
G50T	Setting a workpiece coordinate system	*
G51	Return to the origin for present value display	*

The G50T and G51 commands are both non-modal and valid only in the specified block.

(b) Setting a workpiece coordinate system

```
G50T □□  △△;
     │    └──── Tool offset number designation (00 to 50)
     └───────── Tool coordinate value memory number designation (51 to 99)
```

A workpiece coordinate system is set by using the command format indicated above; the value calculated in the following formula is used as the setting value of the workpiece coordinate system.

Workpiece coordinate system setting value
= [Present position display value in the NC]
+ [Value set at the specified tool coordinate data memory area]
+ [Value set at the specified tool offset memory area]

Here, the "present position display value in the NC" indicates the value displayed at the [EXTERN] function in the [POSIT.] job.

- In normal operation, "00" should be set for △△ (tool offset number designation part).

  Example: G50 T5100 ;

  If "00" is set, a workpiece coordinate system is set assuming that the data set in the tool offset memory area is "0".

When the program as indicated in the example above is set when the tool post or turret is positioned at arbitrary position, the workpiece coordinate system that the operator has determined is set correctly.

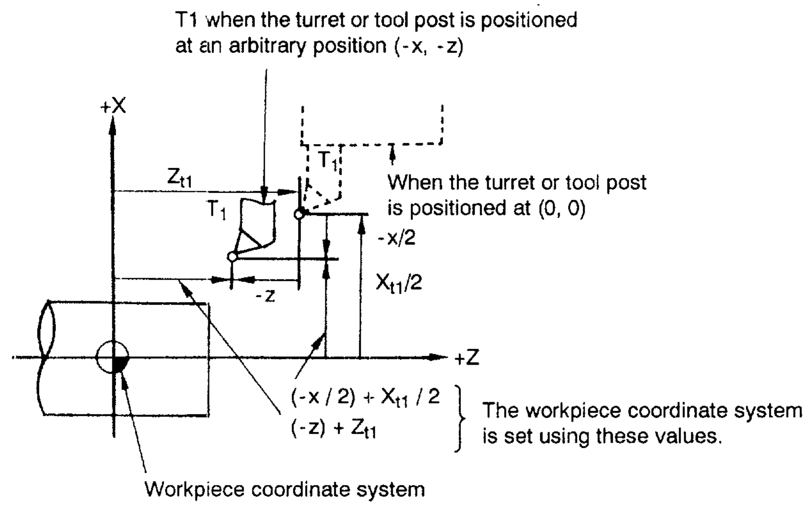


Fig. 3.5 Workpiece Coordinate System Setting

- With the commands of "G50 T0000;", the workpiece coordinate system is canceled. That is, the command T0000 causes the calculation of the present position data with "value in tool coordinate data memory = 0" and "value in tool offset data memory = 0" to set the workpiece coordinate system.

(c) Returning to the origin for present position (G51)

With a machining program that uses the workpiece coordinate system setting function, the start point of machining should be set at the position where the present position data display is (0, 0). Therefore, after the completion of machining, G51 must be specified in the program so that the X- and Z-axis return to the start point of machining accurately. With the G51 command, both of the X- and Z-axis return to the start point at a rapid traverse. Note that G51 should be specified in a block independently without other commands.

3

(3) Example of Programs

(a) Example program using a workpiece coordinate system

An example of program in which a workpiece coordinate system is used is given below.

The start point of machining is (0, 0) of present position display.

N1 G50 T5100; ← Setting of a workpiece coordinate system for tool No. 01

N2 G00 T0101 M03 S100; ← Selection of tool No. 01 (Note)

·
· (Machining using tool No. 01)
·

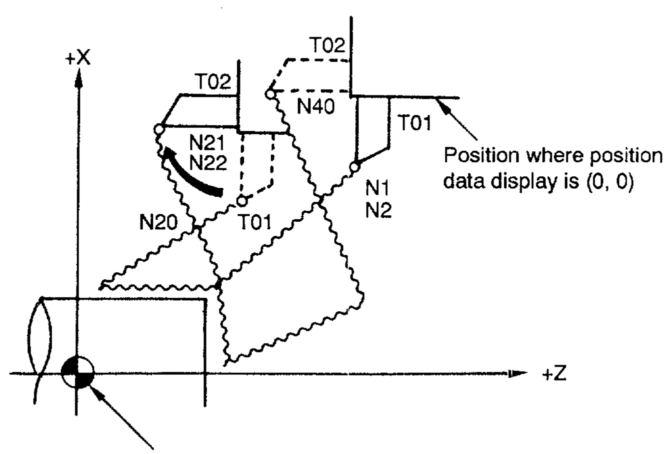
N20 G00 X · · · Z · · · ← Positioning

N21 G50 T5200; ← Setting of a workpiece coordinate system for tool No. 02

N22 G00 T0202; ← Selection of tool No. 02 (Note)

·
· (Machining using tool No. 02)
·

N40 G51; ← Returning to the point of (0, 0) (present position data display)



Note: The tool position offset command in T0101 and T0202 can be used for the compensation for tool wear. It can also be used for offsetting in taper cutting.

Fig. 3.6 Example Program Using Workpiece Coordinate System

(b) Example of program in which operation in a workpiece coordinate system is interrupted

If an operation is restarted from the beginning of the program without returning the cutting tool to the start point of machining after the interruption of the program given below, the cutting tool is positioned correctly at the first approach position.

Example of program in which operation in a workpiece coordinate system is interrupted

N1 G50 T5100;
N2 T0101;
N3 G96 S150 M03;
N4 G00 X20. Z2.5; ← A

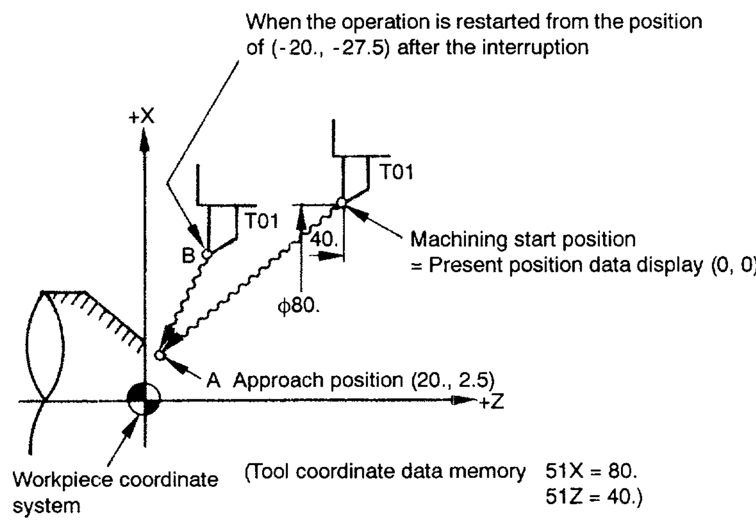


Fig. 3.7 Example of program in which operation in a workpiece coordinate system is interrupted

The commands of "N1 G50 T5100;" executed at point B sets a workpiece coordinate system using the values of X = 60. (-20. + 80.) and Z = 12.5 (-27.5 + 40.). Therefore, the workpiece coordinate system is saved and, accordingly, the approach position point A remains unchanged.

(c) If tool change positions differ in a workpiece coordinate system

An example of program and workpiece coordinate system setting values are indicated below for cases where tool change position differs by tools.

Table 3.5 Tool Coordinate Data Memory

No.	X	Z
51	100.	47.5
52	110.	40.

```
N1 G50 T5100;
N2 G00 T0101 M03 S1000;
    ·
    · (Machining with T01)
    ·
    ·
N25 G50 T0000;
N26 G00 X-50. Z-35.;      ← Tool change position to T02 is (-50., -35.).
N27 G50 T5200;            ← The coordinate system setting values used
N28 G00 T0202 M03 S800;     by these commands are:
    ·                       X = (-50.) + 110. = 60.
    · (Machining with T02)  Z = (-35.) + 40. = 5.
    ·
    ·
N48 G51;
```

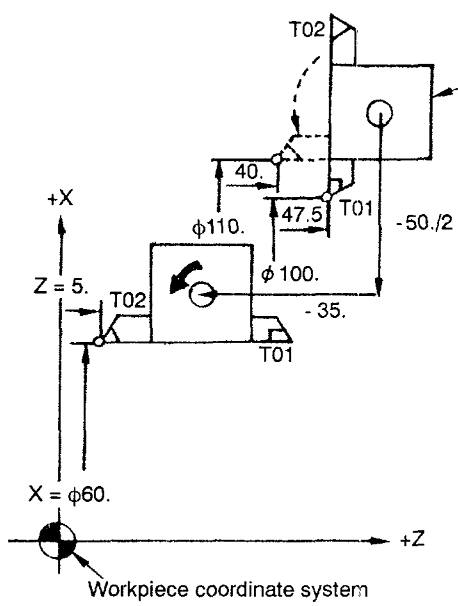


Fig. 3.8 If Tool Change Positions Differ in a Workpiece Coordinate System

(4) Workpiece Coordinate System Shift Amount

The coordinate system that is set using G50 or the workpiece coordinate system setting function can be shifted by the required distance. It is possible to write the shift distance to the workpiece coordinate system shift data memory, which is No. 00 of the offset data memory data, for the X-, Z- and C-axis in the same operation as writing the tool offset data.

(a) Shift data written to the memory

The shift data written to the memory becomes valid at the following timing:

- Execution of the G50 coordinate system setting command
- Execution of the G50 T workpiece coordinate system setting command
- Execution of the automatic coordinate system setting function
- Execution of key operation for setting the coordinate system.

When any of the operation indicated above is executed, a coordinate system is set by simply adding the set shift amount. No cutting tool movement takes place. If a positive value is set for $\triangle X$, $\triangle Z$, and $\triangle C$, the coordinate system is shifted in the direction indicated by the arrow symbol in Fig. 3.9. In this figure, X_0 and Z_0 indicate the original coordinate system setting values.

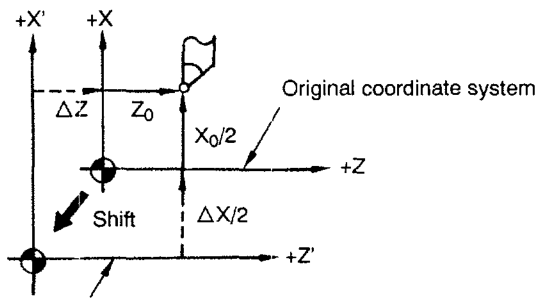


Fig. 3.9 Workpiece Coordinate System Shift Operation

The direction in which the coordinate system should be shifted can be changed by changing the setting for parameter pm4012 D3. By setting D0, it is possible to make the shift amount invalid at the execution of G50.

(b) After changing the data in the workpiece coordinate system shift data memory

If the data in the workpiece coordinate system shift data memory is changed, the new shift amount becomes valid when any of the operations indicated in item (a) above is executed.

SUPPLEMENT

1. Once the coordinate system is shifted by the workpiece coordinate system shift function, it cannot be canceled unless "0" is set. Resetting of the NC cannot cancel the shifting of the workpiece coordinate system.
2. The G50 T□□00; commands do not indicate the offset data memory number of the workpiece coordinate system shift function but it indicates the cancel of the tool position offset function.

3

(5) Supplements to the Workpiece Coordinate System Setting Commands

- To use the G50 T and G50 commands, set "0" for parameter pm3000 D0 (pm3000 D0 = 0; presetting of the external present position data for G50 is OFF).
- The "G51;" command is equivalent to the commands specified in two blocks like "G50 T0000;" and "G00 X0 Z0;". Therefore, execution of the G51 command cancels the tool offset number as well as the workpiece coordinate system and thus the tool offset number becomes "00" after the execution of it.
- The workpiece coordinate system shift function becomes valid when the workpiece coordinate system is set by using the "G50 T" command.
- The present position data of the cutting tool in the set workpiece coordinate system is displayed in the present position display (workpiece coordinate system) and not displayed in the external present position display.
- The workpiece coordinate system set by G50 T is not canceled by the reset operation.

3.2 DETERMINING THE COORDINATE VALUE INPUT MODES

This section describes the commands used to input coordinate values.

3.2.1 Absolute/Incremental Designation

Axis movement data specified following an axis address determines axis movement distance in either incremental or absolute values.

By using addresses X, Z, C*, Y*, U, W, H*, and V*, it is possible to use both incremental and absolute values.

(1) Command Format

3

(a) Absolute commands

To specify axis movement distance in an absolute value, use addresses X, Z, and C.

Example: X · · · Z · · · C · · ·;

(b) Incremental commands

To specify axis movement distance in an incremental value, use addresses U, W, and H.

Example: U · · · W · · · H · · ·;

(c) Use of both incremental and absolute commands in the same block

It is allowed to use both incremental and absolute values in the same block.

Example: X · · · W · · ·;
U · · · Z · · ·;

If addresses that represent the same axis are specified in the same block like "X · · · U · · ·;", the address specified later becomes valid.

These G codes specify whether dimension values specified following an axis address are given in an absolute value or incremental value.

Table 3.6 Absolute and Incremental Commands and Meaning

Address	Command Value		Meaning (Description)
X	Absolute	Diametric value	Position in the X-axis direction
Z		-	Position in the Z-axis direction
*C		-	Position in the C-axis direction
*Y		-	Position in the Y-axis direction
U	Incremental value	Diametric value	Movement distance in the X-axis direction
W		-	Movement distance in the Z-axis direction
*H		-	Movement distance in the C-axis direction
*V		-	Movement distance in the Y-axis direction
I	Incremental value	Radial value	X-axis direction component of the distance to the center of arc viewed from the start point of arc
K		-	Z-axis direction component of the distance to the center of arc viewed from the start point of arc
*J		-	Y-axis direction component of the distance to the center of arc viewed from the start point of arc
R	Incremental value	-	Direct designation of arc radius

Since a diametric value is specified for addresses X and U, actual axis movement distance is a half the specified value.

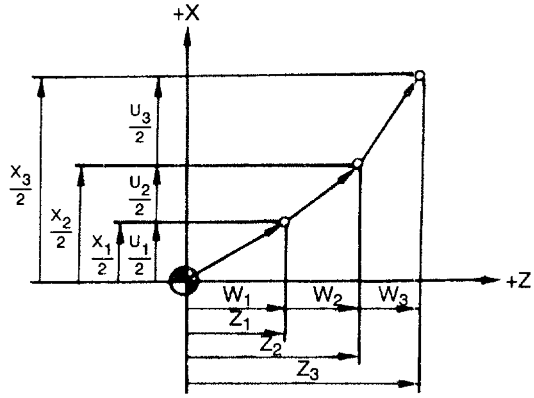


Fig. 3.10 Absolute and Incremental Coordinate Values

(2) Use of G90 and G91

(a) If special G code I (Basic) or II (Option) is selected

G90 and G91 commands can be used when special G code I (basic) or special G code II (option) is selected.

Table 3.7 Function of G90 and G91 Commands

G Code	Function	Group
G90	Absolute designation	03
G91	Incremental designation	03

Note: G90 and G91 are valid only for X, Z, C*, and Y* as indicated in Table 3.8.

Table 3.8 Valid Address for G90/G91 Designation

Mode	Address	G90 Command	G91 Command
TAPE, MEM, and MDI modes	X, Z, C, Y	Absolute	Incremental
	U, W, H, V	Incremental	Incremental

Example: With the commands of "G91 G00 X40. Z50.;" axis movement commands are executed as incremental commands.

(b) Auxiliary data for circular interpolation

The auxiliary circular interpolation data I, J*, K, and R are always interpreted as incremental commands.

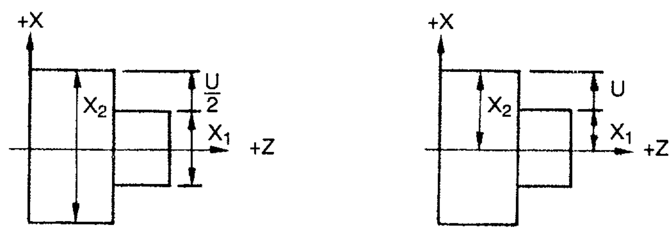
SUPPLEMENT

It is not allowed to specify G90 and G91 in the same block. If both of these G codes are specified in the same block, the one specified later becomes valid. For example, if the commands of "G01 G90 X80. G91 Z60.;" are specified in a block, G91 specified later becomes valid and all axis movement commands (X80. and Z60.) are interpreted as incremental commands.

3.2.2 Diametric and Radial Commands for X-axis

To specify X-axis commands, address X or U is used and dimensions are usually specified in diametric values. This designation method is called the diametric command designation. However, it is also possible to use a radial value to specify X-axis dimensions and which of designation is used is set by parameter pm1000.D1.

pm1000 D1 = 0	Diametric designation
pm1000 D1 = 1	Radial designation



(a) Diametric designation (b) Radial designation

Fig. 3.11 Coordinate Values

Table 3.9 Use of Diametric and Radial Designation

Item	Diametric Designation	Radial Designation
Address X command	Diametric value	Radial value
Address U command	Diametric incremental value	Radial incremental value
X-axis position display	Diametric value	
Tool position offset amount	Diametric value	
Tool coordinate data for tool coordinate system	Diametric value	
Nose R amount	Radial value	
Feedrate F and E in the X-axis direction	Radial value/rev, Radial value/mm	
Radius designation for circular interpolation (I, K, J, R)	Radial value	
G90 to G94, G70 to G76 Chamfering, rounding, multiple chamfering parameters D, I, K, P, Q, R	Radial value	

3.2.3 Inch/Metric Input Designation (G20, G21)

It is possible to select the dimension unit for the input data between "mm" and "inches". For this selection, the following G codes are used.

Table 3.10 Dimension Unit Selection G Codes

G Code	Function	Group
G20	Input in "inch" system	06
G21	Input in "mm" system	06

3

(1) Command Format

G20 and G21 should be specified at the beginning of a program in a block without other commands. When the G code which selects the input dimension unit is executed, the following values are processed in the selected dimension unit: subsequent programs, offset amount, a part of parameters, a part of manual operation, and display.

Example of Programming

```
ER
CR
01234;
G20;  ←——————  Designating the input in "inch" system
 .
 .
 .
```

(2) Supplements to the Dimension Unit Designation Commands

- A parameter is used to select "inch/mm". Therefore, the state when the power is turned ON is determined by the setting for this parameter.
- If the dimension unit system should be switched over during the execution of a program, the tool position offset and nose R offset function must be canceled before the switching over of the dimension unit system.
- After switching over the dimension unit system between G20 and G21, the following processing must be accomplished.
  - Set the coordinate system before specifying axis move commands.
  - If position data are displayed in a workpiece coordinate system, or when an external position data display unit is used, reset the present position data to "0".
- The tool offset amounts stored in memory are treated in a different manner between the G20 and G21 modes.

3

Table 3.11 Tool Offset Amounts in G20 and G21 Modes

Stored Offset Amount	In the G20 (Inch System) Mode	In the G21 (mm System) Mode
150000	1.5000 inch	15.000 mm

3.3 TIME-CONTROLLING COMMANDS

3.3.1 Dwell (G04)

It is possible to suspend the execution of axis move commands specified in the next block for the specified length of time (dwell period).

By specifying "G04 U (P, X, F) · · · ; ", execution of programmed commands is suspended for the length of time specified by address U, P, X or F.

- Command unit of address P is "1 = 0.001 sec". For example, a dwell period of 2.5 seconds is specified by "G04 U2500;". The block used to specify dwell must not include commands other than G04 and P commands.
- The maximum programmable value with address U, P, X, or F is indicated in Table 3.12.

Table 3.12 Dwell Period (Programmable Range of P)

Format	Programmable Range of Dwell Period (P)
U (P, X, F) 63	0 to 999999.999 sec

Note: The value is independent of the input and output unit systems.

3

3.4 TOOL OFFSET FUNCTIONS

The following three kinds of tool offset functions are provided: tool position offset function, nose R offset function, and tool radius offset function*.

3.4.1 Tool Offset Data Memory

The memory area where the data of the offset functions and coordinate system setting is called the tool offset data memory.

	Tool Offset Data Memory No.	X-axis	Z-axis	C-axis	Nose R Offset Data	Control Point	
Workpiece coordinate system shift memory	00						
Memory area for storing tool offset data, tool coordinate data (X- and Z-axis)	01 to 16						Basic
	17 to 99						Option
	100 to 299						Option

Fig. 3.12 Tool Offset Data Memory

(1) Contents of Tool Offset Data Memory (X-/Z-axis) of 001 to 299

The tool offset data memory of 001 to 299 for the X- and Z-axis is usually used to store the tool position offset data. However, there are also cases that the tool coordinate data are stored in this area if workpiece coordinate systems are used. For details of the use of this memory area, refer to the manuals published by the machine tool builder.

SUPPLE-MENT

Even if 299-set option is selected, offset numbers 0 to 99 are allowed if a T4-digit specification is used.

(2) Tool Offset Number Specified by T Function

A "tool offset number" specified by the T function directly corresponds to the "tool offset memory number" and the data stored in the specified area is called out to execute offset functions. Concerning the tool coordinate data memory, the "tool selection designation" number in the T function corresponds to the tool coordinate data memory number. The workpiece coordinate system shift data memory is independent of the T command. The offset data should be stored to the memory area before starting automatic operation.

3.4.2 Tool Position Offset

The tool position offset function adds the offset amount to the coordinate value specified in a program when a tool offset number is specified and moves the nose R to the position obtained by the addition. Therefore, the difference between the coordinate value of the nose R of the cutting tool, assumed in programming, and that of the actual nose R position should be set to the tool offset data memory in advance as the offset amount. If the coordinate value of the nose R is changed due to tool wear or other reasons, the offset amount set in the tool offset data memory must be modified accordingly. By using the tool position offset function, the required dimensions can be obtained without changing numerical values in a program.

(1) Setting Range of Tool Position Offset Amounts

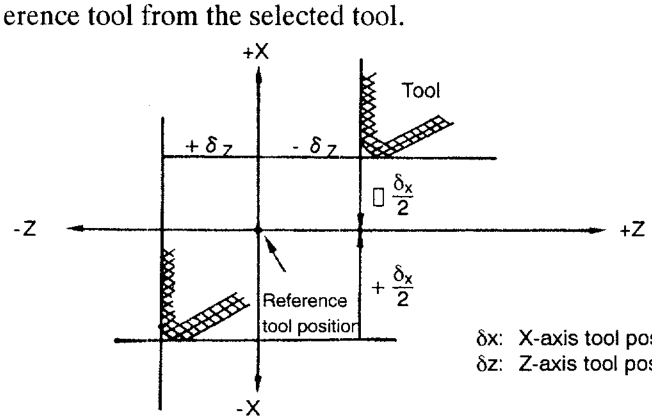
The range of tool position offset amounts that can be set to the memory is indicated in Table 3.13.

Table 3.13 Tool Offset Amount Setting Range

Output	Input	Tool Offset Amount Setting Range
mm output	mm input	0 to ±9999.999 mm
	inch input	0 to ±999.9999 inch
inch output	mm input	0 to ±9999.999 mm
	inch input	0 to ±999.9999 inch

(2) Signs Used in Tool Position Offset Amounts

For the tool position offset amount, difference between the reference tool and the selected tool is set as the signed value with the offset amount of reference tool taken as "0". The sign that precedes the offset amount should be determined by viewing the reference tool from the selected tool.



δx: X-axis tool position offset amount (in diameter)
δz: Z-axis tool position offset amount

Fig. 3.13 Signs Used in Tool Position Offset Amounts

(3) Outline of Tool Position Offsetting Movements

As explained in item (2) above, when the cutting tool selected by "T□□□□" command moves according to the axis movement command specified in a program, the offset amount that corresponds to the specified tool offset number is added to the command and the cutting tool moves to the point obtained after the addition. If no axis movement command in the same block, the cutting tool moves only by the offset amount. Once offset, the cutting tool always moves to the offset position unless other offset number is specified. If other number is specified or offset amount is changed while the tool position is offset, offset is made corresponding to the difference between the previous and new offset amounts.

Example of Programming

```
T0101;
 .
 .
 .
G01 X ··· Z ··· F(E) ··· ;   ←——— ①
T0115;  ←————————————————————— ②
```

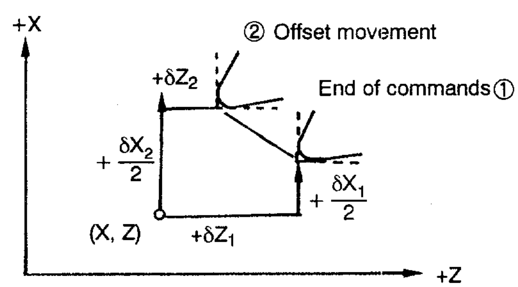


Fig. 3.14 Tool Position Offsetting Movements

(4) Offset Movement Speeds

Offset movement explained above is executed at the feedrate valid at that point. Therefore, the required feedrate must be specified in the same block or a preceding block if the tool position offset function is specified (G00, G01 F..., etc.).

Example of Programming

```
G50 X ··· Z ··· ;
G00 S ··· M03 T0108;   ←———— Only offset movement is executed
X ··· Z ··· ;                at a rapid traverse rate
 .
 .
 .
```

3

(5) Calling the Tool Position Offset Function

The tool position offset command is called when a tool offset number is specified.

(a) Designating a T code

Specify a T code in the block where the tool position offset function should be called. The tool position offset function becomes from the specified block. When a T code is read, a tool selection signal (binary code) is output and, at the same time, offset movements start using the offset amount which corresponds to the specified offset number. T codes are modal and once a T code is specified, it remains valid until another T code is specified. If "G00 T0202;" is specified, for example, tool No. 02 is selected and offset movements occur according to the data set for tool offset No. 2.

(b) Changing the offset amount

To change the offset amount, specify a T code that has another offset number. If a T code with a different tool number is specified, tool selection operation is executed.

Example of Programming

```
G00 T0202;
G01 X · · · Z · · · F · · ·;
  ·
  ·
  ·
G01 T0216;    ←———— Offset No. is changed from 02 to 16 and offset movements
                     occurs at a cutting feedrate.
```

(c) Correcting angle in taper cutting

To correct an angle in taper cutting, specify the T code that changes the offset amount in the same block with the cutting feed commands.

```
G00 T0202;
G01 X · · · Z · · · F · · ·;
  ·
  ·
  ·
G01 U+ · · · W- · · · F · · · T0216;        ←———— ①
```

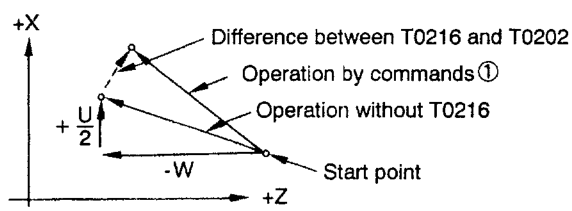



Fig. 3.15 Correcting Taper Angle

If axis movement commands are specified with the T function in the same block, the nose R moves to the offset target position. In this example, cutting is executed by correcting the difference between the offset amounts called by T0202 and T0216.

(d) Canceling the tool position offset function

To cancel the tool position offset function, specify a T code of offset No. 0 or 00 like T□□00. The cancel command becomes valid immediately in the specified block and offset cancel movements occur.

Example of Programming

```
G00 T0202;
G01 X · · · Z · · · F · · ·;
  ·
  ·
  ·
G01 U+ · · · W- · · · F · · · T0216;
  ·
  ·
  ·
G00 X · · · Z · · · T0200; ←———— ① Offset amount is not added to the programmed
                                    axis movement commands and X- and Z-axis
                                    move to the target point specified in the program
```

The commands in block ① can be written in two blocks.

G00 X · · · Z · · · ;
T0200; ← With this command, only cancel movement is executed at a rapid traverse.

The reset operation also cancels the tool position offset function.

(6) Supplements to Tool Position Offset Commands

- Cancel the tool position offset function before specifying the automatic reference point return (G28) operation.
- Cancel the offset by specifying "T□□00" command before specifying the reference point return check (G27) operation. If G27 is executed although the tool position is offset, an error occurs since the offset amount is added to the specified coordinate values.
- Cancel the tool position offset function before specifying M02 or M30.
- If the NC is reset by the reset operation or by the execution of M02 or M30 during the tool position offset mode, the offset function is canceled and offset number becomes "0" or "00".
- The tool position offset function is temporarily canceled if the reference point return (manual or automatic) operation is executed. The processing after that varies depending on the setting for the parameter.

pm4010 D1 = 0	The offset data is saved and recovered in a later block.
pm4010 D1 = 1	The offset data is canceled.

3.4.3 Nose R Offset Function (G40, G41/G42) *

Since the nose of a cutting tool is rounded, overcuts or undercuts occur in taper cutting or arc cutting since offset simply by the tool position offset function is not satisfactory. How such problems occur is shown in Fig. 3.16. The nose R offset function called by G41 and G42 compensates for an error to finish the workpiece to the programmed shape.

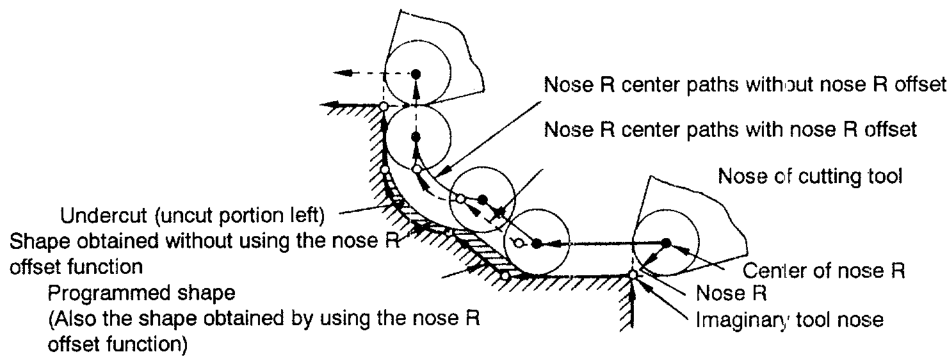


Fig. 3.16 Nose R Offset Function

(1) Nose R Offset Amount

The term "Nose R Offset Amount" means the distance from the too nose to the center of nose R.

(a) Nose R offset data memory

To use the nose R offset function, the nose R amount of the cutting tools to be used must be written to the nose R offset data memory in the NC. The number of pairs of the nose R offset data that can be written to the NC is determined according to the machine model. The memory area used for storing the nose R offset data varies depending whether the basic specification NC is used or option is selected. The allowable maximum value of the nose R offset amount that can be written is $\pm$99.999 mm ($\pm$9.9999 inches).

(b) Nose R offset amount range

The nose R offset amount can be written in the range indicated in Table 3.14.

Table 3.14 Nose R Offset Amount Range

	Nose R Offset Amount Range
mm input	0 to 99.999 mm
inch input	0 to 9.9999 inch

3

(c) Setting the nose R offset amount

For the nose R offset amount, set the radius of the circle of the tool nose without a sign.

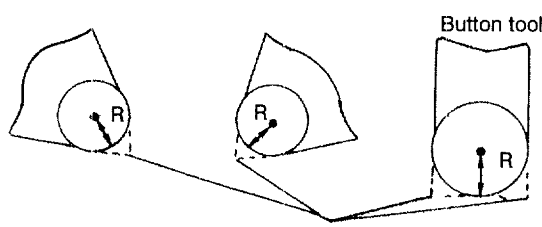


Fig. 3.17 Setting the Nose R Offset Amount and Imaginary Tool Nose

TERM?

◆ Imaginary Tool Nose

With the nose R offset function, the imaginary tool nose which is set at the reference position is taken as the reference point. The present position display given by the NC represents the position of the imaginary tool nose.

(2) Designation of Imaginary Tool Nose Position (Control Point)

(a) Control point memory

The position of the imaginary tool nose viewed from the center of the nose R is expressed using a 1-digit number, 0 to 9. This is called the control point. The control point should be written to the NC memory in advance as with the nose R offset data.

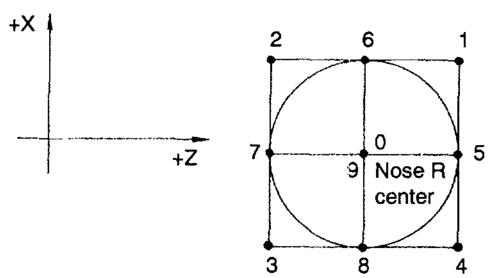


Fig. 3.18 Control Point

(b) Setting the control point

As explained above, the position of the tool nose in reference to the center of nose R is expressed by a 1-digit number as the control point. Control point 0 is treated different manners depending on the setting for a parameter: when "parameter pm4013 D6 = 1", control point 0 is processed in the manner as with control point 9. If "pm4013 D6 = 0", the nose R offset function is invalid. Note that control point data is written using address C.

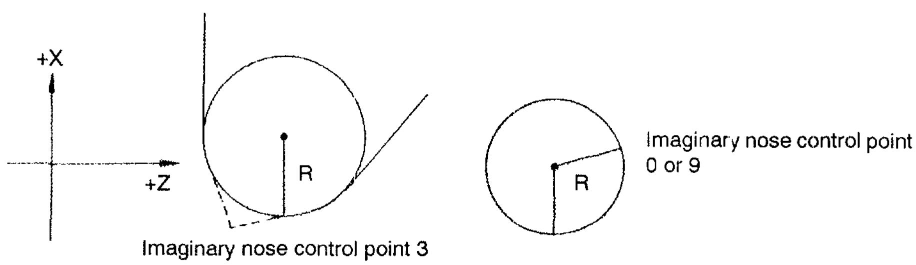
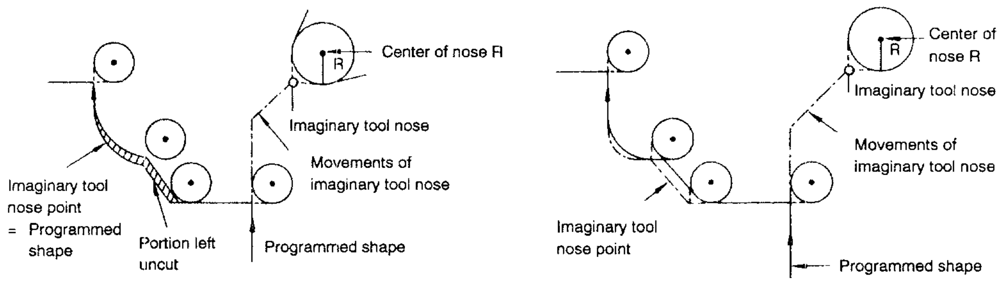


Fig. 3.19 Examples of Control Point Setting

(c) Control points and programs

- When control points 1 to 8 are used, the imaginary tool nose position should be used as the reference to write a program. Write the program after setting a coordinate system.

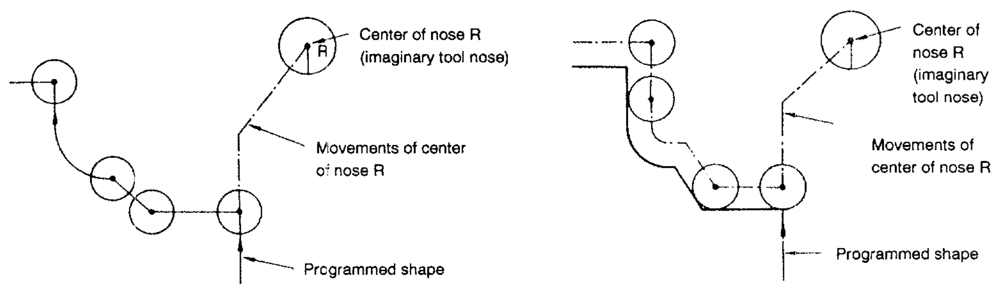


(a) Program without Nose R Offset Function
The imaginary tool nose point follows the programmed shape, causing overcuts and undercuts at tapers and arcs.

(b) Program with Nose R Offset Function
The nose R offset function offsets the tool paths from the programmed shape to eliminated overcuts and undercuts.

Fig. 3.20 Program and Tool Movements for Control Points 1 to 8

- When control points 0 or 9 is used, the center of nose R should be used as the reference to write a program. Write the program after setting a coordinate system. If the nose R offset function is not used, the program shape must not be different from the shape to be machined.



(a) Program without Nose R Offset Function
The center of nose R follows the programmed shape. Therefore, if the coordinate system is set in reference to the center of nose R, the shape to be programmed must be different from the shape to be machined.

(b) Program with Nose R Offset Function
As with the program (b) in Fig. 3.20, appropriate offsetting is made to finish the shape accurately without overcuts and undercuts.

Fig. 3.21 Program and Tool Movements for Control Point 0 or 9

(3) Nose R Offset Commands

(a) Designation of offset amount and control point

To designate the nose R offset amount and control point, specify the offset data memory number where the nose R offset amount and control point are entered by lower 2 digits (3 digits) of a T command.

T □□ □□ (T4-digit command)

Designation of memory number where tool position offset and nose R offset number, and control point are saved

Designation of tool selection

3

(b) Designation of nose R offset function ON and offset direction

To designation ON/OFF of the nose R offset function and the offset direction, use the G codes

Table 3.15 G Codes Used for Turning ON/OFF Nose R Offset Function

G Code	Function	Group
G40	Nose R offset cancel	06
G41	Nose R offset, left (nose R center is at the left side)	06
G42	Nose R offset, right (nose R center is at the right side)	06

G40 and G41/G42 are modal G codes in 06 group, and once designated the specified G code mode remains valid until another G code is specified. When the power is turned ON or the CNC is reset, the G40 mode is set.

To enter the nose R offset mode, specify either G41 or G42 with a T code.

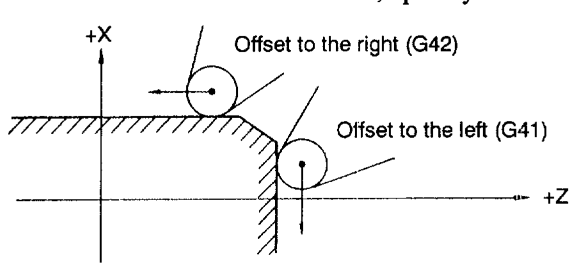


Fig. 3.22 Designation of Nose R Offset Direction

The nose R offset direction can be changed over between "to the right" and "to the left" by specifying G41 or G42 during the execution of a program. It is not necessary to cancel the nose R offset mode by specifying G40 or T00 before changing over direction of offset. To cancel the nose R offset mode, specify G40.

(4) Outline of Nose R Offset Movements

Fig. 3.23 shows how the nose R offset function is executed.

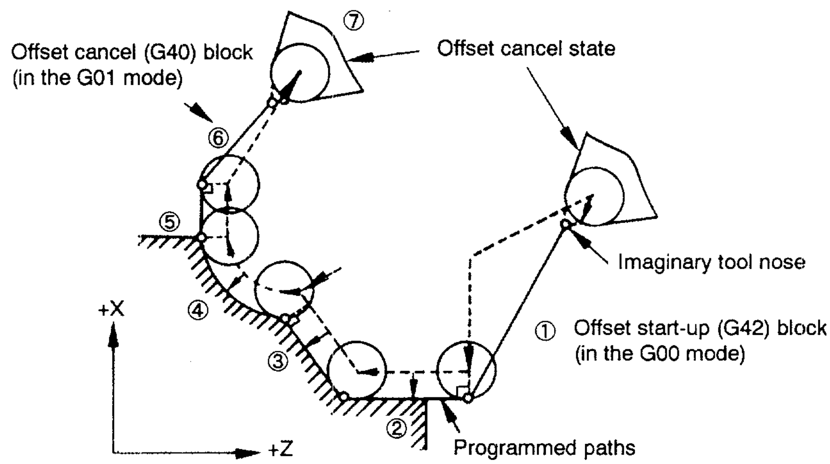


Fig. 3.23 Outline of Nose R Offset Movements (G42, Control Point 3)

- In the offset cancel state, the imaginary tool nose position ⑦ agrees with the point specified in the program ①.
- In the offset mode, the center of nose R is offset by the nose R amount from the programmed paths and it follows the offset paths. Therefore, the imaginary tool nose position does not agree with the programmed point. Note that the present position display shows the position of the imaginary tool nose.
- In the offset mode, at the joints between two blocks, there are two patterns for tool movements: the center of nose R passes the point of intersection between the nose R center paths (M97), or the round-the-arc paths are generated (M98). In Fig. 3.23, round-the-arc path is generated between blocks ③ and ④.
- At the offset start-up block ① and cancel block ⑥, the movements to link the offset mode and offset cancel mode are inserted. Therefore, special attention must be paid for specifying the offset start-up and cancel blocks.

SUPPLEMENT

1. The nose R offset function can be used for circular interpolation specified by radius designation.
2. It is allowed to specify a subprogram (M98, M99) in the offset mode. The nose R offset function is applied to the programmed shape which is offset by the tool position offset function.

(5) Entering the Offset Mode

The offset mode is set when both of a tool offset number (by a T code) and G41 (or G42 to G44) are specified and the nose R offset function is called. More precisely, the offset mode starts at the time when the AND condition of a T code and a G code is satisfied. There are no differences whichever of these codes is specified first (see Fig. 3.24). The initial movement when the offset mode starts in the offset cancel state is called the start-up motion.

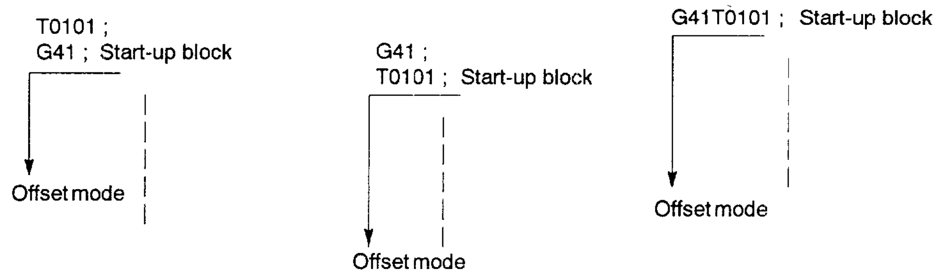


Fig. 3.24 Offset Mode Entry Methods

(6) Start-up of Offset (Axis Movement Command Specified in the Start-up Block)

Since the offset start-up is executed with the offset taken into account, the G code in 01-group must be either G00 or G01. If a G code other than G00 or G01 is specified, alarm "0180" occurs. If the offset starts in the G00 mode, the axes move to the offset point at their individual rapid traverse rates. Therefore, be aware of possible interference of a cutting tool with the workpiece.

There are two types of start-up such as start-up at inside corner and start-up at outside corner.

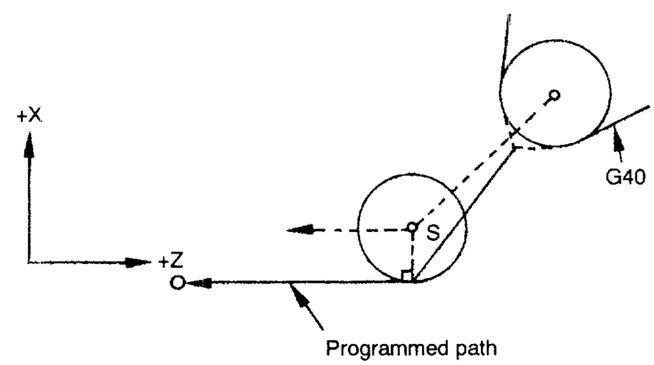
(a) Start-up at inside corner (180° or less)

The center of nose R moves to the offset point (on the normal start point of the vector of movement called up in the block next to the start-up block).

- From straight-line to straight-line

Example of Programming

```
T · · · ;
G01  G42  Z · · · X · · · ;
Z · · · ;
```



Note: "S" indicates the single-block stop point.

Fig. 3.25   Offset Start-up (Straight-Line to Straight-Line)

- From straight-line to arc

Example of programming

```
T · · · ;
G01  G42  Z · · · X · · · ;
G02  Z · · · X · · · R · · · ;
```

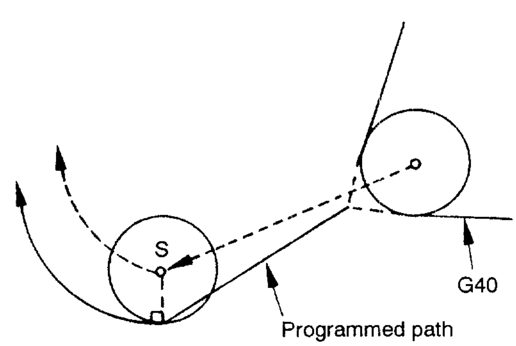


Fig. 3.26   Offset Start-up (Straight Line to Arc)

(b) Start-up at outside corner (180° or larger)

In this case, two kinds of start-up modes (types A and B) are provided and the mode to be used can be selected by the setting for a parameter.

pm4013 D0 = 1	Type A
pm4013 D0 = 0	Type B

- Type A: pm4013 D0 = 1

  The center of nose R moves to the offset point (on the normal start point of the vector of the block next to the start-up block).

  · From straight-line to straight line at outside corner (180° to 270°)

  Example of Programming

  T · · · ;
  G01 G42 Z · · · X · · · ;
  Z · · · ;

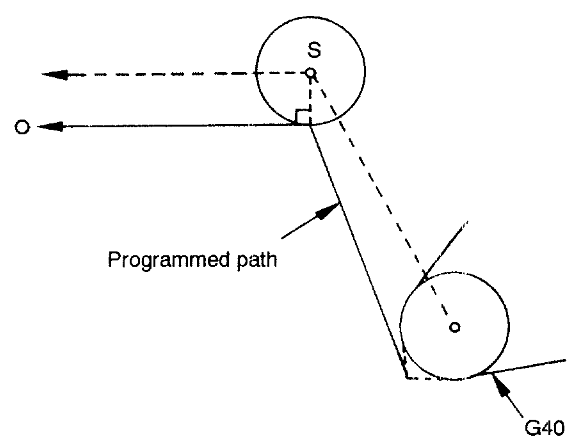


Fig. 3.27 Offset Start-up (Straight-line to Straight-line (1))

· Straight-line to Straight-line at outside corner (270° to 360°)

Example of Programming

T · · · ;
G01 G42 Z · · · X · · · ;
Z · · · ;

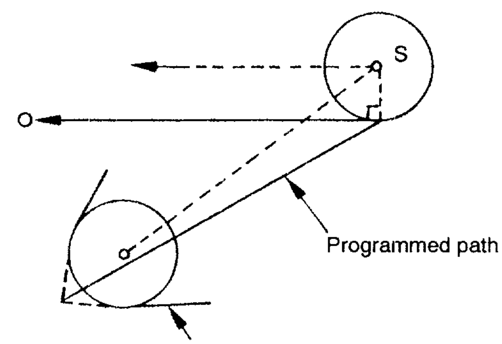


Fig. 3.28 Offset Start-up (Straight-line to Straight-line (2))

· Straight-line to arc at outside corner (270° to 360°)

Example of Programming

T · · · ;
G01 G42 Z · · · X · · · ;
G02 Z · · · X · · · R · · · ;

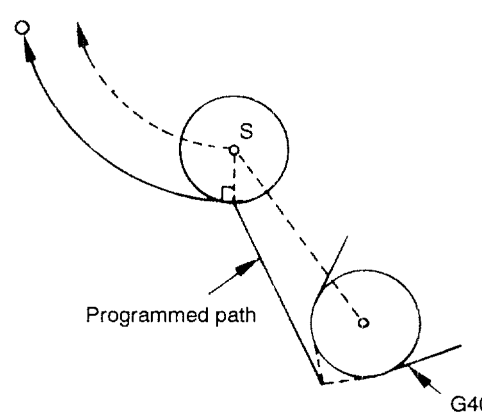


Fig. 3.29 Offset Start-up (Straight-line to Arc)

- Type B: pm4013 D0 = 0

  The offset mode starts in the manner so that overcut does not occur in the movement in the next block.

  · Straight-line to straight-line at outside corner (270° to 360°) in the M97 (round-the-arc motion OFF) mode

  Example of Programming

```
T・・・;
G01 G42 Z・・・X・・・;
Z・・・;
```

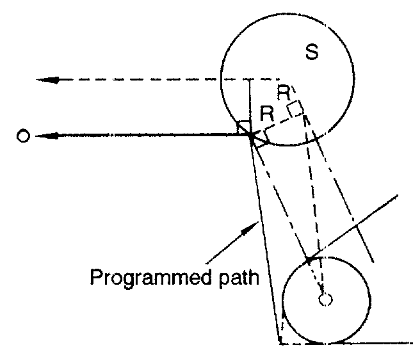


Fig. 3.30 Offset Start-up (Straight-line to Straight-line (1))

  · Straight-line to straight-line at outside corner (180° to 270°) in the M96 (round-the-arc motion ON) mode

  Example of Programming

```
T・・・;
G01 G42 Z・・・X・・・;
Z・・・;
```

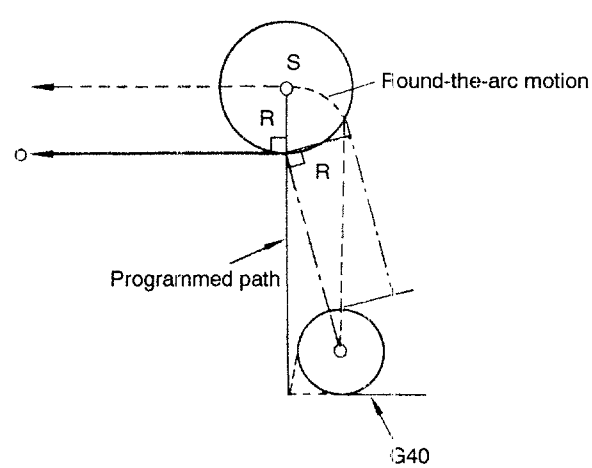


Fig. 3.31 Offset Start-up (Straight-line to Straight-line (2))

・ Straight-line to arc at outside corner (180° to 270°) in the M97 mode

Example of Programming

T・・・;
G01 G42 Z・・・X・・・;
G02 Z・・・X・・・R・・・;

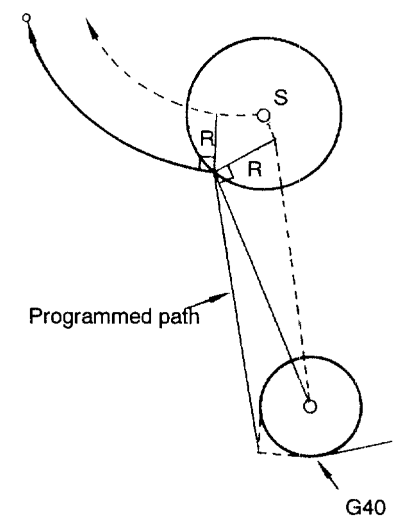


Fig. 3.32 Offset Start-up (Straight-line to Arc (1))

・ Straight-line to arc at outside corner (180° to 270°) in the M96 mode

Example of Programming

T・・・;
G01 G42 Z・・・X・・・;
G02 Z・・・X・・・R・・・;

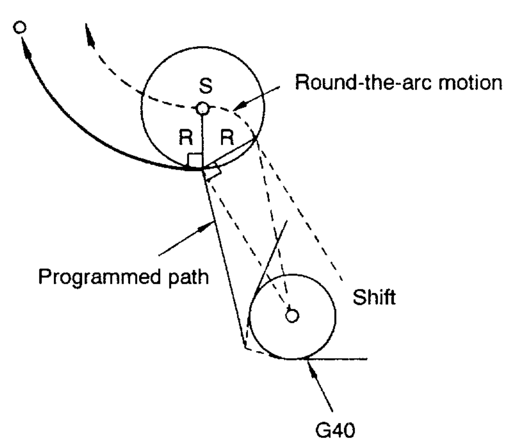


Fig. 3.33 Offset Start-up (Straight-line to Arc (2))

- Straight-line to straight-line at outside corner (270° to 360°) in the M97 mode

In the M96 mode, tool path is generated to connect the end point of the start-up block to the start point of the next block by an arc in the same manner as shown in Figs. 3.31 and 3.33. In the M97 mode, the tool path is generated as shown in Figs. 3.34 and 3.35.

Example of Programming

```
T ··· ;
G01  G42  Z ··· X ··· ;
Z ··· ;
```

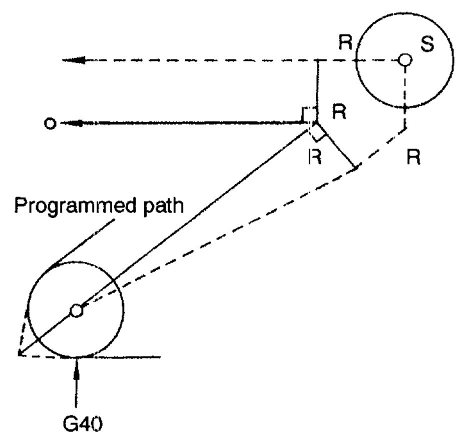


Fig. 3.34　Offset Start-up (Straight-line to Straight-line)

- Straight-line to arc at outside corner (270° to 360°) in the M97 mode

Example of Programming

```
T ··· ;
G01  G42  Z ··· X ··· ;
G02  Z ··· X ··· R ··· ;
```

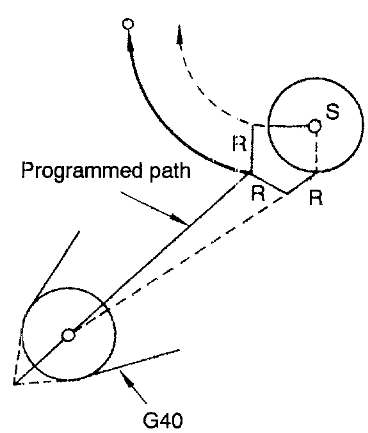


Fig. 3.35　Offset Start-up (Straight-line to Arc)

(7) Start-up of Offset (Axis Movement Command Not Specified in the Start-up Block)

If there are no axis movement commands specified in the start-up block, the center of nose R moves to the point offset by R on the normal at the start point of the next block disregarding of the inside or outside corner and M96 or M97 mode. G codes in the 01 group that can be specified in the start-up block are only G00, G01, and G11, and an alarm "0180" occurs if other G code is specified. When G11 is specified in the start-up block, the offset mode starts when the execution of the first cutting feed command starts. Chamfering operation is executed as in the G01 mode operation in the offset mode.

