CNC SYSTEM

OSP-P200L/P20L
OSP-P200L-R/P20L-R

PROGRAMMING MANUAL

(3rd Edition)

Pub No. 5238-E-R2 (LE33-013-R3) Aug. 2007

OKUMA

SAFETY PRECAUTIONS

The machine is equipped with safety devices which serve to protect personnel and the machine itself from hazards arising from unforeseen accidents. However, operators must not rely exclusively on these safety devices: they must also become fully familiar with the safety guidelines presented below to ensure accident-free operation.
This instruction manual and the warning signs attached to the machine cover only those hazards which Okuma can predict. Be aware that they do not cover all possible hazards.

1. Precautions Relating to Installation

(1) Please be noted about a primary power supply as follows.

- Do not draw the primary power supply from a distribution panel that also supplies a major noise source (for example, an electric welder or electric discharge machine) since this could cause malfunction of the CNC unit.
- If possible, connect the machine to a ground not used by any other equipment. If there is no choice but to use a common ground, the other equipment must not generate a large amount of noise (such as an electric welder or electric discharge machine).

(2) Installation Environment
Observe the following points when installing the control enclosure.

- Make sure that the CNC unit will not be subject to direct sunlight.
- Make sure that the control enclosure will not be splashed with chips, water, or oil.
- Make sure that the control enclosure and operation panel are not subject to excessive vibrations or shock.
- The permissible ambient temperature range for the control enclosure is 5 to 40°C.
- The permissible ambient humidity range for the control enclosure is relative humidity 50% or less at 40°C (no condensation).
- The maximum altitude at which the control enclosure can be used is 1000 m (3281ft.).

2. Points to Check before Turning on the Power

(1) Close all the doors of the control enclosure and operation panel to prevent the entry of water, chips, and dust.

(2) Make absolutely sure that there is nobody near the moving parts of the machine, and that there are no obstacles around the machine, before starting machine operation.

(3) When turning on the power, turn on the main power disconnect switch first, then the CONTROL ON switch on the operation panel.

3. Precautions Relating to Manual/Continuous Operation

(1) Follow the instruction manual during operation.

(2) Do not operate the machine with the front cover, chuck cover, or another protective cover removed.

(3) Close the front cover before starting the machine.

(4) When machining the initial workpiece, check for machine operations, run the machine under no load to check for interference among components, cut the workpiece in the single block mode, and then start continuous operation.

(5) Ensure your safety before rotating the spindle or moving a machine part.

(6) Do not touch chips or workpiece while the spindle is rotating.

(7) Do not stop a rotating part with hand or another means.

(8) Check that the condition of hydraulic chuck jaws as mounted, operating pressure, and maximum permissible revolving speed.

(9) Check the condition and location of the cutting tool as mounted.

(10) Check the tool offset value.

(11) Check the zero offset value.

(12) Check that the SPINDLE OVERRIDE and FEEDRATE OVERRIDE dials on the NC operation panel are set to 100%.

(13) When moving the turret, check the software limits for X- and Z-axes or the locations of limit switch dogs to prevent interference with the chuck and tailstock.

(14) Check the location of the turret.

(15) Check the location of the tailstock.

(16) Cut workpieces with a transmitted power and torque within the permissible range.

(17) Chuck each workpiece firmly.

(18) Check that the coolant nozzle is properly located.

4. On Finishing Work

(1) On finishing work, clean the vicinity of the machine.

(2) Return the ATC, APC and other equipment to the predetermined retraction position.

(3) Always turn off the power to the machine before leaving it.

(4) To turn off the power, turn off the CONTROL ON switch on the operation panel first, then the main power disconnect switch.

5. Precautions during Maintenance Inspection and When Trouble Occurs

In order to prevent unforeseen accidents, damage to the machine, etc., it is essential to observe the following points when performing maitenance inspections or during checking when trouble has occurred.

(1) When trouble occurs, press the emergency stop button on the operation panel to stop the machine.

(2) Consult the person responsible for maintenance to determine what corrective measures need to be taken.

(3) If two or more persons must work together, establish signals so that they can communicate to confirm safety before proceeding to each new step.

(4) Use only the specified replacement parts and fuses.

(5) Always turn the power off before starting inspection or changing parts.

(6) When parts are removed during inspection or repair work, always replace them as they were and secure them properly with their screws, etc.

(7) When carrying out inspections in which measuring instruments are used - for example voltage checks - make sure the instrument is properly calibrated.

(8) Do not keep combustible materials or metals inside the control enclosure or terminal box.

(9) Check that cables and wires are free of damage: damaged cables and wires will cause current leakage and electric shocks.