3

- If the next block calls straight-line motion

  Example of Programming

  T · · · ;
  G01 G42 F · · · ;
  G01 Z · · · X · · · ;

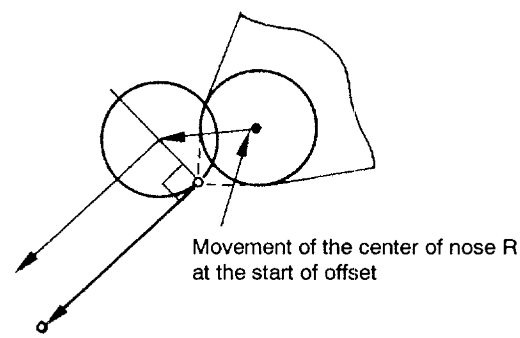


Fig. 3.36 Offset Start-up (No Axis Movement Command in the Start-up Block)

- If the next block calls arc motion

  Example of Programming

  T · · · ;
  G01 G42 F · · · ;
  G03 Z · · · X · · · I · · · ;

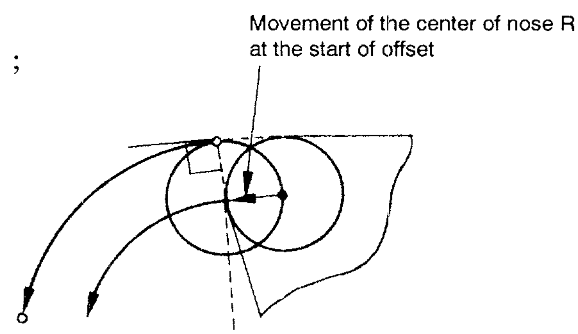


Fig. 3.37 Offset Start-up (No Axis Movement Command in the Start-up Block)

(8) Axis Movements in the Offset Mode

Once the tool radius offset mode is set by the execution of G41 or G42, the center of nose R moves along the paths offset by R from the programmed paths until the tool radius offset mode is canceled by G40. Since the offset paths are automatically generated by the NC, the program should simply define the shape to be machined. The tool paths are controlled according to the angle made between the specified programmed paths.

(a) Inside corner (smaller than 180°)

The center of nose R moves to the position obtained by the calculation for the point of intersection.

- Straight-line to straight-line

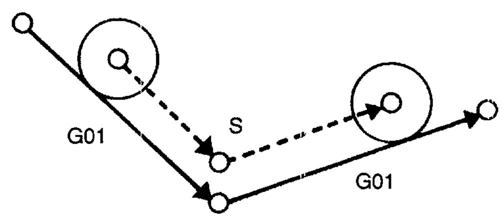


Fig. 3.38 Straight-line to Straight-line

- Straight-line to Arc

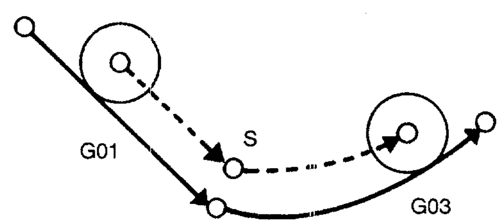


Fig. 3.39 Straight-line to Arc

- Arc to arc

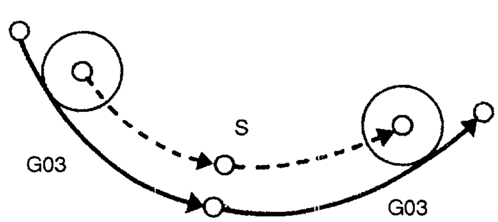


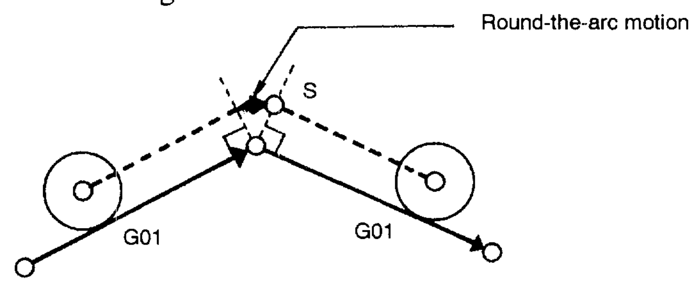
Fig. 3.40 Arc to Arc

(b) Outside corner (larger than 180°)

For this offset, two types of offset modes are provided and the offset mode to be used can be selected by the designation of an M code.

M96	Tool radius offset round-the-arc ON
M97	Tool radius offset round-the-arc OFF (calculation is executed to obtain the point of intersection)

- Tool movements in the M96 (tool radius offset round-the-arc ON) mode
  - Straight-line to straight-line



Note: In this case, round-the-arc motion of a cutting tool is included in the preceding block.

Fig. 3.41 Round-the-arc Motion (Straight-line to Straight-line)

  - Straight-line to arc

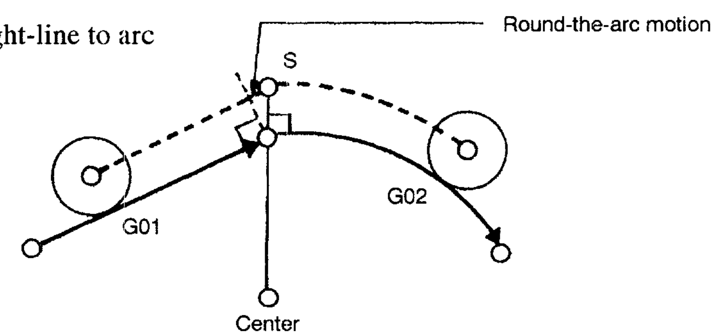


Fig. 3.42 Round-the-arc Motion (Straight-line to Arc)

  - Arc to arc

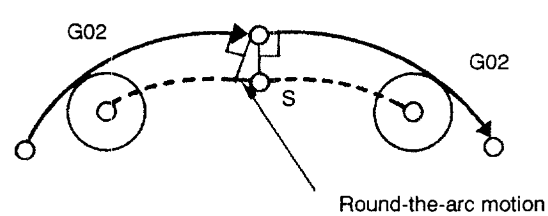


Fig. 3.43 Round-the-arc Motion (Arc to Arc)

- Tool movements in the M97 (tool radius offset round-the-arc OFF) mode
  - Straight-line to straight-line at outside corner (180° to 270°)

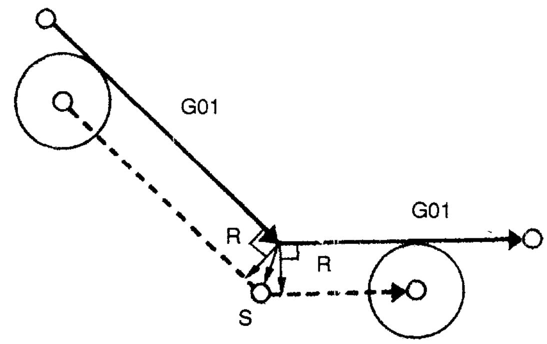


Fig. 3.44 Offset Motion (Straight-line to Straight-line)

- Straight-line to arc at outside corner (180° to 270°)

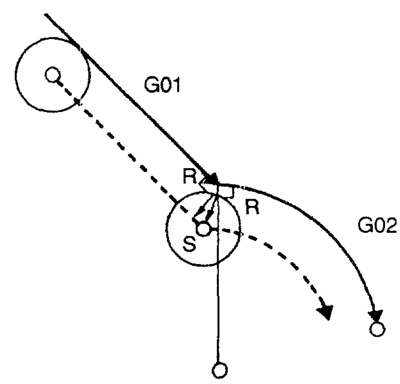


Fig. 3.45 Offset Motion (Straight-line to Arc)

- Arc to arc at outside corner (180° to 270°)

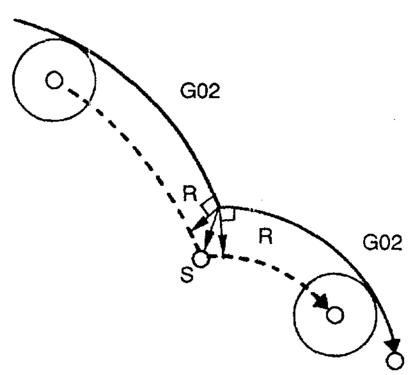


Fig. 3.46 Offset Motion (Arc to Arc)

· Straight-line to straight-line at outside corner (270° to 360°)

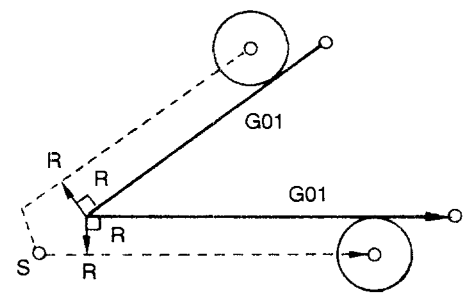


Fig. 3.47 Offset Motion (Straight-line to Straight-line)

· Straight-line to arc at outside corner (270° to 360°)

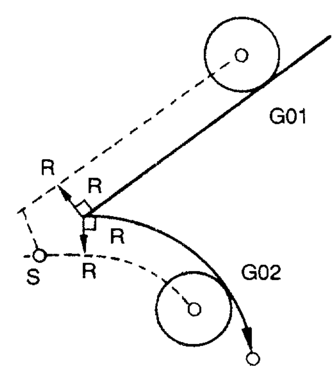


Fig. 3.48 Offset Motion (Straight-line to Arc)

· Arc to arc at outside corner (270° to 360°)

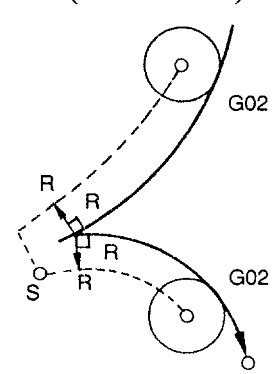


Fig. 3.49 Offset Motion (Arc to Arc)

(c) Special commands and motion in the offset mode

Special commands in the offset mode include the temporary cancel command (type I and II) and re-designation of G41 and G42.

- Temporary cancel command (type I)

  The offset mode is temporarily canceled and offset mode cancel motion is executed if the following commands are specified in the offset mode. In this temporary cancel motion, the center of nose R moves to the point offset by R on the normal at the end point of the preceding block.

  - Three or more blocks that do not include axis movement commands can be specified consecutively. The commands that do not call for axis movement includes G04 (dwell), independent M codes, S code, and axis command with 0 movement distance.
  - It is allowed to specify buffering prohibiting block. The blocks that prohibit buffering include M codes (M00, M01, M02, M30), buffering prohibiting M code set by a parameter, G36 to G39 (stored stroke limit area ON, OFF), and G10 (tool offset amount setting).

  Example of Programming

```
N1 G01 X ··· Z ··· ;
N2 G04 P ··· ;
N3 M15 ;
N4 S1000 ;
N5 G01 X ··· Z ··· ;
```

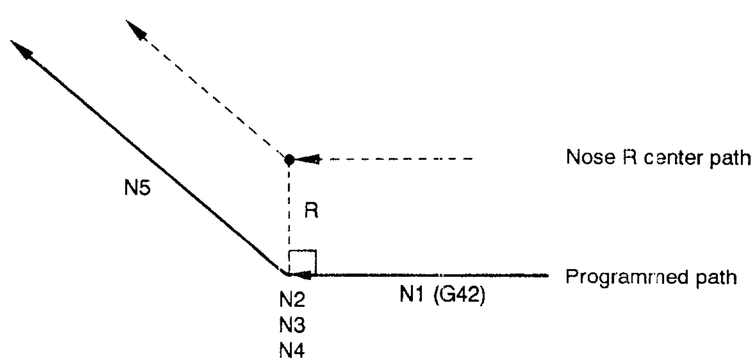


Fig. 3.50 Nose R Offset Temporary Cancel Command
(Consecutive 3 Blocks Not Including Axis Movement Commands)

Example of Programming

N1 G01 X ··· Z ··· ;
N2 G01 ··· ;
N3 G01 X ··· Z ··· ;

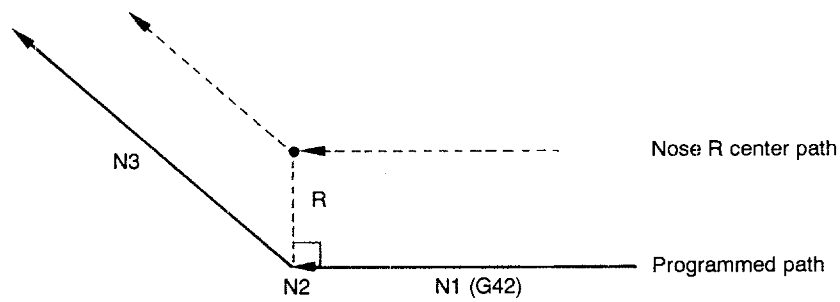


Fig. 3.51 Fig. 3.51 Nose R Offset Temporary Cancel Command (Buffering Prohibiting Block)

- Temporary cancel command (type II)

  With this type of command, the center of nose R moves so that the imaginary tool nose moves to the end point of specified in the program by the automatic reference point return command (G28, G30), thread cutting command (G32, G34, G92), and coordinate system setting command (G50, etc.). The offset mode is automatically set again from the block next to the one that contains such commands. Note that axis movement does not occur if the coordinate system setting command is specified.

- Re-designation of G41 and G42

  It is possible to move the center of nose R to the point offset by R on the normal at the start point of the next block by specifying G41 or G42 in the offset mode, disregarding of the outside or inside corner and M96 or M97 mode.

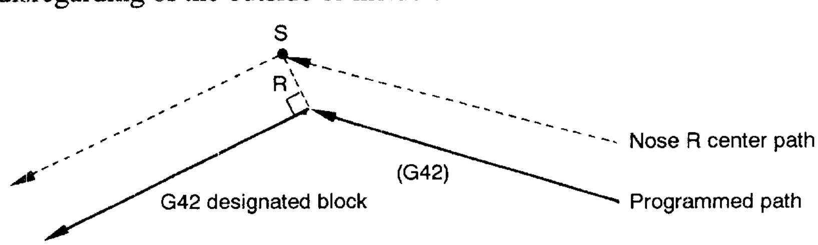


Fig. 3.52 Re-designation of G41, G42 in the Nose R Offset Mode

(9) Axis Movements in the Offset Mode (No Axis Movement Commands)

In the nose R offset mode, the NC generates the tool paths by buffering the data of two blocks. If a block not including axis move commands is read, the NC reads one more block to generate the offset tool paths. Designation of such a block which does not include axis move commands is allowed in the tool radius offset mode for up to two consecutive blocks.

After the designation of G41 or G42, there must not be three or more consecutive blocks that do not include the movement commands of the axes in the offset plane.

(a) Consecutive three or more blocks not including axis move commands

If three or more blocks not containing axis move commands in the offset plane are given consecutively, the cutting tool is moved to the position offset normally by the specified offset amount at the end point of the block immediately preceding such blocks.

Example of Programming

```
T・・・;
G01 G41 Z・・・X・・・F・・・;
Z・・・X・・・;
 .
 .
 .
Z・・・X・・・・;
G04 P1000;
Z・・・X・・・;
 .
 .
 .
Z・・・X・・・;
M・・・;
S・・・;
Z・・・・X・・・;
 .
 .
 .
Z・・・X・・・;
G40 Z・・・X・・・;
```

Blocks not including axis movement commands in the offset plane
(If such blocks continue up to two blocks, the NC can generate tool paths without a problem.)

(b) Insertion of dummy block

If there are no axis move commands in three consecutive blocks, the cutting tool is positioned on the normal end point of the block immediately preceding such blocks. If it is impossible to specify move commands of the axes in the offset plane due to the retraction motion of the third axis or other reasons and if normal positioning is not desirable, a dummy block that includes I or K can be inserted in the program. The dummy block does not call up actual axis movements, but it only gives the data necessary for the calculation to generate the offset tool paths. In the example program given below, a dummy block specifying the same movements as given in the block (N020) where the axis movement restarts in the ZX plane after the Z-axis movement is inserted in a program; addresses I and K are used in the dummy block.

Example of Programming

```
N001 G17 G01 G41 Z · · · X · · · F · · · T · · · ;
N002 Z · · · X · · · ;
 ·
 ·                                   XY plane
 ·
N010 Z · · · Y · · · ;
N011 K · · · I · · · ;               ← Dummy block
N012 M · · · ;
 ·
 ·                                   Z-axis (3 blocks or more)
 ·
N019 S · · · ;
N020 Z · · · X · · · ;
 ·
 ·                                   XY plane
 ·
N029 Z · · · X · · · ;
N030 G40 Z · · · X · · · ;
```

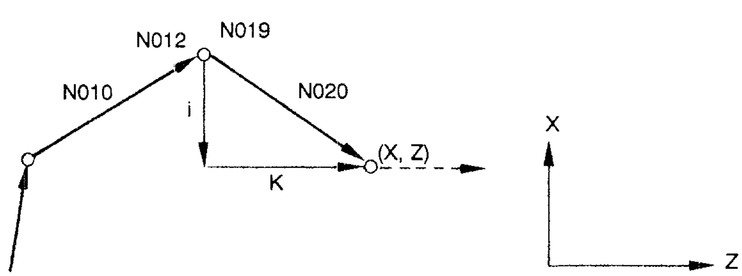


Fig. 3.53 Insertion of Dummy Blocks

- In a dummy block, addresses I and K are used corresponding to X- and Z-axis. Specify these addresses meeting the plane which has been selected as the offset plane. Note that in dummy blocks, commands should be given in incremental commands. With the example program indicated above, if "Z · · · X · · ·" in N20 are specified in absolute values, change them to equivalent incremental values.
- If the object of the dummy block is circular interpolation, enter the dummy block as shown in the example program given below. Insert the dummy block in which the straight line expressing the tangential direction at the start point of circular interpolation is specified as shown in Fig. 3.54. The cutting tool moves to point A as shown in Fig. 3.55 by the execution of the dummy block so that the following circular interpolation can be executed.

3

Example of Programming

```
N050 G01 Z · · · X · · · ;
N051 G01 K(b) I(-a) ; ← Dummy block
N052 M · · · ;        ⎫
 ·                    ⎪
 ·                    ⎬ Z-axis
 ·                    ⎪
N059 S · · · ;        ⎭
N060 G03 Z · · · X · · · K(a) I(b) ; ← Circular interpolation
N061 G01 Z · · · X · · · ;
```

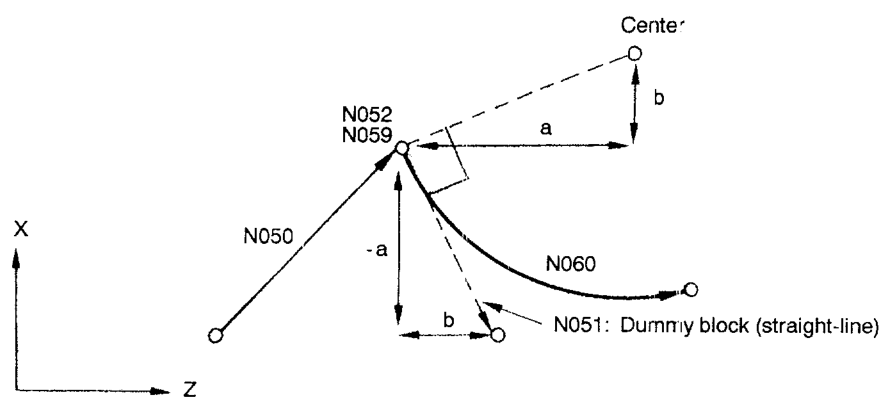


Fig. 3.54 Insertion of Dummy Block

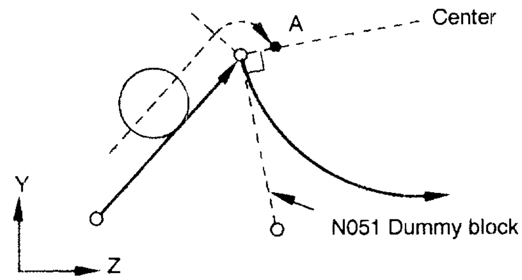


Fig. 3.55 Movement to Point A by Execution of Dummy Block

- If I or K is specified when canceling the offset mode, the offset position is corrected from point ① to point ② according to the direction specified by these addresses.

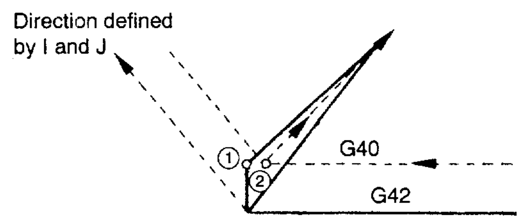


Fig. 3.56 Correction of Offset Position

(10) Switching the G41 and G42 in the Offset Mode

The direction of offset (left side and right side) can be directly switched without canceling the offset mode. There are two kinds of G41/G42 switching methods (types A and B) and the method to be used can be selected by the setting for a parameter.

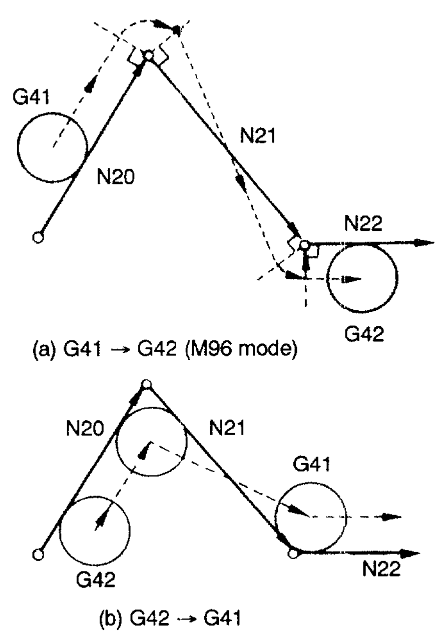
pm4013 D1 = 1	Type A
pm4013 D1 = 0	Type B

(a) Type A: pm4013 D1 = 1

The offset direction is switched at the start and end of the block in which the switching of the offset direction is specified.

Example of Programming

```
N10 T · · · ;
N11 G41 (G42) Z · · · X · · · ;
  .
  .
  .
N20 G01 Z · · · X · · · F · · · ;
N21 G42 (G41) Z · · · X · · · ;    ← Offset direction switching block
N22 Z · · ·;
```



Note: If the contents of N21 block are expressed in two blocks as indicated below
G42 (or G41) ;
Z · · · X · · · ;
the offset direction is switched in the same manner.

Fig. 3.57 Switching the Offset Direction at the Start and End of the Block

(b) Type B: pm4013 D1 = 0

Direction of offset is switched at the point of intersection of the offset tool paths.

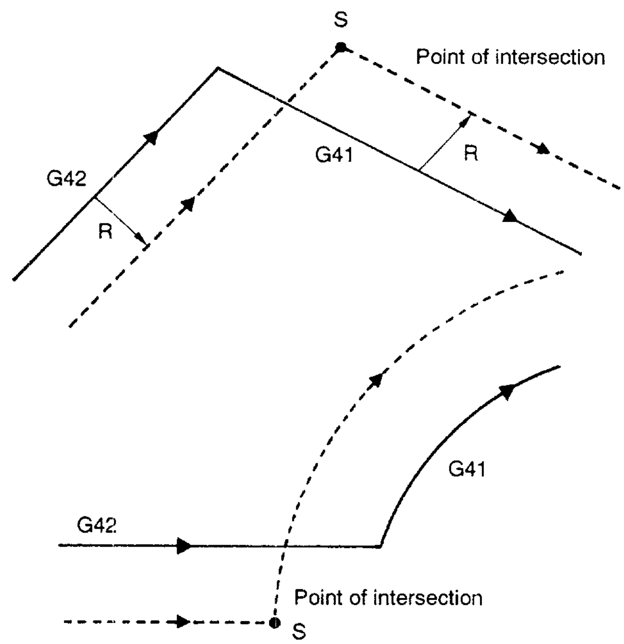


Fig. 3.58 Switching of the Offset Direction at Point of Intersection of Offset Tool Paths

If there is no point of intersection, the offset direction is switched according to type A.

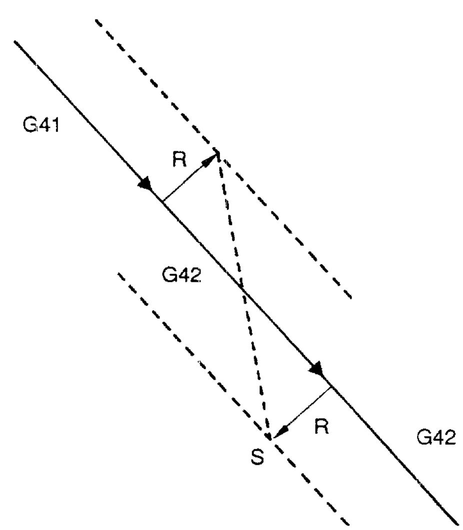


Fig. 3.59 Switching of the Offset Direction when There is No Point of Intersection

(11) Changing the Tool Offset Amount in the Offset Mode

There are two kinds of offset amount changing methods (types A and B), and the method to be used can be selected by the setting for a parameter.

pm4013 D2 = 0	Type A
pm4013 D2 = 1	Type B

(a) Type A: pm4013 D2 = 0

When a new T code is specified, the new offset data are calculated from the axis move commands given in the block including the new D code and the next block.

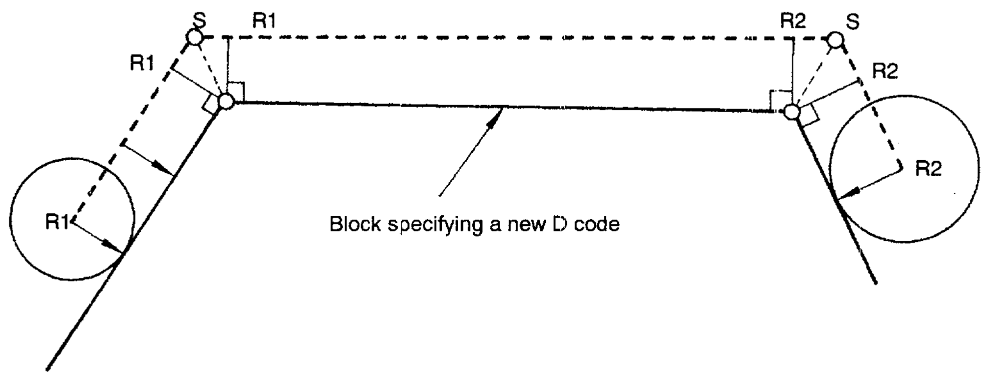


Fig. 3.60 Calculating the New Offset Data from the Axis Move Commands in the New T Code Specifying Block and the Next Block

(b) Type B: pm4013 D2 = 1

When a new T code is specified, the new offset data are calculated from the axis move commands given in the block including the new T code and the preceding block.

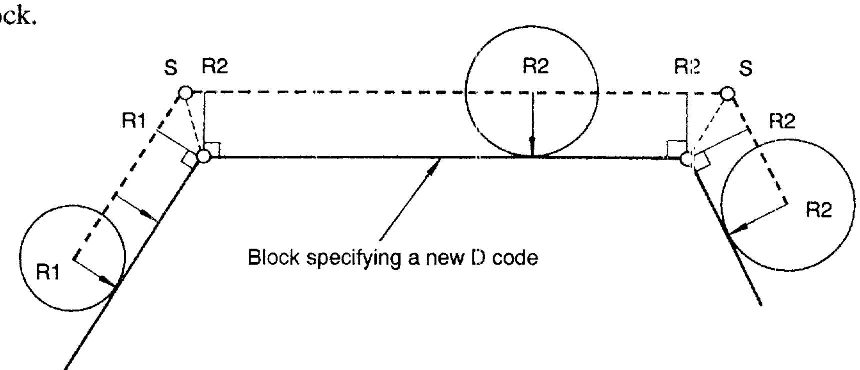


Fig. 3.61 Calculating the New Offset Data from the Axis Move Commands in the New T Code Specifying Block and the Previous Block

3

(12)Canceling the Offset Mode

If "T**00" or G40 command is specified in the offset mode, the offset mode is canceled from the block in which such a command is specified. The first block of the offset cancel mode is called the cancel block.

(13)Offset Mode Cancel Movements (Axis Movement Command Specified in the Cancel Block)

(a) Canceling the offset mode at inside corner (smaller than 180°)

The center of nose R moves to a point on the normal at the end of the block immediately before the cancel block. In the cancel block, the imaginary nose R agrees with the command value.

- Straight-line to straight-line

Example of Programming

```
(G42)
G01 X · · · Z · · · ;
    X · · · Z · · · ;
G40 X · · · Z · · · ;
```

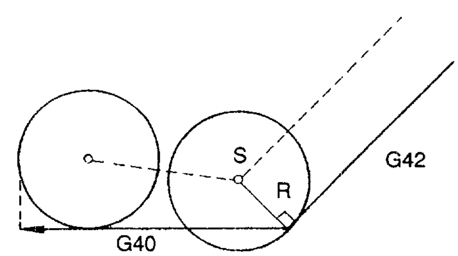


Fig. 3.62 Canceling the Offset Mode at Inside Corner (Straight-line to Straight-line)

- Arc to straight-line

Example of Programming

```
(G42)
 .
 .
 .
G02 X · · · Z · · · R · · · ;
G40 X · · · Z · · · ;
```

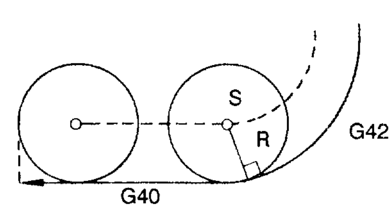


Fig. 3.63 Canceling the Offset Mode at Inside Corner (Arc to Straight-line)

(b) Canceling the offset mode at outside corner (larger than 180°)

There are two types (types A and B) of offset mode cancellation axis movement patterns, and the pattern to be used can be selected by the setting for a parameter. For this selection, the same parameter as used to select the start-up mode is used.

pm4013 D0 = 1	Type A
pm4013 D0 = 0	Type B

- Type A: pm4013 D0 = 1

  The center of nose R is moved normally to the offset position at the end point of the block immediately before the offset mode cancellation block and then to the end point specified in the program.

  · Straight-line to straight-line at outside corner (180° to 270°)

  Example of Programming

```
(G41)
 .
 .
 .
G01 X ··· Z ··· F ··· ;
G40 X ··· Z ··· ;
```

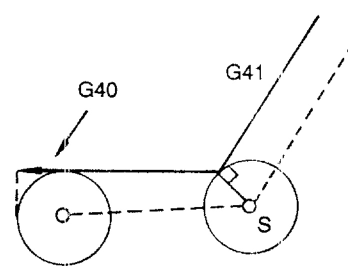


Fig. 3.64 Straight-line to Straight-line at Outside Corner

  · Straight-line to straight-line at outside corner (270° to 360°)

  Example of Programming

```
(G41)
 .
 .
G01 X ··· Z ··· F ··· ;
G40 X ··· Z ··· ;
```

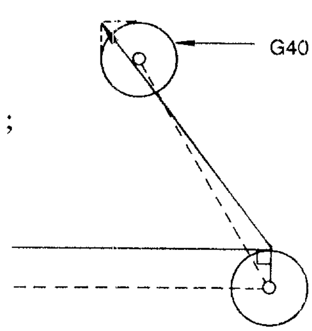


Fig. 3.65 Straight-line to Straight-line at Outside Corner

- From arc to straight-line at outside corner (180° to 270°)

Example of Programming

```
G42
 .
 .
G02 X··· Z··· I··· K···;
G01 G40 X··· Z···;
```

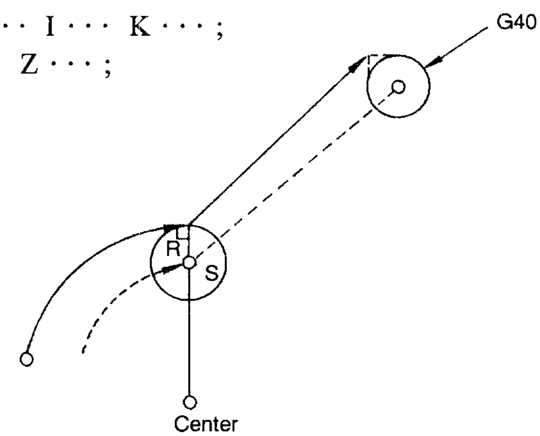


Fig. 3.66 From Arc to Straight-line at Outside Corner

- Arc to straight-line at outside corner (270° to 360°)

Example of Programming

```
G42
 .
 .
G02 X··· Z··· I··· K···;
G01 G40 X··· Z···;
```

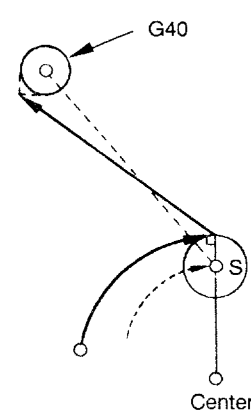


Fig. 3.67 Arc to Straight-line at Outside Corner

- Type B: pm4013 D0 = 0

  The center of nose R moves to the point obtained by the calculation of the point of intersection using the axis move commands in the offset mode cancel block and those in the block immediately before this block, and then to the point specified in the program.

  · Straight-line to straight-line at outside corner (180° to 270°)

    Example of Programming

    G42
    .
    .
    G01 Z · · · F · · · ;
    G40 X · · · Z · · · ;

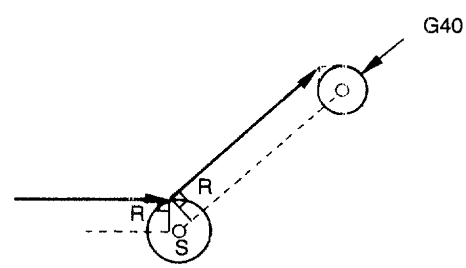


Fig. 3.68 Straight-line to Straight-line at Outside Corner

  · Straight-line to straight-line at outside corner (270° to 360°)

    Example of Programming

    G42
    .
    .
    G01 Z · · · F · · · ;
    G40 X · · · Z · · · ;

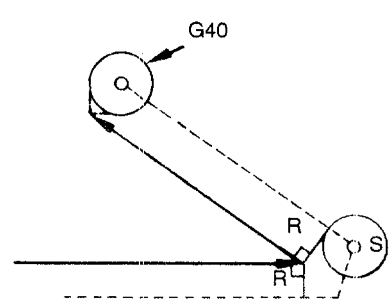


Fig. 3.69 Straight-line to Straight-line at Outside Corner

· Arc to straight-line at outside corner (180° to 270°)

Example of Programming

G42

.

.

G02 X · · · Z · · · I · · · K · · · ;

G01 G40 X · · · Z · · · ;

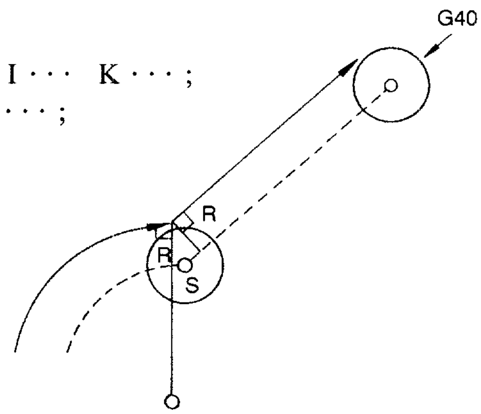


Fig. 3.70 Arc to Straight-line at Outside Corner

· Arc to straight-line at outside corner (270° to 360°)

Example of Programming

G42

.

.

.

G02 X · · · Z · · · I · · · K · · · ;

G01 G40 X · · · Z · · · ;

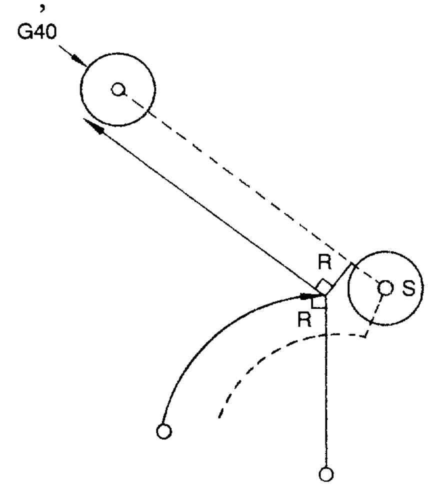


Fig. 3.71 Arc to Straight-line at Outside Corner

(c) Example of programs

Example of Programming 1

(G42, control point 3)

.

.

.

① G02

② G01 U20. F0.25 ;

③ G00 G40 X110. Z40. ; ← Offset mode cancel movement (in the G00 mode)

④ T0100 ;

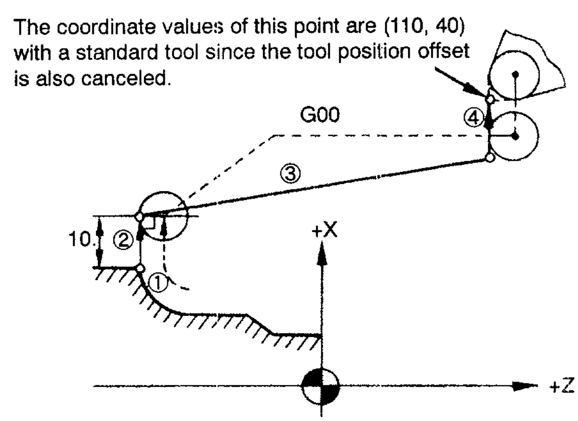


Fig. 3.72 Example of Programming 1

Example of Programming 2

(G42, control point 3)

.

.

.

① G01 X ··· Z ··· F ··· ;

② G01 U24. F0.3 ;

③ G01 G40 X80. Z40. F6. ; ← Offset mode cancel movement (in the G01 mode)

④ G00 T0200 ;

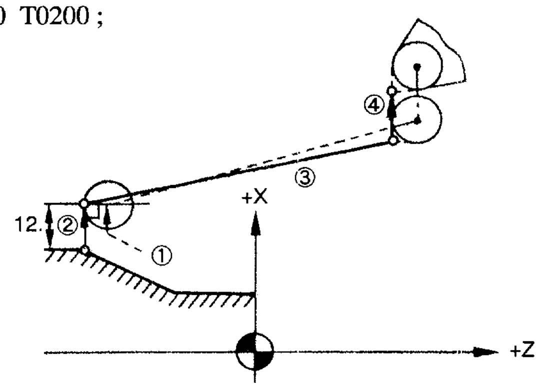


Fig. 3.73 Example of Programming 2

(14) Offset Mode Cancel Movements (Axis Movement Command Specified in the Cancel Block)

If the G40 block that cancels the offset mode does not include an axis movement command, the offset cancel movement in which the imaginary tool nose moves to the end point specified in the program. Since G40 (or T□□00) command calls such axis movements, G00 or G01 must be specified in a preceding or the same block that contains it. If other than G00, G01, or G11 is specified as a G code in 01 group, an alarm "0181" occurs.

Example of Programming

(G41, control point 4)

.

.

.

① G01 X · · · Z · · · F · · · ;
② G01 G40 F · · · ;
G00 T0300 ;

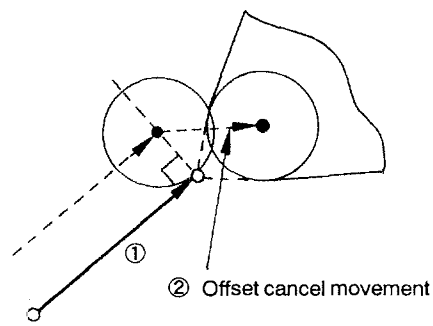


Fig. 3.74 Offset Mode Cancel Movements

(a) Canceling the offset mode by T□□00 command

If the nose R offset mode is canceled by "T□□00" command, tool position offset cancel movements occur simultaneously with the nose R offset cancel movements. In these movements, the imaginary nose R moves to the position, where the tool position offset is canceled, specified in the program. If these two cancel movements should not be executed simultaneously, use G40 to cancel the nose R offset mode.

(b) Canceling the offset mode by "G40 X · · · Z · · · I · · · K · · · ;"

A special cancel movement can be called by specifying I and K commands with G40 in the same block. The point of intersection is calculated using the commands specified in the block immediately preceding the G40 block and the vector defined by I and K specified in the G40 block; the offset mode is canceled in the manner the center of nose R passes through the calculated point of intersection.

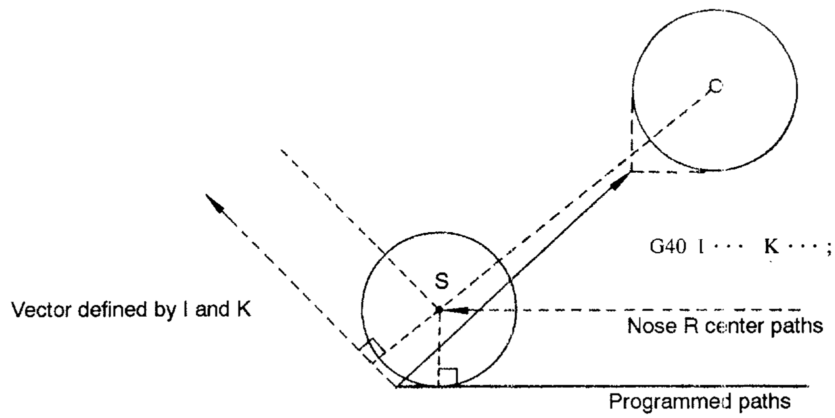


Fig. 3.75 Canceling the Offset Mode by G40 I · · · K · · ·

In this offset mode cancel movements, the center of nose R passes the point of intersection determined by the two blocks disregarding of whether the corner made by the preceding block and the vector (defined by I and K) is inside or outside corner, and also disregarding of the M96 and M97 mode. If the point of intersection cannot be calculated, the center of nose R moves to the point offset by R on the normal at the end point of the preceding block as shown in Fig. 3.76.

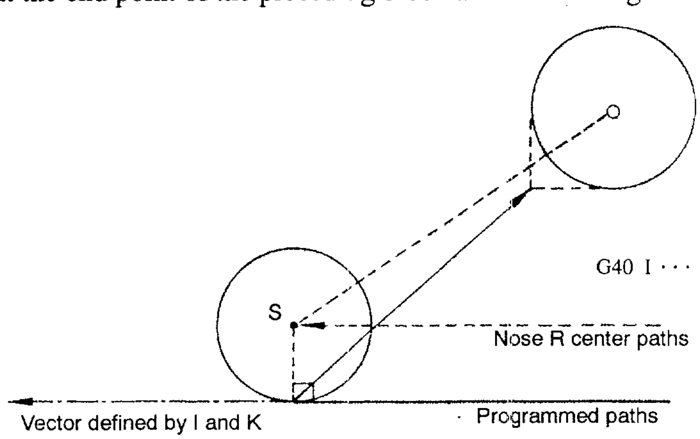


Fig. 3.76 Canceling the Offset Mode by G40 I · · · K · · ·
(No Point of Intersection)

3

Example of Programming

```
N1  G50  X140.  Z20. ;
N2  G00  S1700  M03  T0202 ;
N3  (G00)  G42  X0  Z5. ; ———————— Nose R offset start-up block
N4  G01  Z0  F0.2 ;
N5  X20. ;
N6  Z-20. ;
N7  X30.  W-15.  S1100 ;
N8  G12  W-20.  I3. ; ——————————— (Rounding)
N9  G11  X50.  K-3.  S700 ; ——————— (Chamfering)
N10  G01  Z-70. ;
N11  G02  X90.  Z-90.  R20.  S360 ; —— (Arc designated by R command)
N12  G01  X110.  S300 ;
N13  G04  U0 ; ——————————————— (Dwell for making sharp edge)
N14  (G01)  Z-110. ;
N15  X120. ;
N16  G00  X140.  Z30.  T0200 ; ——— Nose R offset cancel block
N17  G40 ;
```

In the nose R offset mode (N3 to N16)

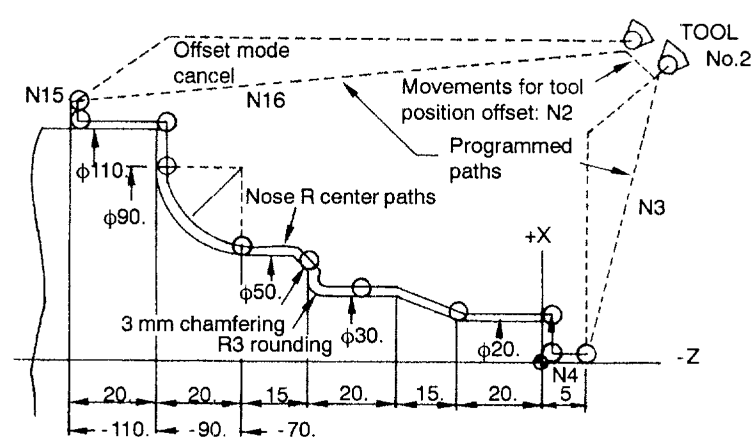


Fig. 3.77 Example of Programming

(15) Interference Check

The interference check function prevents the cutting tool from cutting into or interfering with the workpiece. However, this check is not made at the start-up of the tool radius offset mode. The process to be taken in case of interference is detected from the blocks read into the buffer memory can be selected by the setting for a parameter as indicated below.

Type A	pm4013 D4 = 1	Alarm occurs and operation stops.
Type B	pm4013 D4 = 0	Tool paths are corrected.

Whether or not the interference check is executed is also determined by the setting for a parameter.

pm4013 D3 = 0	Interference check is not made.
pm4013 D3 = 1	Interference check is made.

The illustration in Fig. 3.78 shows how the interference check function operates. In reference to the programmed paths, offset tool paths are generated according to the set nose R offset amount. With nose R offset amount Ra, tool paths $f_1 \rightarrow f_2 \rightarrow f_3 \rightarrow f_4 \rightarrow f_5 \rightarrow f_6$ is generated and with Rb tool paths $f_1' \rightarrow f_6'$ are generated. However, in the tool paths generated with the offset amount of Rb, path $f_3' \rightarrow f_4'$ shows 180° reversed movement from the correct programmed path direction $f_3 \rightarrow f_4$. The function assumes this interference and generates an alarm.

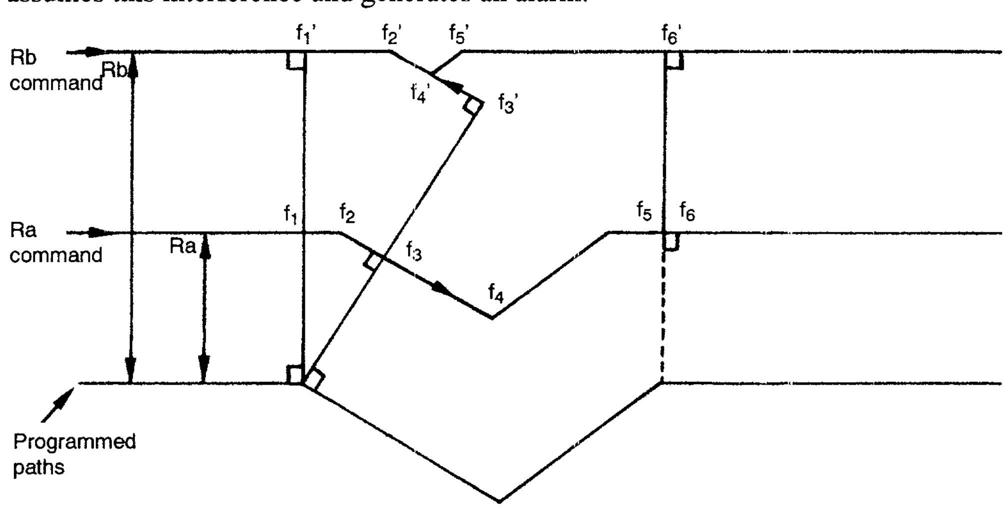


Fig. 3.78 Definition of Interference

(a) Type A: pm4013 D4 = 1 Generation of alarm

The following programs give examples in which the function determines that interference (overcuts) will occur due to considerable differences between the programmed paths and the tool paths generated after offset.

- Example program 1

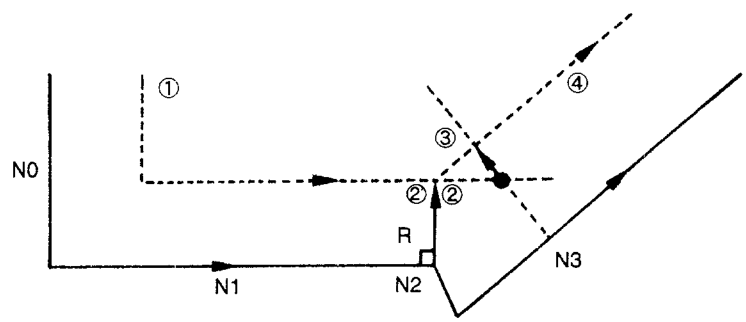
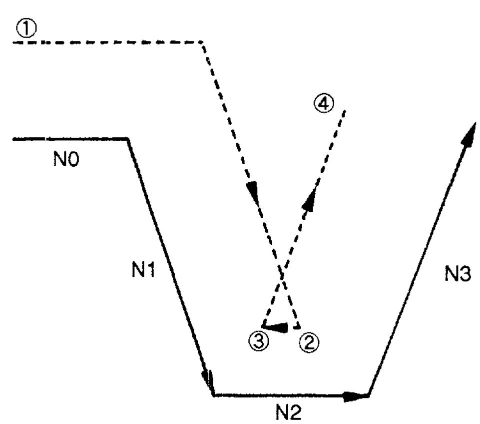


Fig. 3.79 Example Program 1

- Example program 2



Note: Since the cutting tool cuts into the workpiece excessively at the end point② of block N1, alarm "0187" occurs. The operation stops when the cutting tool reaches the end point of block N0.

Fig. 3.80 Example Program 2

(b) Type B: pm4013 D4 = 0 Correcting the tool paths

If the function detects possible interference after the calculation of the offset tool paths, the function clears the nose R center paths that might cause interference and generates the paths that are free of interference.

- Generating interference-free paths for straight-line to straight-line motion

  For the programmed paths as shown in Fig. 3.81, three points f_1, f_2, and f_3 are generated at the joint of blocks N1 and N2 according to the nose R offset function.

  Point f_4 is also generated at the joint of N2 and N3. Interference check is made using these four points f_1 to f_4 and the points causing interference are erased one by one until the tool paths that are free of interference are generated.

  Checked for f_3 - f_4: Erasing f_3 since interference occurs.
  Checked for f_2 - f_4: Erasing f_2 since interference occurs.
  Checked for f_1 - f_4: No interference

  Tool paths are generated as ① → f_1 → f_4 → ②.

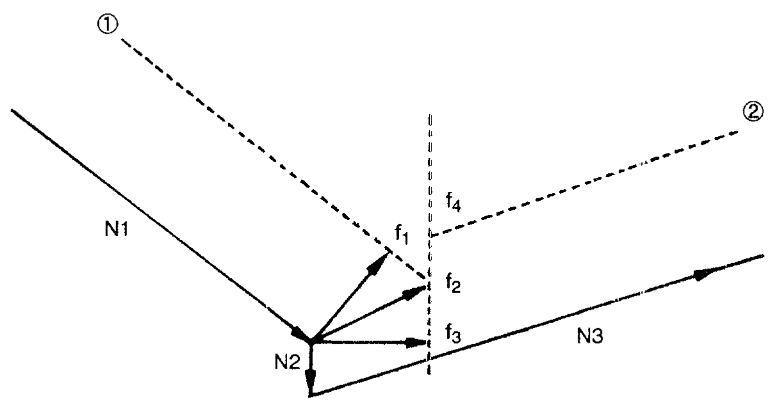


Fig. 3.81 Generating Tool Paths without Interference (Straight-line to Straight-line)

3

3

- Generating interference-free paths for arc to arc motion

For the programmed paths as shown in Fig. 3.82, four points f_1, f_2, f_3, and f_4 are generated at the joint of N1 and N2 according to the nose R offset function. At the joint of N2 and N3, another four points f_5 to f_8 are generated. Interference check is made using these eight points f_1 to f_8 and the points causing interference are erased one by one until the tool paths that are free of interference are generated.

Checked for f_4 - f_5: f_4 and f_5 are erased since interference occurs.
Checked for f_3 - f_6: f_3 and f_6 are erased since interference occurs.
Checked for f_2 - f_7: No interference

Tool paths are generated as $f_1 \rightarrow f_2 \rightarrow f_7 \rightarrow f_8$.

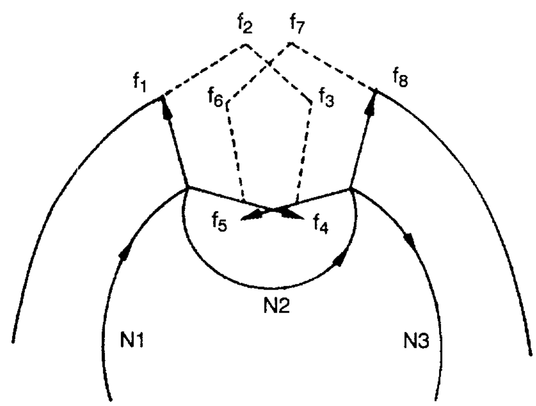


Fig. 3.82 Generating Interference-free Paths for Arc to Arc Motion

- Example where interference-free tool paths cannot be generated

For the programmed paths as shown in Fig. 3.83, three points f_1, f_2, and f_3 are generated at the joint of N1 and N2 according to the nose R offset function. At the joint of N2 and N3, another three points f_4 to f_6 are generated. Interference check is made using these six points f_1 to f_6 and the points causing interference are erased one by one until the tool paths that are free of interference are generated.

Checked for f_3 - f_4: f_3 and f_4 are erased since interference occurs.
Checked for f_2 - f_5: f_2 and f_5 are erased since interference occurs.
Checked for f_1 - f_6: f_1 and f_6 are erased since interference occurs.

Occurrence of alarm ("0188"): Operation stops when the cutting tool is positioned at the start point of N1 block.

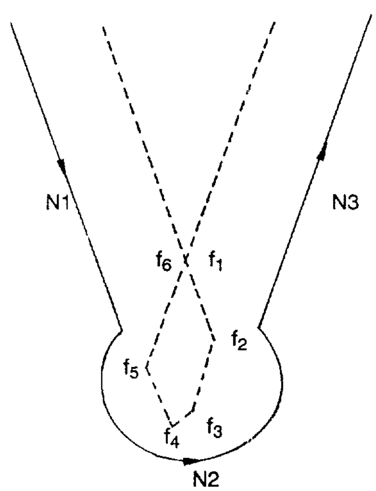


Fig. 3.83 Example where Interference-free Tool Paths cannot be Generated

(16) Internal M codes for judging round-the-arc ON/OFF (M96/M97)

M96 and M97 commands are modal and the M96 mode is set when the power is turned ON. The round-the-arc judgment internal M codes are indicated in Table 3.16.

Table 3.16 Round-the-arc Judgment Internal M Codes

M96	Nose R offset round-the-arc ON
M97	Nose R offset round-the-arc OFF (execution of the calculation of point of intersection)

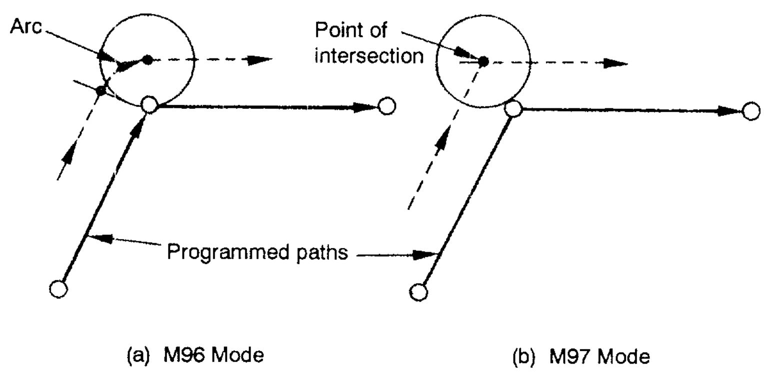


Fig. 3.84 Movements in the M96/M97 Mode

(a) Movements in the nose R offset mode called by G41/G42

In the nose R offset mode, called by G41/G42, when the shape specified by the program has a corner that has the tangential angle of 180° or larger, the cutting tool turns around the corner along an arc if the M96 mode is specified. In the M97 mode, an arc is not generated for the tool paths, but the point of intersection is calculated from the paths that are offset from the programmed paths by the nose R offset amount and the cutting tool moves to the calculated point of intersection when turning around the corner.

(b) Blocks where M96 and M97 commands are valid

The following example shows how the M96 and M97 commands become valid.

Example of Programming

```
G01   Z··· X··· F··· ;     } M96 becomes valid from the tool movements
(G01) Z··· X··· M96 ;      } along the corner, defined by these two blocks.
      Z··· X··· ;          } M97 becomes valid from the tool movements
      Z··· X··· M97 ;      } along the corner, defined by these two blocks.
```

(17)Supplements to the Nose R Offset Commands

- In nose R offset mode, the maximum programmable values specified in Table 3.21 also apply.
- Alarm “0184” occurs if the following shapes are specified.
  - · Circular arc: Radius (r) of arc + 5 ≦ Nose R radius (R)

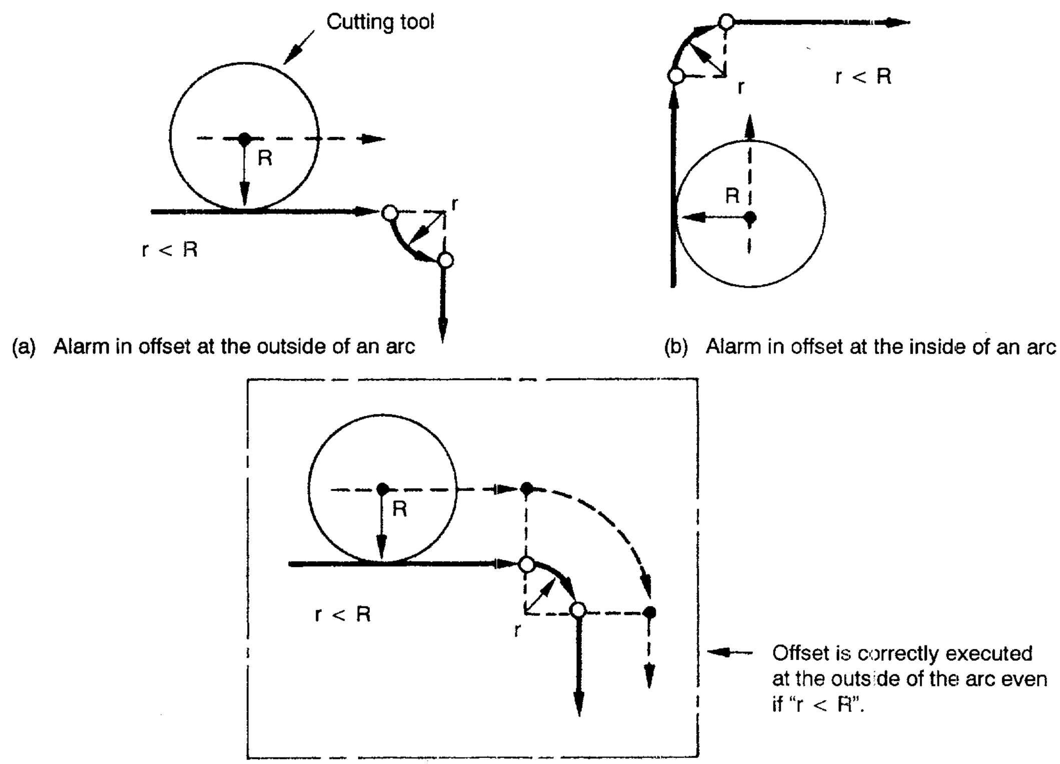


Fig. 3.85 Programmed Paths Causing an Alarm ①

  - · Point of intersection does not lie on the tool paths

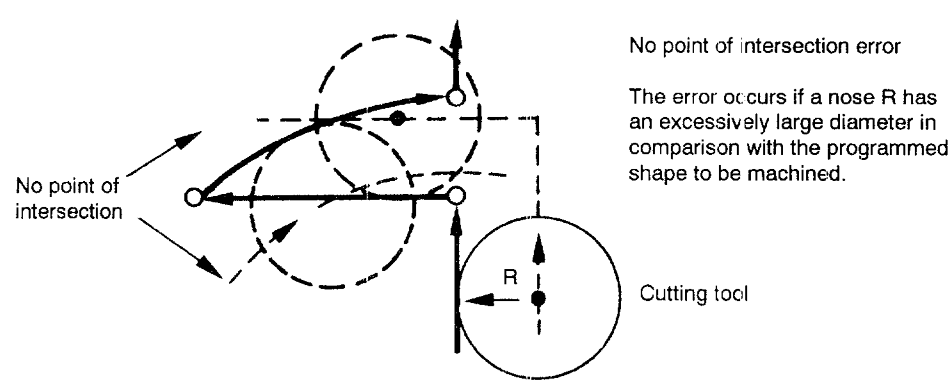


Fig. 3.86 Programmed Paths Causing an Alarm ②

- The G codes that can be used in the nose R offset mode are indicated in Table 3.17. The following G codes must not be used in the nose R offset mode: G31, G74/G75/G76, G68/G69, and G122/G123. If any of these G codes is specified in the nose R offset mode, alarm "0161" occurs.

Table 3.17 G Codes That Can Be Specified in the Nose R Offset Mode

Usable G Codes	Remark
G00, G01, G04, G06, G11 G96, G97 : Constant surface speed control G98, G99 : Feed function designation (G90, G91: Absolute/incremental command)	
G02, G03 G12, G22, G23 : Commands including an arc G70, G71, G72, G73 : Multiple-repetitive cycle G111, G112 : Multiple chamfering, rounding	These G codes must not be specified in the offset start-up and cancel blocks.

- Even in the M96 mode, if both ΔX and ΔZ are smaller than the specified amount as shown in Fig. 3.87, round-the-arc paths are not generated but the cutting tool moves directly to point B. The amount to determine whether or not round-the-arc paths are generated is set for parameter pm4450.

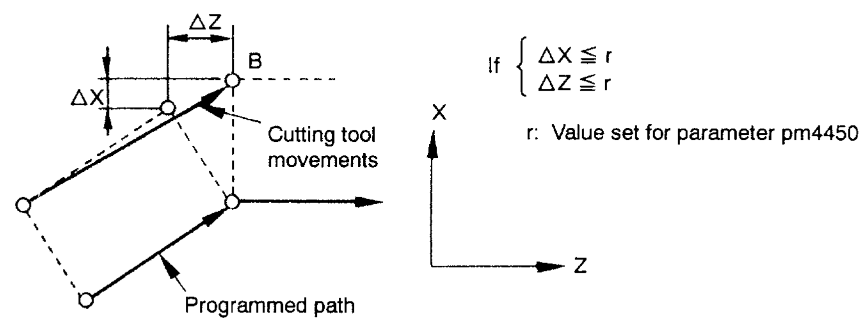


Fig. 3.87 If Both ΔX and ΔZ are Smaller Than Specified Amount

- If offset is made in the M96 mode for the step that is smaller than the nose R offset amount, overcuts will occur. Conversely, uncut portion will be left if the M97 mode is used for offset. In actual operation, it is recommended to use the M97 mode.

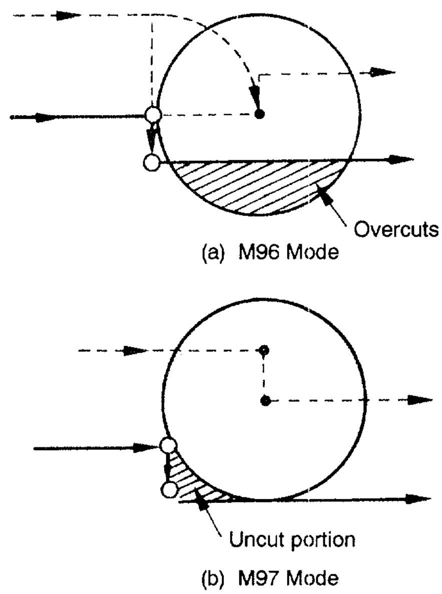


Fig. 3.88 Offsetting Step Smaller Than Nose R Offset Amount

- MDI operation intervention is not allowed in the offset mode. However, one-line MDI operation is possible.
- In the G41 or G42 mode, it is possible to enter the data in the same procedure as in the MDI operation by using the following steps: First turn ON the single block switch. After the machine has stopped in the block stop state, select the RAPID or JOG mode and enter the data. The data that can be entered in this operation are restricted to F, M, S, and T codes. After entering the data, press the cycle start switch without changing the mode (RAPID or JOG) selected for entering the data, and the entered code is executed immediately with the signal such as BIN code output. Return the mode to the automatic operation mode and press the cycle start switch. With this operation, suspended automatic operation can be resumed. Note that M00, M01, M02, M30, and M codes processed in the CNC cannot be entered by this operation.

- The T code command that has the tool offset number of "00" cancels the tool position and nose R offset functions.

Example of Programming

```
N2  G41 ;
N3  G00  T0101 ;    ┐
     .              │ Nose R offset mode with tool No. 01
     .              │
N21  G00  T0100 ;   ┤
     .              │ Cancel } Nose R offset
     .              │          Tool position offset
N25  G00  T0202 ;   ┤
     .              │ Nose R offset mode with tool No. 02
     .              │
N40  G00  T0200 ;   ┘
N41  G40 ;
```

3

3

3.5 SPINDLE FUNCTION (S FUNCTION)

3.5.1 Spindle Command (S5-digit Command)

A spindle speed can be directly specified by entering a 5-digit number following address S (S□□□□□). The unit of spindle speed is "r/min". The specified S value becomes valid from the moment the S command completion input signal (SFIN) is turned ON. If an S command is specified with M03 (spindle forward rotation) or M04 (spindle reverse rotation), the program usually advances to the next block only after the spindle has reached the speed specified by the S command. For details, refer to the instruction manuals published by the machine tool builder.

Example of Programming
S1000 M03;

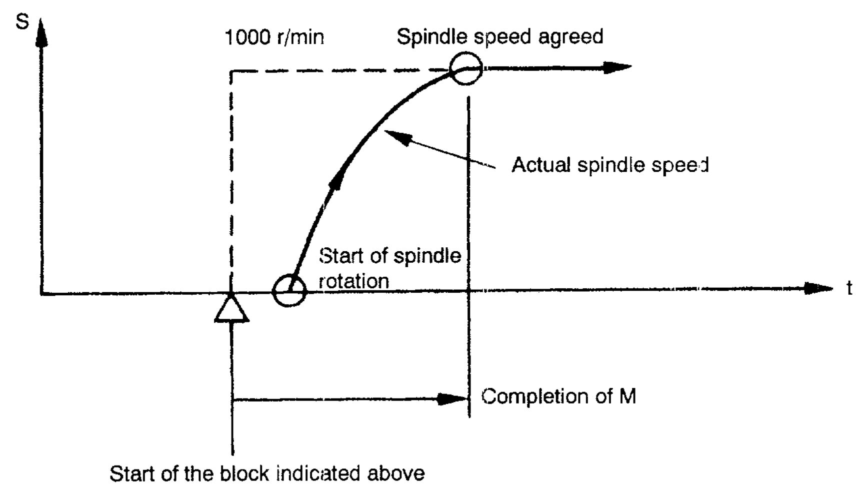


Fig. 3.89 Spindle Speed Command

- For the output of S5-digit commands, it is possible to add the control function implemented by the PLC can be added by the NC. In this case, it is possible to set the spindle speed in manual operation to the speed that corresponds to the specified S command by using the rotary switch on the machine operation panel. For details, refer to the manuals published by the machine tool builder.
- An S command is modal and, once specified, it remains valid until another S command is given next. If the spindle is stopped by the execution of M05, the S command value is retained. Therefore, if M03 or M04 is specified without an S command in the same block, the spindle can start by using the S command value specified before.
- The lower limit of an S command (S0 or an S command close to S0) is determined by the spindle drive motor and spindle drive system, and it varies with each machine. Do not use a negative value for an S command. For details, refer to the instruction manuals published by the machine tool builder.
- Spindle speed override is possible for the specified S code.
- For the machine that has the gearbox with which gear range can be changed by specifying an M code, specify the M code to select an appropriate gear range before specifying an S code. For the number of gear ranges and the available spindle speed range in the individual gear ranges, refer to the manuals published by the machine tool builder.

3.5.2 Maximum Spindle Speed Command (G50 S)

By the commands of "G50 S · · · ;", the clamp speed of the spindle can be set by specifying the allowable maximum spindle speed in a 5-digit number following address S. Once the clamp speed is set, it is not influenced by the reset operation.

If a spindle speed that exceeds the specified speed is entered, the spindle speed is clamped at the specified clamp speed. To cancel the clamp speed, specify "G50 S0 ;".

- The clamp speed specified with G50 can be displayed on the screen.
- If the PLC-based control function is added to the S code output, the unit of S codes is not always "r/min". For the unit system used for spindle speed, refer to the manuals published by the machine tool builder.

3.5.3 Constant Surface Speed Control (G96, G97) *

The G codes indicated in Table 3.18 are used for the constant surface speed control function. G96 and G97 are modal G code of 02 group. The initial state when the power is turned ON is the G97 (cancel) mode.

Table 3.18 G Codes for Constant Surface Speed Control

G Code	Function	Group
G96	Constant surface speed control ON	02
G97	Constant surface speed control cancel	02

(1) Constant Surface Speed Control ON (G96)

With the commands of "G96 S · · · (M03) ;", the workpiece surface speed is designated by a maximum 5-digit number following address S. The unit used for specifying the surface speed is indicated in Table 3.19.

Table 3.19 Units of Surface Speed Designation

	Unit
mm	m/min
inch	ft/min

In the constant surface speed control mode, the NC assumes the present value of the X-axis as the workpiece diameter and calculates the spindle speed every 32 msec so that the specified surface speed is maintained. The result of calculation is output in analog voltage. The specified surface speed can be changed by specifying a required S code in the following blocks.

(a) Spindle gear range selection

For the machine that has the gearbox with which gear range can be changed by specifying an M code, specify the M code to select an appropriate gear range before specifying G96. For details, refer to the manuals published by the machine tool builder.

Example of Programming

```
 .
 .
 .
N8 M△△ ;          ←————— M code for selecting gear range
N9 G96 S100 M03 ;         (Example: Gear range No. 4)
 .
 .
 .
```

(b) Designation of "G50 S"

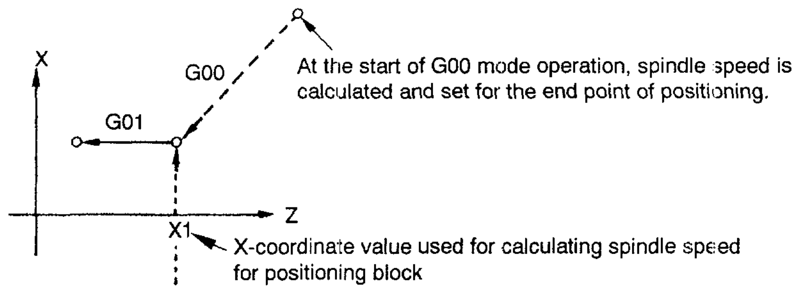
In the constant surface speed control operation, the allowable maximum spindle speed must be specified following G50 before the G96 designation block so that the spindle speed becomes abnormally high as the X-axis present value becomes smaller.

```
N10 G50 S2000 ;     ←————— Designation of clamp speed (r/min)
N11 M△△ ;
N12 G96 S150 M03 ;
```

(c) Constant surface speed control in positioning mode blocks

If the setting for parameter pm4011 D4 is "1" (pm4011 D4 = 1), the constant surface speed control is applied even to positioning mode (G00, G06) blocks. In this case, however, the spindle speed is calculated based on the coordinate values of the end point of positioning. Therefore, spindle speeds are constantly calculated and controlled only in the cutting mode.

If "pm4011 D4 = 0", constant surface speed control is applied only to the cutting feed blocks and the positioning block immediately before the cutting feed block. For the positioning block, spindle speed is calculated based on the coordinate values of the end point of positioning.




```
N4 G50 S1500 ;              ← Spindle speed clamp value
N5 MΔΔ ;                    ← Gear range selection M code
N6 G96 S150 M03 ;           ← Designation of surface speed of 150 m/min
N7 G00 X40. Z5. ;           ┐
N8 G01 Z0 F0.15 ;           │
N9 X80. Z-30. ;             │ Constant surface speed control mode
N10 W-10. ;                 │
N11 G22 X120. W-20. R20. ;  │
N12 G01 U10. ;              ┘
N13 G97 S500 ;
N14 G50 S2000 ;             ← Cancel of constant surface speed control
```

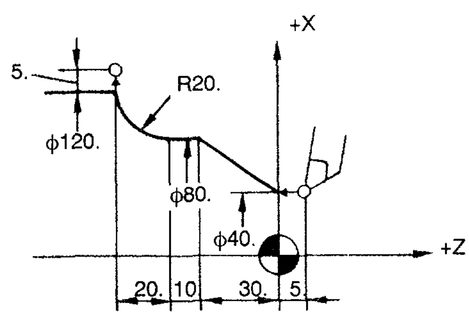


Fig. 3.90 Constant Surface Speed

(2) Canceling the Constant Surface Speed Control (G97)

Specify a spindle speed (r/min) by a maximum of 5-digit number following address S with the commands "G97 S · · · (M03) ;". The constant surface speed control mode is canceled, and the spindle rotates at the specified spindle speed.

(3) Supplements to the Constant Surface Speed Control Commands

- To execute the constant surface speed control, set the G50 coordinate system or a workpiece coordinate system so that the X-coordinate value of the centerline of the spindle will be "0" and program the operation on this coordinate system. With this, X-coordinate values in a program represent the diameter of workpiece accurately.
- Set "1" for parameter pm4011 D5 (pm4011 D5 = 0) to execute the constant surface speed control. In this setting, spindle speed is calculated without adding the tool position offset amount to the coordinate values specified in a program. If a large value is set for tool offset data, the tool position offset function is executed correctly and the constant surface speed control is also executed correctly.
- With the setting of "pm4011 D5 = 1", the "coordinate value in a program + tool position offset amount" is taken as the diameter of a workpiece for the calculation of spindle speed to execute the constant surface speed control. If the constant surface speed control is executed under such setting, it is necessary to set a coordinate system for the individual tools. It is also necessary to use the tool position offset only for compensation for tool wear so that a large value will not be set for offset data.
- For spindle gear ranges, up to four steps is allowed.
- Setting for parameter pm4011 D5 and that for pm3000 D2 are not related to each other. Switching over the setting for these parameters (calculation for the constant surface speed control, present position data display on the screen) is processed independently.

pm3000 D2 = 0	In the present position data (workpiece coordinate system), coordinate values are displayed with the tool position offset amount and nose R offset amount included.
pm3000 D2 = 1	In the present position data (workpiece coordinate system), coordinate values are displayed without including the tool position offset amount and nose R offset amount.

3.5.4 Rotary Tool Spindle Selection Function *

By selecting this option, it is possible to add the rotary tool spindle to the main spindle. In this case, spindle speed commands (S codes) are applied to the main or rotary tool spindle according to the specified G code as indicated below. G132 and G133 are modal; when the NC is reset or when the power is turned ON, the G133 mode is set.

After switching over the G code between G132 and G133, make sure to specify a spindle speed before rotating the spindle.

Table 3.20 Spindle Mode Selection G Codes

G Code	Function	Group
G132	Spindle speed commands are used as those for the rotary tool spindle.	22
G133	Spindle speed commands are used as those for the main spindle.	22

- It is not possible to use the rotary tool spindle for the reference spindle where feed per revolution control is executed.
- The constant surface speed control is not valid for the rotary tool spindle.

3

3.6 TOOL FUNCTION (T FUNCTION)

The tool function has two command designation types as T4-digit commands and T6-digit commands.

3.6.1 T4-digit Command

A tool number and a tool offset number are specified by a 4-digit number following address T (T□□□□).

```
T  □□  □□
   |   |
   |   └─── Tool offset number
   |
   └─────── Tool number
```

The range of numbers that can be specified differ depending on whether the option is selected or not. For details, refer to the manuals published by the machine tool builder. Concerning the details of tool offset, refer to 3.4 "TOOL OFFSET FUNCTIONS".

SUPPLE-MENT

1. When a command that changes the selected tool is given, turret indexing operation starts immediately to select the specified tool with the turret type NC lathe. Therefore, move the axes to the position where rotation of the turret does not cause interference before specifying such a command.
2. Tool offset number "00" indicates "cancellation" of the tool offset function.

3.6.2 T6-digit Command *

A tool number and a tool offset number are specified by a 6-digit number following address T (T□□□□□□). In a T command, leading zeros may be omitted. In comparison to the T4-digit commands, only the number of digits is increased and functions and other details are the same as T4-digit commands.

```
T  □□□  □□□
    |    |
    |    └─── Tool offset number
    |
    └──────── Tool number
```

3.7 MISCELLANEOUS FUNCTION (M FUNCTION)

The miscellaneous function is specified by a maximum of a three-digit number (M□□□) following address M. With the exception of specific M codes, the functions of M00 to M89 codes are defined by the machine tool builder. Therefore, for details of the M code functions, refer to the instruction manuals published by the machine tool builder.

The M codes specific to the NC are described below.

3.7.1 M Codes Relating to Stop Operation (M00, M01, M02, M30)

When an M code relating to stop is executed, the NC stops buffering. Whether spindle rotation, coolant discharge or another operation stops in response to the execution of such an M code is determined by the machine tool builder. For details, refer to the instruction manuals published by the machine tool builder. For these M codes, a code signal is output independently in addition to M2-digit BIN code.

3

(1) M00 (Program Stop)

If M00 is specified during automatic operation, automatic operation is interrupted after the completion of the commands specified with M00 in the same block and the M00R signal is output. The interrupted automatic operation can be restarted by pressing the cycle start switch.

(2) M01 (Optional Stop)

If M01 is executed with the optional stop switch ON, the same operation as with M00 is executed. If the optional stop switch is OFF, M01 is disregarded.

(3) M02 (End of Program)

M02 should be specified at the end of a program. When M02 is executed during automatic operation, automatic operation ends after the commands specified with M02 in the same block have been completed. The NC is reset. The state after the end of a program varies with each machine. For details, refer to the instruction manuals published by the machine tool builder.

(4) M30 (End of Tape)

Normally, M30 is specified at the end of tape. When M30 is executed during automatic operation, automatic operation ends after the commands specified with M30 in the same block have been completed. The NC is reset and the tape is rewound. The state after the execution of M30 varies with each machine. For details, refer to the instruction manuals published by the machine tool builder.

When M00, M01, M02, or M30 is specified, the NC stops buffering. For these M codes, the NC output the independent decode signal in addition to the M2-digit BIN code.

Refer to the manuals published by the machine tool builder concerning whether or not the spindle and/or coolant supply is stopped by the M00, M01, M02, and M30.

3

3.7.2 Internally Processed M Codes

M codes in the range of M90 to M99 and M190 to M199 are processed by the NC internally and the corresponding output signal (BIN code and decode output) is not output even when these M codes are executed.

Table 3.21 Internally Processed M Codes

M Code	Function	Setting at Power-ON
* M92	Multi-active registers OFF	○
* M93	Multi-active registers ON	
* M96	Nose R offset, round-the-arc mode	○
* M97	Nose R offset, point of intersection calculation mode	
M98	Subprogram call	
M99	End of subprogram	
* M191	Comment output function	

Note 1: M190 to M199 are used for extension M codes.

2: When the power is turned ON, the M code mode indicated by "○" symbol is set. This is not influenced by the reset operation.

3.7.3 General Purpose M Codes

(1) Other General M Codes

The functions of the M codes other than the specific M codes are determined by the machine tool builder. The representative use of several general M codes is given below. For details, refer to the instruction manuals published by the machine tool builder. If an M code is specified with axis move commands in the same block, whether the M code is executed with the axis move commands simultaneously or it is executed after the completion of the axis move commands is determined by the machine tool builder. For details, refer to the instruction manuals published by the machine tool builder.

3

Table 3.22 Other General M Codes

M Code	Function	Remarks
M03	Spindle start, forward direction	Generally, M state between M03 and M04 cannot be switched directly. To change the M code state, execute M05 once.
M04	Spindle start, reverse direction	
M05	Spindle stop	
M08	Coolant ON	
M09	Coolant OFF	

(2) Designation of Multiple M Codes in a Single Block*

It is possible to specify up to five M codes in a single block. The specified M codes and sampling output are output at the same time. Concerning the combinations of the M codes that can be specified in the same block, refer to the manuals published by the machine tool builder for restrictions on them.

4

ENHANCED LEVEL COMMANDS

Chapter 4 describes the program support functions, automation support functions, and macro programs.

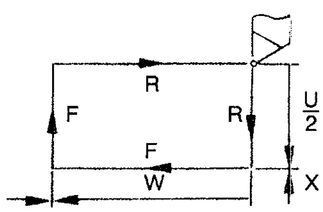
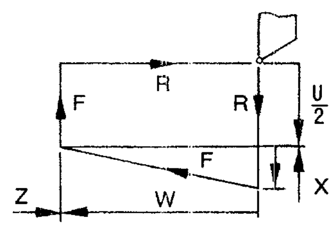
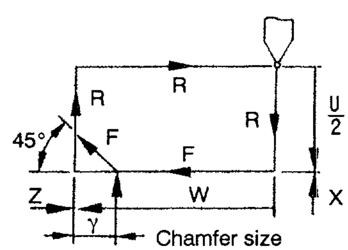
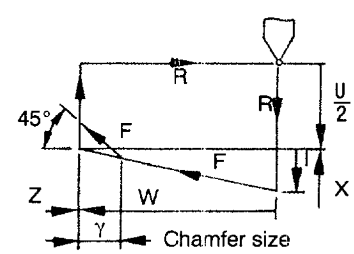
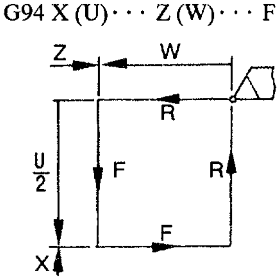
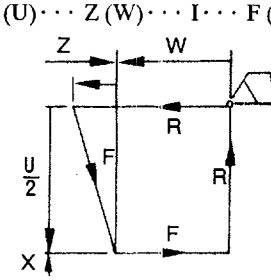
4

4.1 PROGRAM SUPPORT FUNCTIONS (1)

4.1.1 Canned Cycles (G90, G92, G94)

The canned cycle function defines the four block operations of basic cutting operation, infeed, cutting (or thread cutting), retraction, and return, in one block (to be called as one cycle).

Table 4.1 Tale of Canned Cycles

G Code	Straight Cycle	Taper Cycle
G90 Cutting cycle A (OD cutting)	G90 X (U)··· Z (W)··· F (E)··· ; 	G90 X (U)··· Z (W)··· I··· F (E)···; 
G92 Thread cutting cycle	G92 X (U)··· Z (W)··· F (E)··· ; 	G92 X (U)··· Z (W)··· I··· F (E)··· ; 
G94 Cutting cycle B (face cutting)	G94 X (U)··· Z (W)··· F (E)··· ; 	G94 X (U)··· Z (W)··· I··· F (E)··· ; 

(1) Cutting Cycle A (G90) Commands

The cutting cycle A is used for outside diameter (OD) cutting and has two kinds of cycles – straight cutting cycle and taper cutting cycle.

(a) Straight cutting cycle

With the commands of "G90 X(U)··· Z(W)··· F(E)··· ;", straight cutting cycle is executed as indicated by sequence ① to ④ shown in Fig. 4.1.

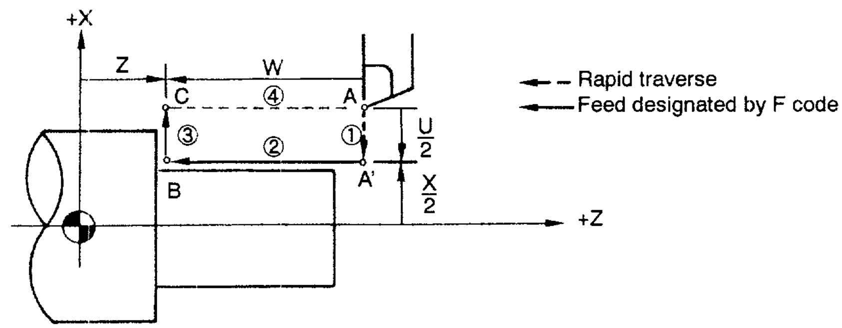


Fig. 4.1 Straight Cutting Cycle

Since G90 is a modal G code, cycle operation is executed by simply specifying in-feed movement in the X-axis direction in the succeeding blocks.

Example of Programming

```
N10 G00 X94. Z62. ;
N11 G90 X80. W–42. F0.3 ;     ←  Start of G90 cycle
N12 X70. ;                    ←  Executes G90 cycle by changing
N13 X60. ;                    ←  the cutting paths.

N14 G00··· ;
```

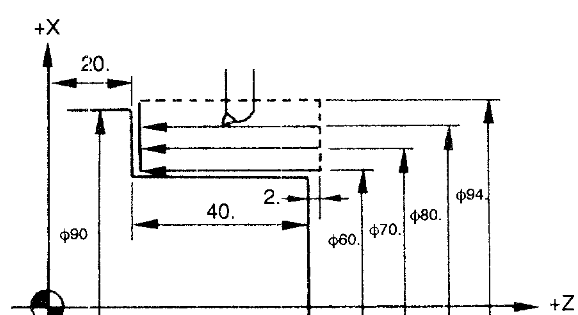


Fig. 4.2 Straight Cutting Cycle

(b) Taper cutting cycle

With the commands of "G90 X(U)· · · Z(W)· · · I· · · F(E)· · · ;" taper cutting cycle is executed as indicated by sequence ① to ④ shown in Fig. 4.3.

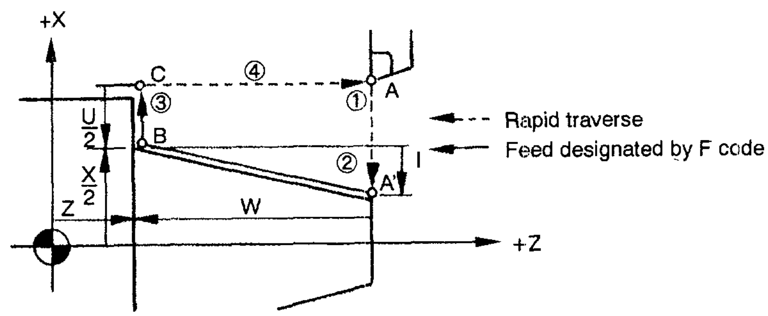


Fig. 4.3 Taper Cutting Cycle

The sign of address I is determined by the direction viewing point A' from point B.

Example of Programming

N20 00 X87. Z72. ;

N21 G90 X85. W–42. I–10.5 F0.25 ;
N22 X80. ;
N23 X75. ;
N24 X70. ;

N25 G00 · · ·;

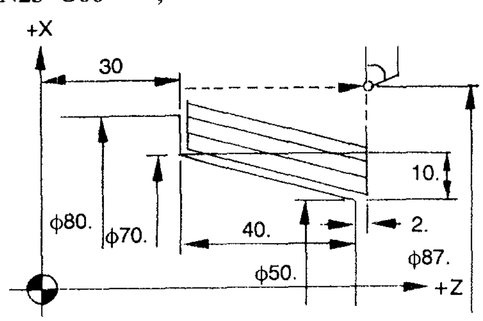


Fig. 4.4 Taper Cutting Cycle

- If the G90 cycle is executed with the single block function ON, the cycle is not interrupted halfway but it stops after the completion of the cycle consisting of sequence ① to ④.
- The S, T, and M functions that are used as the cutting conditions for the execution of the G90 cycle should be specified in blocks preceding the G90 block. However, if these functions are specified in a block independently without axis movement commands, such designation is valid if the block is specified in the G90 mode range.

```
G90  X ··· Z ··· I ··· F(E) ··· ;          ┐
     X ··· ;                               │ G90 valid range
     X ··· ;                               │
     X ··· T0505 M05 ; ← Error             ┘
G00  X ··· Z ··· ;

G90  X ··· Z ··· I ··· F(E) ··· ;          ┐
     X ··· ;                               │ G90 valid range
     X ··· ;                               ┘
G00  X ···· T0505  M05 ; ← Correct
     X ··· Z ··· ;
```

The G90 mode is valid up to the block immediately before the one in which a G code of 01 group is specified.

(2) Thread Cutting Cycle (G92) Command

For thread cutting operations, four kinds of thread cutting cycles are provided – two kinds of straight thread cutting cycles and two kinds of tapered thread cutting cycles.

(a) Straight thread cutting cycle

```
G92  X(U) ··· Z(W) ··· F(E) ··· ;
                       └──────────── Designation of thread lead (L)
```

With the commands indicated above, straight thread cutting cycle ① to ④, shown in Fig. 4.5, is executed.

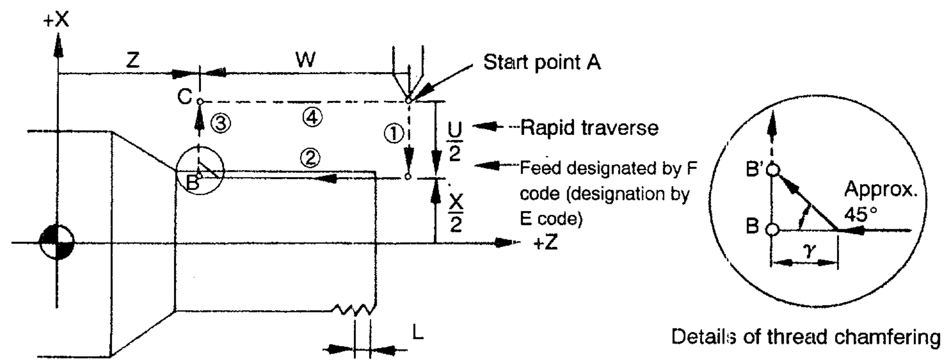


Fig. 4.5 Straight Thread Cutting Cycle

4

Since G92 is a modal G code, thread cutting cycle is executed by simply specifying depth of cut in the X-axis direction in the succeeding blocks. It is not necessary to specify G92 repeatedly in these blocks.

Example of Programming

N30 G00 X80. Z76.2 M○○ ;	← M○○; Thread chamfering ON
N31 G92 X66.4 Z25.4 F6. ; N32 X65. ; N33 X63.8 ; N34 X62.64 ;	Thread cutting cycle, in four in-feeds
N35 G00 X100. Z100. M△△ ;	← M△△; Thread chamfering OFF

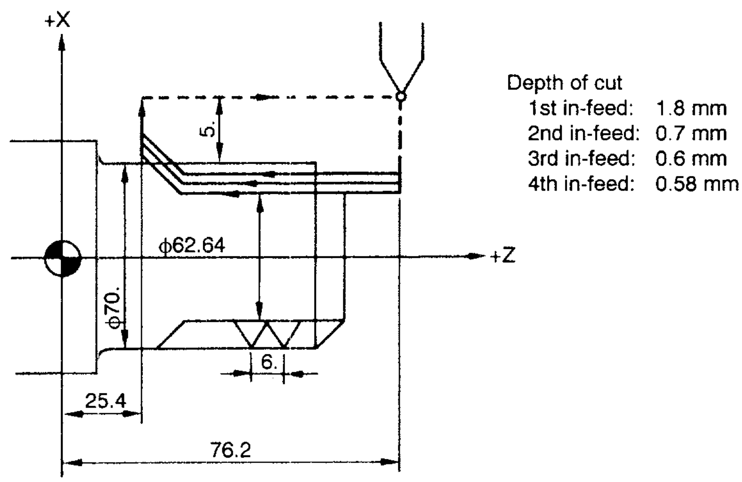


Fig. 4.6 Straight Thread Cutting Cycle

- When the G92 cycle is executed with the single block function ON, the cycle is not suspended halfway, but it stops after the completion of the cycle consisting of sequence ① to ④.
- If "thread chamfering input (CDZ)" is ON at the time G92 is specified, thread chamfering is executed. Thread chamfering size γ can be set for parameter pm0100 in increments of 0.1L in the range from 0 to 25.5L. Here, "L" represents the specified thread lead.

  It is recommended to program the sequence that turns ON and OFF the "thread chamfering input (CDZ)" by using appropriate M codes.

(b) Straight thread cutting cycle (in-feed along thread angle)

With the commands of "G92 X(U) · · · Z(W) · · · K · · · F(E)· · · ;", straight thread cutting cycle of ① to ④ as shown in Fig. 4.7 is executed. In this cycle, in-feed is made along the thread angle.

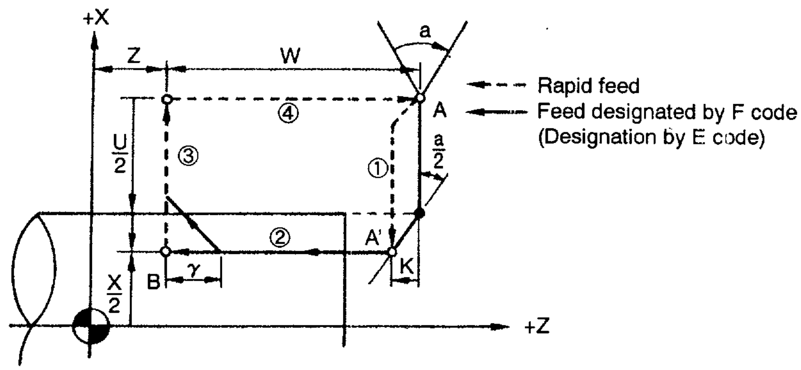


Fig. 4.7 Straight Thread Cutting Cycle (In-feed along Thread Angle)

The sign of address K is determined by the direction viewing point A' from point A.

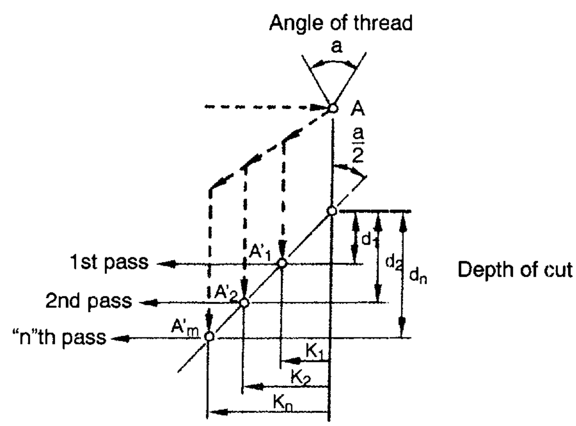


Fig. 4.8 Designation of Shift Amount K in Z-axis Direction from Point A to Point A'

- When the G92 cycle is executed with the single block function ON, the cycle is not suspended halfway, but it stops after the completion of the cycle consisting of sequence ① to ④.
- To execute in-feeding along the angle of thread (a), calculate the value of K using the following formula.

$$|K_n| = dn \tan\left(\frac{a}{2}\right)$$

Table 4.2 Quick Reference – Angle of Thread (a) and tan $\left(\frac{a}{2}\right)$

a	$\tan\left(\frac{a}{2}\right)$
29°	0.258618
30°	0.267949
55°	0.520567
60°	0.577350
80°	0.839100

- In the multiple repetitive cycle (G76), thread angles are restricted to six kinds. However, the cycle called by G92 allows cutting of thread which has an optional thread angle.

  Calculation of value $|K| = d \tan(60°/2)$

  $K_1 = -1.8 \times 0.57735 = -0.866$ mm
  $K_2 = -2.5 \times 0.57735 = -1.443$ mm
  $K_3 = -3.1 \times 0.57735 = -1.790$ mm
  $K_4 = -3.68 \times 0.57735 = -2.125$ mm

  From the calculation indicated above, the program should be as indicated below.

```
N40  G00  X80.  Z76.2  M○○ ;

N41  G92  X66.4  Z25.4  K–0.87  F6. ;
N42  X65.               K–1.44 ;
N43  X63.8              K–1.79 ;
N44  X62.64             K–2.13 ;

N45  G00  X100.  Z100.  M△△ ;
```

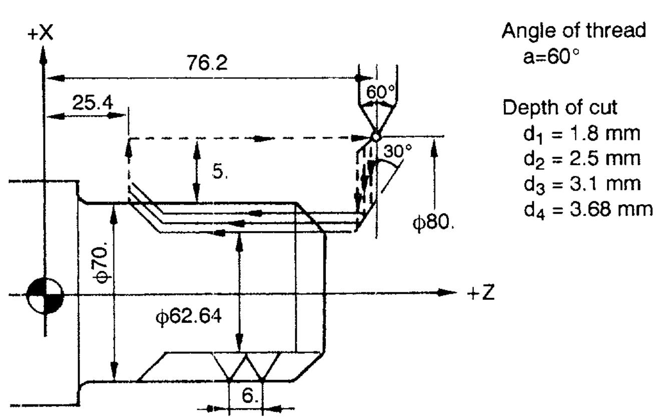


Fig. 4.9 Straight Thread Cutting Cycle (In-feed Along Thread Angle)

(c) Tapered thread cutting cycle

With the commands of "G92 X(U) ··· Z(W) ··· I ··· F(E) ··· ;" tapered thread cutting cycle of ① to ④ as shown in Fig. 4.10 is executed.

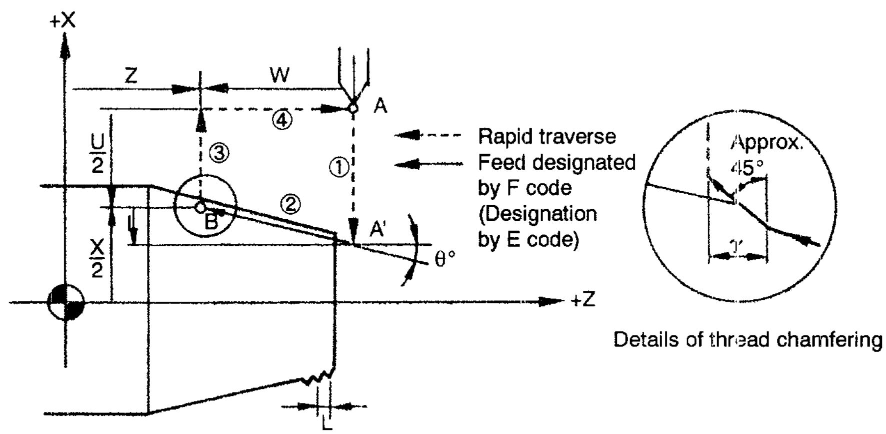


Fig. 4.10 Tapered Thread Cutting Cycle

The sign of address I is determined by the direction viewing point A' from point B. Since G92 is a modal G code, thread cutting cycle is executed by simply specifying depth of cut in the X-axis direction in the succeeding blocks. It is not necessary to specify G92 repeatedly in these blocks.

Example of Programming

N50 G00 X80. Z80.8 M○○ ;

N51 G92 X70. W–50.8 I–1.5 F2. ;
N52 X68.8 ;
N53 X67.8 ;

N54 G00 X100. Z100. M△△ ;

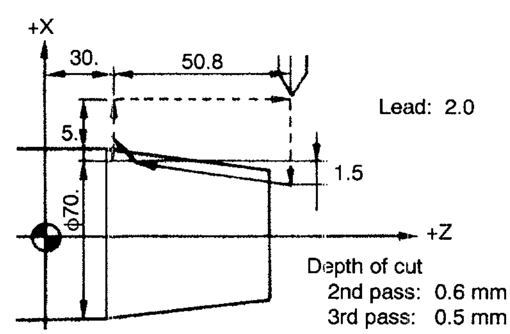


Fig. 4.11 Tapered Thread Cutting Cycle

- When the G92 cycle is executed with the single block function ON, the cycle is not suspended halfway, but it stops after the completion of the cycle consisting of sequence ① to ④.

(d) Tapered thread cutting cycle (in-feed along thread angle)

With the commands of "G92 X(U) · · · Z(W) · · · I · · · K · · · F(E) · · · ;", tapered thread cutting cycle of ① to ④ as shown in Fig. 4.12 is executed. In-feed is made along the angle of thread. The sign of address K is determined by the direction viewing point A' from point A.

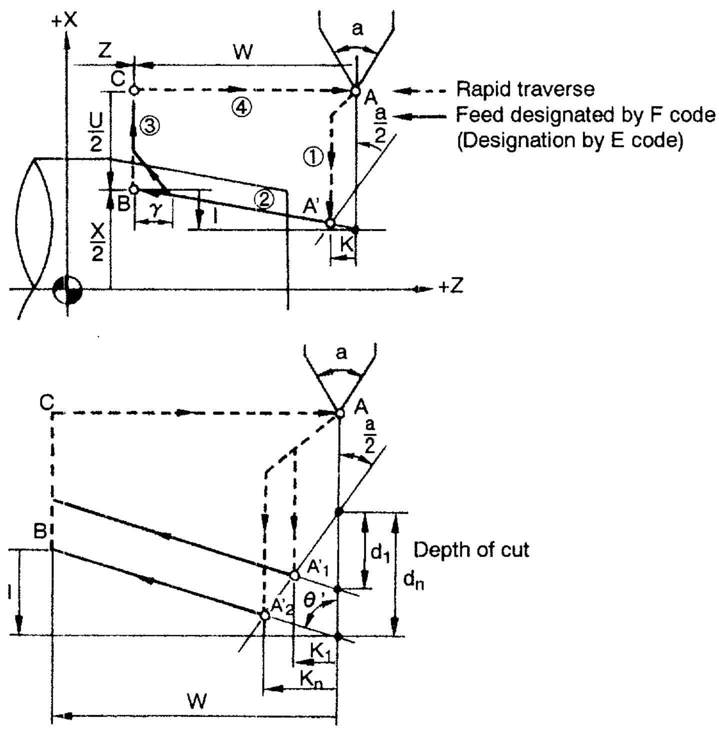


Fig. 4.12 Tapered Thread Cutting Cycle (In-feed Along Thread Angle)

- When the G92 cycle is executed with the single block function ON, the cycle is not suspended halfway, but it stops after the completion of the cycle consisting of sequence ① to ④.
- To execute in-feeding along the angle of thread (a), calculate the value of K using the following formula.

$$\ell|K_n| = \frac{dn \tan(\frac{a}{2})}{1 \pm |\frac{I}{W}| \cdot \tan(\frac{a}{2})}$$

The sign of denominator depends on the value of θ'.

If θ' < 90°: "+"
If θ' > 90°: "–"

Since this makes calculation complicated, it is recommended to use the G76 automatic thread cutting cycle if the NC has the multiple repetitive cycle function. With the G76 cycle, the calculation indicated above is automatically executed by the NC.

- In the multiple repetitive cycle (G76), thread angles are restricted to six kinds. However, the cycle called by G92 allows cutting of thread which has an optional thread angle.
- The S, T, and M functions that are used as the cutting conditions for the execution of the G92 cycle should be specified in blocks preceding the G92 block. However, if these functions are specified in a block independently without axis movement commands, such designation is valid if the block is specified in the G92 mode range.
- If the thread cutting feed hold option is selected, thread chamfering is executed immediately when the FEED HOLD button is pressed during the execution of thread cutting cycle. After the completion of chamfering, the cutting tool returns to the start point A. If the setting for parameter pm4011 D1 is "1" (pm4011 D1 = 1), the cutting tool stops at the point B where chamfering is completed.

  When the CYCLE START button is pressed while the cutting tool is at start point A or chamfering completion point B, the suspended cycle is executed again from the beginning.

  If the thread cutting feed hold option is not selected, the thread cutting cycle is continued even if the FEED HOLD button is pressed during the execution of thread cutting cycle. In this case, the operation is suspended upon completion of retraction operation after finishing the thread cutting cycle.

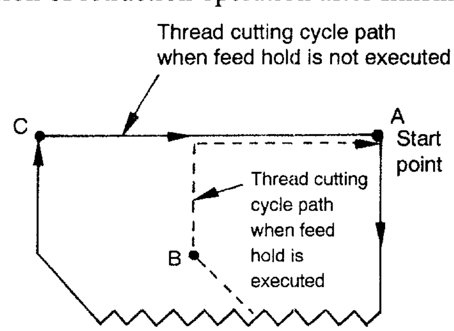


Fig. 4.13 Feed Hold during Thread Cutting Cycle

- If chamfer size is "0" when the G92 cycle is executed with chamfering ON, alarm "0454" occurs.

(3) Cutting Cycle B (G94) Commands

(a) Straight facing cycle

With the commands of "G94 X(U) · · · Z(W) · · · F(E) · · · ;", straight facing cycle of ① to ④ as shown in Fig. 4.14 is executed.

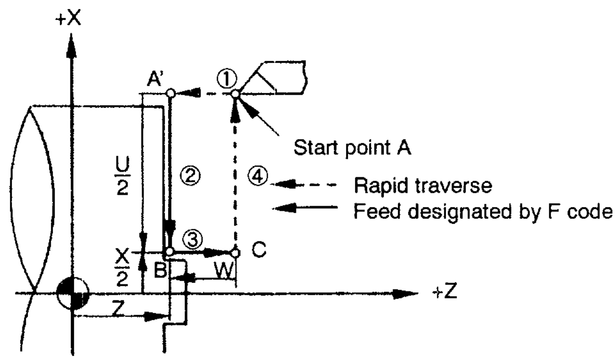


Fig. 4.14 Straight Facing Cycle

Since G94 is a modal G code, thread cutting cycle is executed by simply specifying depth of cut in the Z-axis direction in the succeeding blocks. It is not necessary to specify G94 repeatedly in these blocks.

Example of Programming

```
N60 G00 X65. Z42. ;

N61 G94 X20. Z38. F0.35 ;   ]
N62 Z34. ;                  ] Cutting in 3 cycles in the G94 mode
N63 Z30. ;                  ]

N64 G00 ;
```

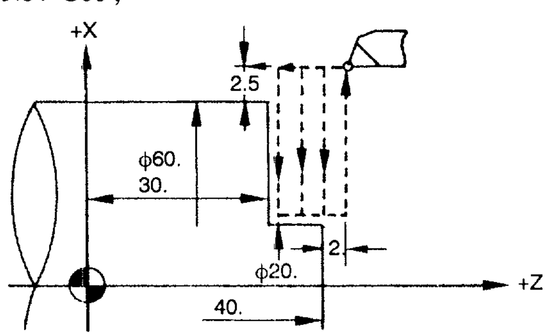


Fig. 4.15 Straight Facing Cycle

(b) Taper facing cycle

With the commands of "G94 X(U) · · · Z(W) · · · K · · · F(E) · · · ;", taper facing cycle of ① to ④ as shown in Fig. 4.16 is executed.

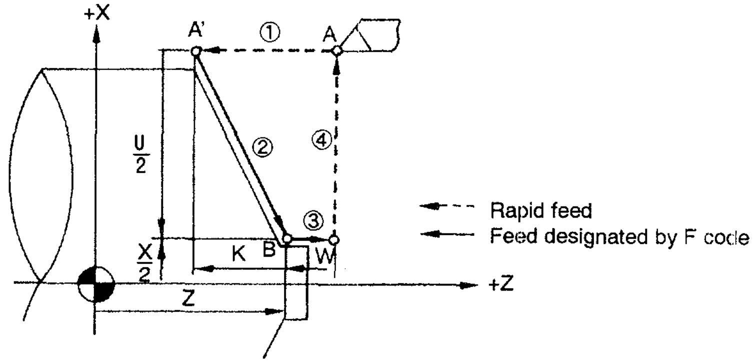


Fig. 4.16 Taper Facing Cycle

The sign of address K is determined by the direction viewing point A' from point B.

Example of Programming

N70N G00 X74. Z32. ;

N71 G94 X20. Z30. K–5.29 F0.3 ;
N72 Z25. ;
N73 Z20. ;

Taper cutting in 3 cycles in the G94 mode

N74 G00 ;

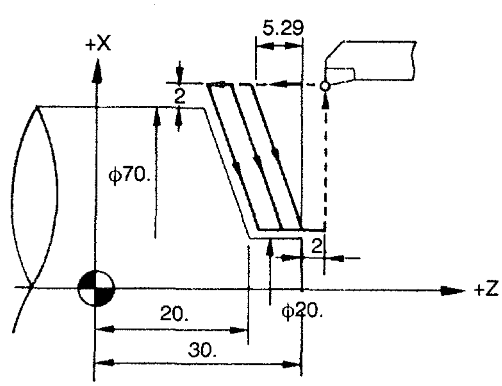


Fig. 4.17 Taper Facing Cycle

- The S, T, and M functions that are used as the cutting conditions for the execution of the G94 cycle should be specified in blocks preceding the G94 block.

However, if these functions are specified in a block independently without axis movement commands, such designation is valid if the block is specified in the G94 mode range.

- If the G94 cycle is executed with the single block function ON, the cycle is not interrupted halfway but it stops after the completion of the cycle consisting of sequence ① to ④.

4.1.2 Multiple Repetitive Cycles (G70 to G76) *

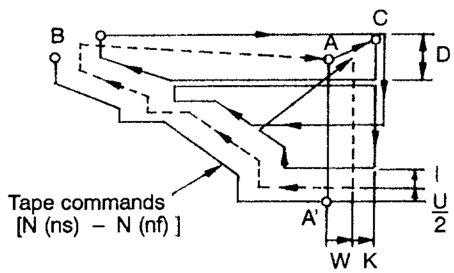
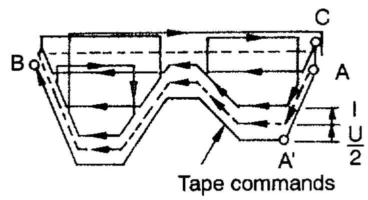
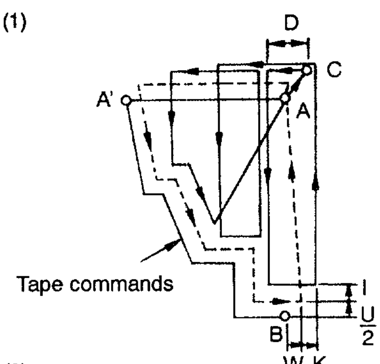
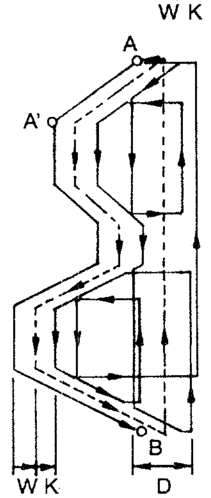
By using the multiple repetitive cycles, programming steps can be considerably reduced due to the features that both rough and finish cutting cycles can be executed by simply defining the finishing shape, and the like.

For the multiple repetitive cycles, seven kinds of cycles (G70 to G76) are provided as indicated in Table 4.3. Note that these G codes are all non-modal G code.

Table 4.3 Cycles Called by G70 to G76

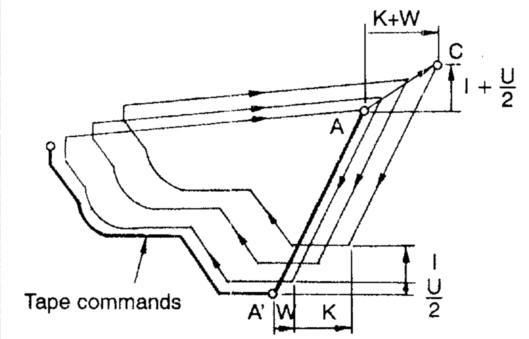
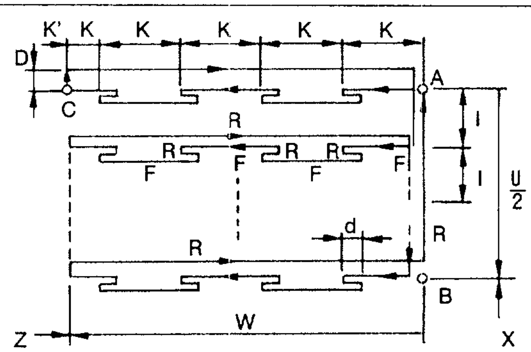
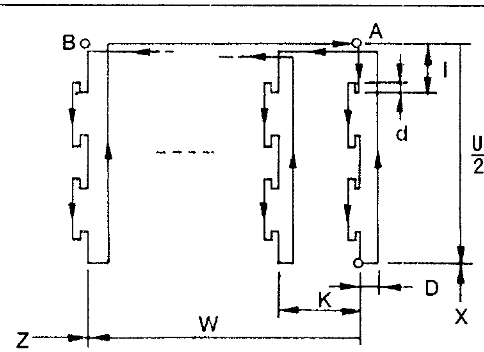
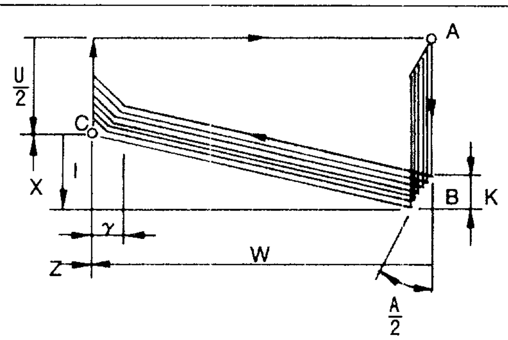
<table>
<tr><th>G Code</th><th>Cycle Name</th><th colspan="2">Remark</th></tr>
<tr><td>G70</td><td>Finishing cycle</td><td></td><td rowspan="4">Nose R offset possible</td></tr>
<tr><td>G71</td><td>OD stock removal cycle</td><td rowspan="3">G70 cycle can be used for finishing</td></tr>
<tr><td>G72</td><td>Face rough turning cycle</td></tr>
<tr><td>G73</td><td>Pattern repeating cycle</td></tr>
<tr><td>G74</td><td>Face cut-off cycle</td><td colspan="2" rowspan="3">Nose R offset impossible</td></tr>
<tr><td>G75</td><td>OD cut-off cycle</td></tr>
<tr><td>G76</td><td>Automatic thread cutting cycle</td></tr>
</table>

Table 4.4 Table of Multiple Repetitive Cycles (G70-76)

	Cutting Cycle	Programming Format
G71 OD stock removal cycle	(1)  (2)  	(1) Monotonous increase/monotonous decrease shape G71 Pns Qnf U · · W · · I · · K · · D · · F(E) · · S · · ; Nns · · · · ; . . . Nnf · · · · ; } Finishing shape (2) Shape with recesses G71 Pns Qnf U · · I · · D · · F(E) · · S · · R1 ; Nns · · · · ; . . . Nnf · · · · ; } Finishing shape (U, W, I, and K: Signed commands)
G72 Face rough turning cycle	(1)  (2)  	(1) Monotonous increase/monotonous decrease shape G72 Pns Qnf U · · W · · I · · K · · D · · F (E) · · S · · ; Nns · · · · ; . . . Nnf · · · · ; } Finishing shape (2) Shape with recesses G72 Pns Qnf W · · K · · D · · F(E) · · S · · R1 ; Nns · · · · ; . . . Nnf · · · · ; } Finishing shape (U, W, I, and K: Signed commands)

4

Table 4.4 Table of Multiple Repetitive Cycles (G70-76) (cont'd)

	Cutting Cycle	Programming Format
G73 Pattern repeating cycle		G73 Pns Qnf U ·· W ·· I ·· K ·· D ·· F(E) ·· S ··; Nns ·· ; · } Finishing shape · Nnf ·· ; (U, W, I and K: Signed commands)
G70	Execution of finishing cutting defined by Nns to Nnf	G70 Pns Qnf ;
G74 Face cut-off cycle		G74 } X (U) ·· Z (W) ·· I ·· K ·· D ·· G75 } F (E) ·· R1 ; (1) Operation as shown in the left is executed if "R1" command is not specified. (2) If "R1" command is specified, retraction amount "d" for each in-feed is disregarded and the axis returns to point A level after each in-feed. d: Setting parameter pm0864 (G74) pm0865 (G75) (I, D and K: Unsigned commands) (3) Using address A, it is possible to specify the number of in-feed steps instead of depth of cut. (4) It is possible to shift the axis at the start and end of operation by specifying the width of cutting tool by address B. (I, K, D, A, and B: Unsigned commands)
G75 OD cut-off cycle		
G76 Automatic thread cutting cycle		G76 X (U) ·· Z (W) ·· I ·· K ·· D ·· F (E) ·· A ·· ; A: Angle of thread (0°, 29°, 30°, 55°, 0°, 60°, 80°) K and D: Unsigned commands 1 / 6 K ≦ D ≦ K

(1) OD Stock Removal Cycle (G71)

With the G71 command, stock removal cycle and rough finishing cycle in which finishing allowance is left on OD or ID can be specified. The programming format differs depending on the finishing shape, whether it is of monotonous increase/monotonous decrease shape or it has recesses in it.

(a) For the workpiece with monotonous increase/monotonous decrease finishing shape

If the finishing shape is monotonous increase/monotonous decrease, the following commands are used.

ns: Sequence number of cycle start

nf: Sequence number of cycle end

G71 Pns Qnf U± · · · W± · · · I± · · · K± · · · D · · · F(E) · · · S · · · ;

(Note)

Nns · · · · · · · · ;
· · · · · · · · ;
· · · F · · · ;
· · · S · · · ;
Nnf · · · · · · · · ;

D: Depth of cut in the X-axis direction in each in-feed (unsigned)

K: Rough finishing allowance in the Z-axis direction

I: Rough finishing allowance in the X-axis direction (radius designation)

W: Finishing allowance in the Z-axis direction

U: Finishing allowance in the X-axis direction (diameter designation)

Nns to Nnf: Finishing shape program (Max. 45 blocks)

This program defines the shape to be finished (A → A' → B) and it should start with sequence number "ns" and end with "nf". Among the commands specified in this program, the F and S commands are valid only when the G70 finishing cycle is executed.

Note: Specify the feed command (F (E)) and spindle command (S) that are used for the execution of the OD stock removal cycle.

- The operation starts from point A; after executing the stock removal cycle (——) and rough finishing cycle (- - -), the cutting tool returns to point A and the operation ends.

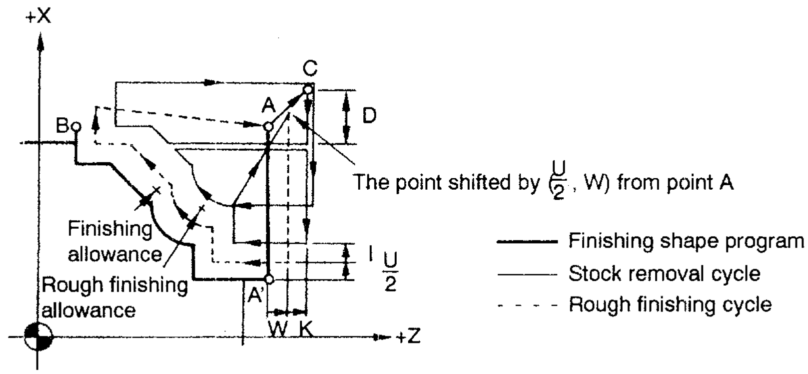


Fig. 4.18 Execution of the Cycle

- If "I = 0, K = 0 (or no designation)", the cycle finishes by skipping the rough finishing cycle as shown in Fig. 4.19.

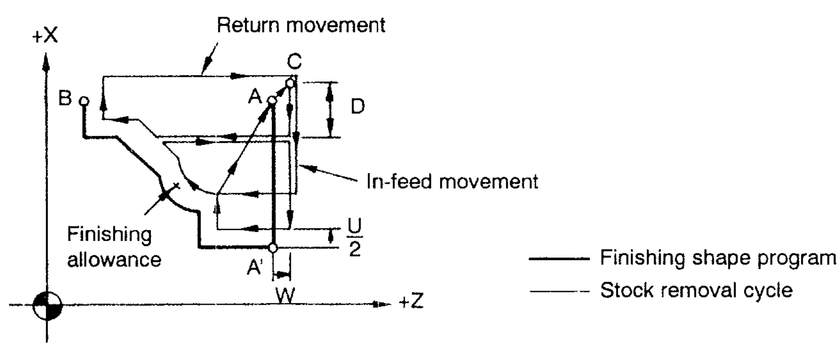


Fig. 4.19 Skipping the Rough Finishing Cycle

- The "return movement" is executed in the G00 (rapid traverse) mode. Concerning the "in-feed movement", it is executed at the feedrate (G00 or G01) specified in the program for $\overline{AA'}$. For feedrate for in-feeding by depth of cut D in the X-axis direction, override setting is possible in 21 steps in units of 10% in the range from 0 to 200% by the setting for a parameter.

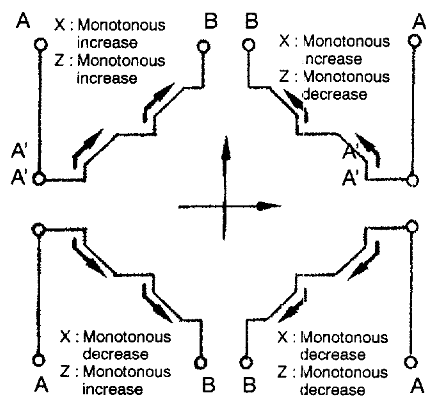


Fig. 4.20 Examples of monotonous increase/monotonous decrease shape

- The following restrictions apply to the start (Nns· · ·) and end (Nnf· · ·) blocks of the finishing shape program.

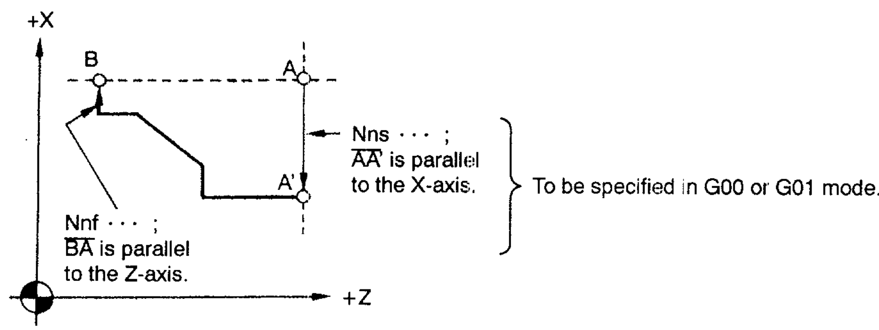


Fig. 4.21 Restrictions on Start and End Blocks

(b) For the workpiece with recesses in the finishing shape

If the finishing shape has recesses in it, the following commands are used.

ns: Sequence number of cycle start

nf: Sequence number of cycle end

G71 Pns Qnf U± · · · I± · · · D · · · F(E) · · · S · · · R1 ;

(Note 1) (Note 2)

```
Nns · · · · · · · · ;
    · · · · · · · · ;
    · · · F · · · ;
    · · · S · · · ;
Nnf · · · · · · · · ;
```

D: Depth of cut in the X-axis direction in each in-feed (unsigned)

I: Rough finishing allowance in the X-axis direction (radius designation)

U: Finishing allowance in the X-axis direction (diameter designation)

Nns to Nnf: Finishing shape program (Max. 45 blocks)
Program that defines the shape of A → A' → B

Note 1: Specify the feed command (F (E)) and spindle command (S) that are used for the execution of the OD stock removal cycle.

2: If "R1" is designated in the program, tool paths are calculated for the finishing shape program which has recesses.

- The operation starts from point A; after executing the stock removal cycle (——) and rough finishing cycle (- - -), the cutting tool returns to point A and the operation ends. If address I is not designated, the rough finishing cycle is skipped.

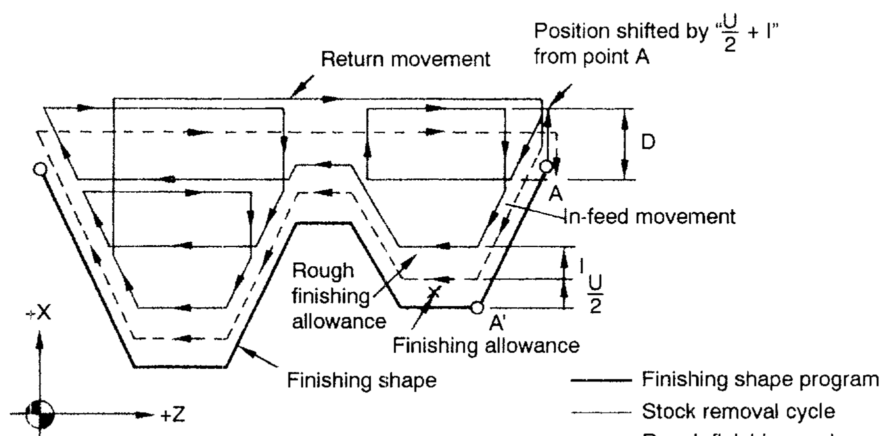


Fig. 4.22 Cycle Execution

POINT

Each block specified in the finishing shape program must define monotonous increasing or monotonous decreasing shape. An arc that extends over multiple quadrants must be programmed in two or more blocks.

- The "return movement" is executed in the G00 (rapid traverse) mode. Concerning the "in-feed movement", it is executed at the feedrate (G00 or G01) specified in the program for $\overline{AA'}$. For feedrate for in-feeding by depth of cut D in the X-axis direction, override setting is possible in 21 steps in units of 10% in the range from 0 to 200% by the setting for a parameter.
- In the stock removal cycle, cutting starts from the recess closest to the start point.

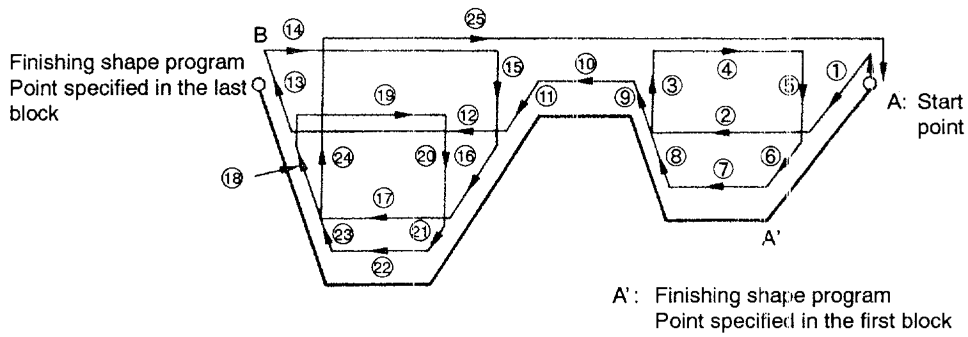


Fig. 4.23 Stock Removal Cycle

Since the defined shape is cut from the recess located closest to the start point, if the cutting path being generated crosses the projection lying next to the recess as shown in Fig. 4.24, the cutting path is interrupted. Then, the new cutting paths are generated until the deepest point in the recess is finished. After that the cutting path generation returns to the interrupted point and cutting paths are generated continuously from the interruption point.

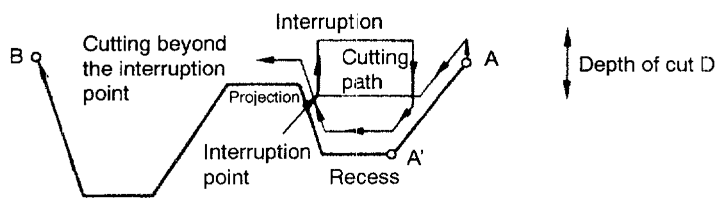


Fig. 4.24 Example of Cutting – Cutting Path Crosses the Projection beyond the Recess

- If a recess has projection and recess in it as shown in Fig. 4.25, interruption will occur again during the cutting of a recess appearing in the recess being defined.

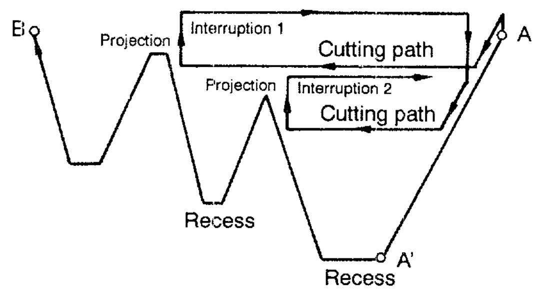


Fig. 4.25 Cutting a Complicated Recess

If interruption points appear repeatedly during the cutting of a recess, appearance of up to three interruption points is allowed before the cutting path returns to the first interruption point. If such interruption points appear at more than three positions, alarm "0469" occurs. There are no special restrictions on the number of recesses as long as this requirement is satisfied.

- The shape that has overhang cannot be cut. Therefore, the Z-axis commands in the finishing shape program must change monotonously.

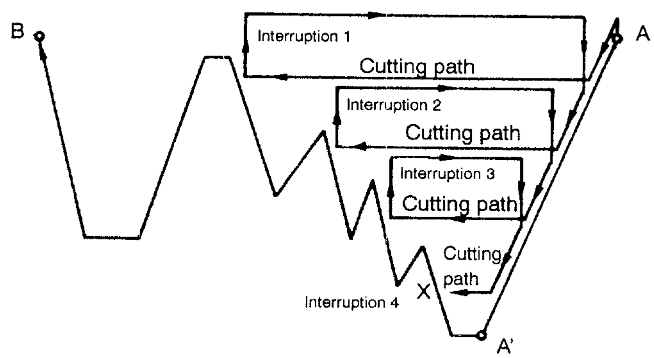


Fig. 4.26 An Example of Shape that Cannot be Cut

- For the end block of the finishing shape program, the restrictions shown in Fig. 4.27 apply. Therefore, the G command to be specified in the end block (Nnf· · · ;) must be either G00 or G01.

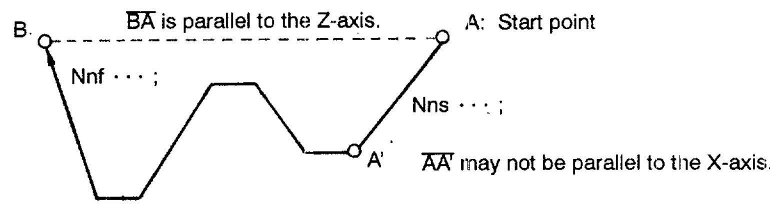


Fig. 4.27 Restrictions on the End Block of Finishing Shape Program

- The retraction amount in each in-feed cycle can be set for a setting parameter.

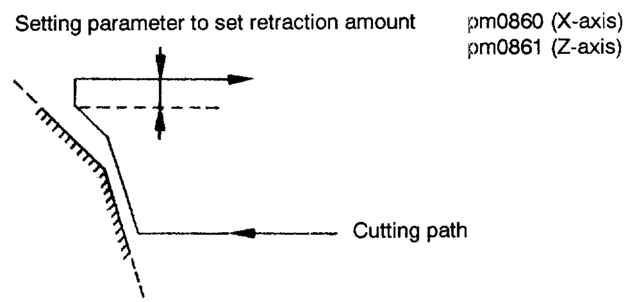


Fig. 4.28 Setting the Retraction Amount

- The finishing allowance (W, K) in the Z-axis direction must not be specified. If such finishing allowance is specified, overcuts into the wall at one side occurs.

- Approach is executed in the cutting feed mode and not influenced by the G code specified in the finishing shape program. Therefore, with some finishing shape programs, positioning could be executed at a rapid traverse rate after an approach in a cutting feedrate.

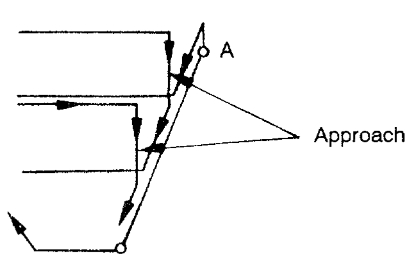


Fig. 4.29 Approach

- When parameter setting is "pm4026 D1 = 1", the cutting tool could interfere with the workpiece if the end point of finishing shape program for the monotonous decreasing shape or the shape with recess lies lower than the start point.

(c) Supplements to OD stock removal cycle

- U, W, I, and K are signed commands. If a wrong sign is used in the designation of these commands, overcuts will occur. The depth of cut D to be specified for each in-feed operation should be specified without a sign.

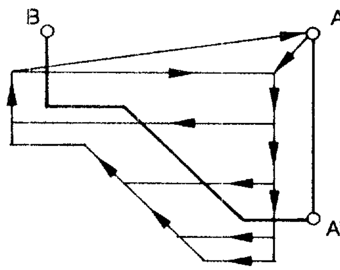


Fig. 4.30 In-feed Caused by Wrong Sign Used for U, W, I, and K

- Write the finishing shape program immediately after the G71 or G72 block. Blocks written between the G71/G72 block and the finishing shape program are disregarded.
- If F, S, and/or T code is not specified in the G71/G72 block, these codes specified in the preceding blocks are applied for the execution of the OD stock removal cycle. The F, S, and T codes specified in the finishing shape program are valid only for the execution of the finishing cycle (G70), and they are disregarded during OD stock removal cycle.
- The G codes that can be specified in blocks in the finishing shape program, excluding Nns and Nnf, are indicated in Table 4.5.

Table 4.5 Usable G Codes

Usable G Codes	Remark
G00, G01, G02, G03, G22, G23, G41, G42	–
G11, G12	To be counted as equivalent to two blocks
G111	To be counted as equivalent to four blocks
G112	To be counted as equivalent to five blocks

- For in-feed movement by D, override setting is possible in units of 10% in 21 steps in the range from 0 to 200%. Seting parameter pm0023 D0 to D4 is used (setting is made in a 5-bit code).
- If both I and K are omitted, it is possible to execute the cycle by using the finishing allowance U and W for the rough finishing allowance. (Valid by the setting of parameter pm4026 D0 = 1)

- If the nose R offset mode has been set before the execution of G71 or G72, the nose R offset is valid for the G71/G72 cycle.

  Therefore, in the program where rough finishing cycle is omitted (I = 0, K = 0), the nose R offset function is invalid. For the G70 to G73 cycles, it is possible to execute the nose R offset function in the finishing shape programs. Accordingly, G41 and G42 can also be specified in blocks in the finishing shape program with an exception of Nsf and Nnf blocks. In addition, in the rough finishing cycle and finishing cycle, the nose R offset function becomes valid from the block where G41 or G42 command is specified.

  If G41 or G42 is specified in the first block of the finishing shape program, G00 or G01 must also be specified in the same block along with an axis movement command. Designation of G41 or G42 in a block without other commands is not allowed.

- When the nose R offset function is called up for the finishing shape that has no recess in it

Example of Programming

```
N1  G50 X260. Z220. ;
N2  G00 S1000 M03 T0101 ;
N3  G42 ;
N4  X145. Z180. ;                                    OD stock removal cycle
                                                              ↓
N5  G71 P6 Q13 U1. W0.5 I2. K2. D4. F0.3 S800 ;
N6  G00 X40. S800 ;   ← In-feed at rapid traverse   ┐
N7  G01 W-40. F0.15 ;                               │
N8  X60. W-30. S600 ;                               │
N9  G12 W-20. I5. ;   ← Equivalent to 2 blocks      │
N10 G01 X100. W-10. S300 ;                          │  Finishing shape
N11 W-20. ;                                         │  = 9 blocks
N12 X140. W-20. S200 ;                              │
N13 X145. ;                                         ┘

N14 G40 ;
N15 G00 X260. Z220. T0100 ;
```

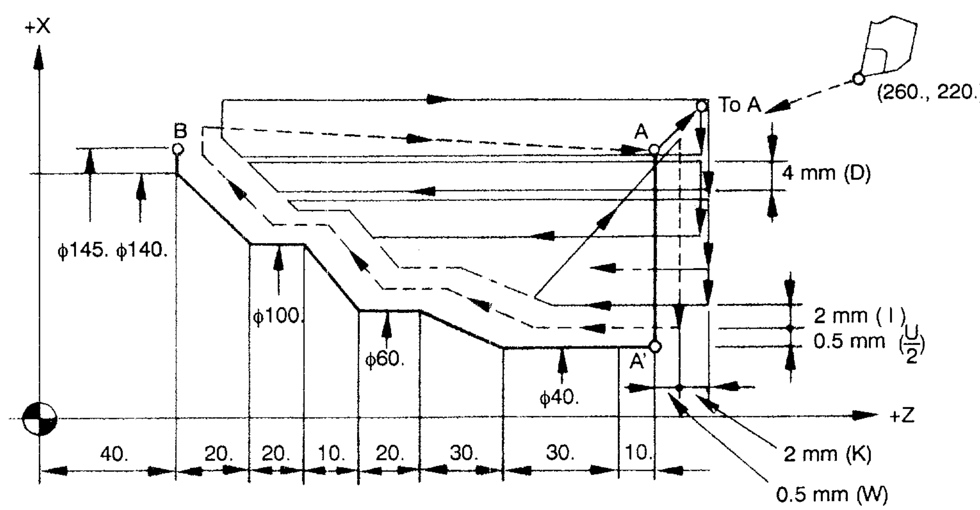


Fig. 4.31 When Nose R Offset is Called Up for the Finishing Shape that Has No recess in It

- When the nose R offset function is not called for the finishing shape that has recess in it

Example of Programming

```
N01 G50 X260. Z70. ;
N02 G00 S500 M03 T0101 ;
N03 X124. Z-10. ;

N04 G71 P5 Q14 U2. D6. F0.2 S250 R1 ;      ← OD stock removal cycle
N05 G01 X120. ;                              ┐
N06 X80. Z-50. F0.1 S500 ;                   │
N07 W-10. ;                                  │
N08 X110. W-10. ;                            │
N09 W-10. ;                                  │
N10 G02 X90. W-20. I15. K-20. ;              │ Finishing shape
N11 X110. W-20. I25. ;                       │
N12 G01 W-5. ;                               │
N13 X120. W-5. ;                             │
N14 X124. ;                                  ┘

N15 G00 X260. Z70. T0100 ;
N16 T0202 ;
N17 G50 X255. Z70. ;
N18 X124. Z-10. ;      ← Execution of the finishing cycle for G71
N19 G70 P5 Q14 ;
```

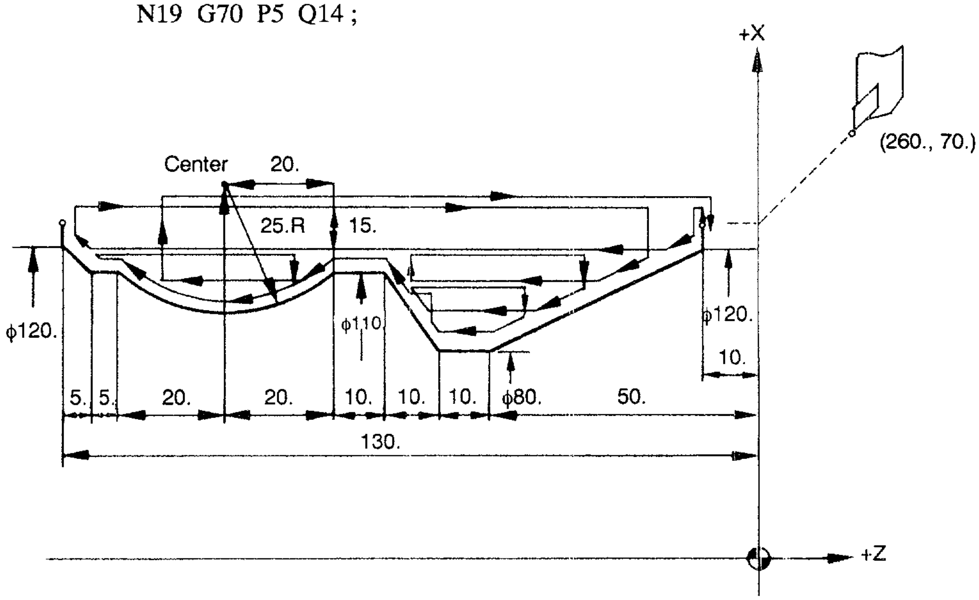


Fig. 4.32 When Nose R Offset is not Called for the Finishing Shape that Has Recess in It

(2) Face Rough Turning Cycle (G72)

With the G72 command, stock removal cycle and rough finishing cycle in which finishing allowance is left on face can be specified. In comparison to the cycle called by G71, which carries out cutting by the movement in parallel to the Z-axis, the G72 cycle carries out cutting by the movements parallel to the X-axis. Therefore, the cycle called by G72 executes the same operation as with the cycle called by G71 in a different direction. Read the supplements described for the G71 cycle before attempting programming for the G72 cycle.

(a) For the workpiece with monotonous increase/monotonous decrease finishing shape

If the finishing shape is monotonous increase/monotonous decrease, the following commands are used.

ns: Sequence number of cycle start

nf: Sequence number of cycle end

G72 Pns Qnf U± · · · W± · · · I± · · · K± · · · D · · · F(E) · · · S · · · ;

(Note)

Nns · · · · · · · · ;
· · · · · · · · ;
· · · F · · · ;
· · · S · · · ;
Nnf · · · · · · · · ;

D: Depth of cut in the X-axis direction in each in-feed (unsigned)

K: Rough finishing allowance in the Z-axis direction

I: Rough finishing allowance in the X-axis direction (radius designation)

W: Finishing allowance in the Z-axis direction

U: Finishing allowance in the X-axis direction (diameter designation)

Finishing shape program (Max. 45 blocks)

This program defines the shape to be finished (A → A' → B) and it should start with sequence number "ns" and end with "nf". Among the commands specified in this program, the F and S commands are valid only when the G72 finishing cycle is executed.

Note: Specify the feed command (F (E)) and spindle command (S) that are used for the execution of the OD stock removal cycle.

- The operation starts from point A; after executing the stock removal cycle (——) and rough finishing cycle (- - -), the cutting tool returns to point A and the operation ends.

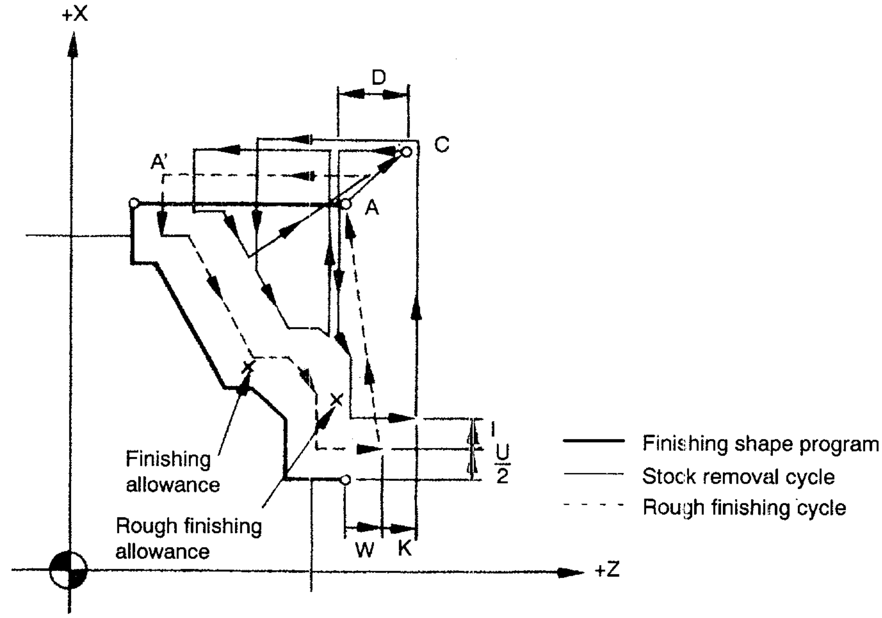


Fig. 4.33 Execution of the Cycle

- If "I = 0, K = 0 (or no designation)", the cycle finishes by skipping the rough finishing cycle as shown in Fig. 4.34.

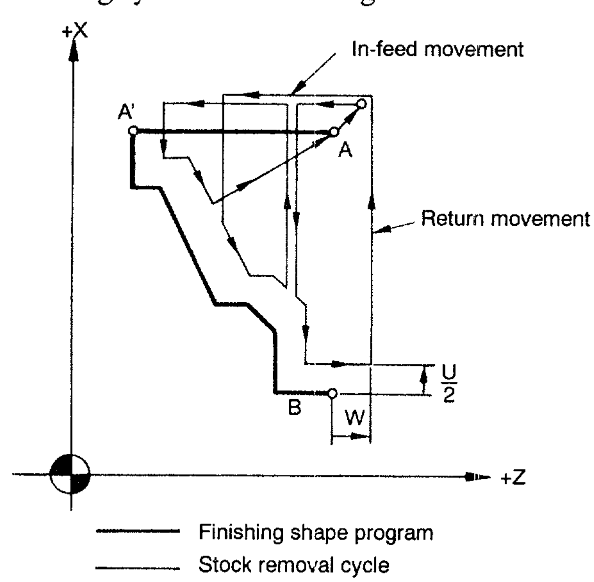


Fig. 4.34 Skipping the Rough Finishing Cycle

- The "return movement" is executed in the G00 (rapid traverse) mode. Concerning the "in-feed movement", it is executed at the feedrate (G00 or G01) specified in the program for $\overline{AA'}$. For feedrate for in-feeding by depth of cut D in the Z-axis direction, override setting is possible in 21 steps in units of 10% in the range from 0 to 200% by the setting for a parameter.
- The following restrictions apply to the start (Nns· · ·) and end (Nnf· · ·) blocks of the finishing shape program.

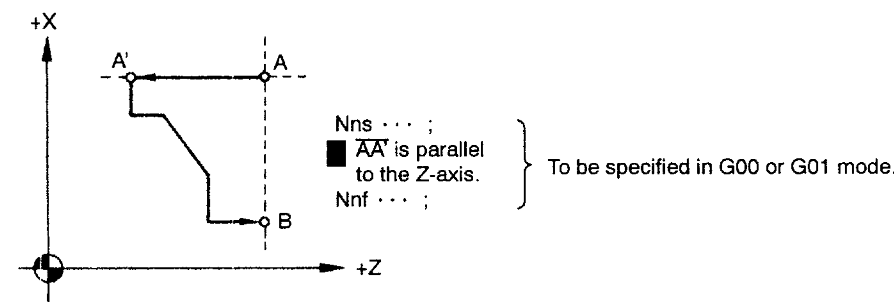


Fig. 4.35 Restrictions on Start and End Blocks

(b) For the workpiece with recesses in the finishing shape

If the finishing shape has recesses in it, the following commands are used.

ns: Sequence number of cycle start

nf: Sequence number of cycle end

G72 Pns Qnf W± · · · K± · · · D · · · F(E) · · · S · · · R1 ;

(Note 1) (Note 2)

```
Nns · · · · · · · · ;
    · · · · · · · · ;
    · · · · · · · · ;
    · · · · · · · · ;
Nnf · · · · · · · · ;
```

D: Depth of cut in the Z-axis direction in each in-feed (unsigned)

K: Rough finishing allowance in the Z-axis direction (radius designation)

W: Finishing allowance in the Z-axis direction (diameter designation)

Nns to Nnf: Finishing shape program (Max. 45 blocks) Program that defines the shape of A → A' → B

Note 1: Specify the feed command (F (E)) and spindle command (S) that are used for the execution of the OD stock removal cycle.

2: If "R1" is designated in the program, tool paths are calculated for the finishing shape program which has recesses.

- The operation starts from point A; after executing the stock removal cycle (——) and rough finishing cycle (– - -) , the cutting tool returns to point A and the operation ends. If address I is not designated, the rough finishing cycle is skipped.

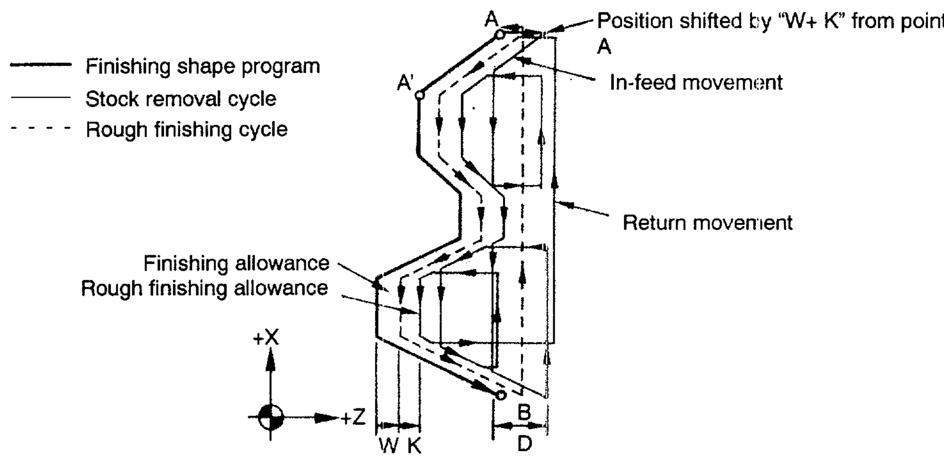


Fig. 4.36 Cycle Execution

POINT

Each block specified in the finishing shape program must define monotonous increasing or monotonous decreasing shape. An arc that extends over multiple quadrants must be programmed in two or more blocks.

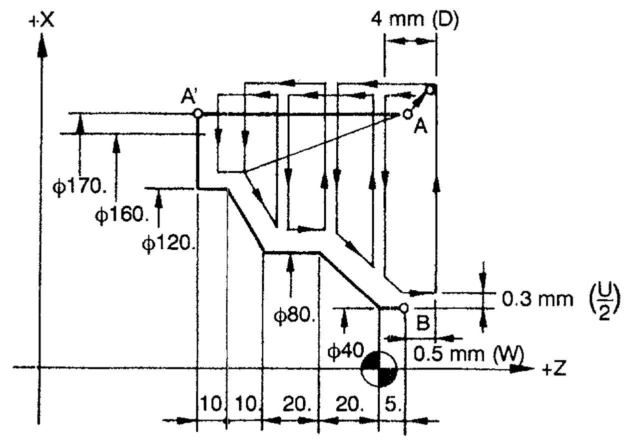
- The "return movement" is executed in the G00 (rapid traverse) mode. Concerning the "in-feed movement", it is executed at the feedrate (G00 or G01) specified in the program for $\overline{AA'}$. For feedrate for in-feeding by depth of cut D in the X-axis direction, override setting is possible in 21 steps in units of 10% in the range from 0 to 200% by the setting for a parameter.

SUPPLE-MENT

The retraction amount in G72 can be set for pm0862 (X-axis) and pm0863 (Z-axis).

4

- When the nose R offset function is not called up by "I = 0, K = 0 (or no designation)"



Example of Programming

```
N1  G50 X260. Z60. ;
N2  G00 S1000 M03 T0202 ;
N3  X170. Z5. ;                          Face stock removal cycle
N4  G72 P5 Q11 U0.6 W0.5 I0 K0 D4.0 F0.3 S200 ;
N5  G01 Z-60. F0.15 ;   ← In-feed at cutting feedrate
N6  X120. S250 ;
N7  Z-50.
N8  X80. Z-40. S400 ;                    Finishing shape program
N9  Z-20. ;
N10 X40. Z0 S800 ;
N11 Z5. ;
N12 G00 X260. Z60. ;
N13 T0303 ;
N14 X170. Z5. ;
N15 G70 P5 Q11 ;        ← Executes the finishing cycle
```

Fig. 4.37 When Nose R Offset is not Called Up by "I = 0, K = 0 (or No Designation)"

- An example of finishing shape for G71 and G72 is shown below.

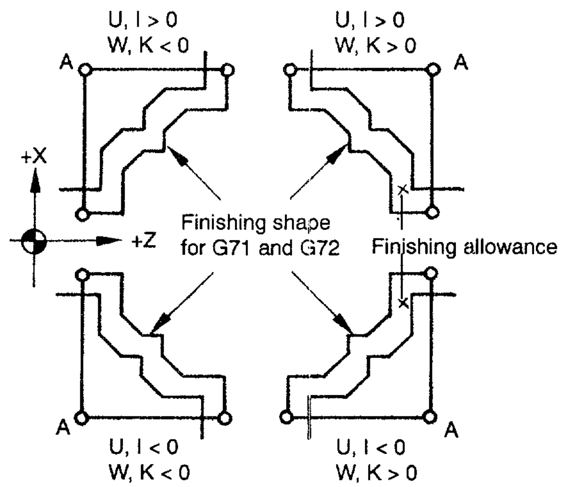


Fig. 4.38 Relationships between Addresses U, W, I, and K and Finishing Shape Programs for G71 and G72

(3) Pattern Repeat Cycle (G73)

The G73 pattern repeat cycle is effective when machining a workpiece that has similar shape to the finishing shape like cast and forged workpieces. The following commands are used to execute the pattern repeat cycle.

ns: Sequence number of cycle start

nf: Sequence number of cycle end

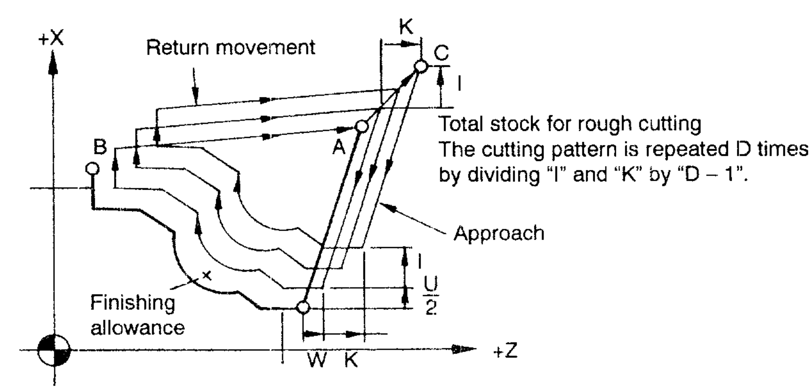
G72 Pns Qnf U +· · · W + · · · I + · · · K +· · · D · · · F(E) · · · S · · · ;

(Note)

Nns · · · · · · · · ;
· · · · · · · · ;
· · · F · · · ;
· · · S · · · ;
Nnf · · · · · · · · ;

- D: Number of rough cutting cycles (1 ≦ D ≦ 127; unsigned)
- K: Total Z-axis stock for rough cutting (signed)
- I: Total X-axis stock for rough cutting (signed) (radius designation)
- W: Finishing allowance in the Z-axis direction (signed)
- U: Finishing allowance in the X-axis direction (signed) (diameter designation)
- Nns … Nnf: Finishing shape programs (Max. 39 blocks)
  This is the finishing shape program for A → A' → B and the program should start with the sequence number of "ns" and ends with "nf". The F and S commands specified in these blocks are valid for the execution of the finishing cycle called by G70.

Note: Specify the feedrate (F (E)) and spindle speed (S) to be used for the execution of the pattern repeat cycle.



—— Finishing shape program
—— Pattern repeating stock removal cycle · · · The operation ends by executing this cycle. The operation starts and ends at point A.

Fig. 4.39 Pattern Repeating Cycle

- The "return movement" is executed in the G00 (rapid traverse) mode. Concerning the "approach", it is executed at the feedrate (G00 or G01) specified in the program for $\overline{AA'}$.

Example of Programming

```
N10 G50 X260. Z220. ;
N11 G00 S300 M03. T0303 ;
N12 X220. Z160. ;
N13 G73 P14 Q19 U2. W1. I8. K8. D3 F0.3 S200 ;  ← Pattern repeat cycle
N14 G00 X80. W-40. S400 ;        ┐
N15 G01 W-20. F0.15 ;            │
N16 X120. W-10. S300 ;           │ Finishing shape program
N17 W-20. ;                      │
N18 G02 X160. W-20. R20. S200 ;  │
N19 G01 X180. W-10. ;            ┘
N20 G00 X260. Z220. ;
```

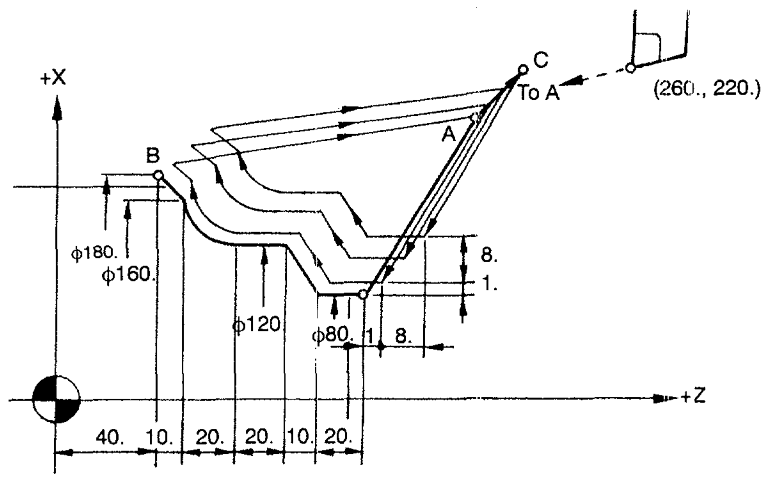


Fig. 4.40 Pattern Repeat Cycle

- The number of rough cutting pattern to be repeated (D) should be specified by an unsigned value. The following restriction applies to the designation of address D.
  - $1 \leqq D \leqq 127$
    Alarm "0467" occurs if a value outside the range indicated above is specified. With the designation of "D = 1", rough cutting is executed one time at depth of cut of I and K to leave the finishing allowance.
  - It is necessary to specify the finishing shape immediately after the G73 block.
  - The start (Nns · · ·) and end (Nnf · · ·) blocks of the finishing shape program must be designated in either the G00 or G01 mode. Note that tool paths may not be parallel to the X- or Z-axis.
  - The shape defined by the finishing shape program may not be monotonous increasing or monotonous decreasing shape.

4

- If F, S, and/or T code is not specified in the G73 block, these codes specified in the preceding blocks are applied for the execution of the OD stock removal cycle. The F, S, and T codes specified in the finishing shape program are valid only for the execution of the finishing cycle (G70), and they are disregarded during OD stock removal cycle.
- The G codes that can be specified in blocks in the finishing shape program, excluding Nns and Nnf, are indicated in Table 4.6.

Table 4.6 Usable G Codes

Usable G Codes	Remark
G01, G06, G02, G03, G22, G23, G41, G42	–
G11, G12	To be counted as equivalent to two blocks
G111	To be counted as equivalent to four blocks
G112	To be counted as equivalent to five blocks

- If designation of I and K (total stock for rough cutting) is both "0" or neither of these addresses are specified, alarm "0467" occurs.

  $\triangle$I and $\triangle$K, which indicate stock removal per one cycle of rough cutting, are calculated by the following:

  $\triangle I = \frac{I}{D-1}$, $\triangle K = \frac{K}{D-1}$ (where, $D \leqq 2$)

  In this calculation, a value smaller than 0.001 mm is rounded off. Do not write a program in which a value of $\triangle$I and/or $\triangle$K becomes smaller than 0.001 mm.

  (Example 1) With the program in which I = 0.005 mm, K = 0.005 mm, and D = 7,

  $\triangle I = \frac{0.005}{6} = 0$, $\triangle K = \frac{0.005}{6} = 0$ is obtained.

  Thus, alarm "0467" occurs.

  (Example 2) With the program in which I = 0.01 mm, K = 0.01 mm, and D = 7,

  $\triangle I = \frac{0.01}{6} = 0.001$ mm, $\triangle K = \frac{0.01}{6} = 0.001$ mm

  is obtained. Thus, each cycle is executed with the stock amount indicated below.

  1st to 6th cycle : $\triangle I = \triangle K = 0.001$ mm

  7th cycle : $\triangle I = \triangle K = 0.004$ mm

  If the nose R offset mode has been set prior to the designation of the G73 cycle, the nose R offset function is valid for all G73 cycles.

4

(4) Finishing Cycle (G70)

After carrying out rough cutting cycle by using the G71, G72, and G73 cycles, finish cutting can be carried out by specifying the G70 cycle.

G70 Pns Qnf ;

- nf: Finishing cycle end sequence number
- ns: Finishing cycle start sequence number

Only the finishing shape program, specified before the G71, G72, or G73 cycle, is executed by the commands indicated above. During the execution of the finishing cycle G70, the F(E), S, and T codes specified in the finishing shape program are valid. Those specified in the G71, G72, or G73 block for rough turning are invalid for a finishing cycle.

(a) Prohibited commands and operation

It is not necessary to specify the G70 block immediately after the designation of the G71, G72, or G73 cycle. For example, it is allowed to enter commands to change the cutting tool from a roughing tool to a rough finishing tool between them. However, the commands or the operation indicated in Table 4.7 must not be entered between them.

Table 4.7 Prohibited Commands and Operation

Prohibited Commands and Operation	Results
M02 and M30 which are associated with the internal reset processing	The finishing shape program is deleted from the memory.
Resetting operation	

(b) Save and search function for the finishing shape program

The processing for the finishing shape program differs depending on the operation mode -- TAPE mode or MEM mode.

TERM?

◆ Finishing Shape Program Memory

The "finishing shape program memory" is the special memory provided in the NC to store binary converted program so that the processing time for the stock removal cycle is shortened.

- In the TAPE mode

```
┌─ G71 Pns Qnf ··· ;        ← OD stock removal cycle commands
│  Nns ··· ;      ─┐
│   ·              │
│   ·              │← Finishing shape program (A)
│   ·              │
└─ Nnf ··· ;      ─┘
   G70 Pns Qnf ··· ;        ← Execution of finishing by program (A)
┌─ G72 Pns' Qnf' ··· ;      ← Face rough turning cycle commands
│  Nns' ··· ;     ─┐
│   ·              │
│   ·              │← Finishing shape program (B)
│   ·              │
└─ Nnf' ··· ;     ─┘
```

After the execution of the commands indicated above, the finishing shape program (A) is cleared and finishing shape program (B) remains in the internal memory. Therefore, the finishing cycle specified by the G70 can be used for the finishing shape program (B). If the sequence number specified in the G70 block does not agree with the sequence number in the finishing shape program memory, alarm "0462" occurs.

4

- In the MEM (memory) mode

If the sequence number specified in the G70 block and the one in the finishing shape program memory agree with each other, the finishing cycle is executed. If they do not agree with each other, the finishing shape program is searched in the part program; the found program is once saved to the internal memory and then executed. This function is called the "finishing shape program search function". This search function is executed only in the part program of the program number in which the G70 command has been specified. If this function is used, cycle time will be longer than the time required in executing the program without using this function.

Only in the MEM mode operation, this function allows more than two stock removal cycles (pattern repeating cycles) to be executed which are followed by the respective finishing cycles as indicated below:

Stock removal cycle (A)
↓
Stock removal cycle (B)
↓
Finishing cycle (A)
↓
Finishing cycle (B)

(c) Supplements to the finishing cycle

- If the sequence numbers "ns" and "nf" of the start and end of the finishing cycle are as indicated below, an alarm occurs.
  - If sequence numbers "ns" and "nf" specified with G70 do not agree with the sequence numbers stored in the finishing shape program memory (TAPE mode operation).
  - In the finishing shape program, if the sequence number "ns" appears before "nf", both specified in the G70 block. If "ns = nf", it also causes an alarm.
- If the nose R offset mode has been set before the designation of G70, the nose R offset function is executed for the finishing cycle of G70.

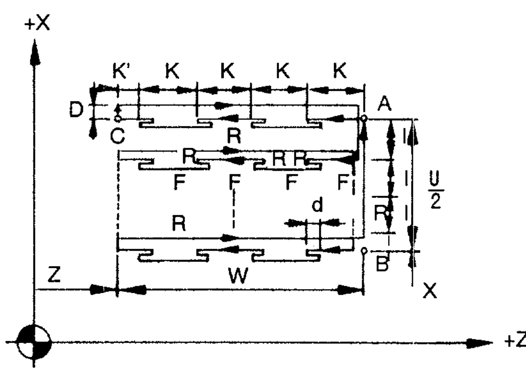
4

(5) Face Cut-off Cycle (G74)

In the cycle called by G74, peck feed operation parallel to the Z-axis is repeated to carry out face cut-off cycle. For the execution of the face cut-off cycle, the following commands are used.

G74 X(U)± · · · Z(W)± · · · I · · · K · · · D · · · F(E) · · · (R1) ;

- F(E) — Feed command
- D — Retraction amount at cutting bottom (unsigned)
- K — Depth of cut in the Z-axis direction (unsigned)
- I — Move distance in the X-axis direction (unsigned)
- Z(W) — Z-coordinate of point C
- X(U) — X-coordinate of point B



R : Rapid traverse
F : Feed designated by F code
d : Retraction amount
(setting parameter pm0864)
Dwell at the cutting bottom
(setting parameter pm0402)

Note: The illustration above indicates the operation when "R1" is not specified. If "R1" is specified, the cutting tool returns to the in-feed start point, point A level, for each in-feed operation disregarding of the retraction amount (d). The cycle starts and ends at point A.

Fig. 4.41 Face Cut-off Cycle

TERM?

◆ Peck Feed Operation

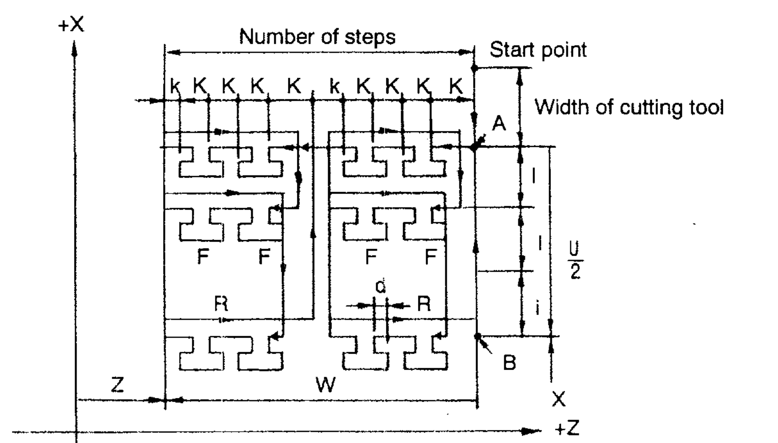
Peck feed operation indicates the operation in which the cutting axis repeats advance and retraction movements to carry out the specified cutting.

(a) Grooving canned cycle

By specifying addresses A and B with the G74 command, grooving canned cycle is executed with the number of in-feed steps and the width of cutting tool taken into consideration.

G74 X(U) ··· Z(W) ··· I ··· K ··· D ··· A ··· B ··· F(E) ··· (R1) ;

- F(E): Feed command
- B: Width of cutting tool (unsigned)
- A: Number of in-feed steps (unsigned)
- D: Retraction amount at cutting bottom (unsigned)
- K: Depth of cut in the Z-axis direction (unsigned)
- I: Move distance in the X-axis direction (unsigned)
- Z(W): Z-coordinate of point C (signed)
- X(U): X-coordinate of point B (signed)



R : Rapid traverse
F : Feed designated by F code
d : Retraction amount
(setting parameter pm0864)
Dwell at cutting bottom
(setting parameter pm0402)

Fig. 4.42 Grooving Canned Cycle

- If none of A and B is specified, the same cycle as the normal G74 cycle is executed.

- If only address B is specified, shift operation by the width of cutting tool is executed at the start and end of the G74 cycle.

  The shift movement at the start of the G74 cycle is made from the point where positioning has been made in the block immediately before the G74 block by the width of the cutting tool in the specified X-axis direction.

  At the end of the G74 cycle, the cutting tool is first positioned to the point where the shift has been made at the start of the G74 cycle and then returns to the position where positioning was made by the commands in the block immediately preceding the G74 block.

- If only address A is specified, shift operation is not executed but only grooving operation is executed.
- If address A is specified, the retraction amount is as set for parameter pm0867. If "0" is set for this parameter, peck feed operation is not executed.
- Alarm "0472" occurs if groove width < B (width of cutting tool).

Example of Programming

G74 X40. Z50. I4. K15. D1. F0.25 ;

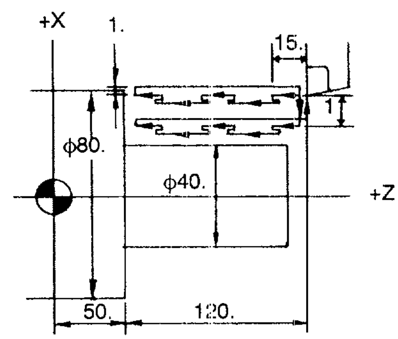


Fig. 4.43

(b) Supplements to face cut-off cycle

- If the given commands are "I > | U/2 |", peck feed operation starts from and ends at point A.
- If "K > | W |" or the parameter setting is "pm0864 = 0", cutting is executed to the bottom in one time without including peck feed operation.
- If "D = 0" or address D is not specified, retraction movement is not executed at the bottom.
- The final depth of cut K' in the Z-axis direction and the final movement amount I' in the X-axis direction are automatically calculated.
- If X (U), I, and D are not specified, such designation calls out 1 cycle operation with only Z-axis. This cycle can be used for deep-hole drilling cycle.
- The nose R offset function is invalid for the G74 cycle.

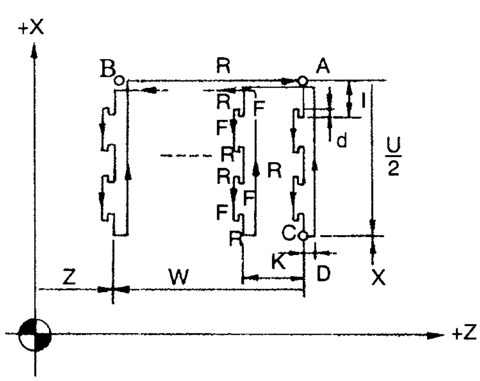
(6) OD Cut-off Cycle (G75) Commands

The G75 cycle executes an OD cut-off cycle while carrying out peck feed operation parallel to the X-axis. In comparison to the G74 cycle in which the OD cut-off cycle is executed in parallel to the X-axis, the G75 cycle executes virtually the same operation excluding that the cycle is executed in parallel to the X-axis. Read the supplements to the G74 commands given before.

The OD cut-off cycle is executed by using the following commands.

G75 X(U)± · · · Z(W)± · · · I · · · K · · · D · · · F(E) · · · (R1) ;

- F(E) · · · — Feed command
- D · · · — Retraction amount at cutting bottom (unsigned)
- K · · · — Move distance in the Z-axis direction (unsigned)
- I · · · — Depth of cut in the X-axis direction (unsigned)
- Z(W)± · · · — Z-coordinate of point B
- X(U)± · · · — X-coordinate of point C



R : Rapid traverse
F : Feed designated by F code
d : Retraction amount
(setting parameter pm0865)
Dwell at cutting bottom
(setting parameter pm0402)

Note: The illustration above indicates the operation when "R1" is not specified. If "R1" is specified, the cutting tool returns to the in-feed start point, point A level, for each in-feed operation disregarding of the retraction amount (d). The cycle starts and ends at point A.

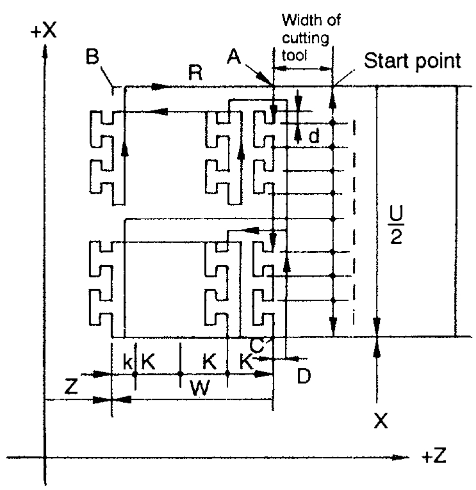
Fig. 4.44 OD Cut-off Cycle

(a) Grooving canned cycle

By specifying addresses A and B with the G75 command, grooving canned cycle is executed with the number of in-feed steps and the width of cutting tool taken into consideration.

G74 X(U) ··· Z(W) ··· I ··· K ··· D ··· A ··· B ··· F(E) ··· (R1) ;

- X(U): Z-coordinate of point C (signed)
- Z(W): X-coordinate of point B (signed)
- I: Move distance in the X-axis direction (unsigned)
- K: Depth of cut in the Z-axis direction (unsigned)
- D: Retraction amount at cutting bottom (unsigned)
- A: Number of in-feed steps (unsigned)
- B: Width of cutting tool (unsigned)
- F(E): Feed command



R : Rapid traverse
F : Feed designated by F code
d : Retraction amount (setting parameter pm0865)
Dwell at cutting bottom (setting parameter pm0402)

Fig. 4.45 Grooving Canned Cycle

4

- If none of A and B is specified, the same cycle as the normal G75 cycle is executed.
- If only address B is specified, shift operation by the width of cutting tool is executed at the start and end of the G75 cycle.

  The shift movement at the start of the G75 cycle is made from the point where positioning has been made in the block immediately before the G75 block by the width of the cutting tool in the specified Z-axis direction.

  At the end of the G75 cycle, the cutting tool is first positioned to the point where the shift has been made at the start of the G75 cycle and then returns to the position where positioning was made by the commands in the block immediately preceding the G75 block.
- If only address A is specified, shift operation is not executed but only grooving operation is executed.
- If address A is specified, the retraction amount is as set for parameter pm0868. If "0" is set for this parameter, peck feed operation is not executed.
- Alarm "0472" occurs if groove width < B (width of cutting tool).

Example of Programming

N1 G00 X86. Z70.
N2 G75 X50. Z40. I6. K4. (D0) F0.2 ;

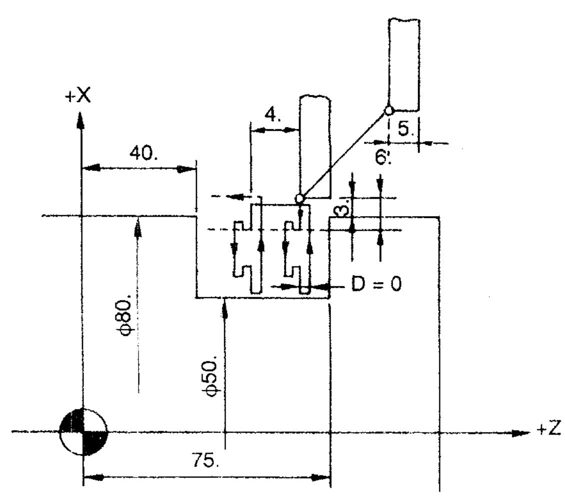


Fig. 4.46

(7) Automatic Thread Cutting Cycle (G76) Commands

G76 calls an automatic thread cutting cycle for cutting straight or taper thread in which in-feed is made along thread angle. The following commands are used for the execution of the automatic thread cutting cycle.

G76 X(U) ± · · · Z(W) ± · · · I ± · · · K · · · D · · · F(E) · · · A · · · ;

- X(U) ± · · · — X-coordinate of point C
- Z(W) ± · · · — Z-coordinate of point C
- I ± · · · — X-axis distance at taper (radius designation)
- K · · · — Height of thread (unsigned)
- D · · · — 1st pass in-feed (unsigned)
- F(E) · · · — Thread lead
- A · · · — Angle of thread (deg)

4

- The sign of address I is determined by the direction viewing point B' from point C. The automatic thread cutting cycle starts and ends at point A.

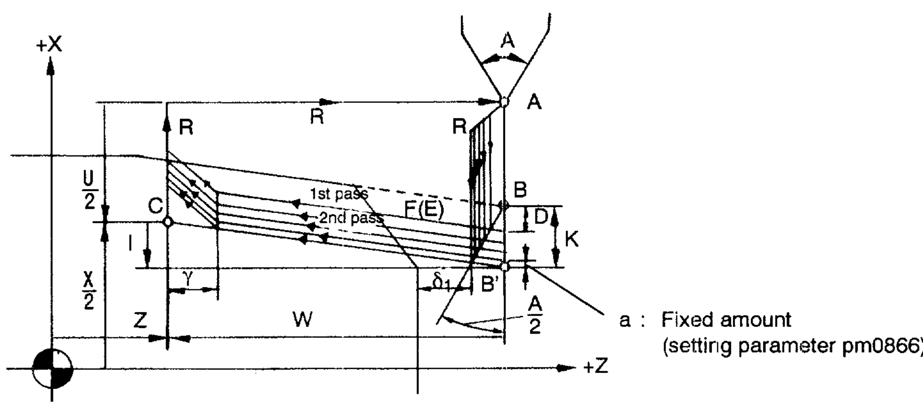


Fig. 4.47 Execution of Automatic Thread Cutting Cycle

- How the cutting is tool is moved near point B is shown in Fig. 4.48 (taper thread).

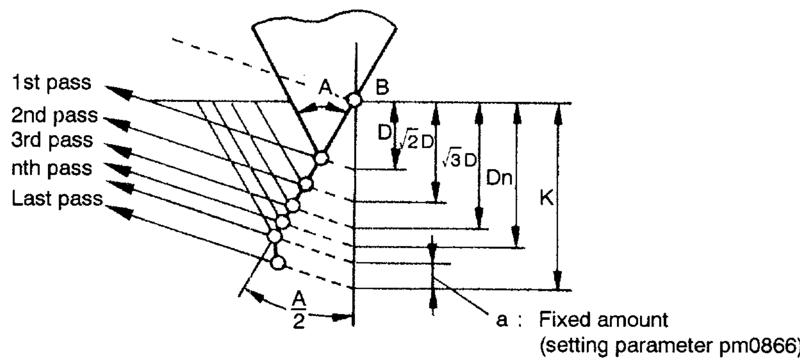


Fig. 4.48 In-feed Near Point B

4

- Depth of cut Dn for the nth in-feed movement is $Dn = \sqrt{n}\,D$

  For angle of thread, designation is possible from the following six angles: 0°, 29°, 30°, 55°, 60°, 80°. In the last pass of thread cutting cycle, the depth of cut is fixed to the predetermined value "a" (in the X-axis direction) which is set for setting parameter pm0866.

(a) Straight thread cutting

If address I is "0" or address I not specified, straight thread cutting cycle as shown in Fig. 4.49 is executed.

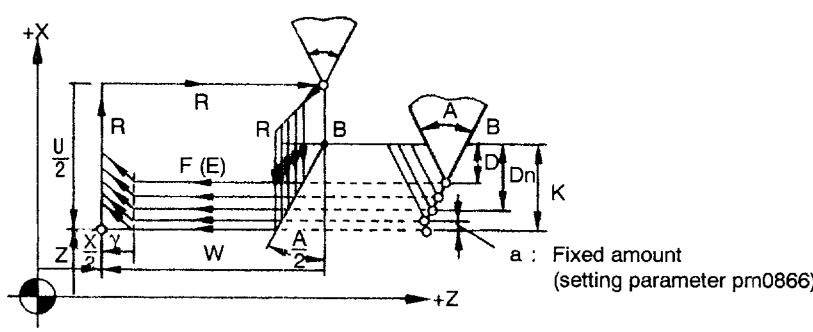


Fig. 4.49 Straight Thread Cutting

Example of Programming

G00 X66. Z115. ;
G76 X56.2 Z30. K3.9 D2. F6. A60 ;
G00 · · ·
.
.
.

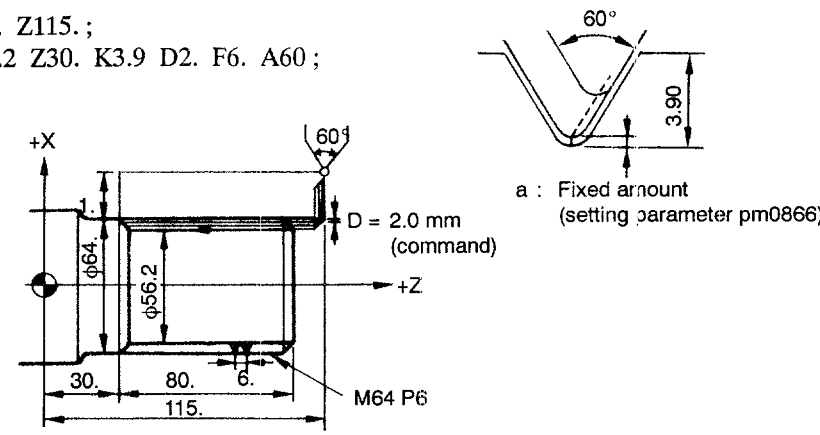


Fig. 4.50 Example of Programming

If fixed depth of cut "a" is set to 0.2 mm, the depth of cut in the individual thread cutting passes is as indicated below.

1st pass .... 1.700 mm
2nd pass .... 2.528 mm
3rd pass .... 3.164 mm
4th pass .... 3.700 mm
5th pass .... 3.900 mm

Although 2.00 mm is specified in the program for the depth of cut in the first pass, actual depth of cut is determined to 1.7 mm as the result of calculation of $\sqrt{nend}$ D which calculates the difference.

- If the "thread chamfering input (CDZ)" is ON when G76 is specified, thread chamfering is executed. Thread chamfering size γ can be set for parameter pm0100 in increments of 0.1L in the range from 0 to 25.5L. Here, "L" represents the specified thread lead.

- By adding an L command to the G76 mode commands, it is possible to execute the cycle by n times counted from the final pass.

  G76 X(U) ± · · · Z(W) ± · · · I ± · · · K · · · D · · · F(E) · · · A · · · L · · · ;

  L0 = The commands of the final pass are executed.
  L1 = The cycle one before the final pass and the final pass are executed.

  .
  .
  .

  Ln= The cycle is executed from the "n" times before the final pass to the final pass.
  (If value "n" is greater than the normal number of cycle execution times (N), normal thread cutting cycle is executed.)

- It is possible to execute zig-zag in-feed mode thread cutting cycle with constant metal removal amount by adding a P command.

  G76 X(U) ± · · · Z(W) ± · · · I ± · · · K · · · D · · · F(E) · · · A · · · P · · · ;

  The P command determines how the in-feed is made for thread cutting operation as indicated below.

No P command	:	Constant metal removal amount, in-feed on one side
P1	:	Constant metal removal amount, in-feed on one side
P2	:	Constant metal removal amount, zig-zag in-feed mode
P3 or greater	:	Constant metal removal amount, in-feed on one side

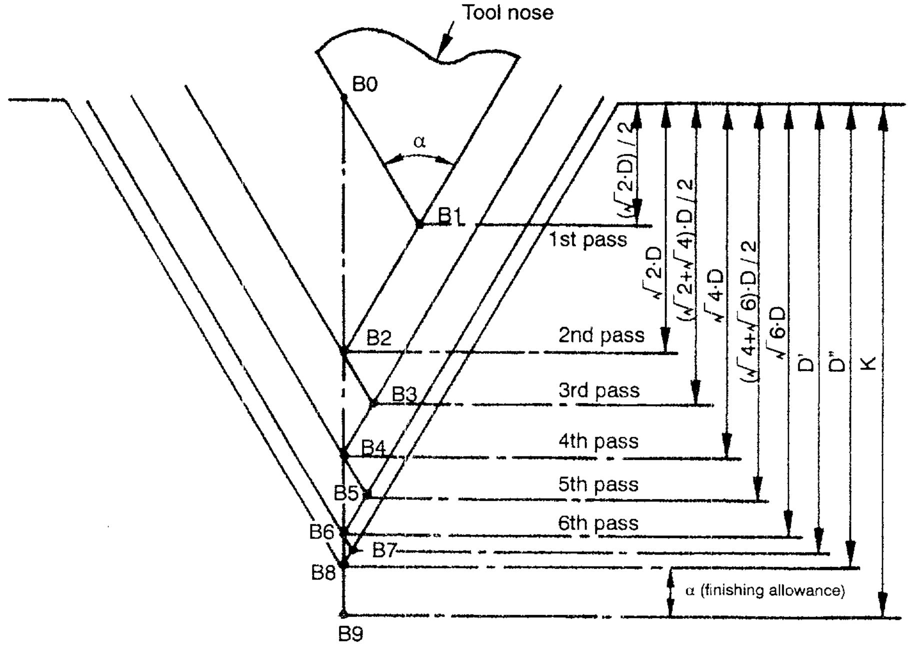


Fig. 4.51 Constant Metal Removal Amount, Zig-zag In-feed Mode Thread Cutting

(b) Taper thread cutting

If a tapered thread is designated with "A ≠ 0", the X-coordinate value of the thread cutting start point is not always Dn, which expresses the depth of cut.

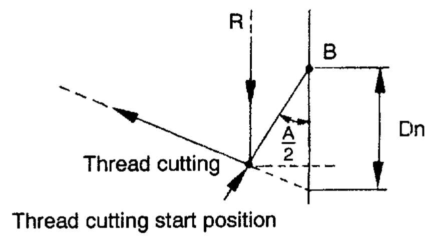


Fig. 4.52 Thread Cutting Start Position for Tapered Thread Designated with "A ≠ 0"

(c) Supplements to automatic thread cutting cycle

- If an angle other than six allowable angle values for the angle of thread (0°, 29°, 30°, 55°, 60°, 80°) is specified, the larger closest angle is selected.

  (Example) If "A15" is specified, actual thread cutting cycle is executed with "A29". However, if the specified value A is "A > 80°", it is replaced with "A80" to execute thread cutting cycle.

- If depth of cut along the angle of thread in the last thread cutting pass, $\sqrt{nend}\,D$, does not agree with the value of "K – a", the difference between these two values is deducted from the depth of cut applied to the first pass. The depth of cut in the first pass is not greater than the specified value D in any case.

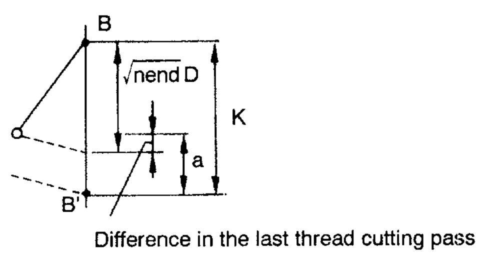


Fig. 4.53 Depth of Cut in the 1st Thread Cutting Pass when "$\sqrt{nend}\,D \neq (K-a)$"

(Example) D = 5.0 mm, K = 9.8 mm
If fixed amount is "a = 0.2 mm":

$\sqrt{nend}\,D = \sqrt{4} \times 5.000 = 10.000$ mm

Difference $= \sqrt{nend}\,D - (K - a) = 10.000 - (9.800 - 0.200) = \underline{0.400} > 0$

As the result of calculation indicated above, the depth of cut in each pass is determined as indicated below.

1st pass .... $5.000 - 0.400 = 4.600$ mm
2nd pass .... $\sqrt{2} \times 5.000 - 0.400 = 6.671$ mm
3rd pass .... $\sqrt{3} \times 5.000 - 0.400 = 8.261$ mm
4th pass .... $\sqrt{4} \times 5.000 - 0.400 = 9.600$ mm
5th pass .... $9.600 + 0.200 = 9.800$ mm

- If the thread cutting feed hold option is selected, thread chamfering is executed immediately when the FEED HOLD button is pressed during the execution of thread cutting cycle. After the completion of chamfering, the cutting tool returns to the start point A. If the setting for parameter pm4011 D2 is "1" (pm4011 D2 = 1), the cutting tool stops at the point B where chamfering is completed. The cutting tool returns to point A when the CYCLE START button is pressed after that.

  If the thread cutting feed hold option is not selected, the thread cutting cycle is continued even if the FEED HOLD button is pressed during the execution of thread cutting cycle. In this case, the operation is suspended upon completion of retraction operation after finishing the thread cutting cycle.

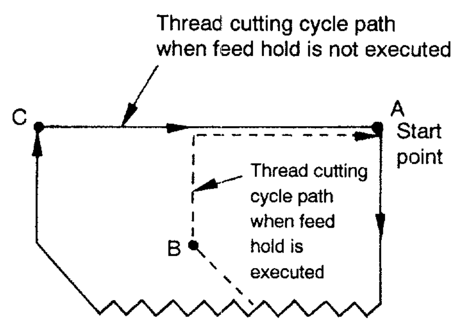


Fig. 4.54 Feed Hold during Thread Cutting Cycle

- Nose R offset is invalid for the G76 cycle.
- In the block immediately after the G76 block, it is necessary to newly specify a G code of 01 group.

  (Example) G76 · · · · · · · · · ;
  G00 M30 ;

(8) Supplements to Multiple Repetitive Cycles

- In the multiple repetitive cycle mode (G70 to G76), MDI mode operation is not allowed.
- It is not possible to execute G70 to G76 cycles by the MDI mode operation.
- If G70 to G76 cycle is executed with the SINGLE BLOCK switch ON, the cycle is executed n the manner as indicated in Table 4.8.

Table 4.8 Single Block Operation

G Code	Operation
G70, G71, G72, G73, G74, G75	Block stop occurs in minimum units of blocks into which the cycle is broken down.
G76	Block stop occurs at point A for the execution of each cycle.

- If the block specified immediately after the designation of G70 to G76 cycle, it is necessary to specify a G code of 01 group again. This is because the G code in this group could have been changed from the one set before the entry to the G70 to G76 cycle due to the execution of the cycle.
- For the commands specified in the G71 to G76 cycle, it is possible to specify the symmetric patterns as shown in Fig. 4.55. With G71 to G73, this is specified by the direction the finishing shape program in reference to point A. With G74 to G76, four patterns can be specified by the commands position of point (X, Z) or (U, W) for point A.

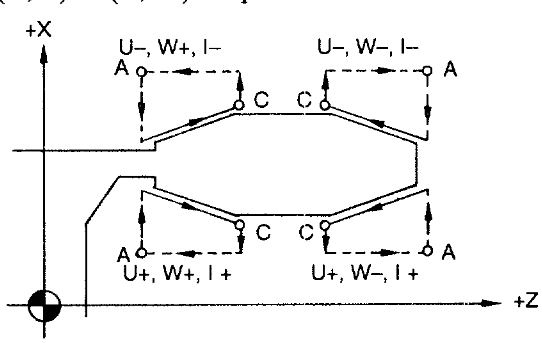


Fig. 4.55 Four Patterns

4.1.3 Multiple Chamfering/Rounding on Both Ends of Taper (G111) *

The following four movements can be specified by the commands in a single block if G111 function is used: taper → chamfering/rounding → taper → chamfering/rounding. Representative shapes for which the multiple chamfering/rounding are used are shown in Fig. 4.56.

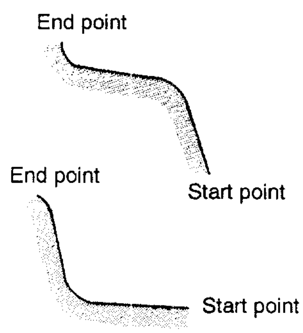


Fig. 4.56 Examples of Multiple Chamfering

G111 is a non-modal G code and valid only in the specified block.

(1) Programming Format

(a) Rounding

Example of Programming

G111 X (U) · · · I · · · A · · · B · · · P · · · Q · · · ;
G111 X (U) · · · K · · · A · · · B · · · P · · · Q · · · ;
G111 X (U) · · · I · · · K · · · B · · · P · · · Q · · · ;

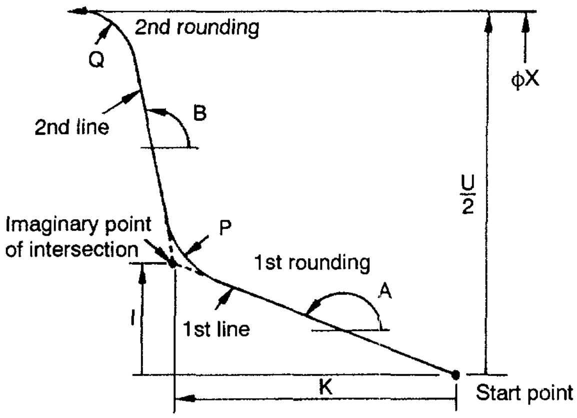


Fig. 4.57 Rounding on Both Ends of Taper and Programming Format (X-axis Commands)

Example of Programming

G111 Z (W) · · · I · · · A · · · B · · · P · · · Q · · · ;
G111 Z (W) · · · K · · · A · · · B · · · P · · · Q · · · ;
G111 Z (W) · · · I · · · K · · · B · · · P · · · Q · · · ;

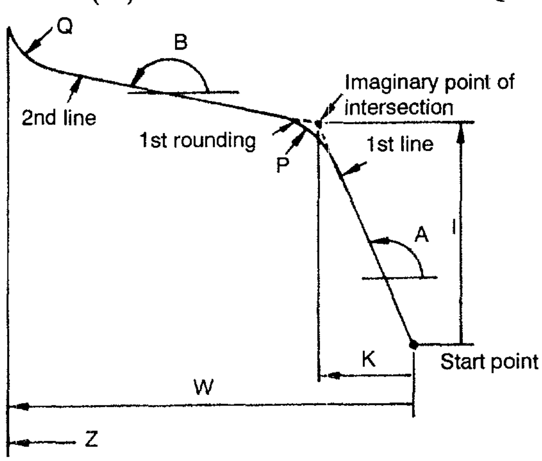


Fig. 4.58 Rounding on Both Ends of Taper and Programming Format (Z-axis Commands)

(b) Chamfering

Example of Programming

G111 X (U) · · · I · · · A · · · B · · · C · · · D · · · ;
G111 X (U) · · · K · · · A · · · B · · · C · · · D · · · ;
G111 X (U) · · · I · · · K · · · B · · · C · · · D · · · ;

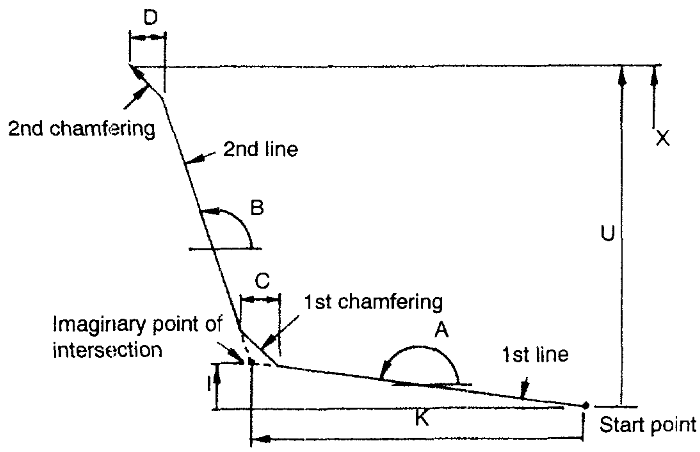


Fig. 4.59 Chamfering on Both Ends of Tapers and Programming Format (X-axis Commands)

Example of Programming

G111 Z (W) · · · I · · · A · · · B · · · C · · · D · · · ;
G111 Z (W) · · · K · · · A · · · B · · · C · · · D · · · ;
G111 Z (W) · · · I · · · K · · · B · · · C · · · D · · · ;

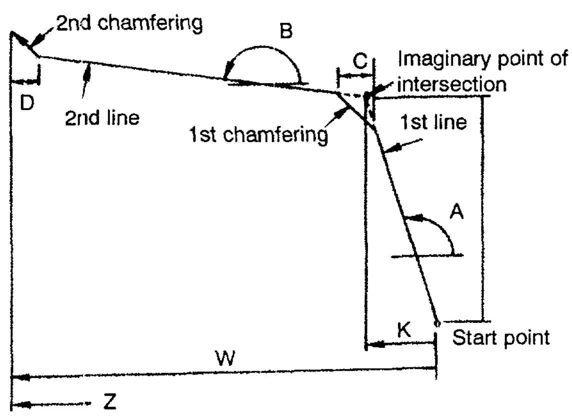


Fig. 4.60 Chamfering on Both Ends of Tapers and Programming Format (Z-axis Commands)

(c) Addresses

The addresses indicated in Table 4.9 are used when designating multiple chamfering/rounding on both ends of taper function. The required movements can be executed by specifying only addresses that define the required shape.

Table 4.9 Addresses and Meaning

Address	Description	Input Increment Unit
X (U)	X-coordinate of end point (U: Incremental amount from the start point)	1 = 0.001 mm or 1 = 0.0001 inch
Z (W)	Z-coordinate of end point (W: Incremental amount from start point)	
A	Move angle of the 1st line	1 = 0.001 deg
B	Move angle of the 2nd line	
I	Imaginary point of intersection between the 1st and 2nd lines, X-axis distance from the start point (radius value)	1 = 0.001 mm or 1 = 0.0001 inch
K	Imaginary point of intersection between the 1st and 2nd lines, Z-axis distance from the start point (radius value)	
P	1st rounding radius (unsigned)	
Q	2nd rounding radius (unsigned)	
C	1st chamfering size (unsigned)	
D	2nd chamfering size (unsigned)	

4

(2) Defining the Shape

To define the shapes of tapers and chamfering/rounding shapes, follow the instructions given in Table 4.10.

Table 4.10 Defining the Shapes

Shape	Definition	
1st line	A : Move angle of the 1st line I : X-axis distance from the start point to the imaginary point of intersection K : Z-axis distance from the start point to the imaginary point of intersection	Specify two of these addresses.
1st chamfering/ rounding	C : 1st chamfer size P : 1st rounding radius	Specify either of these addresses.
2nd line	B : Move angle of the 2nd line X (U) : X-coordinate of the end point (U : End point of X-axis Incremental amount from the start point) Z (W): Z-coordinate of end point (W : End point of Z-axis Incremental amount from the start point)	Specify either of these. Note that the following designation is not allowed: • Combination of X and U commands • Combination of Z and W commands
2nd chamfering or rounding	D : 2nd chamfer size Q : 2nd rounding radius	Specify either of these addresses.

(a) 1st rounding

The 1st rounding indicates rounding at the corner made by the 1st and 2nd lines.

4

(b) 2nd chamfering/rounding

The 2nd chamfering/rounding is made according to the commands defining the 2nd line as shown in Fig. 4.61.

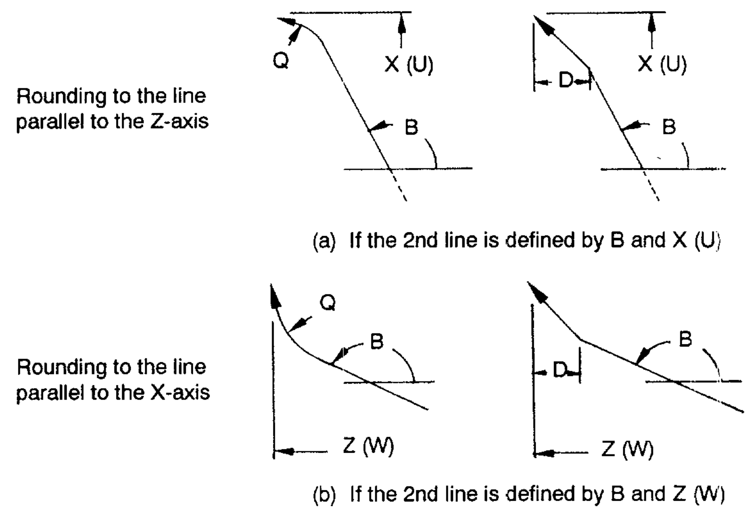


Fig. 4.61 2nd Chamfering/Rounding

(c) Direction of the 2nd chamfering/rounding

The direction of the 2nd chamfering/rounding is the same direction as the 2nd line advancing direction. For details, see Tables 4.11 and 4.12.

Table 4.11 Direction of 2nd Chamfering

Move Angle of the 2nd Line B Command Value	Chamfering Direction +X +Z	Other Conditions
B = 0, –360.000, 360.000	Chamfering in the X+, Z+ direction	The 1st line moves in the positive (+) direction of the X-axis.
	Chamfering in the X–, Z+ direction	The 1st line moves in the negative (–) direction of the X-axis.
0 < B < 90.000 –360.000 < B < –270.000	Chamfering in the X+, Z+ direction	–

Move Angle of the 2nd Line B Command Value	Chamfering Direction +X +Z	Other Conditions
B = 90.000, –270.000	Chamfering in the X+, Z+ direction	The 1st line moves in the positive (+) direction of the Z-axis.
	Chamfering in the X+, Z– direction	The 1st line moves in the negative (–) direction of the Z-axis.
90.000 < B < 180.000 –270.000 < B < –180.000	Chamfering in the X+, Z– direction	–
B = 180.000, –180.000	Chamfering in the X+, Z– direction	The 1st line moves in the positive (+) direction of the X-axis.
180.000 < B < 270.000 –180.000 < B < –90.000	Chamfering in X–, Z– direction	The 1st line moves in the negative (–) direction of the Z-axis.
	Chamfering in the X+, Z+ direction	–
	Chamfering in the X–, Z– direction	The 1st line moves in the negative (–) direction of the Z-axis.
B = 270.000, –90.000	Chamfering in the X–, Z+ direction	The 1st line moves in the positive (+) direction of the Z-axis.
270.000 < B < 360.000 –90.000 < B < 0	Chamfering in the X–, Z+ direction	–

Table 4.12 Direction of 2nd Rounding

Move Angle of the 2nd Line B Command Value	Chamfering Direction +X +Z	Other Conditions	
B = 0, –360.000, 360.000	Rounding in the X–, Z+ direction	The 1st line moves in the negative (–) direction of the X-axis.	X (U) command cannot be used.
	Rounding in the X+, Z+ direction	The 1st line moves in the positive (+) direction of the X-axis.	
0 < B < 90.000 –360.000 < B < –270.000	Rounding in the X+, Z+ direction Rounding to the line parallel to the X-axis	The 2nd line is defined by B and Z (W).	
	Rounding to the line parallel to the Z-axis	The 2nd line is defined by B and X (U).	
B = 90.000, –270.000	Rounding in the X+, Z+ direction	The 1st line moves in the positive (+) direction of the Z-axis.	Z (W) command cannot be used.
	Rounding in the X+, Z– direction	The 1st line moves in the negative (–) direction of the Z-axis.	
90.000 < B < 180.000 –270.000 < B < –180.000	Rounding in the X+, Z– direction Rounding to the line parallel to the Z-axis	The 2nd line is defined by B, X (U).	
	Rounding to the line parallel to the X-axis.	The 2nd line is defined by B, Z (W).	
B = 180.000, –180.000	Rounding in the X+, Z– direction	The 1st line moves in the positive (+) direction of the X-axis.	X (U) command cannot be used.
	Rounding to the X–, Z– direction	The 1st line moves in the negative (–) direction of the X-axis.	
180.000 < B < 270.000 –180.000 < B < –90.000	Rounding to the X–, Z– direction Rounding to the line parallel to the X-axis	The 2nd line is defined by B, Z (W).	
	Rounding to the line parallel to the Z-axis	The 2nd line is defined by B, X (U).	
B = 270.000, –90.000	Rounding in the X–, Z– direction	The 1st line moves in the negative (–) direction of the Z-axis.	Z (W) command cannot be used.
	Rounding to the X–, Z+ direction	The 1st line moves in the positive (+) direction of the Z-axis.	

Move Angle of the 2nd Line B Command Value	Chamfering Direction +X +Z	Other Conditions
270.000 < B < 360.000 −90.000 < B < 0	Rounding in the X–, Z+ direction Rounding to the line parallel to the Z-axis	The 2nd line is defined by B, X (U).
	Rounding to the line parallel to the X-axis	The end line is defined by B, Z (W).

(d) Supplements to shape definition

- If all of B, X (U), and Z (W) are specified to define the 2nd line, the 1st line can be defined by specifying only one of A, I, and K.
- For the multiple chamfering/rounding on both ends of taper function, the 1st and 2nd lines are defined by selecting appropriate addresses from X, Z, I, K, A, and B. Note that omission of an address and designation of "0" for it have different meaning. Therefore, differing from other G codes, designation of "0" for an address cannot be omitted.

Table 4.13 Omission of Address with Value "0"

Address	Omission of Address with Value "0"
X Z I K A B	Omission is not allowed.
P Q C D	Omission is allowed. (chamfer size and rounding radius are "0".)

- If X (U) and Z (W) are used to define the 2nd line, 2nd chamfering/rounding is not allowed. If 2nd chamfering/rounding is specified although X (U) and Z (W) are used to define the 2nd line, an error occurs.
- Chamfering and rounding can be combined as needed such as 1st chamfering and 2nd rounding, and 1st rounding and 2nd chamfering.
- If all of A, I, and K are used to define the 1st line, address A is disregarded and the 1st line is defined by I and K.
- If all of B, X (U) and Z (W) are used to define the 2nd line and two of A, I, and K are used for the definition of the 1st line, B is disregarded for the definition of the 2nd line, and the 2nd line is defined by X (U) and Z (W).
- The direction of rotation is defined in reference to the positive (+) direction of the Z-axis; positive value for the counterclockwise direction and negative value for clockwise rotation. (Programmable range: $-360.000 \leqq A, B \leqq 360.000$)

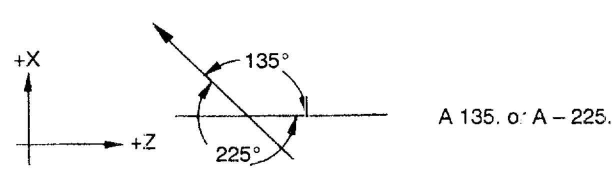


Fig. 4.62 Designation of Move Angle A and B of Line

(3) Examples of Programming

Example 1

(G01 W · · · ;) → Commands for φ50 portion (broken lines) in the illustration below

G111 W–100. I15. A90. B165. C3. D5. ;
or G111 W–100. I15. K0. B165. C3. D5. ;
] Commands for the portion indicated by solid lines in the illustration below

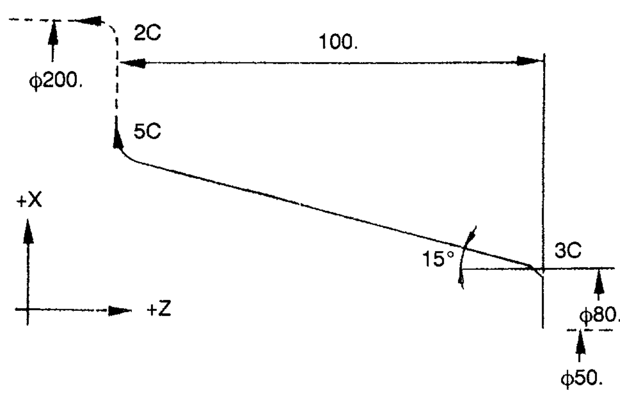


Fig. 4.63 Multiple Chamfering on Both Ends of Taper

Example 2

·
·
·

(G01 W · · · ;)

G111 W–100. I15. A90. B165. P3. Q5. ;
or G111 W--100. I15. K0. B165. P3. Q5. ;
] Commands for the portion indicated by solid lines in the illustration below

(G12 X200. K–2. ;) ← Commands for broken line portion after solid line portion: 2R

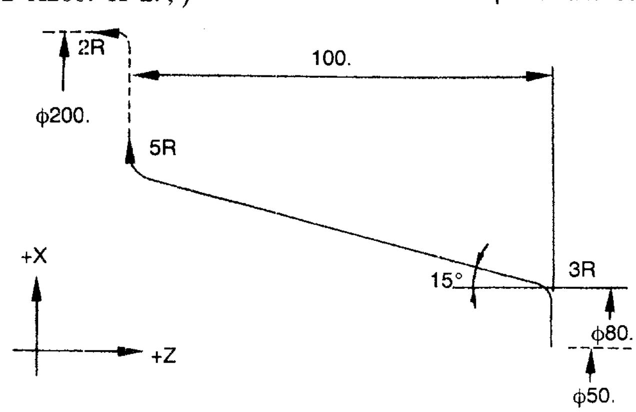


Fig. 4.64 Continuous Rounding on Both Ends of Taper

(4) Supplements to Multiple Chamfering/Rounding on Both Ends of Taper

- It is not allowed to specify addresses M, S, and T in the block containing G111.
- If the 1st chamfering specified by address C in the G111 block has the shape as shown in Figs. 4.65 and 4.66, such designation is not possible.

  The end point is that of the 2nd line assuming that chamfering/rounding is not made.

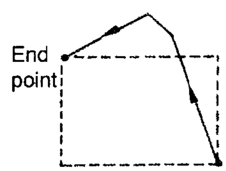


Fig. 4.65 Outside the Rectangle Defined by the Start and End Points

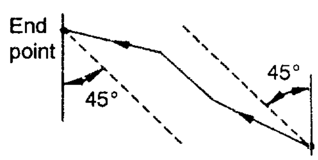


Fig. 4.66 Inside the Range between 45° Line from Start Point to End Point and 45° Line from End Point to Start Point

- The NC processes all operations to define the 1st and 2nd lines and the 1st and 2nd chamfering/rounding when it reads the G111 block into buffer. It could take more than 500 msec depending on the shape to be defined. If the time required for the axes to execute the commands in the preceding block is shorter than the time required for the necessary operation, axis movement will be suspended to impair the surface being machined. To prevent such suspension of axis movement due to long operation time, place the NC in the buffering mode (M93 mode) several blocks before the G111 block.
- If the G111 block is executed with the single block function ON, the movements to the end point of the G111 block are divided into a maximum of 4 blocks.

- If the 1st line is defined as the "45° line from the start point to the end point" or the 2nd line is defined as the "45° line from the end point to the start point", such designation could or could not cause an error depending on the shape being defined.

Example 1 Program that does not Cause an Error

G111 Z–100. I–25. A225.
B180. C5. D5. ;

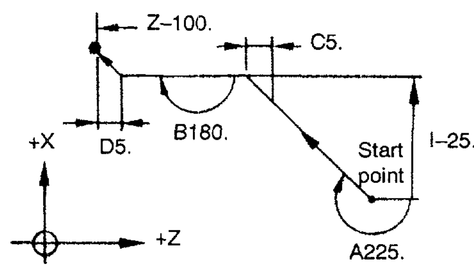


Fig. 4.67 Example of Program that does not Cause an Error

Example 2 Program that Causes an Error

G111 Z–100. I–25. A225.
B270. C5. D5. ;

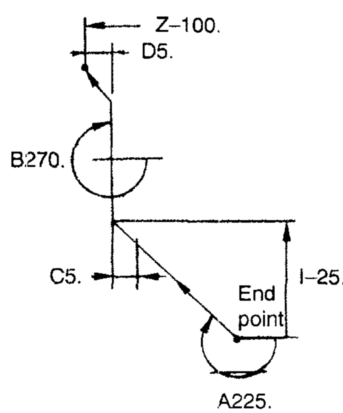
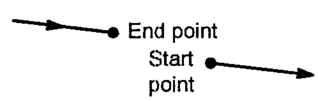
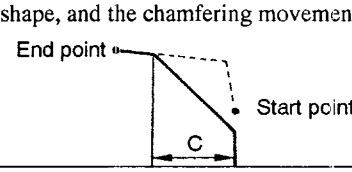
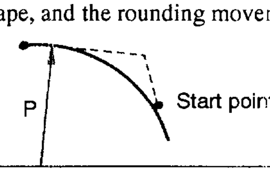


Fig. 4.68 Example of Program that Causes an Error

- Alarms Caused by Incorrect G111 Command Designation

Table 4.14 Table of Alarms Caused by Incorrect G111 Command Designation

Alarm Code	Description
0281	• For the definition of the 2nd line, only one address is specified among addresses B, X (U), and Z (W). • For the definition of the 2nd line, two addresses are specified among addresses B, X (U), and Z (W) while only one or none of addresses A, I, and K is specified to define the 1st line. • Both address C (1st chamfering) and address P (1st rounding) are specified. • Both address D (2nd chamfering) and address Q (2nd rounding) are specified. • The 2nd line is defined using addresses X and Z with the 2nd chamfering / rounding defined using addresses Q and D.
0282	The value specified for addresses A and B (line move angle) is outside the range of $-360.000 \leqq A, B \leqq 360.000$.
0283	• The 1st chamfering portion is outside the rectangle defined by the start and end points. • The 1st chamfering portion is inside the area between the 45° line from the start to the end points and the 45° line from the end to the start points.
0284	• There is no point of intersection between the 1st and 2nd lines. 
	• The 1st and 2nd lines are on the same line. • Since addresses A, I, and K are specified in the following manner to define the 1st line, the shape cannot be defined When value A is –360.000, –180.000, 0, 180.000, or 360.000: Address I is specified for the definition of the 1st line. When value A is –270.000, –90.000, 0, 90.000, or 270.000: Address K is specified for the definition of the 1st line. • Since addresses B, X (U), and Z (W) are specified in the following manner to define the 1st and the 2nd line, the shape cannot be defined. When value B is –360.000, –180.000, 0, 180.000, or 360.000: Address X (U) is specified for the definition of the 2nd line. When value B is –270.000, –90.000, 0, 90.000, or 270.000: Address Z (W) is specified for the definition of the 2nd line.
	• The value specified for addresses C and D (chamfer size) is too large in comparison to the specified shape, and the chamfering movement is not possible. 
	• The value specified for addresses P and Q (rounding radius) is too large in comparison to the specified shape, and the rounding movement is not possible. 
0285	• M, S, and/or T command is specified in the G111 block.

4.1.4 Multiple Chamfering/Rounding on Arc Ends (G112) *

The following four movements can be specified by the commands in a single block if G112 function is used: line → chamfering/rounding → arc → chamfering/rounding. Depending on the direction of arc, two kinds of chamfering/rounding on arc ends are possible – on the periphery and on the face.

G112 is a non-modal G code and valid only in the specified block.

(1) Programming Format

(a) Continuous rounding on arc ends (on periphery)

Example of Programming

G112 X (U) · · · I · · · K · · · P · · · Q · · · R · · · ;

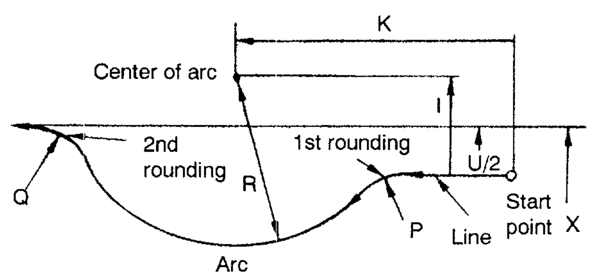


Fig. 4.69 Continuous Rounding on Arc Ends (on Periphery)

(b) Multiple chamfering on arc ends (on periphery)

Example of Programming

G112 X (U) · · · I · · · K · · · C · · · D · · · R · · · ;

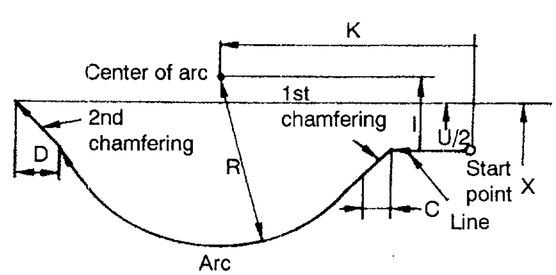


Fig. 4.70 Multiple Chamfering on Arc Ends (on Periphery)

(c) Continuous rounding on arc ends (on face)

Example of Programming

G112 Z (W) · · · I · · · K · · · P · · · Q · · · R · · · ;

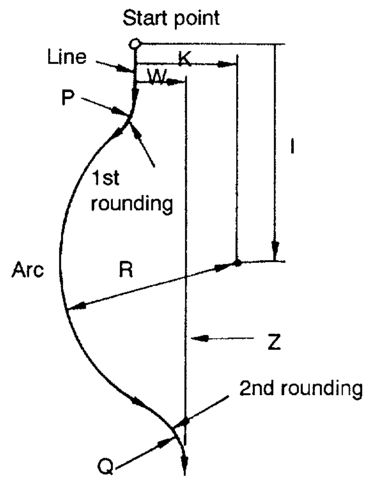


Fig. 4.71 Continuous Rounding on Arc Ends (on Face)

(d) Multiple chamfering on arc ends (on face)

Example of Programming

G112 Z (W) · · · I · · · K · · · C · · · D · · · R · · · ;

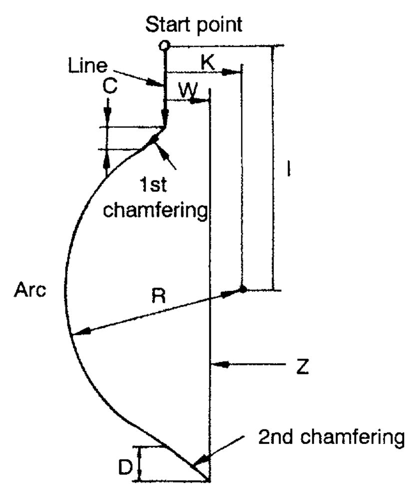


Fig. 4.72 Multiple Chamfering on Arc Ends (on Face)

(e) Addresses

The addresses indicated in Table 4.15 are used when designating multiple chamfering/rounding on arc ends.

Table 4.15 Addresses and Meaning

Address	Meaning	Input Increment Unit
X (U)	X-coordinate of end point in multiple chamfering/rounding on arc ends (on periphery) (U: Incremental amount from the start point)	1 = 0.001 mm or 1 = 0.0001 inch (decimal point input allowed)
Z (W)	Z-coordinate of end point in multiple chamfering/rounding on arc ends (on face) (W: Incremental amount from the start point)	
I	Distance along the X-axis from the center of arc or start point	
K	Distance along the Z-axis from the center of arc or start point	
R	Radius of arc	
P	1st rounding radius (unsigned)	
Q	2nd rounding radius (unsigned)	
C	1st chamfer size (unsigned)	
D	2nd chamfer size (unsigned)	

4

(2) Defining the Shape

(a) Shapes in multiple chamfering/rounding on arc ends

How the shapes appearing in the multiple chamfering/rounding on arc ends function is defined is indicated in Table 4.16.

Table 4.16 Description of Shapes

Shape	Description
Line	The line extending from the start point, and parallel to the Z-axis (arc on periphery) or the X-axis (arc on face)
Arc	The arc which has the center defined by I and K in reference to the start point.
1st chamfering	Chamfering executed at the corner made between the line and the arc; the size of chamfer is specified by C.
1st rounding	Rounding executed at the corner made between the line and the arc; the radius of rounding is specified by P.
2nd chamfering	Chamfering executed at the corner made between the line defined by X (U), which is parallel to the Z-axis (arc on periphery), or the one defined by Z (W), which is parallel to the X-axis (arc on face), and the arc; the size of chamfering is specified by D.
2nd rounding	Rounding executed at the corner made between the line defined by X (U), which is parallel to the Z-axis (arc on periphery), or the one defined by Z (W), which is parallel to the X-axis (arc on face), and the arc in the manner that the rounding arc is tangent to both elements; the radius of rounding is specified by Q.

4

(b) Arc cutting direction

The rotating direction for arc cutting is determined so that the arc lies at the opposite side to the center of arc in reference to the line drawn from the start point as shown in Fig. 4.73.

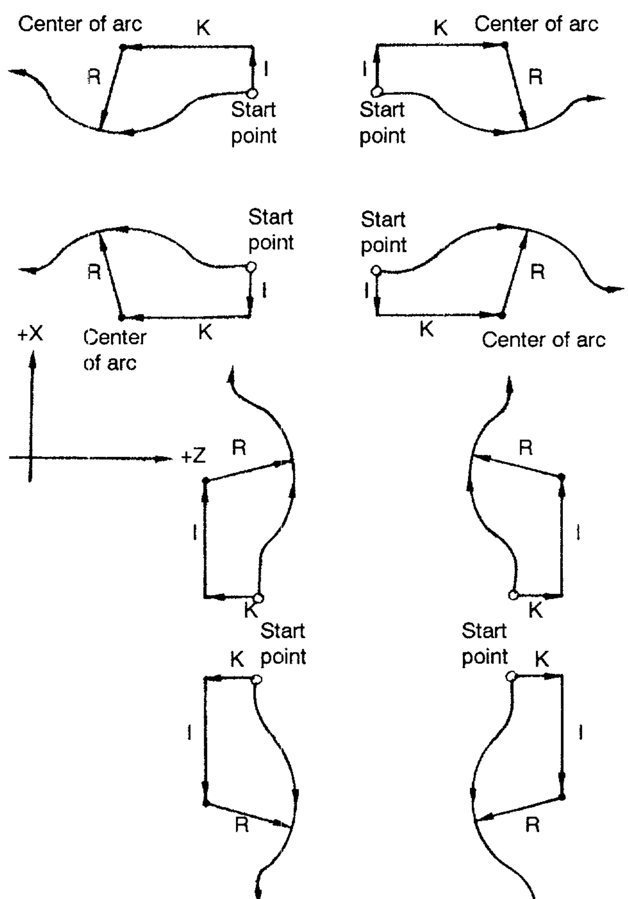


Fig. 4.73 Arc Cutting Direction

How the direction of rotation in arc cutting is determined in the NC is indicated in Table 4.17.

Table 4.17 Commands and Arc Cutting Directions

I and K Command Values	Arc Cutting Direction	
	Arc on periphery	Arc on face
$I \geqq 0, K \geqq 0$	Counterclockwise: CCW (same as G03)	Clockwise: CW (same as G02)
$I \geqq 0, K < 0$	Clockwise: CW (same as G02)	Counterclockwise: CCW (same as G03)
$I < 0, K \geqq 0$		
$I < 0, K < 0$	Counterclockwise: CCW (same as G03)	Clockwise: CW (same as G02)

The arc cutting direction indicated above can be reversed by specifying a negative value for arc radius R as shown in Fig. 4.74.

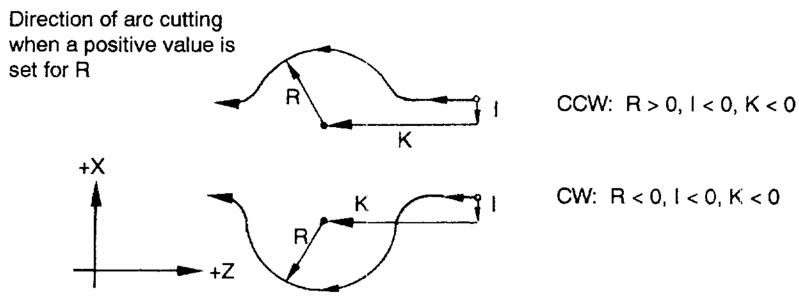


Fig. 4.74 Arc Cutting Direction with a Negative R Value

(c) Omission of addresses

Addresses X (U) and Z (W) are used to determine the arc location (on periphery or on face). Therefore, they cannot be omitted even if the start and end points are on the same position; specify "U0" or "W0". Concerning other addresses, how their omission is treated is indicated in Table 4.18.

Table 4.18 Omission of Addresses

Address	Processing at Omission
I	Equivalent of "I0"
K	Equivalent to "K0"
R	Equivalent to "R0" and causes an alarm.
P Q C D	Equivalent to "0" designation and chamfering/rounding is not executed.

(3) Examples of Programming

Example 1

(G01 X100. Z–50. ;) ← Line in broken line (preceding the arc)
G112 U0 I10. K–50. P5. Q5. R30. ;
(G01 Z–150. ;) ← Line in broken line (succeeding the arc)

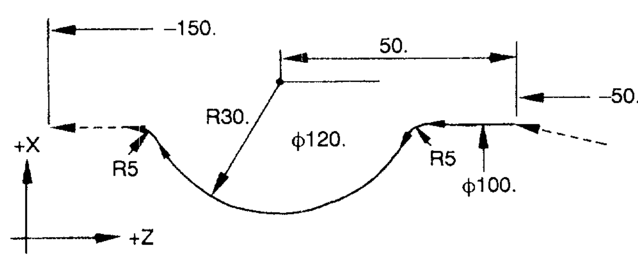


Fig. 4.75 Continuous Rounding on Arc Ends

Example 2

(G01 X100. Z–50. ;) ← Line in broken line (preceding the arc)
G112 U0 I10. K–50. C5. D5. R30. ;
(G01 Z–150. ;) ← Line in broken line (succeeding the arc)

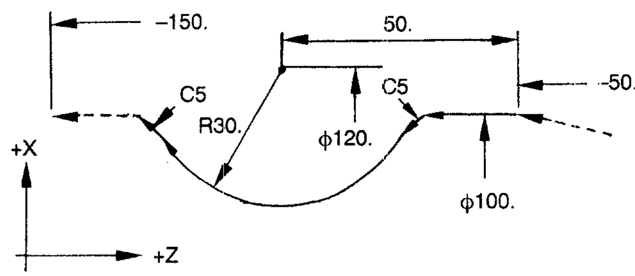


Fig. 4.76 Multiple Chamfering on Arc Ends

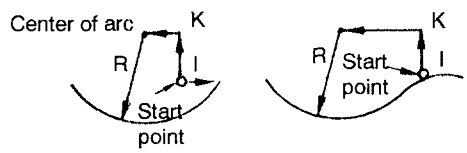
(4) Supplements to Multiple Chamfering/Rounding on Arc Ends Function

- It is not allowed to specify addresses M, S, and T in the block containing G112.
- No other G codes may be specified in the G112 block. An error occurs if a G code is specified with G112 in the same block.
- The NC processes all operations to define the 1st and 2nd lines and the 1st and 2nd chamfering/rounding when it reads the G112 block into buffer. It could take more than 500 msec depending on the shape to be defined. If the time required for the axes to execute the commands in the preceding block is shorter than the time required for the necessary operation, axis movement will be suspended to impair the surface being machined. To prevent such suspension of axis movement due to long operation time, place the NC in the buffering mode (M93 mode) several blocks before the G112 block.
- If the G112 block is executed with the single block function ON, the movements to the end point of the G112 block are divided into a maximum of 4 blocks.
- If G112 is specified in the finishing shape defining block for the multiple repetitive cycles G71 (OD stock removal cycle), G72 (face rough turning cycle), and G73 (pattern repeating cycle), the G112 block is equivalent to five blocks.

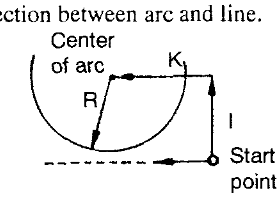
- Alarms Caused by Incorrect G112 Command Designation

Table 4.19 Table of Alarms Caused by Incorrect G112 Command Designation

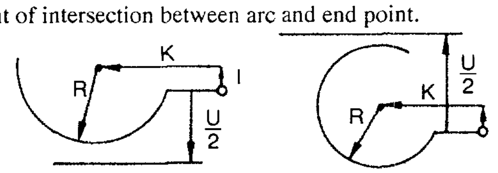
<table>
<tr><th>Alarm Code</th><th>Description</th></tr>
<tr><td>0285</td><td>M, S, and/or T code is specified in the G112 block.</td></tr>
<tr><td>0286</td><td>• X (U) or Z (W) is not specified.
• Both of X (U) and Z (W) are specified.
• R is not specified, or R0 is specified.
• I and K are not specified, or “0” is specified for both of I and K.
• Both P and C are specified.
• Both Q and D are specified.</td></tr>
<tr><td>0287</td><td>• Movement in the direction opposite to the direction from the start point to the center of arc



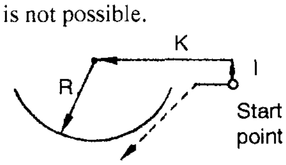
• There is no point of intersection between arc and line.



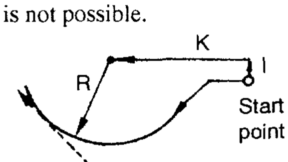
• There is no point of intersection between arc and end point.



• Chamfering specified by C is not possible.



• Chamfering specified by D is not possible.



</td></tr>
</table>

4

4.1.5 Hole-machining Canned Cycles (G80 to G89, G831, G841, G861) *

Hole-machining canned cycles (G80 to G89, G831, G841, G861) can define specific movements for machining holes that usually require several blocks of commands by single-block commands. Fourteen kinds of canned cycles are provided and G80 cancels the called out canned cycle program.

(1) G Codes Calling Canned Cycles and Axis Movement Patterns of Canned Cycles

G codes that call out a canned cycle and the axis movement pattern of the called canned cycle are indicated in Table 4.20.

Table 4.20 Hole-machining Canned Cycles

G Code	Axis Feed	Processing at Hole Bottom	Retraction	Applications
G80	–	–	–	Cancel
G81	Cutting feed	–	Rapid traverse	Drilling
G82	Cutting feed	Dwell	Rapid traverse	Spot facing
G83	Intermittent feed	–	Rapid traverse	Deep hole drilling
G831	Intermittent feed	–	Rapid traverse	High-speed deep hole drilling
G84	Cutting feed	Spindle reverse rotation after dwell	Cutting feed → Dwell → Spindle forward rotation	Tapping
G841	Cutting feed	Spindle forward rotation after dwell	Cutting feed → Dwell → Spindle reverse rotation	Reverse tapping
G85	Cutting feed	–	Cutting feed	Boring
G86	Cutting feed	Spindle stop	Rapid traverse → Spindle forward rotation	Boring
G861	Cutting feed	Spindle indexing → Shift	Rapid traverse → Shift → Spindle forward rotation	Boring
G87	Spindle indexing → Shift → Rapid traverse → Shift → Spindle forward rotation → Cutting feed	Spindle indexing → Shift	Rapid traverse → Shift → Spindle forward rotation	Back boring

G Code	Axis Feed	Processing at Hole Bottom	Retraction	Applications
G88	Cutting feed	Spindle forward rotation after dwell	Manual return → Spindle forward rotation	Boring
G89	Cutting feed	Dwell	Cutting feed	Boring

(2) Programming Format

```
G17 G··· X(Z)··· C··· Z(X)··· R±··· P···
(G19)    (U)(W)   (H)  (W)(U)
```

- G17 (G19): Designation of a plane (determination hole machining axis)
- G: G code calling up a canned cycle
- X(Z)··· C··· (U)(W) (H): Designation of hole machining point coordinate values
- Z(X)··· (W)(U): Designation of hole-machining axis coordinate value
- R±···: Designation of R-point level coordinate value
- P···: Length of dwell at hole bottom

```
Q··· L··· F··· ;
```

- Q···: G83, G831: Cut-in depth
  G861: Shift distance
  G84, G841: Dwell time at R-point level
- L···: Number of repetitions
- F···: Feedrate

The following four steps are executed as one cycle with the commands indicated above.

- Positioning at the hole machining position
- Rapid traverse to R-point level
- Hole machining up to the bottom
- Return to R-point or initial point level

TERM?

◆ Initial Point Level

The initial point level is the absolute position of the point where the hole machining axis is located when the NC mode enters the canned cycle mode from the canned cycle cancel state. The initial point level is not changed if a canned cycle is executed in the G199 (R-point level return) mode.

(a) Addresses

- Positioning axes : The hole machining position is specified by either incremental or absolute values. The positioning axes are those included in the selected plane.
- Hole machining axis : The position of the hole bottom is specified by either an absolute value or an incremental value referenced to the R-point level. Axis move from the R-point level to the hole bottom is controlled in the G01 mode using the feedrate specified by an F code. With some types of canned cycles, G00 operation is included (intermittent feed, for example). Return motion from the bottom of the hole to the R-point level is controlled in the G00 or G01 mode according to the type of the canned cycle. The hole machining axis is an axis not included in the selected plane as shown in Table 4.21.

Table 4.21 Plane Selection G Codes and Hole Machining Axes

G Code	Selected Plane (Positioning Plane)	Hole Machining Axis
G17	XY	Z
G18	ZX	Y
G19	YZ	X

Note 1: Generally, the G18 plane is selected for normal machining carried out in a two-axis NC lathe. Before starting a hole machining canned cycle, G17 or G19 must always be specified, and when canceling the hole machining cycle the plane must be returned to the XZ plane by specifying G18.

2: The C-axis can be used as the positioning axis disregarding of the plane designation (G17 to G19).

3: In a hole-machining canned cycle, if the selection of the plane and the designation of the hole machining axis do not agree with each other while parameter setting is "pm4017 D6 = 1", an alarm occurs.

- R (hole machining feed start level) : The position of the R-point level is specified by either an absolute or incremental value. The feed axis is the hole machining axis. Return operation from the R-point level to the initial point level is controlled in the G00 mode. With the standard G code, the R-point level is always specified by an absolute value. If the X-axis is taken as the hole machining axis, the unit system of the hole machining axis is the same as selected for the X-axis (diametric value when pm1000 D1 = 0, and radial value when pm1000 D1 = 1).
- L (number of repetitions) : The number of repetitions is specified by address L. If designation of address L is omitted, "L1" is assumed. If "L = 0" is specified, only positioning at (X, Z) is executed.
- P (dwell time) : The length of dwell at the bottom of the hole is specified in units of 1 msec. Designation of "P1.0" executes dwell for 1 second. If address P is not designated, dwell is not executed.
- Q (depth of cut, shift amount) : Address Q is used to specify depth of cut for G83 and G831 cycles and shift amount for G861 and G87 cycles. An unsigned incremental value is used; to specify X-axis component, a radial value is used.

(3) Designation of Return Mode

In the execution of a hole-machining canned cycle, the return mode after the completion of a cycle differs depending on which of the following G codes is specified.

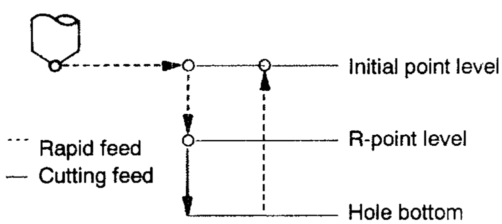
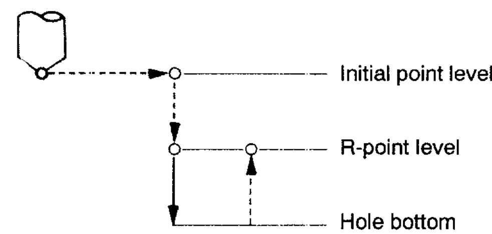
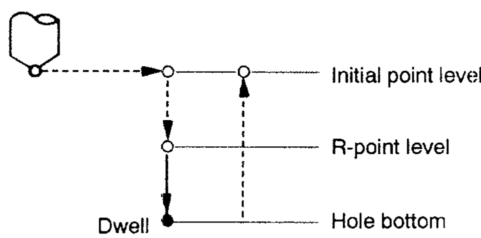
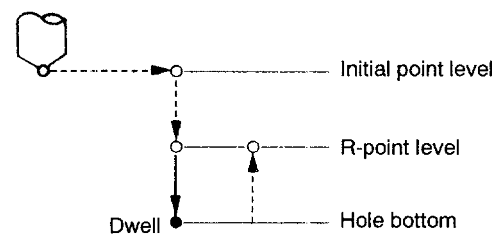
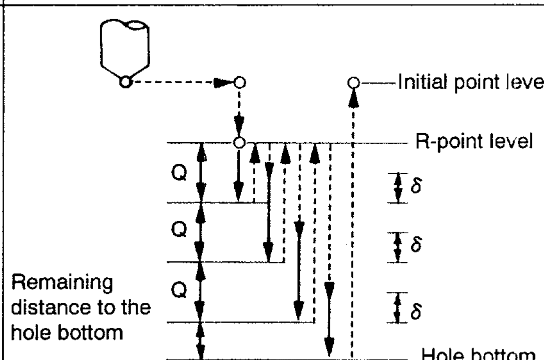
G198	Returns to the initial point level.
G199	Returns to the R-point level.

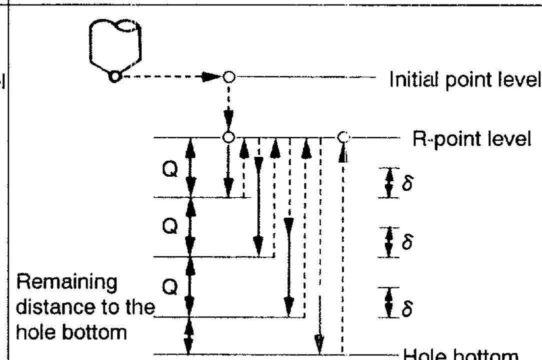
These G codes are modal.

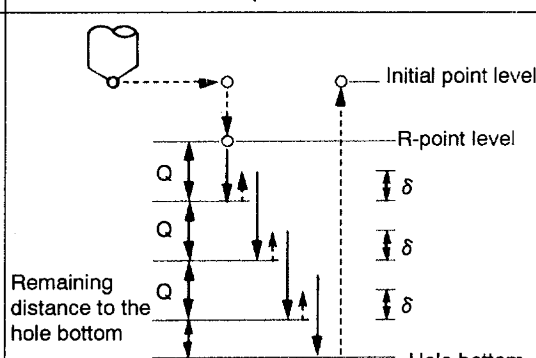
(4) Table of Operation

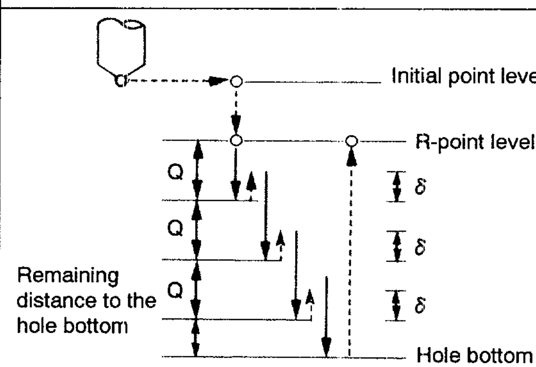
Table 4.22 Table of Normal Hole-machining Canned Cycles

● Dwell
○ Single-block stop

<table>
<tr><th></th><th>G198 (Initial Point Level Return) Mode</th><th>G199 (R-point Level Return) Mode</th></tr>
<tr><td>G81
Drilling</td><td>
</td><td>
</td></tr>
<tr><td>G82
Drilling
Spot facing</td><td>
</td><td>
</td></tr>
<tr><td>G83
Deep hole drilling</td><td>

δ: Setting G17: pm 0870
G18: pm 0871
G19: pm 0872</td><td>

δ: Setting G17: pm 0870
G18: pm 0871
G19: pm 0872</td></tr>
<tr><td>G831
High-speed deep hole drilling</td><td>

δ: Setting G17: pm 0870
G18: pm 0871
G19: pm 0872</td><td>

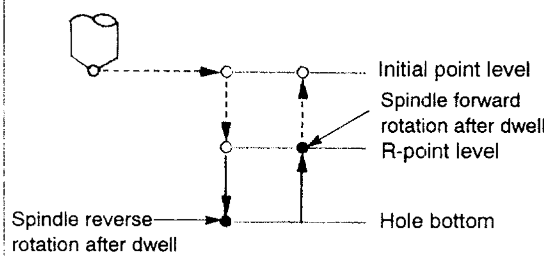
δ: Setting G17: pm 0870
G18: pm 0871
G19: pm 0872</td></tr>
</table>

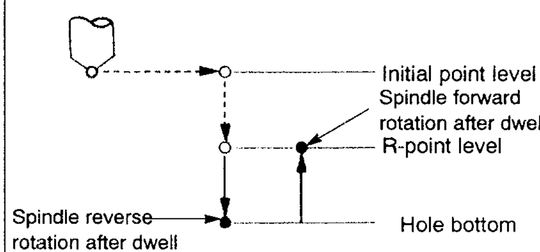
4

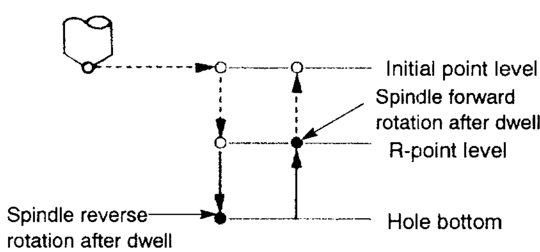
4

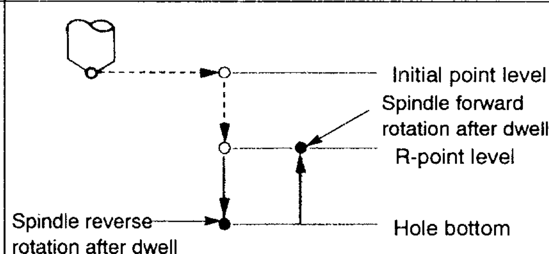
Table 4.22 Table of Normal Hole-machining Canned Cycles (cont'd)

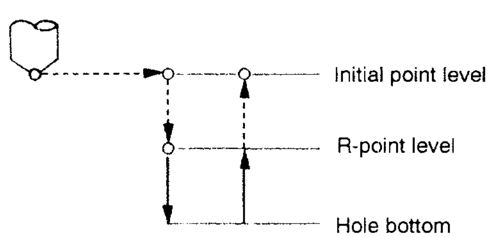
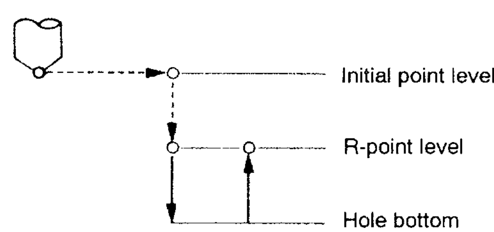
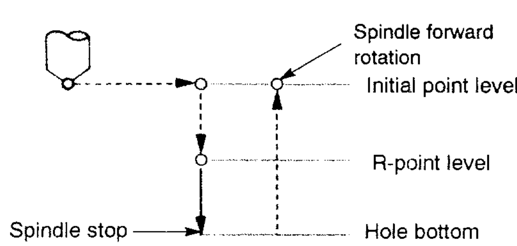
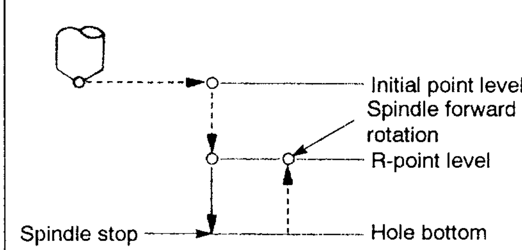
● Dwell
○ Single-block stop

<table>
<tr><th></th><th>G198 (Initial Point Level Return) Mode</th><th>G199 (R-point Level Return) Mode</th></tr>
<tr><td>G84
Tapping</td><td>

P : Dwell time at hole bottom
Q : Dwell time at R-point level</td><td>

P : Dwell time at hole bottom
Q : Dwell time at R-point level</td></tr>
<tr><td>G841
Reverse tapping</td><td>

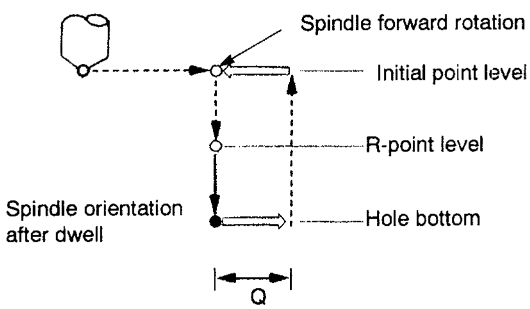
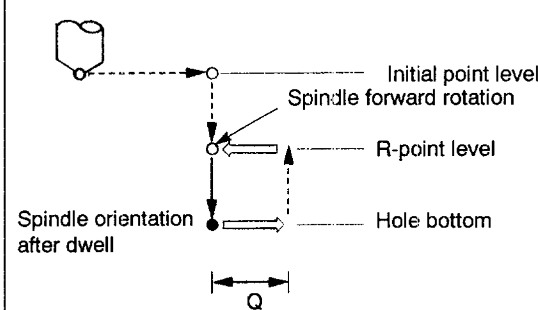
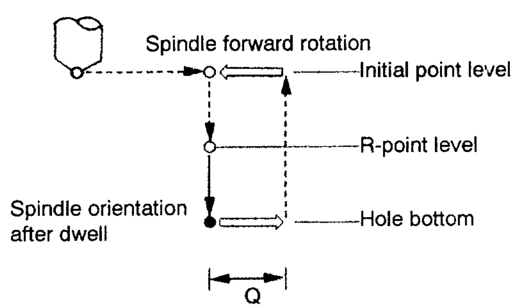
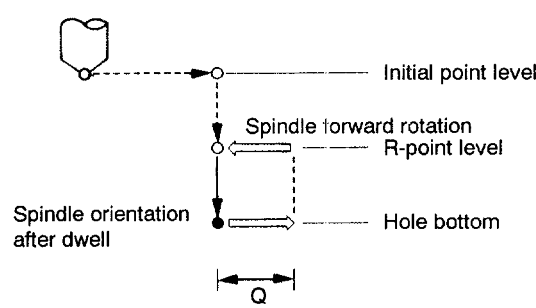
P : Dwell time at hole bottom
Q : Dwell time at R-point level</td><td>

P : Dwell time at hole bottom
Q : Dwell time at R-point level</td></tr>
<tr><td>G85
Boring</td><td>
</td><td>
</td></tr>
<tr><td>G86
Boring</td><td>
</td><td>
</td></tr>
</table>

Note: For the spindle control, refer to (5) "Spindle Control in Hole-machining Canned Cycles".

Table 4.22 Table of Normal Hole-machining Canned Cycles (cont'd)

● Dwell
○ Single-block stop

	G198 (Initial Point Level Return) Mode	G199 (R-point Level Return) Mode
G861 (constant shift) Boring	 Q: Shift distance (unsigned incremental value) Shifting feedrate: pm2864 Shifting direction: pm4028 Dwell time: P command	 Q: Shift distance (unsigned incremental value) Shifting feedrate: pm2864 Shifting direction: pm4028 Dwell time: P command
G861 (variable shift) Boring	 Q : Shift distance (specified by i, j, and k) G17: By i and j G18: By k and i G19: By j and k i : X-axis incremental value (signed) (radial value) j : Y-axis incremental value (signed) k : Z-axis incremental value (signed) Shifting feedrate: pm2864 Dwell time: P command	 Q : Shift distance (specified by i, j, and k) G17: By i and j G18: By k and i G19: By j and k i : X-axis incremental value (signed) (radial value) j : Y-axis incremental value (signed) k : Z-axis incremental value (signed) Shifting feedrate: pm2864 Dwell time: P command

4

Table 4.22 Table of Normal Hole-machining Canned Cycles (cont'd)

● Dwell
○ Single-block stop

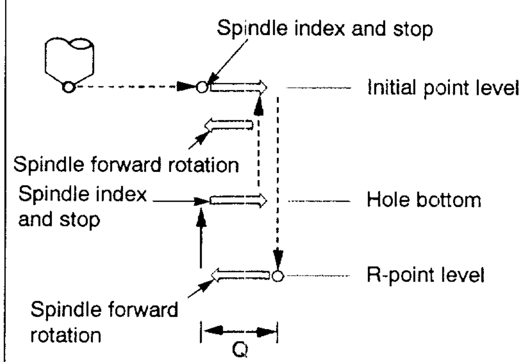
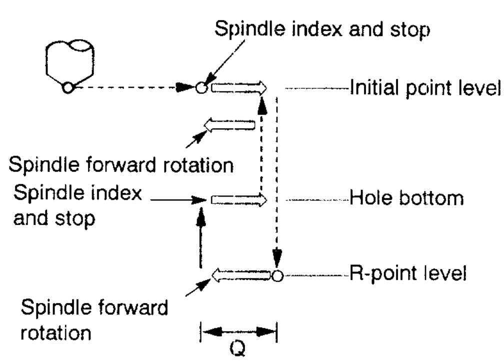
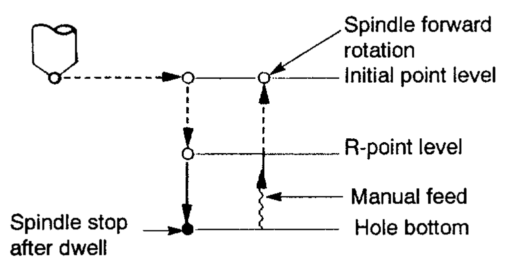
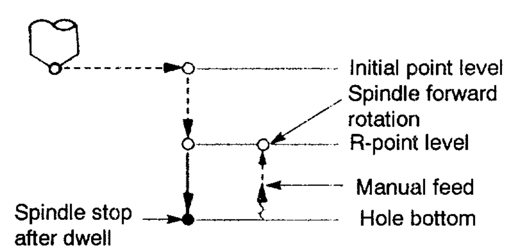
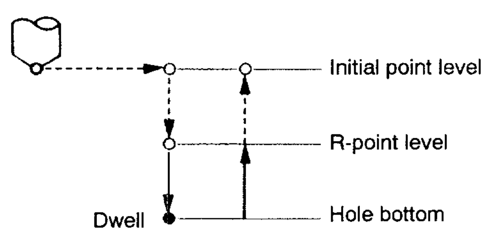
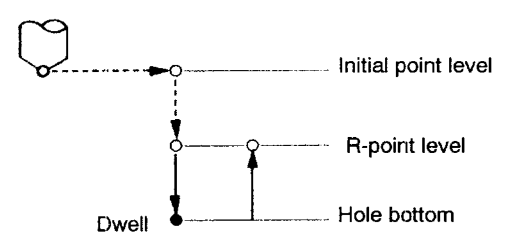
	G198 (Initial Point Level Return) Mode	G199 (R-point Level Return) Mode
G87 (constant shift) Back boring	 Q : Shifting distance (unsigned incremental value) Shifting feedrate: pm2864 Shifting direction: pm4028	Not used
G87 (variable shift) Back boring	 Q : Shift distance (specified by i, j, and k) G17: By i and j G18: By k and i G19: By j and k i : X-axis incremental value (signed) (radial value) j : Y-axis incremental value (signed) k : Z-axis incremental value (signed) Shifting feedrate: pm2864	Not used

Table 4.22 Table of Normal Hole-machining Canned Cycles (cont'd)

● Dwell
○ Single-block stop

	G198 (Initial Point Level Return) Mode	G199 (R-point Level Return) Mode
G88 Boring		
G89 Boring		

4

(5) Spindle Control in Hole-machining Canned Cycles

(a) Tapping cycle (G84) and Reverse tapping cycle (G841)

Table 4.23 Spindle Control in Tapping and Reverse Tapping Cycles

	G84	G841
Bottom of hole	(Spindle stop) ↓ Spindle reverse rotation	(Spindle stop) ↓ Spindle forward rotation
Retraction	(Spindle stop) ↓ Spindle forward rotation	(Spindle stop) ↓ Spindle reverse rotation

For the control of the spindle, a value is set for the parameters indicated in Table 4.24 and the set value is output as an M code.

Table 4.24 Spindle Control Parameters for Tapping and Reverse Tapping Cycles

	Parameter	Default M Code (No Parameter Setting)
Spindle start (forward)	pm 4430	M03
Spindle start (reverse)	pm 4431	M04
Spindle stop	pm 4432	M05

When changing the spindle rotating direction from forward to reverse or from reverse to forward, whether the spindle is stopped once or the direction of rotation is changed directly without stopping the spindle is selected by the setting for the following parameter.

pm4016 D3 = 0	Spindle stop M code is not output.
pm4016 D3 = 1	Spindle stop M code is output.

(b) Boring cycle (G86)

Table 4.25 Spindle Control in Boring Cycle

	G86
Bottom of hole	Spindle stop
Retraction	Spindle forward rotation

Note: The parameters used to output the spindle control M codes are the same as indicated in Table 4.24.

(c) Boring cycle (G861) and back boring cycle (G87)

Table 4.26 Spindle Control in Boring and Back Boring Cycles

	G861	G87
Initial point level	–	Spindle index and stop
R-point level	–	Spindle forward rotation
Bottom of hole	Spindle index and stop	Spindle index and stop
Retraction	Spindle forward rotation	Spindle forward rotation

Note: The M code for spindle forward rotation is the same as indicated in Table 4.24.

For the parameter used to set the spindle index and stop M code, the following parameter is used.

Table 4.27 Spindle Index and Stop Parameter

	Parameter	Default M Code (No Parameter Setting)
Spindle index and stop	pm 4445	M19

(6) C-axis Clamp/Unclamp

It is possible to clamp the C-axis during hole machining. By setting "1" for parameter pm4017 D4 (parameter pm4017 D4 = 1), the C-axis clamp/unclamp M codes which are set for the parameters indicated below are output at the positions shown in Fig. 4.77. It is also possible to execute dwell after clamping the C-axis by the setting for the setting parameter indicated below.

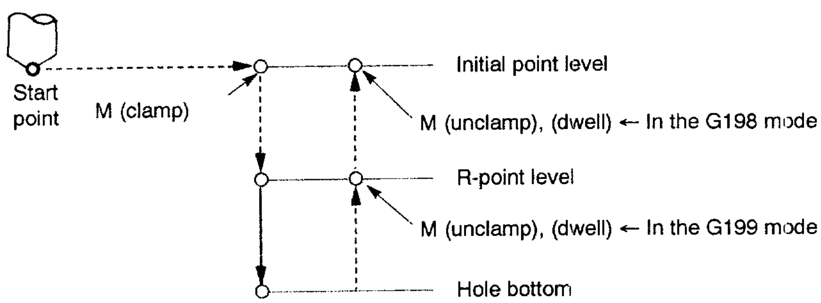


Fig. 4.77 C-axis Clamp/Unclamp

Table 4.28 Parameters Used for C-axis Clamp/Clamp Setting

	Parameter	Default (No Parameter Setting)
Clamp M code	pm 4439	No default value is set. Always set these parameters.
Unclamp M code	pm 4440	
Dwell time after clamp	pm 0400	
C-axis is clamped/not clamped	pm 4017 D4 0: C-axis is not clamped 1: C-axis is clamped	

(7) Shift in Boring (G861) and Back Boring (G87) Cycles

(a) Direction of shift in G861/G87 (when Q command is used)

Specify the shift amount with a Q command and set the direction of shift for parameter pm4028.

Table 4.29 Direction of Shift and Parameters

Plane Selection	Hole Machining Axis	Shift Direction Setting Parameter	Description		
			D1	D0	
			0	0	+X
G17	Z-axis	pm4028 D1, D0	0	1	–X
			1	0	+Y
			1	1	–Y
			D3	D2	
			0	0	+Z
G18	Y-axis	pm4028 D3, D2	0	1	–Z
			1	0	+X
			1	1	–X
			D5	D4	
			0	0	+Y
G19	X-axis	pm4028 D5, D4	0	1	–Y
			1	0	+Z
			1	1	–Z

(b) Direction of shift for G861/G87 cycle (designation by I, J, and K)

It is possible to specify the shift in linear interpolation by using I, J, and K. The shift amount is specified in the following manner according to the plane selected for the operation.

G17 (XY plane) : Specify with I and J.
G18 (ZX plane) : Specify with K and I.
G19 (YZ plane) : Specify with J and K.

I : X-axis incremental value (signed) (radius value)
J : Y-axis incremental value (signed)
K : Z-axis incremental value (signed)

The shift speed is set for parameter pm2864 in either case (a) or (b).

$$\text{(G17)}\ X \cdots\ Z \cdots\ R \cdots\ \underbrace{I \cdots}_{\text{Shift amount}}$$

(8) Supplements to Hole-machining Canned Cycles

- G codes that call up a hole-machining canned cycle (G81 to G89, G831, G841, and G861) are modal and once specified, the specified G code remains valid until another G code in the same G code group, a G code in 01 group, or G80 is specified.
- The hole machining data are modal while a hole-machining canned cycle mode remains valid. It is possible to call up a new hole-machining canned cycle while in the hole-machining canned cycle mode previously set up. If address data to be used for the execution of a newly called hole-machining canned cycle are omitted, the modal data specified in the previous blocks are used.
- When the program is written using incremental commands, the bottom of the hole to be machined is defined by the distance referenced from the R-point level. If the R-point level is changed during the execution of a hole-machining canned cycle, the bottom of the hole is defined in reference to the new R-point level. Therefore, to prevent an error, always specify both the R-point level and the bottom of the hole.
- The L command that specifies how many times the hole-machining canned cycle should be repeated is non-modal. However, there is a case that the specified L command is saved temporarily as indicated below. Note that the L command remains valid until it is actually executed.

  Example of Programming
  G81 U10. R–20. Z–30. F100 ;

L3 ;	The hole-machining canned cycle is not executed since none of X (U), Z (W), C (H), Y (V), and R are specified.
X20. ;	The G81 cycle is executed three times as specified by "L3" which has been saved. After the completion of the cycle, "L3" is cleared.

- Before starting a hole-machining canned cycle, the spindle must have been started in the automatic operation by executing M03 or M04. Never start a hole-machining canned cycle after starting the spindle manually.
- Before entering the hole-machining canned cycle mode, define the R-point level (hole bottom) newly by specifying the bottom of the hole. Note that the R-point level data are cleared when the hole-machining canned cycle mode is canceled.

4

- To execute a hole-machining canned cycle after changing the address data, the block in which the new address data are specified must include any of the following addresses: X (U), Z (W), C (H), Y (V), and R. The hole-machining canned cycle is not executed unless any of these addresses is specified.
- If M, S, and/or T code is specified in the block where a hole-machining canned cycle commands are specified, the specified codes are output at the first positioning operation. They are also output in the first positioning operation if "L" is specified. Therefore, these codes must be specified independently.
- If following G codes are specified in the hole-machining canned cycle mode, alarm "0170" occurs. The hole machining canned cycle mode must be canceled before specifying these G codes.
  - · G codes in * group, excluding G04
  - · G codes (G41, G42) that call up the nose R offset mode.
- In the hole-machining canned cycle mode, it is possible to call a subprogram by specifying the subprogram call command (M98). The hole-machining canned cycle can be continuously executed in the called subprogram. In this case, although the P command (dwell time) for the hole-machining canned cycle is temporarily destroyed by the P command (jump destination program number) specified with M98, the previous P command value is automatically recovered after the jump to the specified subprogram.

  The restrictions on M98, such as the maximum four nesting levels and the command input function from punched tape, are the same as applied to M98 in other than the hole-machining canned cycle mode.

  An alarm occurs if the hole-machining canned cycle G code and M98 are specified in the same block.
- The hole-machining canned cycle mode is canceled when G80 or a G code in 01 group is specified.

  If a G code in 01 group is specified with a G code that calls up a hole-machining canned cycle in the same block, alarm "0170" occurs. Note that if G80 is specified with a G code in 01 group in the same block, an alarm does not occur but the specified commands are executed normally.

- If a hole-machining canned cycle is executed with the SINGLE BLOCK switch set ON, the operation is suspended at the timing indicated below and the FEED HOLD lamp on the machine operation panel lights. The single block stop at the completion of a hole-machining canned cycle is the same as the single block stop in other than a hole-machining canned cycle; the FEED HOLD lamp does not go on.
  - After the completion of positioning at the specified point
  - After the completion of positioning at the R-point level
  - After the completion of one cycle if an L command is specified
- In the hole-machining canned cycle mode, it is possible to insert the dwell (G04) command block, that contains only dwell command. In this case, the specified dwell is executed normally.
- The F command specified as a hole-machining canned cycle command remains valid even after the cancellation of the hole-machining canned cycle.
- If address search operation is attempted by suspending (block stop) a hole-machining canned cycle, it causes an alarm. Address search during block stop, which is specified in a program, is allowed. If the called hole-machining canned cycle is executed more than one time for one block of commands in a program (L command designation), address search at the completion of each cycle causes an alarm.

4.2 PROGRAM SUPPORT FUNCTIONS (2)

4.2.1 Solid Tap Function (G84, G841) *

The solid tapping function executes tapping by synchronizing the feed of the hole machining axis with the rotation of the rotary tool spindle. If tapping is executed by using this function, a floating chuck is not necessary any more and, at the same time, accurate tapping is made possible at a high speed.

(1) Commands Used for Solid Tap Cycle

To execute solid tapping, change the mode to the solid tap mode and then specify the solid tap cycle.

(a) Solid tap mode commands

4

The following G codes are provided to determine which of the tapping mode is called up, solid tap or conventional tapping cycle. These G codes are modal. When the power is turned ON or the NC is reset, G94 mode is set.

- Solid tap mode command (G93)

  Once G93 is executed, the tapping cycle commands (G84/G74) are executed in the solid tap mode. In this mode, Z-axis feed is controlled in the "feed per revolution" mode. In the solid tap mode, no machining other than solid tapping is allowed.

- Solid tap mode cancel command (G98, G99)

  The solid tap mode is canceled and the conventional tapping mode is called up. Once G94 is executed, tapping cycles are executed in the conventional mode, in which the Z-axis feed is controlled in the "feed per minute" mode.

(b) Programming for the solid tap cycle

After executing G93, solid tapping is enabled by specifying the commands indicated below.

- Programming for tapping cycle

```
G84 X (U) ± · · C (H) ± · · Z (W) ± · ·  R ± · ·  (P · ·) F (E) · ·  (S · ·) (L · ·) ;
```

G84 — G code calling up a tapping cycle

X (U) ± · · C (H) ± · · Z (W) ± · · — Designation of hole position and hole bottom level (U, H, and W: incremental command)

R ± · · — R-point level coordinate value (absolute command)

(P · ·) — Dwell time at the hole bottom

F (E) · · — Tap pitch (mm/rev)

(S · ·) — Spindle speed

(L · ·) — Number of repetitions

- Programming for reverse tapping cycle

```
G841 X (U) ± · ·  C (H) ± · ·  Z (W) ± · ·  R ± · ·  (P · · ) F (E) · ·  (S · ·) (L · ·);
```

G841 — G code calling up reverse tapping cycle

(2) Part Program Using Solid Tap Commands

(a) M** command

This command selects the gear range used for solid tap cycle. If no such M code is specified, A gear is selected.

(b) G93 command

When the G93 command is executed, the spindle stops and the solid tap mode is established with the position loop set for the control of the spindle.

It is also possible to execute spindle indexing to position the spindle at a fixed position before establishing the solid tap mode after the spindle has been stopped. To execute spindle indexing, change the setting for parameter (pm1053 D2 = 1). Note that spindle indexing to the fixed position is possible only when the spindle and the spindle PG rotate at a 1 : 1 ratio.

The solid tap mode is canceled by G84 (or G841).

(c) G84 (G841) command

How the tapping cycle is executed in the solid tap mode is described below.

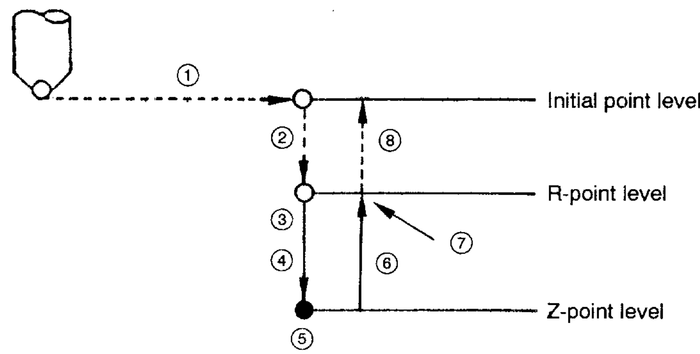


Fig. 4.78 Tapping Cycle Executed in the G84 (G841) Mode

4

① Positioning at the position specified by X and Z

② Positioning at the R-point level

③ Spindle rotating in the forward direction, and cutting to the hole bottom (Z-point level) at the specified feedrate (spindle rotating in the reverse direction if G841 is specified).
(At the start of this block, the number of servo lag pulses is checked for the spindle and the Z-axis whether it is within the allowable number: error detect ON)

④ Spindle stop (the number of servo lag pulses is checked for the spindle.)

⑤ Dwell (if P command is specified.)

⑥ Spindle rotating in the reverse direction, and cutting to the R-point level at the specified feedrate

⑦ Spindle stop

⑧ Positioning at the initial point level (in the G198 mode)

(3) Example of Programming and Description of Operation

Example of Programming

N1 G17 ; ....①
N2 G93 ; ....②
N3 G84 (or G841) X100. C10. Z-20. R-10. F1 S3000 ; ......③
N4 G20. ; ....④
N5 X150. ; ....⑤
N6 G80. ; ....⑥
N7 G99 ; ....⑦

Operation

① Hole machining axis

② Solid tap mode ON (spindle positioning mode ON)

③ Tapping in the solid tap mode at (100., 10.) in 1 mm pitches.

④ Tapping in the solid tap mode at (100., 20.) in 1 mm pitches.

⑤ Tapping in the solid tap mode at (150., 20.) in 1 mm pitches.

⑥ Canceling the canned cycle

⑦ Solid tap mode OFF, mm/rev mode designation

4

(4) Relationship between the Solid Tap and Other Operation

(a) Dry run

Whether the solid tap cycle or the conventional tapping should be executed when G93 is executed with the DRY RUN switch ON can be selected by the setting for a parameter.

pm4016 D6 = 0	Conventional tapping cycle
pm4016 D6 = 1	Solid tap cycle

If a tapping cycle is called up with the setting of "pm4016 D6 = 0", the cycle is executed in the following manner.

If G93 is executed with the DRY RUN switch set ON, G93 is invalid and the G84/G841 command called up in the solid tap mode is processed as the G84/G841 command called up in the conventional tapping mode. The feedrate for this operation is determined by the setting of the JOG switch. Once the solid tap mode is entered with the DRY RUN switch ON, the G84/G841 command is processed as if the command were called up in the conventional tapping mode even if the DRY RUN switch setting is changed from ON to OFF during the execution of the cycle. Therefore, the spindle does not start even when G93 is executed.

For the execution of the G93 block, whether the DRY RUN switch is ON or not is judged when the G93 code is read. Usually, this state is determined while the commands in the preceding block are executed. Therefore, when G93 is operated with DRY RUN ON for program check, etc., be sure that the DRY RUN switch is ON from the beginning and do not change it before completion.

(b) MST function lock

If G93 is executed with the MST FUNCTION LOCK switch set ON, G93 is invalid and the G84/G841 cycle is executed as the conventional tapping cycle. Note that feedrate command is executed in the feed per revolution mode and thus the spindle position is not controlled.

Whether the solid tap cycle is executed with the MST function lock state or not is judged when the G93 code is read. Therefore, if the G93 program should be checked in the MST function lock state, the MST LOCK switch must be set ON from the beginning of operation so that it will not be turned ON during the operation.

(c) Machine lock

If tapping cycle is executed in the solid tap mode with the MACHINE LOCK switch set ON, although the spindle rotates, Z-axis does not move but only the position data are updated.

(d) Feedrate override and spindle override

During tapping cycle in the solid tap mode (G84 or G74), feed override is fixed at 100%. Note that override for rapid traverse is valid. Concerning the spindle override, the value set for parameter pm4017 D2 is used; the parameter setting is read when the solid tap commands are read and the spindle override is clamped at this value during solid tapping.

(e) Feed hold

During tapping cycle in the solid tap mode, feed hold is invalid. If the FEED HOLD button on the machine operation panel is pressed during tapping in the solid tap mode, tapping is executed up to the point-R level and stops there.

(f) Mode change

During tapping cycle in the solid tap mode, mode change is invalid.

(g) Program re-start

If the program is restarted from a block in the solid tap mode, G93 is not executed. Therefore, to restart a program from a block in the solid tap mode, it is necessary to enter and execute G93 in the MDI mode.

(5) Supplements to the Solid Tap Function

- In the G93 block, only S, F, and N codes can be specified. If other commands are specified in the G93 block, alarm "0250" occurs.
- In the G93 mode, an S code is processed as the S command for solid tap operation.
- In the G93 mode, only G codes indicated below can be specified. If a G code not indicated below is specified, alarm "0250" occurs. Concerning G01, although it can be specified in the G93 mode, axis move commands cannot be specified.

  G codes that can be specified in the G93 mode
  G00, G01, G04, G70, G71, G72, G74, G80, G84, G90, G91, G98, G99
- Spindle indexing called in the G93 mode is the indexing to the fixed position in reference to the zero point pulse (C-axis) that is output from the spindle PG.
- To specify G98/G99 after the completion of solid tap, make sure to cancel the canned cycle by specifying G80.
- When G98/G99 is specified after the completion of solid tap, an F command value is reset to "0". Make sure to specify an F command when a program including cutting feed is specified after the designation of G98/G99.
- Solid tapping is executed by the combination of the control of the spindle and the hole machining axis. Two combinations of the spindle and the hole machining axis can be set using parameters (pm1240 to pm1243). If the solid tap command is specified for the axis not set for the parameters, alarm "0161 occurs.
- F values are displayed in the order they are specified in a program. To change the display, it is necessary to specify the F value again.
- For the parameter where the spindle speed that corresponds to the 10 V command is set in the gear range used for solid tap, make sure to set a value larger than the value set for the parameter where the allowable maximum spindle speed for solid tap is set.

  Example: pm1415 = 3200, pm1416 = 3000

(6) Functions Related to Solid Tap

If the options related to the solid tap function are selected, the following functions are added or modified.

(a) Display of synchronization error in solid tapping

- While in the solid tap mode, the servo lag error display screen displays the following data.

  X-axis ... Spindle servo lag error

  Z-axis .... Synchronization error between the spindle and the hole machining axis

  C-axis .... Hole machining servo lag error

  Note that the names of axes displayed on the screen vary depending on the machine configuration.

- If "pm4015 D6 = 1", it is possible to display the peak value of the number of synchronization error pulses for the spindle and the hole machining axis. (X-axis: Positive peak value; C-axis: Negative peak value)

  Note that the names of axes displayed on the screen vary depending on the machine configuration.

4

(b) Error detect in the solid tap mode

By setting "1" for parameter pm4015 D5, it is possible to set the error detect OFF mode for rapid traverse (X- , Z-axis positioning) during solid tap.

By this setting, cycle time can be reduced. In this case, the program must be made carefully since the program advances to the Z-axis block immediately after the completion of pulse distribution for the positioning of the X-axis.

(c) High-speed return speed in solid tap

- By setting a numeral "n" for parameter pm1252, the cutting speed is controlled during solid tap so that the feedrate in the return motion will be "n" times ($0.1 \leqq n \leqq 25.5$) the feedrate applied in the cutting motion.

pm1252	Programmable range:	0 to 255
	Setting:	1 = 0.1 times (If "pm1252 = 0", setting of "1" is equivalent to 1 time.)

- For spindle speed, the value of "S command value × multiplication ratio" is clamped at the maximum speed for solid tap.

(d) Alarm code

Table 4.30 List of Alarm

Alarm Code	Description	Causes
0250	A command not allowed in the solid tap mode	• In the G93 block, a command other than G93, S, F, and N is specified. • G93 is specified in other than the G00 or G01 mode established by the 01-group G code. • A G code not allowed in the G93 mode is specified.
2191	Solid tap input/ output error	• When solid tap operation is executed, the position control loop is not established for the spindle. • When spindle indexing is executed, SLPC is turned OFF before the completion of indexing.

(7) Parameters Related to Solid Tap Function

Table 4.31 List of Parameters

Item	Parameter	Setting and Setting Range
Spindle PG mounting position	pm1039 : 1st spindle pm1040 : 2nd spindle	D1, D2 = 1, 0 : Spindle side = 0, 1 : Motor side
Spindle feedback multiplication ratio setting	pm1053 : 1st spindle pm1054 : 2nd spindle	D1, D0 = 0, 0 : ×1 = 0, 1 : ×2 = 1, 0 : ×4 = 1, 0 : ×8
Spindle encoder type	pm1091 : 1st spindle pm1092 : 2nd spindle	Setting 21 1024 pulses INC (for spindle)
Spindle index check timer	pm1220 : 1st spindle pm1221 : 2nd spindle	Setting 1 = 4 ms Range 1 - 255
Spindle speed for checking spindle stop after indexing	pm1225 : 1st spindle pm1226 : 2nd spindle	Setting 1 r/min Range 1 - 255
Solid tap execution axis	pm1240 : Spindle number for solid tap (1) pm1241 : Servo axis number for solid tap (1) pm1242 : Spindle number for solid tap (2) pm1243 : Servo axis number for solid tap (2)	Spindle 1, 2 Servo axis 1 - 5
Solid tap return speed multiplication ratio setting	pm1252 : Solid tap (1) pm1253 : Solid tap (2)	Setting 1 = 0.1 time (0 = 1 time) Range 0 - 255

Table 4.13 List of Parameters (cont'd)

Item	Parameter	Setting and Setting Range
Spindle speed for command voltage of 10 V with the gear used for solid tap	pm1415 : 1st spindle pm1435 : 2nd spindle	Setting 1 = 1r/min Range 1 - 32767
Maximum spindle speed for solid tap	pm1416 : 1st spindle pm1436 : 2nd spindle	Setting 1 = 1r/min Range 1 - 32767
Spindle position loop gain for solid tap	pm1417 : 1st spindle pm1437 : 2nd spindle	Setting 1 = 0.01 (1 / s) Range 1 - 32767
In-position width for solid tap servo axis to be accelerated to the target point	pm1500 : Solid tap (1) pm1501 : Solid tap (2)	Setting 1 = 1 pulse Range 1 - 32767
Solid tap synchronization compensation parameter (k1)	pm1502 : Solid tap (1) pm1504 : Solid tap (2)	Range –32767 - 32727
Solid tap synchronization compensation parameter (k2)	pm1503 : Solid tap (1) pm1505 : Solid tap (2)	Range –32767 - 32727
Spindle gear ratio for solid tapping	pm1510 : 1st spindle gear A: Number of teeth at the spindle side pm1511 : 1st spindle gear A: Number of teeth of intermediate gear (spindle) pm1512 : 1st spindle gear A: Number of teeth of intermediate gear (motor) pm1513 : 1st spindle gear A: Number of teeth at the motor side pm1514 : 1st spindle gear B: Number of teeth at the spindle side pm1515 : 1st spindle gear B: Number of teeth of intermediate gear (spindle) pm1516 : 1st spindle gear B: Number of teeth of intermediate gear (motor) pm1517 : 1st spindle gear B: Number of teeth at the motor side pm1518 to pm1525 : For 2nd spindle	Setting 1 = 1 tooth Range 0 - 32768
Linear pattern spindle acceleration/deceleration time constant for solid tapping	pm2471 : 1st spindle pm2472 : 2nd spindle	Setting 1 = 1 ms Range 0 - 32767
ON/OFF of G00 error detect during solid tapping	pm4015 D5 : 0 = Error detect ON : 1 = Error detect OFF	Setting 0, 1
Display of synchronization error peak value between spindle and hole machining axis during sold tapping	pm4015 D6 : 0 = Peak value is not displayed. : 1 = Peak value is displayed.	Setting 0, 1
Spindle override clamp in tapping cycle	pm4017 D2 : 0 = Fixed at 100% : 1 = Fixed at the value read first	Setting 0, 1

4.2.2 Programmable Data Input (G10) *

By using the commands of "G10 P··· X(U)··· Z(W)··· R··· C ;", it is possible to write and update the tool offset amount using a part program. If an address is omitted in the designation of data input block, the offset amounts for the omitted addresses remain unchanged.

Table 4.32 Description of Addresses

Address	Description
P	Specifies the tool offset number.
X Z	Updates the present offset amount to the specified value.
U W	Adds the specified value to the present offset amount.
R	Updates the nose R offset amount to the specified value.
C	Tool nose control point data

Example of Programming

G10 P16 X32.5 W0.05 ;

W0.05: Adds 0.05 mm to the offset amount of Z-axis.

X32.5: Updates the present offset amount of X-axis to 32.5 mm.

P16: Declares that the following data are reflected to tool offset number "16".

(1) Tape Format

Punch the tape in the format indicated above. Offset data of different offset numbers can be stored to the offset memory at a time.

Label
% ;
G10 P··· X··· Z··· R··· ;
G10 P··· X··· Z··· R··· ;
G10 P··· X··· Z··· R··· ;
%

(2) Setting the Workpiece Coordinate System Shift Data

With the commands of "G10 P00 X (U) · · · Z (W) · · · C (H) · · · ;", it is possible to write and update the workpiece coordinate system shift data using a part program. If an address is omitted in the designation of data input block, the offset amounts for the omitted addresses remain unchanged.

X, Z, C	:	Absolute setting data of the workpiece coordinate system shift amount
U, W, H	:	Incremental setting data of the workpiece coordinate system shift amount

4.2.3 Subprogram Call Up Function (M98, M99)

This function can be used when subprograms are stored in the part program memory. Subprograms registered to the memory with program numbers assigned can be called up and executed as many times as required.

The created subprograms should be stored in the part program memory before they are called up.

```
O ········· ;      ←—— Program number
············ ;
············ ;
············ ;
M99 ;              ←—— End of subprogram
```

4

(1) Commands

The M codes indicated in Table 4.33 are used.

Table 4.33 Subprogram Call M Code

M Code	Function
M98	Subprogram call up
M99	End of subprogram

(a) Subprogram call (M98)

- By specifying "M98P · · · L · · · ; ", the subprogram of the program number that is specified by P is called up and executed by the number specified by L. If L is omitted, the subprogram is executed once. If the specified program number is not found, alarm "0390" occurs.
- Nesting of subprograms is possible - the allowable nesting level is four. If the nesting level exceeds this limit, an alarm occurs.
- By specifying "M98P · · · L · · · Q · · · ; " the subprogram specified by the P command is executed from the block specified by the Q command. A sequence number must be specified by a numeral of up to five digits.

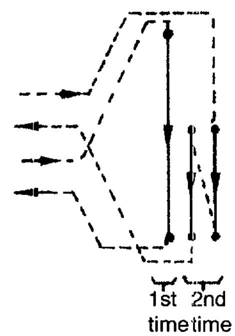
(b) End of Subprogram Code (M99)

At the end of a subprogram, M99 must be specified in a block without other commands.. Upon execution of M99, the program automatically returns to the block in the main program next to the one where the subprogram has been called up. Fig. 4.79 shows how a subprogram called up from the main program is executed.

Example of Programming

Main Program

```
O100 ;
N1 G00 ··· ;
N2 M98 P200 Q3 L2 ;
N3 ··· ;
N4 M98 P200 ;
N5 ··· ;
 ·
 ·
 ·
```



Subprogram

```
O200;
N1 ··· ;
N2 ··· ;
N3 ··· ;
 ·
 ·
 ·
N20 M99 ;
```

Fig. 4.79 Execution of A Subprogram

- By specifying "M99P · · · · · ; ", the program returns to the block specified by the P command in the main program instead of the block next to the one where the subprogram has been called up.
- If M99 is specified in a main program, the program returns to the beginning of that main program and the program is repeatedly executed.

4.2.4 Stored Stroke Limit B (G36 to G39)

The stored stroke limit function checks whether the present position of axes operated manually or automatically enters the stored stroke limit (entry prohibited area) which is set by G36 to G39. The No. 2 to No. 5 entry prohibited area is called the stored stroke limit B. If an axis has entered the stroke end limit, operation is stopped and alarm occurs.

(1) Programming Format

- G36 U··· W··· I··· K··· (P···) ;
  - U··· W···: Coordinate value of point C
  - I··· K···: Coordinate value of point D
  - (P···): By setting 3, 4, and 5, No. 3 to No. 5 entry prohibited areas can also be set.

  With the commands indicated above, the function checks the entry of axes into the entry prohibited area No. 2.

- The commands of "G37 (P···);" cancel the function to check entry into the prohibited area No. 2.

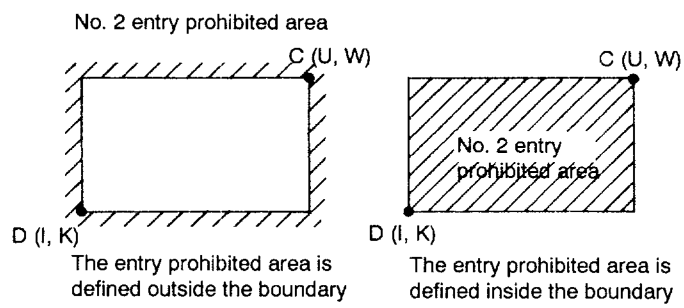


Fig. 4.80 Stored Stroke Limit B

• G38 U··· W··· I··· K··· ;

- U··· W··· — Coordinate value of point E
- I··· K··· — Coordinate value of point F

With the commands indicated above, the function checks the entry of axes into the entry prohibited area No. 3.

- The command of "G39;" cancels the function to check entry into the prohibited area No. 3.

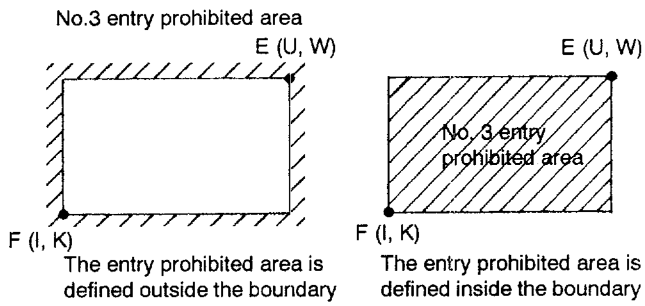


Fig. 4.81 Stored Stroke Limit B

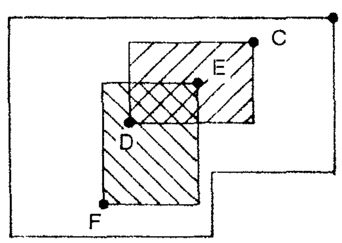
(a) Setting the boundary and area

The parameter numbers used for area designation of stored stroke limit B are indicated in Table 4.34.

Table 4.34 Parameter Numbers Used for Area Designation

Boundary	Check Axis No.		
	No. 1	No. 2	No. 3
No. 2 prohibited area (+): C point	pm0831	pm0832	pm0833
No. 2 prohibited area (–): D point	pm0834	pm0835	pm0836
No. 3 prohibited area (+): E point	pm0837	pm0838	pm0839
No. 3 prohibited area (–): F point	pm0840	pm0841	pm0842
No. 4 prohibited area (+)	pm0843	pm0844	pm0845
No. 4 prohibited area (–)	pm0846	pm0847	pm0848
No. 5 prohibited area (+)	pm0849	pm0850	pm0851
No. 5 prohibited area (–)	pm0852	pm0853	pm0854

It is possible to define the entry prohibited areas so that they overlap partially with each other by setting the corresponding parameter.



Note: Points D and E set the positive (+) side boundary and those D and F the negative (–) side boundary in the machine coordinate system.

Fig. 4.82 Overlapping of Entry Prohibited Areas

(b) Designation of check axes

The axes for which stored stroke limit B (No. 2 to No. 5 entry prohibited areas) is checked is designated by using parameters (maximum of three axes).

Table 4.35 Stored Stroke Limit Check Axis Numbers for No. 2 to No. 5 Entry Prohibit Areas

Check Area	Stroke Limit Check Axis No. (Note)		
	No. 1	No. 2	No. 3
No. 2 entry prohibited area	pm6111	pm6112	pm6113
No. 3 entry prohibited area	pm6114	pm6115	pm6116
No. 4 entry prohibited area	pm6117	pm6118	pm6119
No. 5 entry prohibited area	pm6120	pm6121	pm6122

Note: Setting: 1 = X-axis, 2 = Z-axis

(c) Parameters used for setting the entry prohibited area outside/inside of the boundary

Table 4.36 indicates the parameter numbers used for setting the entry prohibited area outside or inside the boundary.

Table 4.36 Outside/Inside Designation of Entry Prohibited Area (No. 2 to No. 5)

Check Area	Parameter No.	Description
No. 2 entry prohibited area	pm0008 D4	0: Inside the specified area 1: Outside the specified area
No. 3 entry prohibited area	pm0008 D5	
No. 4 entry prohibited area	pm0008 D6	
No. 5 entry prohibited area	pm0008 D7	

(d) Turning ON/OFF the Stored Stroke Limit Check

Whether or not the entry to No. 2 to No. 5 entry prohibited area should be checked (stored stroke limit B) can be designated by the setting of setting parameters.

Table 4.37 Turning ON/OFF the Stored Stroke Limit B

Check Area	Parameter No.	Description
No. 2 entry prohibited area	pm0008 D0	0: Invalid (OFF) 1: Valid (ON)
No. 3 entry prohibited area	pm0008 D1	
No. 4 entry prohibited area	pm0008 D2	
No. 5 entry prohibited area	pm0008 D3	

When a G code (G36 to G39) is specified, the setting for these setting parameters is automatically changed. Therefore, the ON or OFF state that is specified last by either the G code or the setting for the setting parameters becomes valid. Concerning the No. 1 entry prohibited area, the check is always ON.

- For the coordinate values of the points that define the boundaries, always use the absolute value in the machine coordinate system. That is the distance [1 = least output (movement) increment] from the first reference point should be written. Therefore, the stored stoke limit check function is not made valid unless the manual or automatic reference point return is executed after turning the power ON.
- Upon completion of the first manual or automatic reference point return after power ON, the stored stroke limit check function becomes valid to check the entry of axes into the entry prohibited areas. Therefore, if the reference point is located in the entry prohibited area, it causes stored stroke limit error immediately. If this occurs, turn the stored stroke limit check function OFF and correct the set data.

4

(2) Supplements to the Stored Stroke Limit B Function

- If a cutting tool enters the entry prohibited area, it stops at the position slightly inside the entry prohibited area beyond the boundary and the stored stroke limit error occurs. In this state, the cutting tool is allowed to be moved only in the opposite direction manually.
- If the MACHINE LOCK switch is ON, the check is made based on the coordinate values in the machine coordinate system.

Example of Setting

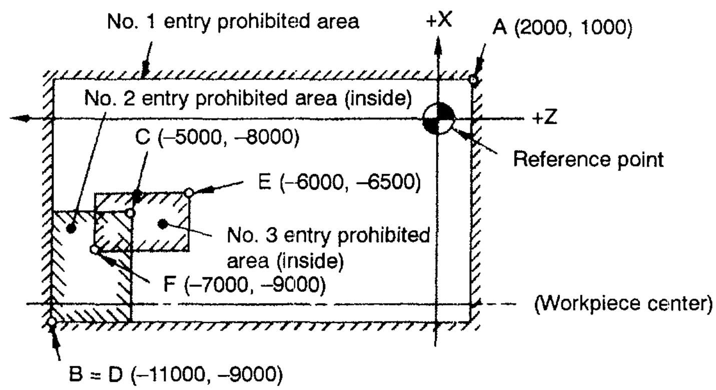


Fig. 4.83 Area Setting

	Parameter	Coordinate Values	
Inside/outside	pm0008 D4	0 (No. 2)	
	pm0008 D5	0 (No. 3)	
No. 2 area	pm0831	−5000	C
	pm0832	−8000	
	pm0834	−11000	D
	pm0835	−10000	
No. 3 area	pm0837	−6000	E
	pm0838	−6500	
	pm0840	−7000	F
	pm0841	−9000	
No. 1 area	pm6901	2000	A
	pm6902	1000	
	pm6911	−11000	B
	pm6912	−10000	

4.3 AUTOMATING SUPPORT FUNCTIONS

4.3.1 Skip Function (G31) *

By specifying "G31 X(U) ··· Z(W) ··· F(E) ···;", special linear interpolation is executed. If a skip signal is input during the execution of linear interpolation, linear interpolation is interrupted and the program advances to the next block without executing the remaining linear interpolation.

Delay from the input of the skip signal to the start of processing corresponding to the input signal is shorter than 0.5 msec; this is processed at extremely high speed.

(1) Programming Format

(a) Feedrate

For the execution of the G31 block, feedrate can be selected from the following two methods according to the setting for parameter pm2001 D0.

pm2001 D0 = 0	To specify the feedrate with F as another ordinary program
pm2001 D0 = 1	To use the feedrate preset for parameter pm2440

(b) If skip signal is turned ON

When the skip signal is input, the coordinate values of the point where the skip signal is input are automatically saved to the parameters. Therefore, the coordinate values of the skip point can be used as the coordinate data in macro programs.

pm0811	Saving the X-axis coordinate value
pm0812	Saving the Z-axis coordinate value

(c) If skip signal is not turned ON

If the skip signal is not turned ON during the execution of the commands specified in the G31 block, the operation stops upon completion of these commands and alarm "0491" occurs. Note that G31 is a non-modal G code.

Note that G31 is a non-modal code.

pm0007 D2 = 0	An alarm occurs if the skip signal is not input until the completion of the G31 block.
pm0007 D2 = 1	An alarm does not occur if the skip signal is not input until the completion of the G31 block. The program advances to the next block.

(2) Operation after Skip Signal ON

How the axes move after the turning ON of the skip signal varies depending on the commands specified in the block to be executed next.

(a) When axis move commands in the next block are incremental commands

The position where the skip signal is turned ON is taken as the reference point to execute the incremental commands in the next block.

Example of Programming

G31 W120.;
G01 U100.;

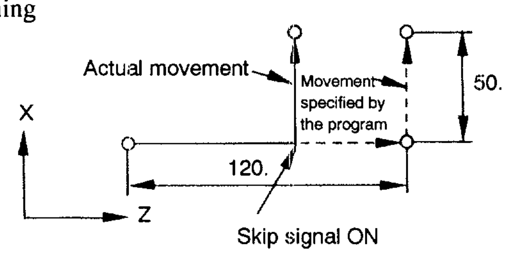


Fig. 4.84

(b) When axis move command in the next block is absolute command (one axis)

The axis specified in the next block moves to the specified position and the other axis remains at the position where the skip signal has turned ON.

Example of Programming

G31 Z400.;
G01 X100.;

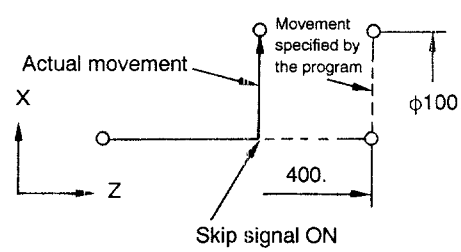


Fig. 4.85

(c) When axis move commands in the next block are absolute command (two axes)

The axes move to the specified position when the skip signal is turned ON.

Example of Programming

G31 W100.;
G01 X300. Z200.;

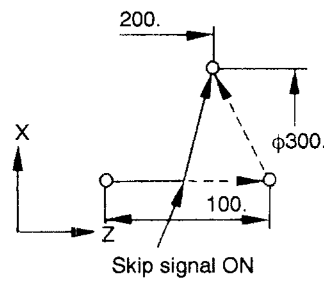


Fig. 4.86

4

SUPPLE-MENT

Before specifying G31, cancel the nose R offset mode by specifying G40. If G31 is specified without canceling the nose R offset mode, alarm “0182” occurs.

4.3.2 Tool Life Control Function (G122, G123) *

When the tool life control function is used, the tools are controlled in groups and the service life of the tool is set for individual tools. Tools are selected by tool group and if a selected tool is used to the preset life, it is replaced with another tool in the same tool group according to the preset tool selection order.

(1) Tool Life Control Specifications

(a) Number of tool life controlled tools

With the tool life control function, tool life can be controlled for up to 256 tools. The tools are grouped for this function. Parameter pm0009 is used to set the number of tool groups and the number of tools in each group.

Table 4.38 Number of Tool Life Controlled Tools

pm0009		Number of Groups	Number of Tools in Group
D7	D6		
0	0	64	4
0	1	32	8
1	0	16	16
1	1	8	32

POINT

Do not change the number of tool groups during the execution of the tool life management function.

4

(b) Tool life control data

The data used by the tool life control function is stored as the tool group file. The data in this file is retained even after the power is turned OFF.

Table 4.39 Tool Life Control Data

Title	Description
T NO (tool number)	The number assigned to the tool for which tool life is controlled.
LIFE (tool life)	The service life of the tool set for the tool number
USED (tool used data)	Counted used data of the tool set for the tool number
STS (status)	Status of the life of the tool set for the tool number OVER/SKIP/USE/NOT
LIFE TYPE	Tool life control data for each tool group TIME: 1 to 9999 COUNT: 1 to 9999

POINT

The data in the tool file and other files become valid at the timing when the tool selection command is executed after modifying the data. Even if the NC is reset after modifying the data, the new data are not reflected to the tool life management information presently executed.

(c) Counting the tool life

Tool life is counted for the life-controlled tools while they are actually called up and used in the tool life control mode. It is possible to set the tool life counting type for each of tool groups.

- Control by time

  Total cutting time of the specified tool is counted. Although counting is made in units of seconds, the counted value is stored in units of minutes. The fraction of data (data in seconds) are retained until the power is turned OFF and the tool life time data are counted continuously when the same tool is selected next before turning the power OFF. The counted data are cleared when the power is turned OFF.

  Cutting time means the following:

  · Cutting in the G01, G02/G03 mode

  · Time until the skip signal is input by the G31 command

  · Thread cutting in the G32, G34 mode

  · The time in which axis movements are controlled at a cutting feedrate in a canned cycle, etc.

  · The time in which axis movements are controlled in the thread cut mode in a canned cycle, etc.

- Control by the number of use

  The number of tool use is counted by specifying a predetermined code in a program. The code to be used for this function is specified by the setting for the parameter.

pm0009 D1 = 0	By M code
pm0009 D1 = 1	By T9999

Since tool life is counted in the buffering processing, if life exceeded status occurs during the execution of the next one block, the life exceeded status is triggered before the execution of that block.

POINT

During the operation, registration or deletion of a tool file is not allowed.

SUPPLE-MENT

1. It is possible to select the tool life counting (number of uses) objective groups by using the parameter indicated below.

pm4029 D3 = 0	Counting only for the specified tool groups
pm4029 D3 = 1	Counting for all registered tool groups

2. If "pm4029 D3 = 1", tool life is counted for the tools for which "TIME" is set for "LIFE TYPE" and the STS of them is "NOT", "USED" or "OVER".
3. If the last tool in a group is skipped, the one previously used is called. If the previous tool has been skipped, the one used before the previous tool is called.

4

(d) Life count conditions

Although tool life count processing is executed automatically, life count processing is not executed in the following cases.

When the life count ignore request input signal is ON	Life count data (time and use count) are not counted while this signal is ON.
When the MST lock input signal is ON	Life count data (time and use count) are not counted while this signal is ON and also during the period until a T command is input after this signal is turned OFF. This is because the actual tool number and the tool number specified in the program could differ from each other.
When the machine lock signal is ON	Life count data (time and use count) are not counted while this signal is ON.
When the dry run input signal is ON	Life count data (time and use count) are not counted while this signal is ON.
When the feed hold input signal is ON	Life count data (time) are not counted while this signal is ON and also during the period until the cycle start signal is input after this signal is turned OFF.
When the operation mode is changed over	Life count data (time) are not counted while in the manual operation mode and during the period until the cycle start signal is input after the recovery to the automatic operation mode. The use count data are counted upon the input of the operation completion signal if MST is saved or at the input of the cycle start signal after recovery to the automatic operation mode if operation has been finished forcibly.
When the internal toggle switches of MST function lock, machine lock, and dry run are turned ON	Especially when the MST function lock switch is ON, the life count data are not counted until a T command is input correctly.

(2) Setting of Tool Life Control Data

The tool life control data are set by manually inputting them on the Tool life screen. In addition to this method, the data can also be set by using the following methods.

(a) Using the tool data registration commands (G122, G123)

G122	Start of tool data registration
G123	End of tool data registration

In each tool group, the tool data are stored in the order they are specified. Disregarding of whether or not the tool data exist in the tool data registration area, the specified tool data are stored from the beginning of the area to overwrite the existing data. Concerning the data that are not overwritten, the previous data remain as they are. After storing the necessary tool data, specify "T0" or clear the tool data of the tool group for which the tool data are going to be registered in advance. If the tool data exceeding the allowable number of tools are specified, the data of the tools exceeding the limit are discarded. If no setting data are specified following tool number, "0" is set for both SET and USED. Note that STS data cannot be set using a program.

Example of Programming

```
O0001 ;
G122 P1 I1 ;   ←——   Start of tool data registration for 01 group tools
T0101 L0100 U0100;
T0202 L0200 U0200;
T0303 L0300 U0300;
T0404 L0400 U0123;
T0;
P2 I0 ;        ←——   Start of tool data registration for 02 group tools
  .
  .
  .
G123 ;         ←——   End of tool data registration
M30 ;
```

P : Tool group number (1 to 64: Max. group number)

I : Life kind (0: Count, 1: Time)
A space is entered if a number other than "0" or "1" is set.

T : Tool number (0 to 9999)

L : Life setting (0 to 9999)

U : Tool use data (0 to 9999)

SUPPLE-MENT

Alarms related to the tool life control function

- If T9999 is registered using the tool data registration program, alarm "0300" occurs.
- If an address other than P, I, T, L, and U is specified in tool data registration commands, or if no tool is registered to the group that is selected by the tool group selection command, alarm "0301" occurs.

(b) Using the user macroprogram command

For the data in the tool life control function tool group file, system variable numbers of the user macroprogram are assigned. By setting a system variable with the user macroprogram command, tool change commands, etc. can be changed. The macro system numbers of the data used by the tool life control function are indicated below.

Tool No.	#60001 to #60256
SET data	#60301 to #60556
USED data	#60601 to #60856
STS data	#60901 to #61156
TYPE data	#61201 to #61456

(c) Using tape format

If tool group data are output to tape using the external communication function, there is no distinction of tool groups and the data are output from the tools in 01 group successively by the number of tools that can be registered. If there are vacant areas where tool data are not registered, the information of such areas is output as tool number "0000". If the tool data are input using tape, the tool data area stored in the same format.

Example of Output to Tape

```
% ;
$1 ;
T0101 L0100 U0100 S3 I1 ;
T0202 L0200 U0200 S3 I1 ;
T0303 L0300 U0300 S3 I1 ;
T0404 L0400 U0123 S4 I1 ;
T0505 L0500 U0045 S2 I1 ;
T0606 L0600 U0000 S0 I1 ;
T0707 L9999 U0000 S0 I1 ;
T0000 L0000 U0000 S0 I1 ;
T1121 L0123 U0000 S0 I0 ;
T1222 L1234 U0000 S0 I0 ;
:
:
:
%
```


(3) Tool Selection Command

To execute a tool selection command using the tool life control function, specify a T code in the following format.

T99 □□;

└── Tool group number

Tools in the group specified by "□□" are selected in order among available ones. Alarm "0302" occurs if the status of all tools in the selected tool group is "skip".

(4) Relation with Coordinate System Setting Command

(a) Workpiece coordinate system setting function

The format for the workpiece coordinate system setting function is changed when the tool life control function is added.

G50 T99 □□;

Group number

When the tool life control function is executed, the workpiece coordinate system number is determined as follows: value "50" is added to the higher two digits of the registered tool number of the tool selected in the tool group which is specified by "T99□□". In this case, however, lower two digits (workpiece coordinate system offset number) of the registered command value are assumed to be "0". For example, if the following program is executed when the tool number of the tool selected in 01 group is "T0203",

```
G50 T9901 ;
T9901 ;
```

this is equivalent to the following program.

```
G50 T5200 ;
T0203 ;
```

(b) Tool coordinate system setting function*

If the optional function to set a coordinate system automatically when a T code is executed, the coordinate system is set according to the registered command value after the execution of the tool life control group selection command. In this case, the registered value is directly used as the coordinate system setting value.

SUPPLE-MENT

1. When the tool life control function is used, a T code in a program does not indicate a specific tool, but it indicates a tool group, and a tool is selected from the specified tool group to execute the program. If a tool command does not satisfy the requirements for a tool group command, the T command in a program is directly executed.
2. While the 1-line MDI function is executed, do not specify the spare tool selection command or the life counting command of the tool life control function. T99□□, T9999, and M△△ are processed as normal commands; execution of such commands finishes immediately by outputting an external signal. Accordingly, the next tool search and life counting are not executed.
3. If the turret is rotated manually, actual tool number of the tool at the cutting position differs from the tool number specified in a program. Even in this state, the function counts tool life data for the tool number specified in the program. Since this state continues until a T command is executed in automatic operation mode, do not execute the tool life control function until the tool number of the tool actually indexed to the cutting position agrees with the tool number specified in the program.

4.4 MACROPROGRAMS

The NC has a set of instructions that can be used by the machine tool builders and the users to implement the original functions. The program created by using these instructions is called a macroprogram, which can be called and executed by the commands specified in a block with G65 or G66.

A macroprogram provides the following.

- Variables can be used.
- Arithmetic and logical operations using variables and constants are possible.
- Control commands for branch and repeat can be used.
- Commands to output messages and data can be used.
- Arguments can be specified.

This makes it possible to create a program in which complicated operations and operations requiring conditional judgment are included.

4

4.4.1 Differences from Subprograms

Differences between macroprograms and subprograms are indicated below.

- With macroprogram call up commands (G65, G66), arguments can be specified. However, with subprogram call up command (M98), it is not possible to use arguments.
- If commands other than P, Q, and L are specified in the M98 block, the program jumps to the specified subprogram after executing these commands. With G65 and G66, commands other than P and L are regarded as argument specification and the program jumps to the specified macroprogram immediately. In this case, however, the commands specified preceding G65 and G66 are executed normally.
- With a macroprogram, local programs at the same level as the level of the macroprogram are used. However, with subprograms, levels of local variables are not changed.

  In other words, local variables in a macroprogram are different before and after the call up of the macroprogram and those in a subprogram remain the same before and after the call up of the subprogram.

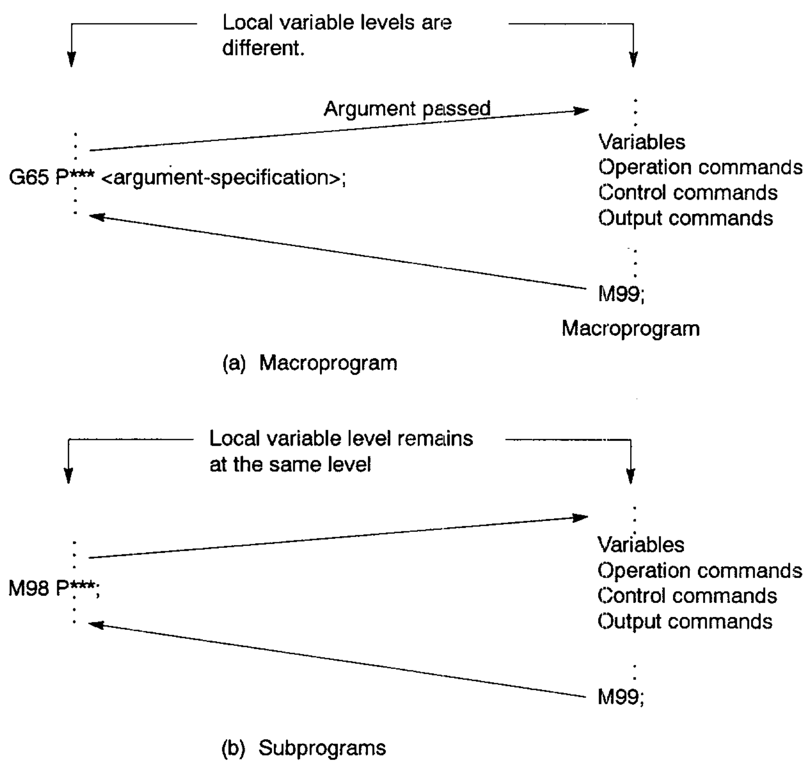


Fig. 4.87 Differences between Macroprograms and Subprograms

4.4.2 Macroprogram Call (G65, G66, G67) *

Macroprograms are usually executed after being called up.

The procedure used for calling up a macroprogram is indicated in Table 4.40.

Table 4.40 Macroprogram Calling Format

Calling Up Method	Command Code	Remarks
Simple call up	G65	
Modal call up (a)	G66	Canceled by G67
G code call up	G***	G command: 3 digits
M code call up	M***	M command: 2 or 3 digits
T code call up	T****	T command: 4 digits or 6 digits

POINT

By specifying these codes in the order of G, M, and T, the M and T codes are disregarded while only G code is valid. This specification does not cause an alarm.

If macroprogram call commands of G, M, and T are specified in one block, the priority of the commands is "G · · · M · · · T · · · ;" disregarding of the order they are specified in the same block.

(1) Simple Call Up (G65)

By specifying "G65 P ··· L··· <argument specification>; ", the macroprogram which is assigned the program number specified with P is called up and executed L times.

If it is necessary to pass arguments to the called up macroprogram, these arguments can be specified in this block.

Table 4.41 P and L Commands

Address	Description	Number of Digits
P	Program number	5 digits
L	Number of repetitions	9 digits

(2) Modal Call Up (G66, G67)

The modal call up commands set the mode for calling up a macroprogram. The specified macroprogram is called up and executed when the specified conditions are satisfied.

- By specifying "G66 P ··· L··· <argument-specification>; ", the mode for calling up the macroprogram is set. Once this block is executed, the macroprogram which is assigned the program number specified with P is called up and executed L times after the completion of move commands.

  If an argument is specified, the argument is passed to the macroprogram each time it is called up as with the simple call up of a macroprogram. The correspondence between the address of argument and local variables is the same as in the case of simple call up (G65).

- G67 cancels the G66 mode. When arguments are specified, G66 must be specified before all arguments. If G66 is specified, G67 must be specified in the same program corresponding to it.

Table 4.42 Modal Call Up Conditions

Call Up Conditions	Mode Setting Code	Mode Cancel Code
After the execution of move command	G66	G67

TERM?

◆ Argument Specification

A real number is assigned to the local variable that corresponds to the level of the called up macroprogram.
When specifying arguments, G65 must be placed before all arguments.
Commands specified before G65 are processed as normal commands and the program jumps to the called up macroprogram after the completion of these commands.
For details, refer to item (6) "Specifying Argument".

(3) Macroprogram Call Up by G code

By specifying "G*** <argument-specification>;", the macroprogram/subprogram of the program number that corresponds to the number specified by G code is called up and executed.

For the G code used to call up a macroprogram/subprogram, a maximum of 24 pairs of G codes can be set; each G code has a maximum of 3 numerical digits that are not used by the NC. The program numbers of programs to be called up can be set corresponding to the set G codes.

Table 4.43 Parameters for Setting the Correspondence

Number of Pairs	Macroprogram Call Up G Code	Program No. to be Called Up
1	pm4480	pm4840
2	(max. 3 digits)	(max. 5 digits)
.	.	.
.	.	.
.	.	.
23	.	.
24	pm4503	pm4863

↑
Max. 24 pairs

(4) Macroprogram Call Up by M Codes

(a) Macroprogram call up format

By specifying "M*** < argument >;", the macroprogram of the program number that corresponds to the specified M code is called up and executed.

In this case, if a move command is specified in the same block, the macroprogram is executed after the completion of the axis move command.

For the M code used to call up a macroprogram/subprogram, a maximum of 24 pairs of M codes can be set excluding such M codes as M00, M01, M02, M30, and those used for internal processing. The program numbers of programs to be called up can be set corresponding to the set M codes.

Table 4.44 Parameters for Setting the Correspondence

Number of Pairs	Macroprogram Call Up M Code	Program No. to be Called Up
1	pm4504	pm4864
2	(max. 3 digits)	(max. 5 digits)
.	.	.
.	.	.
.	.	.
23	.	.
24	pm4527	pm4887

↑
Max. 24 pairs

(b) Specifying arguments

It is possible to specify arguments in the M code macroprogram call up block. In this case, it is not allowed to specify axis move commands in the same block.

pm4020 D5 = 0	Call up without argument specification
pm4020 D5 = 1	Call up with argument specification

If more than one M code is specified in a single block, the first M code is checked whether it is for macroprogram call up. Concerning the second and later M codes, if the setting for parameter pm4020 D5 is “call up with argument specification”, the use of them is determined by the setting for a parameter whether they are treated as a normal M code or as M code used for specifying an argument.

pm4020 D6 = 0	Normal M code
pm4020 D6 = 1	M code for specifying argument

When an M code used for macroprogram call up is executed, M code or MF which is output for normal M code is not output.

4

(5) Macroprogram Call Up by T Code

By specifying "T***** ", it is possible to determine whether the specified T command should be treated as a normal T command or a macroprogram call up T command by the setting for parameter pm4889. If "pm4889 = 0", the T command is treated as a normal T command.

When using a T command for calling up a macroprogram, one required program number can be set. In this case, the T command value is used as the argument of common variable #149. Designation of other arguments is not allowed.

Table 4.45 Parameter Used for Macroprogram Call Up by T Code

Command Selection	Call Up Program Number
Normal T command	pm4889 = 0
Macroprogram call up T command	pm4889 max. 5 digits

When a macroprogram call up T command is executed, T code and TF are not output as a normal T code. The T command is a 4-digit or 6-digit command.

(6) Nesting of Macroprogram Call Up

As with subprograms, it is possible to call up a macroprogram from another macroprogram. In this type of call up, nesting level increases one each time a macroprogram call up is executed by G65, G66, G, M, or T code. The allowable maximum nesting level of macroprogram call up is four.

(a) Nesting level of macroprogram call up

With a macroprogram called up by G, M, and T codes, the allowable nesting level is one. In other words, from a macroprogram called up by G, M, or T code, call up of another macroprogram using G, M, or T code again is not allowed. If G, M, or T code is specified in a macroprogram which has been called up by the execution of macroprogram call up G, M, or T code, an alarm occurs in the case of a G code and with other codes (M and T), they are executed as normal M and T codes.

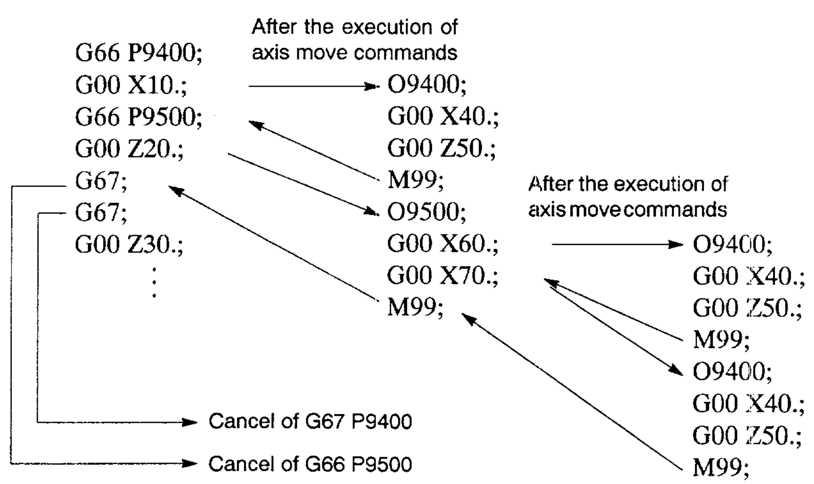
(b) Modal call up (G66)

In the modal call up mode, the specified macroprogram is called up and executed at each execution of a move command. If more than one G66 is specified in the same program, the prior G66 command specified is valid during the execution of a macroprogram called up by the G66 command given later. Therefore, after the execution of a move command given in the macroprogram called up by G66 specified later, the macroprogram specified with the previous G66 is also executed. In other words, the macroprograms are executed sequentially starting with the one specified last.

Example of Programming

:		
G66 P9400;	O9400;	O9500;
G00 X10.;	G00 X40.;	G00 X60.;
G66 P9500;	G00 Z50.;	G00 Z70.;
G00 Z20.;	M99;	M99;
G67;		
G67;		
G00 Z30.;		
:		

Execution order of programs



Note: If macroprogram call up is nested by specifying more than one G66, cancel code G67, cancels G66 sequentially beginning with the one specified last. It is not allowed to specify G66 in the macroprogram which is called up by G66.

Fig. 4.88 Nesting of Macroprogram Call

(7) Specifying Argument

The term "to specify argument" means "assigning a real number" for local variables used in a macroprogram. There are two types of argument specifications: type I and type II. These types can be used as required, including a combination of the two types.

(a) Correspondence between addresses and local variables (Type I)

Table 4.46 Address - Variable Correspondence and Usable Addresses for Call Up Commands (Type I)

Address - Variable Correspondence		Address - Variable Correspondence	
Address in Type I	Local Variable	Address in Type I	Local Variable
A	#1	Q	#17
B	#2	R	#18
C	#3	S	#19
D	#7	T	#20
E	#8	U	#21
F	#9	V	#22
H	#11	W	#23
I	#4	X	#24
J	#5	Y	#25
K	#6	Z	#26
M	#13		

4

(b) Correspondence between addresses and local variables (Type II)

To use I, J, and K, they must be specified in the order of I, J, and K. Suffixes 1 to 10 specified in the table below indicate the order they are used in a set, and the suffix is not written in actual commands.

For addresses for which argument specification is not required, the commands can be omitted. In this case, local variables corresponding the addresses without commands are <empty>.

Table 4.47 Address - Variable Correspondence and Usable Addresses for Call Up Commands (Type II)

Address - Variable Correspondence		Address - Variable Correspondence	
Address in Type II	Local Variable	Address in Type II	Local Variable
A	#1	K_5	#18
B	#2	I_6	#19
C	#3	J_6	#20
I_1	#4	K_6	#21
J_1	#5	I_7	#22
K_1	#6	J_7	#23
I_2	#7	K_7	#24
J_2	#8	I_8	#25
K_2	#9	J_8	#26
I_3	#10	K_8	#27
J_3	#11	I_9	#28
K_3	#12	J_9	#29
I_4	#13	K_9	#30
J_4	#14	I_{10}	#31
K_4	#15	J_{10}	#32
I_5	#16	K_{10}	#33
J_5	#17		

Note 1: If more than one address is specified for one variable number, the one specified later is valid.

2: If more than one set of I, J, or K is specified, the order of sets is determined for each I/J/K set, so that variable numbers are determined corresponding to that order.

(c) Example of argument specification

When arguments are specified, the macroprogram call up code must always be specified before the specification of arguments. If argument specification is given before the macroprogram call up code, an alarm occurs. The value of argument specification can include a sign and decimal point independent of the address.

If no decimal point is used, the value is saved to the variable as the value with a decimal point according to the normal number of digits of that address.

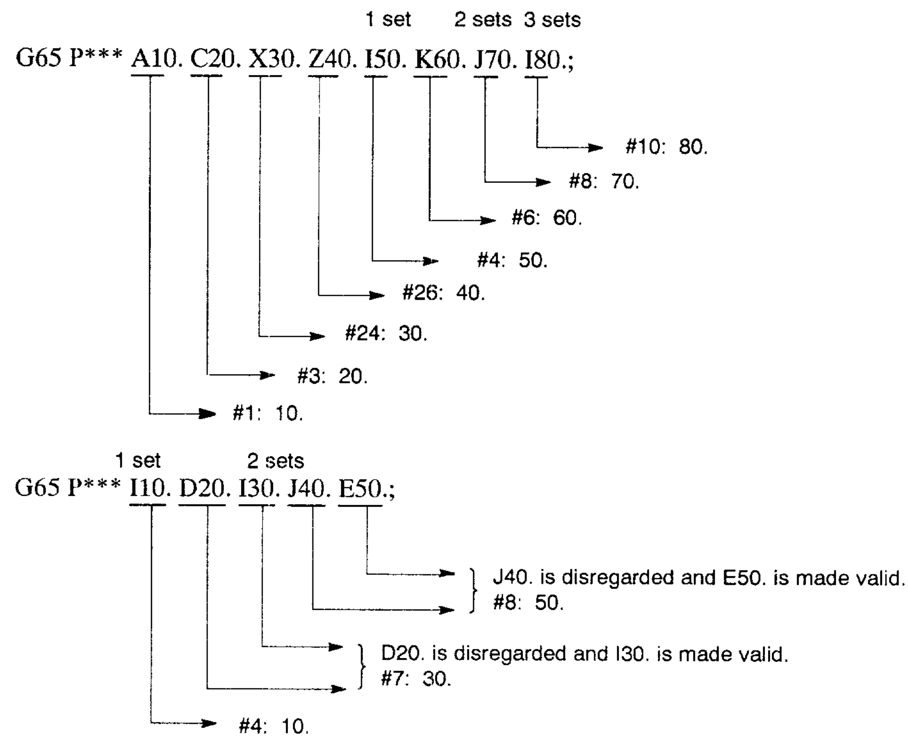


Fig. 4.89 Example of Argument Specification

(d) Decimal point position in argument

An argument is usually specified with a sign and a decimal point. If a decimal point is not specified, the decimal point position is assumed at the position indicated in Table 4.48.

Table 4.48 Decimal Point Position in Argument

Address for Argument Specification	Input in mm	Input in inches
A, B	3	3
D, H	0	0
E	4	6
F (in the G99 mode)	3	4
F (in the G98 mode)	3	4
I, J, K, C	3 (2)	4 (3)
M, S, T	0	0
Q	0	0
R	3 (2)	4 (3)
U, V, W	3 (2)	4 (3)
X, Y, Z	3 (2)	4 (3)

Note 1: The number indicates the position of the decimal point counted from the lowest position.

2: Numbers in ( ) indicate the number of digits right to the decimal point when the setting of parameter "pm1000 D0 = 1".

4.4.3 Variables

Three types of variables are provided: local variables, common variables and system variables.

(1) Local Variables (#1 to #33)

Local variables are used locally for each macroprogram. Each time a macroprogram is called up, new local variables (#1 to #33) are secured independently for that macroprogram. For the local variables, values specified using arguments are saved or the results of operation executed in the macroprogram are saved.

For those for which an argument is passed, the value is saved and those for which argument is not passed, the contents are <empty>. When execution of a program returns from the called up macroprogram by the execution of M99, the local variables secured for that macroprogram become <empty>. They are also <empty> when the power is turned ON or the NC is reset.

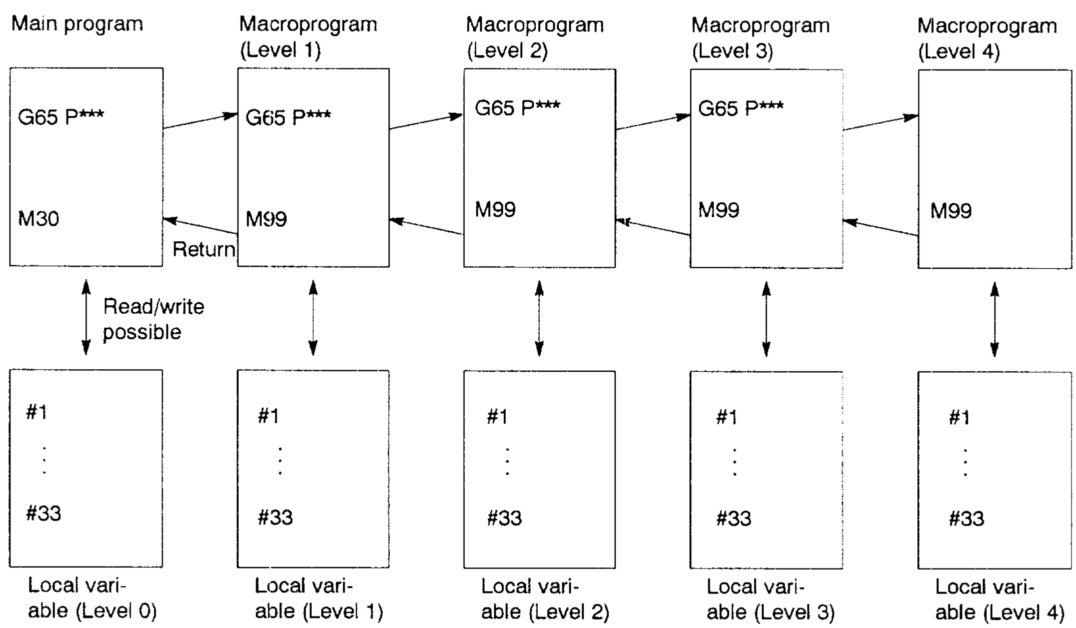


Fig. 4.90 Local Variables

① Local variables of level 0 are secured for the main program. For macroprograms, local variables are secured corresponding to the level (level 1 to level 4) of the called up macroprogram.

② If a macroprogram is called up by G65, for example, the local variables used for the program where macroprogram call up is executed are saved and the local variables are secured for the called up macroprogram corresponding to its level. In this case, the arguments are passed to the called up macroprogram. Consequently, even with the same macroprogram, the local variables of the level of that macroprogram have different values if the macroprogram is called up in different timing.

③ When the execution of a macroprogram returns to the macroprogram one level above by the execution of M99, the local variables of the previous macroprogram level are reset to <empty> and the local variables having been saved are recovered.

④ You should not change the contents of local variables while a macroprogram is being executed. If they are changed after interrupting the operation by single block stop, make sure that the new contents do not cause problems before restarting the operation.

⑤ Local variables can be used in a subprogram. In this case, the local variables of the present macroprogram level are used. Argument specification is not allowed when calling up a subprogram. The contents of the local variables are not reset to <empty> when the execution of a program returns from the subprogram by the execution of M99.

(2) Common Variables (#100 to #299, #500 to #999)

Common variables means the variables that can be used in common in main programs, subprograms, macroprograms, and those called up in nesting. Therefore, the common variable where the result of an operation executed in a macroprogram is saved can be used in another macroprogram. For common variables, argument specification is not allowed.

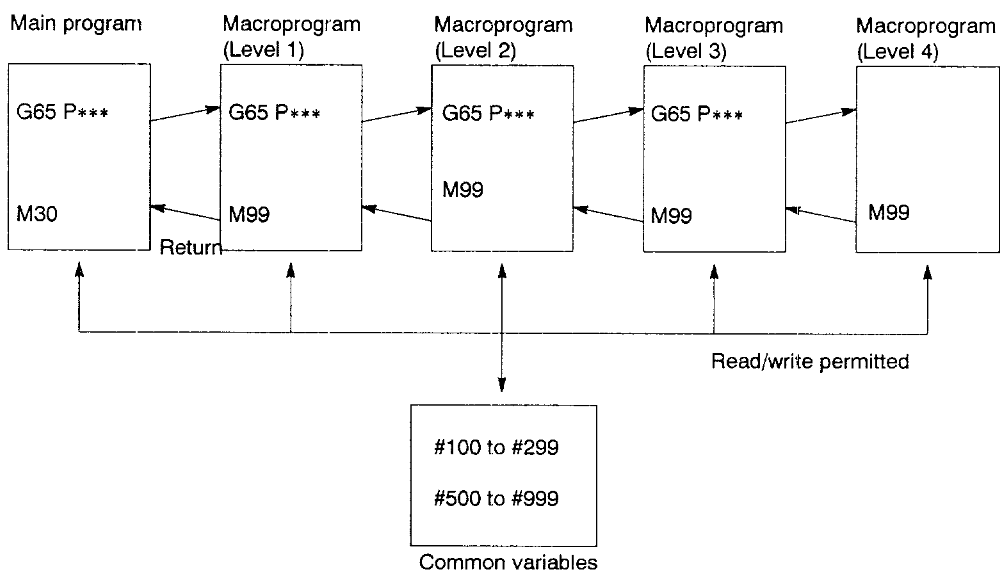


Fig. 4.91 Common Variables

- Common variables are classified into two types according to the state they are in when the NC is reset.

Table 4.49 Common Variables

#100 to #299	The content is <empty> when the power is turned ON or the NC is reset. By setting parameter "pm4009 D1 = 1", the content is not cleared to <empty>.
#500 to #999	The content is saved and not cleared to <empty> when the power is turned ON or the NC is reset.

- The number of sets of the common variables can be optionally expanded.

Table 4.50 Option Type and Expanded Common Variables

Option Type	Number of Sets
a	#100 to #149 (50 sets) #500 to #559 (60 sets)
b	#100 to #199 (100 sets) #500 to #599 (100 sets)
c	#100 to #199 (100 sets) #500 to #699 (200 sets)
d	#100 to #299 (200 sets) #500 to #999 (500 sets)

(3) System Variables

With the system variables, their use is predetermined as indicated in Table 4.51.

Table 4.51 System Variables

Type of System Variable	System Variable No.
Interface input signals	#1000 to #1031, #1032
Interface output signals	#1100 to #1131, #1132
Tool offset amount, workpiece coordinate system shift distance	#2001 to #2499, #14101 to #14112 #12001 to #13499
Alarm message display	#3000
Clock	#3001, #3002 to #3010
Control for single-block stop, and miscellaneous function complete wait	#3003
Control for feed hold, feedrate override, and exact stop	#3004
RS-232C data output	#3100
Modal information	#4000 to #4999
Position information	#5000 to #5999

4

(a) Interface input signals

- By entering system variables #1000 to #1031 in the right side of an operation expression, it is possible to read the ON/OFF state of each of 32-point input signal exclusively used for a macroprogram.

  The relationship between the input signals and system variables is indicated in Table 4.52.

Table 4.52 Interface Input Signals and System Variables

System Variables	#1007	#1006	#1005	#1004	#1003	#1002	#1001	#1000
Input Signals	UI 7 2^7	UI 6 2^6	UI 5 2^5	UI 4 2^4	UI 3 2^3	UI 2 2^2	UI 1 2^1	UI 0 2^0
System Variables	#1015	#1014	#1013	#1012	#1011	#1010	#1009	#1008
Input Signals	UI 15 2^{15}	UI 14 2^{14}	UI 13 2^{13}	UI 12 2^{12}	UI 11 2^{11}	UI 10 2^{10}	UI 09 2^9	UI 08 2^8
System Variables	#1023	#1022	#1021	#1020	#1019	#1018	#1017	#1016
Input Signal	UI 23 2^{23}	UI 22 2^{22}	UI 21 2^{21}	UI 20 2^{20}	UI 19 2^{19}	UI 18 2^{18}	UI 17 2^{17}	UI 16 2^{16}
System Variables	#1031	#1030	#1029	#1028	#1027	#1026	#1025	#1024
Input Signals	UI 31 2^{31}	UI 30 2^{30}	UI 29 2^{29}	UI 28 2^{28}	UI 27 2^{27}	UI 26 2^{26}	UI 25 2^{25}	UI 24 2^{24}

The value read to the system variables indicated above is either "1.0" or "0.0" according to the ON/OFF state of the corresponding input signals.

Table 4.53 Value of Variables

Input Signal State	Variable Value
ON	1.0
OFF	0.0

- By entering system variable #1032 in the right side of an operation expression, it is possible to read the ON/OFF state of all of 32 points of input signals (U10 to U131) collectively as a positive decimal value.

$$\#1032 = \sum_{i=0}^{31} \#[1000 + i] \times 2^i$$

- Note that it is not possible to enter a value by entering a system variable (#1000 to #1032) in the right side of an operation expression.

4

(b) Interface output signals

- By entering system variables #1100 to #1131 in the right side of an operation expression, it is possible to output the ON/OFF state to each of 32-point output signal exclusively used for a macroprogram. The relationship between the output signals and system variables is indicated in Table 4.54.

Table 4.54 Interface Output Signals and System Variables

System Variables	#1107	#1106	#1105	#1104	#1103	#1102	#1101	#1100
Output Signals	UO 7 2^{7}	UO 6 2^{6}	UO 5 2^{5}	UO 4 2^{4}	UO 3 2^{3}	UO 2 2^{2}	UO 1 2^{1}	UO 0 2^{0}
System Variables	#1115	#1114	#1113	#1112	#1111	#1110	#1109	#1108
Output Signals	UO 15 2^{15}	UO 14 2^{14}	UO 13 2^{13}	UO 12 2^{12}	UO 11 2^{11}	UO 10 2^{10}	UO 09 2^{9}	UO 08 2^{8}
System Variables	#1123	#1122	#1121	#1120	#1119	#1118	#1117	#1116
Output Signals	UO 23 2^{23}	UO 22 2^{22}	UO 21 2^{21}	UO 20 2^{20}	UO 19 2^{19}	UO 18 2^{18}	UO 17 2^{17}	UO 16 2^{16}
System Variables	#1131	#1130	#1129	#1128	#1127	#1126	#1125	#1124
Output Signals	UO 31 2^{31}	UO 30 2^{30}	UO 29 2^{29}	UO 28 2^{28}	UO 27 2^{27}	UO 26 2^{26}	UO 25 2^{25}	UO 24 2^{24}

4

By entering "1.0" or "0.0" to the system variables indicated in Table 4.54, the corresponding signals are output in the ON or OFF state.

Table 4.55 Value of Variables

Output Signal	Variable Value
ON	1.0
OFF	0.0

- If a value other than "1.0" or "0.0" is set for variables #1100 to #1131, it is treated as indicated below.

  <empty> or less than 0.5: 0.0
  Other than above: 1.0

- By entering system variable #1132 in the left side of an operation expression, it is possible to output the ON/OFF state to the 32 point output signals (U00 to U031) collectively. In this case, a positive decimal value set for #1132 is output after converted into a binary 32-bit value.

$$\#1132 = \sum_{i=0}^{31} \#[1100 + i] \times 2^{i}$$

- By entering system variables #1100 to #1132 in the right side of an operation expression, it is possible to read the ON/OFF state (1.0, 0.0, positive decimal value) output last can be read.

(c) Offset amount and workpiece coordinate system distance

Tool offset amount can be read by entering system variables #12001 to #13499 in the right side an operation expression.

Workpiece coordinate system shift distance can be read by entering system variables #14101 to #14112 in the right side an operation expression.

By entering the system variables indicated above in the left side of an operation expression, it is possible to update the offset values.

Example of Programming

#116 = #12016	:	Enters the content of tool offset number 16 to common variable #116.
#14101 = #4	:	Clears the workpiece coordinate system shift distance of X-axis and sets the content of local variable #4.

4

(d) Correspondence between system variables and tool offset numbers

The correspondence between the tool offset numbers and the system variables is indicated in Table 4.56.

Table 4.56 Tool Offset Numbers and System Variables (16 sets)

Offset Data Name	Offset No.	System Variable
X-axis	01 02 . . . 16	#12001 #12002 . . . #12016
Z-axis	01 02 . . . 16	#12301 #12302 . . . #12316
Nose R offset	01 02 . . . 16	#12901 #12902 . . . #12916
Control point	01 02 . . . 16	#13201 #13202 . . . #13216

4

Table 4.57 Tool Offset Numbers and System Variables (99 sets)

Offset Data Name	Offset No.	System Variable
X-axis	01 02 . . . 99	#12001 #12002 . . . #12099
Z-axis	01 02 . . . 99	#12301 #12302 . . . #12399
Nose R offset	01 02 . . . 99	#12901 #12902 . . . #12999
Control point	01 02 . . . 99	#13201 #13202 . . . #13299

4

4

Table 4.58 Tool Offset Numbers and System Variables (299 sets)

Offset Data Name	Offset No.	System Variable
X-axis	01 02 . . . 99 . . . 299	#12001 #12002 . . . #12099 . . . #12299
Z-axis	01 02 . . . 99 . . . 299	#12301 #12302 . . . #12399 . . . #12599
Nose R offset	01 02 . . . 99 . . . 299	#12901 #12902 . . . #12999 . . . #13999
Control point	01 02 . . . 99 . . . 299	#13201 #13202 (#2302) . . . #13299 . . . #13499

Table 4.59 Workpiece Coordinate System Shift Distance Setting System Variables

	System Variable
X	#14101
Z	#14102
C	#14103

(e) Alarm message display

By specifying "#3000 = <Alarm-number> (<Alarm-message>); ", the NC can be placed in the alarm state. The timing the NC is placed in the alarm state is after the completion of the commands in the block immediately preceding the block including the commands indicated above.

<Alarm-number> : A 4-digit alarm number not used by NC.
Use of a variable is allowed.
(Alarm number range: 5000 to 5999)

<Alarm-message> : ASCII character string with 32 or less characters (alphanumerics and special characters)

(f) Clock

It is possible to read time by entering the system variable used for the clock in the right side of an operation expression. If such a system variable is entered in the left side of an operation expression, it is possible to preset the time.

Table 4.60 System Variables Used for Clock Function

Type	System Variable	Unit	At Power ON	Count Conditions
Clock 1	#3001	1 msec	Preset to "0"	Always
Clock 2	#3002	1 sec	The state immediately before the power was turned OFF is retained	When the STL signal goes ON

(g) Control for single-block stop and waiting for completion of miscellaneous function

By setting an appropriate number for system variable #3003, the following control is possible:

- To make valid/invalid the SINGLE-BLOCK switch setting for the succeeding blocks.
- To advance the program to the next block without waiting for the input of the miscellaneous function (M, T) completion signal (FIN).

If a miscellaneous function is specified with the setting that the input of the completion signal (FIN) is not checked, the distribution complete signal (DEN) is not output and the program advances to the next block without waiting for the input of the FIN signal although the output of the M or T code and M or T read output is executed as normal. When the block in which the setting is made for system variable #3003 so that the state for checking the input of the miscellaneous function complete signal is executed after that, the NC outputs DEN signal and waits for the input of the FIN signal.

When M or T is specified in the state that the miscellaneous function complete is not checked, the first appearance of individual codes is recognized and executed. Then, the second and later appearance are disregarded until the setting is changed to the state in which the complete signal is checked. In the state that the miscellaneous function complete signal is not checked, the program advances to the next block without waiting for the input of the complete signal even when the buffering stop M code (M00, M01, M02, M30) is specified.

When the NC is reset, the setting for #3003 is "0".

Table 4.61 Control for Single-Block Stop and Miscellaneous Function Completion Waiting

#3003	SINGLE-BLOCK Switch	Miscellaneous Function Complete Signal (FIN)
0	Valid	Checked
1	Invalid	Checked
2	Valid	Not checked
3	Invalid	Not checked

Example of Programming

```
M10;          ← M10 code: MF output, DEN output, FIN checked
 :
#3003 = 3
 :
M20;          ← M20 code:  MF not output, FIN not checked
 :
M25;          ← Disregarded                                     "M" on display
 :
#3003 = 0     ← DEN output, FIN checked
 :
M40;          ← M40 code:  MF output, DEN output, FIN checked
```

Single-block stop FIN check not made

Fig. 4.92 Example of Single-block Stop and Miscellaneous Function Completion Waiting Control

(h) Setting for feed hold, feedrate override, and positioning completion control

For the control of feed hold, feedrate override, and positioning completion, system variable #3004 is provided and by setting an appropriate value for this system variable, it is possible to make these functions valid or invalid.

When the NC is reset, the setting is reset to " #3004 = 0".

Table 4.62 Control for Feed Hold, Feedrate Override, and Positioning Completion Functions

#3004	Feed Hold	Feedrate Override	Positioning Completion
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

- For the feed hold function

  The feed hold function is invalid in the following blocks.

  From the block where 1, 3, 5, or 7 is set for #3004

  To the block where 0, 2, 4, or 6 is set for #3004

  The blocks for which the feed hold is made invalid are not accepted and the feed hold signal is not output.

- For feedrate override

  The setting of feedrate override is disregarded in the following blocks.

  From the block where 2, 3, 6, or 7 is set for #3004

  To the block where 0, 1, 4, or 5 is set for #3004

- For the positioning completion function

  The check is not made for the completion of positioning.

  From the block where 4, 5, 6, or 7 is set for #3004

  To the block where 0, 1, 2, or 3 is set for #3004

4

Example of Programming for Special Thread Cutting Cycle (incremental mode)

Macroprogram call command

G65 P9093 U - · · · W - · · · K· · · F· · ·;

- F: #9 Thread lead
- K: #6 Radial value, unsigned
- W: #23 Negative value
- U: #21 Negative value, diametral value

Macroprogram

```
O9093 ;
M93 ;                      ——— 7-block buffering
#10 = ROUND [#6] *2 ;
#11 = ROUND [#21] +#10 ;
#12 = ROUND [#23] +#10 ;
#3003 = 1 ;                ——— SINGLE-BLOCK switch made invalid
G00 U#11 ;                     Feed hold
#3004 = 7 ;                ——— Feedrate override         } Made invalid
G32 U-#10 W-#6 F#9 ;           Positioning completion
G32 W#12 ;
G32 U#10 W-#6 ;
#3004 = 0 ;
G00 U-#11 ;
G00 W-#23 ;
#3003 = 0 ;
M92 ;
M99 ;
```

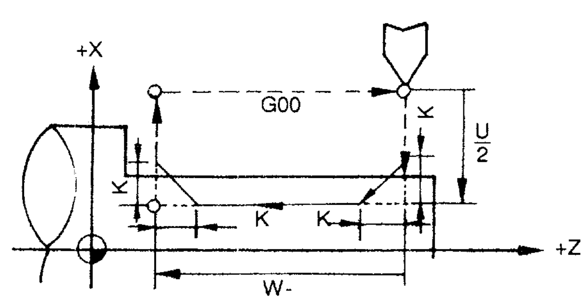


Fig. 4.93

(i) RS-232C data output 1 (#3100)

By using system variable #3100, it is possible to output a message and variable data to an external device via the RS-232C data input/output interface.

- By specifying "#3100 = (<message>);", the message enclosed by the control out and control in codes is output to an external device. The CR and LF (carrier return, line feed) codes are automatically output at the end of the message.

  If no message is input, only the CR and LF codes are output. The term message indicates the ASCII character string (alphanumerics and special characters) consisting of less than 128 characters.

- By specifying "#3100 = [<variable>];", the value of <variable>is output as 9-digit signed decimal data (decimal fraction: 4 digits, integer part: 5 digits). The term variable includes local variables, common variables, and system variables. If five or more digits are specified to the right of a decimal point, the number at the fifth place to the right of the decimal point is rounded off. And if six or more digits are specified in the integer part, an asterisk (*) is output for such digits.

Example of Programming

```
#3100 = (        )  <------  Line feed, carriage return
#3100 = (TOOL OFFSET 01) ;
#3100 = (__ ···X__ ··· Z__ ··· R) ;
#3100 = [#2001] ; ···  = 10.000 mm      ┐
#3100 = [#2101] ; ···  = -10.000 mm     │ In this case
#3100 = [#2201] ; ···  = 0.800 mm       ┘
#3100 = (        )
```

Printout data

```
TOOL OFFSET 01
            X                 Z                      R
_ _ _ _ 10.0000 _ _ _ -10.0000 _ _ _ _ _ _ 0.80000
```

Including a sign, data with up to 6-digit number in the left of a decimal point can be output.

(j) Special codes that can be used in a macroprogram

The allowable special codes are indicated in Table 4.63. For the characters indicated by note) in the table, the tape punch pattern in the EIA code is as indicated in Table 4.63.

Table 4.63 Special Codes

Meaning of Code	Application	EIA Code								ISO Code							
		8	7	6	5	4	3	2	1	8	7	6	5	4	3	2	1
SP	Comments				○					○		○					
note) (	Alarm message and comment				○	○		○				○		○			
note))			○			○		○		○		○		○			○
+	Addition		○	○	○							○		○		○	○
-	Subtraction		○									○		○	○		○
:	Comment		○				○	○				○	○	○		○	
/	Division			○	○				○	○		○		○	○	○	○
note) #	Variable		Parameter setting							○		○				○	○
note) *	Multiplication	○			○	○				○		○		○		○	
note) =	Equal sign	○				○	○			○		○	○	○	○		○
note) [	Brackets	○		○	○					○	○		○	○		○	○
note)]		○		○				○		○	○		○	○	○		○
$	Comment	○			○		○					○			○		
@		○				○	○	○	○	○	○						
?		○			○	○	○	○				○	○	○	○	○	○
.	Decimal point		○	○		○		○	○			○		○	○	○	
note) ,	Comma		○	○	○	○	○			○		○		○			

By using the following parameters, a hole punch pattern different from the pattern indicated above can be set. If the setting for these parameters is "0", the patterns indicated in Table 4.63 are used.

Table 4.64 Hole Punch Pattern Setting Parameters

pm4100	#	pm4104	=
pm4101	[	pm4105	(
pm4102	]	pm4106	)
pm4103	*	pm4107	,

Note: For these parameters, read the required punch hole pattern in a binary number and convert it into a decimal number to set.

(Example) To set "152" for punch hole pattern.

8	7	6	5	4	3	2	1
○			○	○			

(k) Modal information

By entering the system variables indicated below in the right side of an operation expression, it is possible to read the modal value given in blocks up to the immediately preceding block. Note that these system variables cannot be entered to the left side of an operation expression.

Table 4.65 Modal Values and Macro System Variables

Modal Command	Macro System Variable
G code (01-group) to (31-group)	#4001 to #4024
E code	#4108
F code	#4109
Sequence number	#4114
Program number	#4115
S code (1)	#4119
T code	#4120

Note 1: Since an M code is non-modal information, it is not possible to read M codes using system variables.

2: Concerning E (#4108) and F (#4109), either the E or F command specified immediately before the specification of the system variable is saved. Therefore, system variables #4108 and #4109 hold the same value.

Example of Programming

Main program

G65 P9602 <Designation of arguments>;

Macroprogram

```
O9602 ;
#1 = #4001 ;  ←———— Saves G codes (G00 to G03) of 01 group
G00 X · · · Z · · · ;
G01 Z · · · F · · · ;
G03 X · · Z · · R · · ;
G00 Z · · ;
G#1 ;  ←———— Recovers G codes of 01 group
M99 ;
```

4

(l) Position information

By specifying the system variables indicated below, it is possible to read the position information.

Note that these system variables cannot be specified in the left side of an operation expression.

Table 4.66 Position Information

Position Information	Macro System Variable	Reading during Operation
X-axis block end point position (ABSIO) Z-axis block end point position (ABSIO) C-axis block end point position (ABSIO)	#5001 #5002 #5003	Possible
X-axis position in the machine coordinate system (ABSMT) Z-axis position in the machine coordinate system (ABSMT) C-axis position in the machine coordinate system (ABSMT)	#5021 #5022 #5023	Possible (Note 1)
X-axis POS.ABS position (ABSOT) Z-axis POS.ABS position (ABSOT) C-axis POS.ABS position (ABSOT)	#5041 #5042 #5043	Possible (Note 1)
X-axis skip signal input position (ABSKP) Z-axis skip signal input position (ABSKP) C-axis skip signal input position (ABSKP)	#5061 #5062 #5063	Possible
X-axis offset Z-axis offset C-axis offset	#5081 #5082 #5085	Possible
X-axis servo position error Z-axis servo position error C-axis servo position error	#5101 #5102 #5105	Possible (Note 1)

Note 1: When the system variable indicated by (note1) is specified, the position information is read after the completion of the commands specified in the immediately preceding block.

2: If an additional axis is selected, the correspondence between the axis and the system variable could differ from the correspondence indicated above. For details, refer to the manuals published by the machine tool builder.

4

Table 4.67

Abbreviation	ABSIO	ABSMT	ABSOT	ABSKP
Description	End point position of the immediately preceding block	Present position of the command (the same value as the coordinate value in the machine coordinate system)	Present position of the command (the same value as the present position data)	The position where the skip signal is turned ON (G31 block)
Coordinate system	Workpiece coordinate system	Machine coordinate system	Workpiece coordinate system	Workpiece coordinate system
Tool offset amount	Not included	-	Included	Included

Note: The unit of the position information is the specified mm or inch input unit.

Table 4.68

	Unit	
	Microns	Sub Microns
Input in mm	0.001 mm	0.0001 mm
Input in inches	0.0001 inch	0.00001 inch
Input in degrees	0.001 deg	0.0001 deg

Note: If the skip signal is turned ON during the execution of the G31 block, the end point position of this block is the skip signal input position.
If the skip signal is not turned ON, the skip signal input position is the end point of the G31 block.

(4) Expression of Variables

Variables are expressed by variable numbers or alphanumerics specified following #.

- Specifying a variable number directly
  #i (i: variable number)

  (Example) #1, #101, #501, #2001

- Specifying an expression as a variable number
  # [<expression>]

  (Example) # [#101], # [#501+1], # [#1/2]

4

(5) Assigning Variables

A numeric value specified following an address can be replaced with a variable.

By specifying "<address> #i or <address> - #i", the value of the specified variable or its negative value (complement) can be taken as the command value of that address.

(Example)

G#30, #30=1.0	Equivalent to G01
X#101, #101=100.	Equivalent to X100.
Z#103, #103=300.	Equivalent to Z300.
F#140, #140=0.3	Equivalent to F0.3

- For the following addresses, it is not allowed to assign a variable.

(Example)

/#5	It is not allowed to use a variable for "n" of "/n" (n=1 to 9).
O #100	It is not allowed to use a variable for O number (program number).
N #200	If it not allowed to use a variable for N number (sequence number).

- It is not allowed to use a variable to express a variable number.

  When replacing "10" in #10 with #20, for example, expression of ##20 is not allowed. This must be written by # [#20].

- If a variable is used as the address data, values below the minimum input unit are rounded off.

(Example)

X#1, #1 = 45.2346	X45.235 mm (0.001 mm input unit)
F#2, #2 = 0.2555	F0.256 m/rev (F33 format)
G04 P#3, #3 = 5.37672	G04 P5.377 sec
M#4, #4 = 2.7236	M03
G#4, #4 = 2.7236	G03

- It is possible to use <expression> instead of a numeric value to be assigned to an address.

  By specifying "<address> [<expression>], or <address> - [<expression>]", the value or negative value (complement) of the <expression> can be used as the command value for that address.

- The constant used in [ ] without a decimal point is assumed to have a decimal point at the end.

(6) Undefined Variables

Variables which have not been defined yet are called undefined variables, and their values are <empty>. The following variables are treated as undefined variables.

- Local variables and common variables (#100 to #299) when the power is turned ON or the NC is reset.
- Local variables for which arguments are not specified when a macroprogram is called up.
- Local variables which belong to the level of the macroprogram from which the execution of program returns by the execution of M99.
- Local variables and common variables where no values have been set in a macroprogram.
- Common variables where no values have been set at the NC operation panel.
- Variable "#0". (This is always treated as < empty > and must not be entered in the left side of an operation expression.)

(a) Meaning of <empty>

- If an undefined variable is assigned, the address itself for which it is assigned is disregarded.

#2 = <empty>	G00X#2; is equivalent to G00;.
#2 = 0	G00X#2; is equivalent to G00X0;.

- If an undefined variable is used in an operation expression, it is treated to have the variable value of "0" with the exception that it is replaced with <empty>.

#2 = <empty>	#3 = #2; indicates #3 = <empty>.
#2 = <empty>	#3 = # [#2+#2]; indicates #3 = #0 = <empty>.
#2 = <empty>	#3 = #1*#2; indicates #3 = <empty>.
#2 = <empty>	#3 = #2+#2; indicates #3 = <empty>.
#2 = <empty>	#3 = #2/#2; indicates #3 = <empty>.
#2 = <empty>	#3 = 5*#2; indicates #3 = <empty>.
#2 = <empty>	#3 = 2-#2; indicates #3 = 2.
#2 = <empty>	#3 = 5/#2; causes division error.

- If an undefined variable is used in a conditional expression, it is treated to have the variable value of "0" with an exception of EQ and NE.

#3 = 0, #2 = <empty>	#3EQ #2: Not satisfied
#3 = 0, #2 = <empty>	#3NE #2: Satisfied
#3 = 0, #2 = <empty>	#3GE #2: Satisfied
#3 = 0, #2 = <empty>	#3LT #2: Not satisfied

4

4.4.4 Operation Instructions

By performing general arithmetic operations in which local variables, common variables, system variables, and constants are connected with operators and functions, it is possible to set the result of operation to the given variable.

The variables used in the arithmetic operation read the required data from the internal variable data area. The result of the operation is set to a variable to write the result of the operation to the internal variable data area. The write cycle is completed when the execution of one block is completed.

The basic formula of arithmetic operation is " #i = <expression>". The following operations and functions can be used.

4

(1) Definition and Setting of Variables

#i = #j	Definition or setting
#i = # [#j = #k]	Indirect designation

(2) Addition Type Operations

#i = #j + #k	Sum
#i = #j - #k	Difference
#i = #j OR #k	Logical sum (for each bit in 32-bit binary)
#i = #j XOR #k	Exclusive logical sum (for each bit in 32-bit binary)

(3) Multiplication Type Operation

#i = #j * #k	Product
#i = #j/#k	Quotient
#i = #j AND #k	Logical product (for each bit in 32-bit binary)
#i = #j MOD #k	Remainder (With #j and #k, remainder is obtained after rounding the values to an integer. If #j is negative, #i is also negative.)

(4) Functions

#i = SIN [#j]	Sine (in units of degrees)
#i = COS [#j]	Cosine (in units of degrees)
#i = TAN [#j]	Tangent (in units of degrees)
#i = ATAN [#j] or #i = ATAN [#j/#k]	Arctangent
#i = SQRT [#j]	Square root
#i = ABS [#j]	Absolute value
#i = BIN [#j]	Conversion from BCD to binary
#i = BCD [#j]	Conversion from binary to BCD
#i = ROUND [#j]	Conversion into integer by rounding off
#i = FIX [#j]	Cutting off decimal fractions
#i = FUP [#j]	Rounding off decimal fractions
#i = ASIN [#j]	Arcsine
#i = ACOS [#j]	Arccosine
#i = LN [#j]	Natural logarithm
#i = EXP [#j]	Exponent with e (= 2.718· · ·) as a base

4

(5) Combination of Operations

It is possible to combine the operations and functions explained in items (1) to (4) above.

In this case, the priority of operation is in the order of functions, multiplication type operation and addition type operation.

(Example) #i = #j + #k * SIN [#1]
③ ② ①

(6) Changing the Order of Operations by [ ]

By enclosing a part of an expression by brackets ([ ]), that part is given priority for calculation.

The brackets can be nested in up to five levels including the brackets used in functions.

(Example) #i = SIN [[[#j+#k]* #l + #m] * #n]
⑤ ① ② ③ ④

4

(7) Supplements to the Operation Instructions

- A constant used in <expression> without a decimal point is assumed to have a decimal point at the end. The allowable range of the constant is ±99999999.99999999.

- Function ROUND converts a value into integer by rounding off processing. This processing is executed at the digit indicated below.

  · If used in an operation instruction, conditional expression IF or WHILE, decimal fractions are rounded off.

  · If used in address data, the value is rounded off at the digit one place below the minimum input unit of the address.

    (Example 1) When #10 = 12.3758,
    #1 = ROUND [#10] → #1 = 12.0
    ROUND [#10] in IF [#10 GT ROUND [#10]] → 12.0

    (Example 2) When #10 = 12.3758,
    G00 X [ROUND [#10]] is equivalent to G00 X12.376
    (minimum input unit: 0.001 mm).

- Numerical values treated in macroprograms are floating point type values.

  $M*2^E$ M : One sign bit + 52-bit binary data
  E : One sign bit + 10-bit binary data

- With an operation instruction, whether the NC operation stops in the single-block mode or not when the single block input (SKB) is ON is determined by the setting for parameter pm0007 D1.

pm0007 D1 = 0	Does not stop in the single block mode.
pm0007 D1 = 1	Stops in the single block mode.

4.4.5 Control Instructions

To control the program flow of macroprograms, the following two instructions are provided: branch instruction and repetition instruction.

- Branch instruction
- Repetition instruction

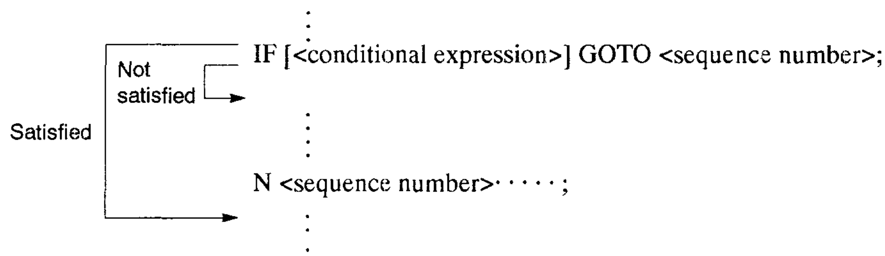
4

(1) Branch Instruction

By specifying "IF [< conditional expression >] GOTO <sequence number>; ", the program jumps to the block of the specified sequence number in the same program if <conditional expression> is satisfied.

If <conditional expression> is not satisfied, the program advances to the next block.

<sequence number> should be placed at the beginning of a block. Even if it is not placed at the beginning of a block, the commands in the block are executed from the beginning. When branch occurs, branch in the reverse direction takes a longer time than branch in the forward direction.



<sequence number> : 5-digit positive integer, variable, [<expression>]

Fig. 4.94 Branch Instruction (Conditional Expression Satisfied, Not Satisfied)

It is possible to omit "IF [< conditional expression >]. In this case, the block indicates a simple jump instruction.

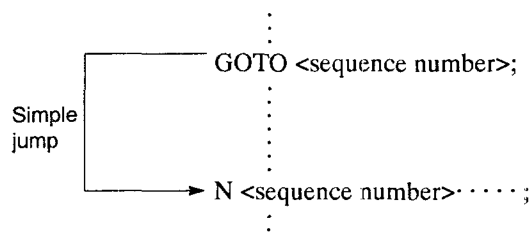


Fig. 4.95 Branch Instruction (Simple Jump Instruction)

Instead of "GOTO <sequence number>", an NC statement or macroprogram statement can be specified in one block. However, the following macroprogram statements cannot be used due to restrictions.

- Control instructions
- RS-232C data output 2
- Status monitoring instruction

```
                 :   Satisfied
                 ┌──────────────┐
                                ▼
Not        ┌──── IF [<conditional expression>] NC statement or macroprogram statement;
satisfied  └───►
                 :
```

Fig. 4.96 Branch Instruction (1-Block Instruction)

The <conditional expression> includes those indicated in Table 4.69.

Table 4.69 Types of Conditional Expressions

Conditional Expression	Description
#i EQ #j	#i = #j
#i NE #j	#i ≠ #j
#i GT #j	#i > #j
#i LT #j	#i < #j
#i GE #j	#i ≧ #j
#i LE #j	#i ≦ #j
A OR B	Logical sum of A and B
A AND B	Logical product of A and B
A XOR B	Exclusive logical sum of A and B

Note: Constants and <expression> can be used instead of #i and #j.

(2) Repeat Instructions

```
WHILE [<conditional expression>] DO<number>;
      :
      :
END <number>;                                    <number> = 1, 2, 3
```

With the commands indicated above, blocks between the block next to the DO block and the END block are repeatedly executed while the <conditional expression> is satisfied.

If the <conditional expression> is not satisfied, the program jumps to the block next to the END block. It is possible to omit "WHILE [<conditional expression>]. In this case, the block between the DO and END blocks is continually repeated.

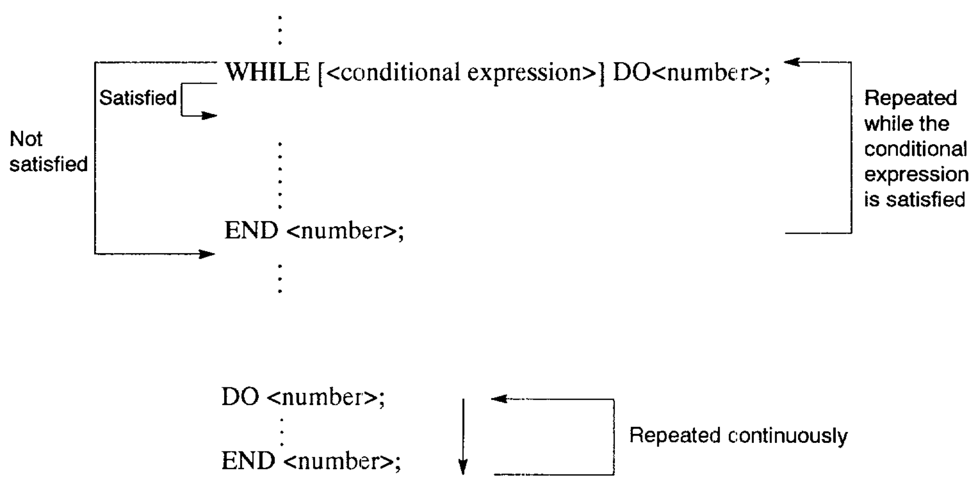


Fig. 4.97 Repeat Instruction

- DO must be specified before END.

DO1 - END1	END1 - DO1
○	×

Note: ○: Correct, × : Incorrect

4

- The <number> in “DO <number>” and “END <number>” must be the same number, and DO and END must be specified as a pair.

DO1 - END1 . . . DO2 - END2	DO1 - END2	DO1 - DO1 - END1	DO1 - END1 - END1
○	×	×	×

Note: ○: Correct, ×: Incorrect

- The same <number> can be used as many times as required. However, the range of repetition must not overlap.

DO1 - END1 . . . DO1 - END1	DO1 - DO1 - END1 - END1	DO1 - DO1 - END1 - END1	DO1 - DO2 - END1 - END2
○	×	×	×

Note: ○: Correct, ×: Incorrect

- Nesting of DO to END loop is allowed for up to three levels in a macroprogram or subprogram.
  From the DO to END loop, it is possible to call up a macroprogram or subprogram. In the call up program, nesting of the DO to END loop is also allowed for up to three levels.

DO1 - DO2 - DO3 - END3 - END2 - END1	DO1 - G65P*** - M98P*** - END1	DO1 - DO2 - DO3 - END3 - END2 - END1
○	○	

Note: ○: Correct

- By specifying “GOTO <sequence number>”, it is possible to jump from the DO to END loop to a block outside the loop. However, jump into the DO to END loop by using “GOTO <sequence number>” is not possible.

DO1 to GOTO <sequence number> to END1 . . . N <sequence number>· · · ;	GOTO <sequence number> . . . DO1 to N <sequence number> to END1
○	×

DO1 to IF [<conditional expression>] GOTO <sequence number> to END1 . . . N <sequence number>· · · ;
○

Note: ○: Correct, ×: Incorrect

SUPPLE-MENT

For the execution of an operation instruction, operation stops or does not stop in the single block stop mode if the single block input (SBK) is ON according to the setting for parameter pm0007 D1.

pm0007 D1 = 0	Does not stop in the single-block mode.
pm0007 D1 = 1	Stops in the single-block mode.

4.4.6 Registering the Macroprogram

Macroprograms can be registered and edited in entirely the same manner as registering and editing normal NC programs and subprograms.

For this registration, there are no limits in the size of macroprograms; NC programs, subprograms and macroprograms can be stored to the limit of the memory capacity.
The program numbers to be used when registering macroprograms are classified as indicated in Table 4.70 according to their applications.

In addition to the classification indicated below, the program numbers to be used specially for macroprograms can be set to clearly identify them from NC programs and subprograms. Whether the program number range should be secured for macroprograms or not can be set by using a parameter.

Table 4.70 Classification of Macroprograms

Program No.	Classification by Applications	Protect
O1 to O7999	There are no restrictions on registration, deletion or editing.	
O8000 to O8999	Macroprograms can be protected from edit and display; edit protect and display protect can be set independently by using a parameter.	Protect 1
O9000 to O9999	Macroprograms can be protected from edit and display; edit protect and display protect can be set collectively by using a parameter.	Protect 2
O10000 to O99999	There are no restrictions on registration, deletion or editing.	

- To disable all of edit, input/output, and display of O8000 to O8999: pm0020 D0 = 1
- To disable edit and output/input and enable display of O8000 to O8999: pm0020 D0 = 0 and pm0021 D0 = 1
- To disable all of edit, input/output, and display of O9000 to O9999: pm3004 D0 = 1 and pm0022 D0 = 0
- To disable edit and output/input and enable display of O9000 to O9999: pm3004 D0 = 0 and pm0022 D0 = 1

For O9000 to O9999, an option is provided to disable all of edit, input/output, and display always disregarding of the setting for pm3004 and pm0022.

4.4.7 RS-232C Data Output 2 (BPRNT, DPRNT)

The macroprogram commands indicated below are possible in addition to the RS-232C data output 1 (see 4.4.3 (3), (k)). These commands are used to output variables and characters through the external device that has the RS-232C interface.

- Open command (POPEN)
- Data output command (BPRNT or DPRNT)
- Close command (PCLOS)

(1) Open Command (POEN)

POPEN [a] ;

a — RS-232C channel number

With the command indicated above, the DC2 control code is output from the NC. This command should be specified prior to the series of data output commands. For the RS-232C channel number, either "1" or "2" (option) can be specified; if no number is specified, No. 1 RS-232C channel is specified.

(Example)

POPEN; Opens the No. 1 RS-232C channel.
POPEN [2]; Opens the No. 2 RS-232C channel.

(2) Data Output Command (BPRNT or DPRNT)

(a) BPRNT

BPRNT [a #b [c] · · ·];

- c — Number of digits right to the decimal point
- b — Variable
- a — Character

With the commands indicated above, the character and the variable are output from the NC.

- Concerning characters, the specified characters are output in the ISO code. The following characters can be specified: alphabets (A to Z), numbers, and special characters (*, /, +, -). Note that "*" is output in a space code.
- Variable value is treated as the 2-word (32 bits) data with the number of digits right to the decimal point taken into consideration, and the value is output in the binary data from the higher byte. Since the values of all variables are saved with the decimal point, it is necessary to specify the number of effective digits right to the decimal point following the variable command in brackets. After the output of the command data, the EOB code is output in the ISO code.

(Example)
With the designation of "BPRNT [C**X#100 [3] Z#101 [3] M#10 [0]];", the following is output if the values of the variables are "#100 = 0.40956", "#101 = -1638.4" and "#10 = 12.34".

C3	A0	A0	D8	00 00 01 9A	5A	FF E7 00 00	4D	00 00 00 0C	0A	(HEX)
"C"	Space	Space	"X"	410	"Z"	-1638400	"M"	12	EOB	

SUPPLE-MENT

1. When outputting the data using the BPRNT command, it is not influenced by the following parameters.
   - Parameters other than pm0004, pm0006 D4 and D6, and pm0009
2. To use the BPRNT command, set the communication control parameters as NO for control code control and YES for RTS control.

(b) DPRNT

DPRNT [a #b [c d] · · ·];

- d: Number of digits right to the decimal point
- c: Number of digits left to the decimal point
- b: Variable
- a: Character

With the commands indicated above, the character and the variable are output from the NC.

- Concerning characters, the specified characters are output in the ISO code as with the BPRNT command.
- Concerning variable value, it is output in the ISO code digit by digit by the specified number of digits from the higher digit position. The decimal point is also output in the ISO code. To output the variable value, specify the variable number following the symbol of "#" and then specify the numbers of effective digits left and right to the decimal point individually in brackets. In this designation, the variable value is assumed to be a maximum of eight digits ($c + d \leqq 8$).

  If the specified number of digits right to the decimal point is not "0", the numeric value is always specified by the specified number of digits. If it is "0", the decimal point is not output.

pm4090 D2 = 0	Space code is output.
pm4090 D2 = 1	Nothing is output.

- After the output of the command data, the EOB code is output in the ISO code.
- The variable of < empty > is regarded as "0".

4

(Example)

With the designation of "DPRNT [X#2 [53] Z#5 [53] T#30 [40]];", the following is output if the values of the variables are "#2 = 128.47398", "#5 = -91.2" and "#30 = 1234.56".

- If the parameter setting is "to output space code" (pm4009 D2 = 0)

```
D8   A0 A0 A0 B1 B2 B8 2E B4 37 B4
          ---128.474
          X

5A   2D A0 A0 A0 39 B1 2E B2 30 30
          - ---91.200
          Z

D4   B1 B2 33 B4   OA
          EOB
          1234
          T
```

- If the parameter setting is "to output nothing" (pm4009 D2 = 1)

```
D8   B1 B2 B8 2E B4 37 B4
          128.474
          X

5A   2D 39 B1 2E B2 30 30
          -91.200
          Z

D4   B1 B2 33 B4   OA
          EOB
          1234
          T
```

(3) Close Command (PCLOS)

PCLOS [a];

└── RS-232C channel number

With the command indicated above, the DC4 control code is output from the NC. For the RS-232C channel number, either "1" or "2" can be specified; if no number is specified, No. 1 RS-232C channel is specified.

(Example)

PCLOS; Closes the No. 1 RS-232C channel.

PCLOS [2]; Closes the No. 2 RS-232C channel.

(4) Supplements to RS-232C Data Output 2

- To output the data using the BPRNT or DPRNT command, set "0" for pm0004 D5. If "1" is set for pm0004 D5, the data cannot be output correctly.

pm0004 D5 = 0	Parity bit output in tape punch in the ISO code
pm0004 D5 = 1	Parity bit not output in tape punch in the ISO code

- It is not necessary to specify the open command (POPEN) and the close command (PCLOS) continuously. Once the open command is executed, the channel remains open until the close command is specified next.
- If the command being output by the data output command is reset, processing stops and the succeeding data is lost. Therefore, if the NC is reset by the M30 command at the end of the program in which data is being output, specify the close command at the end of the program and execute the processing of such as the M30 command only after all data has been output.
- The open and close commands must always be specified in a pair. It is not allowed to specify the close command although the open command is not specified.

4

4.4.8 Macroprogram Alarm Numbers

Alarm numbers related with macroprograms and the cause of them are indicated in Table 4.71.

Table 4.71 Macroprogram Alarm Numbers

Alarm No.	Description	Alarm No.	Description
0210	CONSTANT DATA OUT OF RANGE In a macroprogram, specified constant is outside the allowable range.	0221	0 DIVIDE IN MACRO In a macroprogram, division by "0" is executed.
0211	UNMATCH G67 COMMAND The number of G67 commands is greater than the number of G65 and G66 commands.	0222	ROOT VALUE NEGATIVE A negative value is specified for square root operation.
0212	MACRO FORMAT ERROR There is an error in macroprogram format.	0223	FLOATING DATA OUT OF RANGE Floating point data exceed the allowable range.
0213	UNDEF INED # NO. A value not defined as a variable number is used.	0224	G66-M99 PROG ERROR An axis move command is specified with M99 in the modal call (G66) mode.
0214	ILL LEFT SIDE # NO. A variable that cannot be used is set in the left side of operation expression.	0225	MACRO SYSTEM ERROR Overflow with operation stack.
0215	[ ] LIMIT OVER Nesting level of brackets [ ] exceeds the limit.	0226	ASIN, ACOS, LN, SQRT ERROR The result of function operation (ASIN, ACOS, LN, SQRT) is outside the allowable range.
0216	MACRO CALL LIMIT OVER Macroprogram call up nesting level exceeds the limit.	0227	EXCHANG OVERFLOW Overflow during conversion into integer.
0217	DO-END FORMAT ERROR DO and END instructions are not specified in pairs.	0228	BCD INPUT DATA OVERRFLOW Overflow of input data for BCD function.
0218	[ ] UNMATCH The numbers of left bracket [and right bracket] do not match.	0229	BIN FORMAT ERROR There is an error in format with the BIN function.
0219	DO-END NO.OUT OF RANGE In "DO m" command, "m" is not in the range of $1 \leqq m \leqq 3$.	0230	EXP OUTPUT DATA OVERFLOW Overflow with the EXP function.
0220	GOTO NO.FORMAT ERROR The value of "n" in the "GOTO n" command is outside the allowable range, or the specified "n" is not found.		

4.4.9 Examples of Macroprograms

Some examples of macroprograms are explained below.

(1) Macroprogram in Thread Cutting Canned Cycle

(a) G92 straight thread cutting cycle

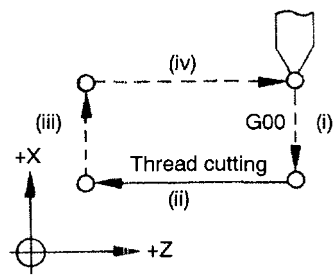
Example of Programming (P1)

G92 U-50. W-60. F6.0 ;

The commands indicated above are executed according to the following processing in the NC. The example is explained assuming that thread chamfering is not executed.

Example of Programming (P2)

(i) G00 U-50. ;
(ii) G32 W-60. F6.0 ;
(iii) G00 U-50. ;
(iv) G00 W-60. ;



① All axis move distances and thread lead should be replaced with variables (local variables: #1 to #33).

② For local variables, type I and type II variables are provided. When the number of local variables to be handled is small, it is recommended to use type I local variables which allow the use of U, W, and F, thus facilitating assigning of arguments.

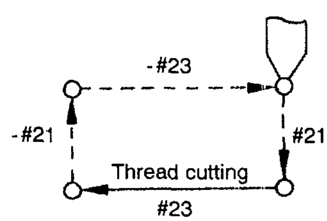
③ By using type I local variables, local variable are assigned to address characters as indicated below.

U-50.	W-60.	F-6.0
↓	↓	↓
#21	#23	#9

④ By using these variables, the example program (P2) can be written in the following manner.

Example of Programming (P3)

(i) G00 U#21 ;
(ii) G32 W#23 F#9 ;
(iii) G00 U-#21 ;
(iv) G00 W-#23 ;
(v) M99 ;



(b) Calling up a macroprogram by using G65

Example of Macroprogram Call Up Program (P4)

```
G65 P9093 U-50.
W-60. F6.0 ;
```

① Use the example program (P4) to call a macroprogram. Macroprogram body is “O9093”.

```
O9093 ;
G00 U#21 ;
G32 W-#23   F#9 ;
G00 U-#21 ;
G00 W-#23 ;
M99 ;
```

② With the macroprogram indicated above, it is necessary to specify the W-point and F-point levels for each execution of the macroprogram. Therefore, another macroprogram which specifies the position of the W-point and F-point levels should be written.

```
O9000 ;
#100 = #23 ;
#101 = #9 ;
M99 ;

O9093 ;
G00 U#21 ;
G32 W#100 F#101 ;
G00 U-#21 ;
G00 W-#100 ;
M99 ;
```

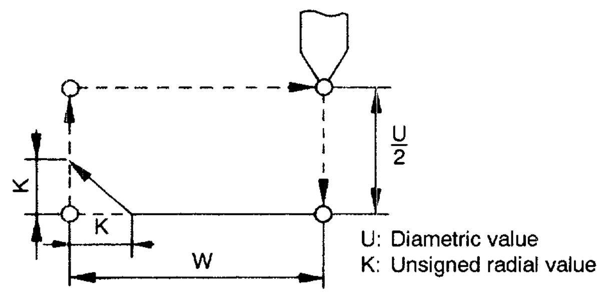
③ The program used to call two macroprograms indicated in item ② above is indicated below.

```
G65 P9000 W-60. F6.0 ;
G65 P9093 U-50. ;
G65 P9093 U-51.4 ;
G65 P9093 U-52.6 ;
G65 P9093 U · · · ;
```

4

(c) Example program for thread chamfering

Specify chamfering distance by address K.



Example of Thread Chamfering Program (P5)

- Example of Macro Program Call Command???

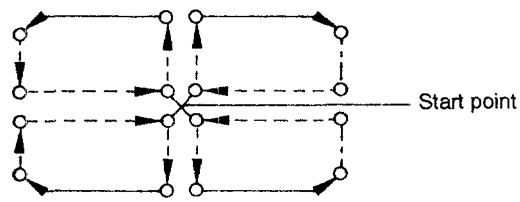
```
G65 P9000 W-60. K4.8 F6.0;
G65 P9093 U-50.;
G65 P9093 U-51.4;
G65 P9093 U ···;
```

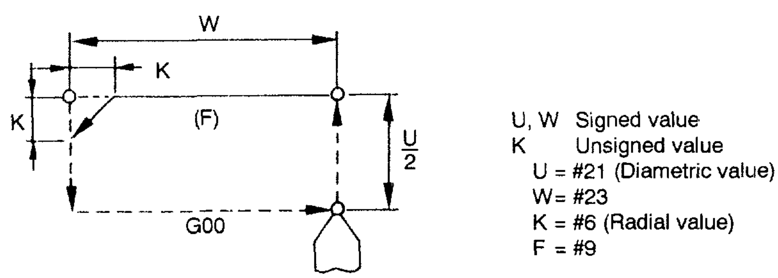
- Example of Macro Program Body

```
O9000;
#100 = #23;
#101 = #9;
#102 = ABS [#6]
M99;

O9093;
#10 = ROUND [#102] *2;
#11 = ROUND [#21] + #10;
#12 = ROUND [#100] + ROUND [#102];
G00 U#21;
G32 W#12 F#101;
G32 U#10 W-#102;   ←Thread chamfering
G00 U-#11;
G00 W-#100;
M99;
```

① Since this program does not include #3000 (single block invalid control) and #3004 (feed hold invalid control), the program is not protected satisfactorily from operation error. In addition, the thread cutting cycle is possible only in the U- and W- axis directions. It is necessary to edit the program so that thread cutting can be executed in four directions.





② An example of macroprogram call up program is indicated below.

```
G65 P9000 W-45. K4.0 F5.0 ;
G65 P9093 U40. ;
G65 P9093 U41.4 ;
G65 P9093 U · · · ;
```

③ An example of macroprogram body is indicated below.

```
O9000 ;
#100 = #23 ;        ← W
#101 = #9 ;         ← F
#102 = ABS [#6] ;     ← |K|
M99 ;
```

```
O9093 ;
#3003 = 1 ;                                   ← Single block invalid
M93 ;                                         ← 7-block buffering
#10 = ROUND [#102] *2 ;
IF [ABS_ [#21] LT#10] GOTO 4 ;
IF [#21 GT 0] GOTO 1 ;
IF [#21 EQ 0] GOTO 4 ;
#11 = ROUND [#21] + #10 ;                     ← If U value is negative
#12 = #10 ;
GOTO 2 ;
N1 #11 = ROUND [#21] -#10 ;                   ← If U value is positive
#12 = -#10 ;
N2 #13 = ROUND [#102] ;
IF [ABS [#100] LT#13] GOTO 4 ;
IF [#100GT0] GOTO 3 ;
IF [#100EQ0] GOTO 4 ;
#14 = ROUND [#100] + #13 ;                    ← If W value is negative
#15 = -#13 ;
GOTO 5 ;
N3 #14 = ROUND [#100] -#13 ;                  ← If W value is positive
#15 = #13
GOTO 5 ;
N4 #3000 = 499 (MACRO INPUT ERR.) ;           ← Error display
N5 G00 U#21 ;                   Feed hold
#3004 = 7 ;                   ← Feedrate override        Invalid
G32 W#14 F#101 ;                Positioning complete
G32 U#12 W#15 ; ← Thread chamfering
#3004 = 0 ;
G00 U-#11 ;
G00 W-#100 ;
M92 ;
#3003 = 0 ;
M99 ;
```

APPENDIX 1

A1

G CODE TABLE

Appendix 1 describes the G code and the functions.

APPENDIX 1.1 G CODE TABLE

Appendix Table 1.1 G Code Table

G Code	Group	Function	Refer to
G00 #1	01	Positioning (rapid traverse)	2.1.1
G01 #1		Linear interpolation	2.1.2
G02		Circular interpolation CW (radius R designation)	2.1.3
G03		Circular interpolation CCW (radius R designation)	2.1.3
G04	*	Dwell	3.3.1
G06		Positioning (error detect OFF)	2.1.1
G10		Programmable data input	4.2.1
G11	01	Chamfering	2.1.4
G12		Rounding	2.1.5
G20 #2	05	Input unit system designation (inch)	3.2.3
G21 #2		Input unit system designation (mm)	3.2.3
G22	05	Circular interpolation radius R designation CW	2.1.3
G23		Circular interpolation radius R designation CCW	2.1.3
G27	*	Reference point return check	2.3.2
G28		Automatic reference point return	2.3.1
G29		Return from reference point	2.3.3
G30		Second, third, and fourth reference point return	2.3.4
G31		Skip function	4.3.1
G32	01	Thread cutting, continuous thread cutting, multiple-thread cutting	2.2.1, 2.2.2
G34		Variable lead thread cutting	2.2.3
G36 #2	07	Stored stroke limit 2nd area ON	4.2.3
G37 #2		Stored stroke limit 2nd area OFF	4.2.3
G38 #2	08	Stored stroke limit 3rd area ON	4.2.3
G39 #2		Stored stroke limit 3rd area OFF	4.2.3
G40 #1	06	Nose R offset, cancel	3.4.3
G41		Nose R offset, left	3.4.3
G42		Nose R offset, right	3.4.3
G50 #1	*	Coordinate system setting, max. spindle speed setting, work-piece coordinate system setting	3.1.4
G51		Actual position display zero return	3.1.5
G65	*	Macroprogram simple call	4.4.1
G66	09	Macroprogram modal call	4.4.1
G67 #1		Macroprogram modal call, cancel	4.4.1

G Code	Group	Function	Refer to
G68	10	Programmable mirror image ON	4.2.4
G69		Programmable mirror image OFF	4.2.4
G70	*	Multiple canned cycle (finish)	4.1.2
G71		Multlple canned cycle (OD, rough)	4.1.2
G72		Multiple canned cycle (Face, rough)	4.1.2
G73		Multiple canned cycle (closed loop cutting)	4.1.2
G74		Multiple canned cycle (face, cut-off)	4.1.2
G75		Multiple canned cycle (OD, cut-off)	4.1.2
G76		Multiple canned cycle (automatic thread cutting)	4.1.2
G80 #1	14	Canned cycle, cancel	4.1.5
G81		Canned cycle (drilling)	4.1.5
G82		Canned cycle (spot facing)	4.1.5
G83		Canned cycle (high-speed deep hole drilling)	4.1.5
G84		Canned cycle (tapping)	4.1.5
G85		Canned cycle (boring)	4.1.5
G86		Canned cycle (boring)	4.1.5
G87		Canned cycle (back boring)	4.1.5
G88		Canned cycle (boring)	4.1.5
G89		Canned cycle (boring)	4.1.5
G90 #2	01	Cutting cycle A	4.1.1
G92		Thread cutting cycle	4.1.1
G94		Cutting cycle B	4.1.1
G96	02	Constant speed control	3.5.3
G97 #1		Constant speed control, cancel	3.5.3
G98 #2	04	Feed per minute (mm/min)	1.2.4
G99 #2		Feed per revolution (mm/rev)	1.2.3
G111	*	Multiple chamfering/rounding of taper	4.2.2
G112		Multiple chamfering/rounding of arc	4.1.3
G122	11	Start of tool registration	4.3.2
G123 #1		End of tool registration	4.3.2
G124	20	Cylindrical interpolation	2.1.6
G125 #1		Cylindrical interpolation cancel	2.1.6
G126	19	Polar coordinate mode ON	2.1.7
G127		Polar coordinate mode OFF	2.1.7
G132	22	Rotary tool S command mode	3.5.4
G133		Rotary tool S command mode cancel	3.5.4
G198 #2	15	Canned cycle (initial point level return)	4.1.5
G199		Canned cycle (R-point level return)	4.1.5

Note 1: The NC establishes the G code modes, identified by #1, when the power is turned ON or when the NC is reset.

2: The NC establishes the G code modes, identified by #2, when the power is turned ON.

APPENDIX 2

INDEX

A2

In Appendix 2, technical terms specific to NC and J300L are arranged in alphabetical order.

Please use this index when looking for descriptions using the technical term as the key code.

A2

YASNAC J300L

PROGRAMMING MANUAL

TOKYO OFFICE New Pier Takeshiba South Tower, 1-16-1, Kaigan, Minato-ku, Tokyo 105 Japan
Phone 81-3-5402-4511 Fax 81-3-5402-4580
YASKAWA ELECTRIC AMERICA, INC.
Chicago-Corporate Headquarters 2942 MacArthur Blvd. Northbrook, IL 60062-2028, U.S.A.
Phone 1-847-291-2340 Fax 1-847-498-2430
Chicago-Technical Center 3160 MacArthur Blvd. Northbrook, IL 60062-1917, U.S.A.
Phone 1-847-291-0411 Fax 1-847-291-1018
MOTOMAN INC.
805 Liberty Lane West Carrollton, OH 45449, U.S.A.
Phone 1-937-847-6200 Fax 1-937-847-6277
YASKAWA ELÉTRICO DO BRASIL COMÉRCIO LTDA.
Avenida Brigadeiro Faria Lima 1664-5° CJ 504/511, São Paulo, Brazil
Phone 55-11-815-7723 Fax 55-11-870-3849
YASKAWA ELECTRIC EUROPE GmbH
Am Kronberger Hang 2, 65824 Schwalbach, Germany
Phone 49-6196-569-300 Fax 49-6196-888-301
Motoman Robotics AB
Box 504 S38525 Torsås, Sweden
Phone 46-486-48800 Fax 46-486-41410
Motoman Robotec GmbH
Kammerfeldstraße 1, 85391 Allershausen, Germany
Phone 49-8166-900 Fax 49-8166-9039
YASKAWA ELECTRIC UK LTD.
Unit2 Centurion Court Brick Close, Kiln Farm, Milton Keynes MK11 3JA, United Kingdom
Phone 44-1908-565874 Fax 44-1908-565891
YASKAWA ELECTRIC KOREA CORPORATION
Paik Nam Bldg. 901 188-3, 1-Ga Euljiro, Joong-Gu Seoul, Korea
Phone 82-2-776-7844 Fax 82-2-753-2639
YASKAWA ELECTRIC (SINGAPORE) PTE. LTD.
151 Lorong Chuan, #04-01, New Tech Park Singapore 556741, Singapore
Phone 65-282-3003 Fax 65-289-3003
YATEC ENGINEERING CORPORATION
Shen Hsiang Tang Sung Chiang Building 10F 146 Sung Chiang Road, Taipei, Taiwan
Phone 886-2-563-0010 Fax 886-2-567-4677
BEIJING OFFICE Room No. 301 Office Building of Beijing International Club, 21
Jianguomenwai Avenue, Beijing 100020, China
Phone 86-10-532-1850 Fax 86-10-532-1851
SHANGHAI OFFICE 27 Hui He Road Shanghai 200437 China
Phone 86-21-6553-6060 Fax 86-21-6553-6060
YASKAWA JASON (HK) COMPANY LIMITED
Rm. 2909-10, Hong Kong Plaza, 186-191 Connaught Road West, Hong Kong
Phone 852-2803-2385 Fax 852-2547-5773
TAIPEI OFFICE Shen Hsiang Tang Sung Chiang Building 10F 146 Sung Chiang Road, Taipei, Taiwan
Phone 886-2-563-0010 Fax 886-2-567-4677

YASKAWA ELECTRIC CORPORATION

YASKAWA

MANUAL NO. TOE-C843-13.21