(10) Maintenance inside the Control Enclosure

a. Switch the main power disconnect switch OFF before opening the control enclosure door.

b. Even when the main power disconnect switch is OFF, there may some residual charge in the MCS drive unit (servo/spindle), and for this reason only service personnel are permitted to perform any work on this unit. Even then, they must observe the following precautions.

- MCS drive unit (servo/spindle)
  The residual voltage discharges two minutes after the main switch is turned OFF.

c. The control enclosure contains the NC unit, and the NC unit has a printed circuit board whose memory stores the machining programs, parameters, etc. In order to ensure that the contents of this memory will be retained even when the power is switched off, the memory is supplied with power by a battery. Depending on how the printed circuit boards are handled, the contents of the memory may be destroyed and for this reason only service personnel should handle these boards.

(11) Periodic Inspection of the Control Enclosure

a. Cleaning the cooling unit
The cooling unit in the door of the control enclosure serves to prevent excessive temperature rise inside the control enclosure and increase the reliability of the NC unit. Inspect the following points every three months.

- Is the fan motor inside the cooling unit working?
  The motor is normal if there is a strong draft from the unit.

- Is the external air inlet blocked?
  If it is blocked, clean it with compressed air.

6. General Precautions

(1) Keep the vicinity of the machine clean and tidy.

(2) Wear appropriate clothing while working, and follow the instructions of someone with sufficient training.

(3) Make sure that your clothes and hair cannot become entangled in the machine. Machine operators must wear safety equipment such as safety shoes and goggles.

(4) Machine operators must read the instruction manual carefully and make sure of the correct procedure before operating the machine.

(5) Memorize the position of the emergency stop button so that you can press it immediately at any time and from any position.

(6) Do not access the inside of the control panel, transformer, motor, etc., since they contain high-voltage terminals and other components which are extremely dangerous.

(7) If two or more persons must work together, establish signals so that they can communicate to confirm safety before proceeding to each new step.

7. Symbols Used in This Manual

The following warning indications are used in this manual to draw attention to information of particular importance. Read the instructions marked with these symbols carefully and follow them.

DANGER

indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

WARNING

indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.

CAUTION

indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

CAUTION

indicates a potentially hazardous situation which, if not avoided, may result in damage to your property.

SAFETY INSTRUCTIONS

indicates general instructions for safe operation.

INTRODUCTION

Thank you very much for purchasing our numerical control unit.
Before using this NC unit (hereafter simply called NC), thoroughly read this programming manual (hereafter called this manual) in order to ensure correct use.
This manual explains how to use and maintain the NC so that it will deliver its full performance and maintain accuracy over a long term.
You must pay particular attention to the cautions given in this manual, read them carefully, and make sure you fully understand them before operating the NC.

Display Screens

The NC display screens vary with the selected NC specifications. The screens shown in this manual, therefore, may not exactly the same with those displayed on your NC.

TABLE OF CONTENTS

SECTION 1 PROGRAM CONFIGURATIONS

1. Program Types

For OSP-P200L, three kinds of programs are used: schedule programs, main programs, and subprograms. The following briefly explains these three kinds of programs.

Schedule Program

When more than one type of workpiece is machined in continuous operation using a bar feeder or other equipment, multiple main programs are used. A schedule program is used to specify the order in which the main programs are executed and the number of times the individual main program is executed. Using a schedule program makes it possible to carry out untended operation easily.
It is not necessary to assign a program name. The END code must be specified at the end of a schedule program. For details, refer to SECTION 14, "SCHEDULE PROGRAMS".

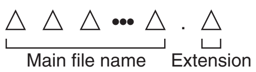
Main Program

A main program contains a series of commands to machine one type of workpiece. Subprograms can be called from a main program to simplify programming.
A main program begins with a program name which begins with address character "O" and ends with M02 or M30.

Subprogram

A subprogram can be called from a main program or another subprogram. There are two types of subprograms: those written and supplied by Okuma (maker subprogram), and those written by the customer (user subprogram).
The program name, which must start with "O", is required at the beginning of the subprogram. The RTS command must be specified at the end of the subprogram. For details, refer to SECTION 13, USER TASK.

- Program file format
  Main file name: Max. 16 alphanumeric characters starting with an alphabet
  Extension: Max. 3 alphabetic characters



LE33013R0300300010001

- Extensions

  SDF : Schedule program file
  MIN : Main program file
  SSB : System subprogram file
  SUB : User subprogram file

2. Program Name

With the OSP-P200L, programs are called and executed by designating the program name or program number assigned to the beginning of individual programs. This simplifies programs.
A program name that contains only numbers is called a program number.

Program Name Designation

- Enter letters of the alphabet (A to Z) or numbers (0 to 9) following address character "O". Note that no space is allowed between "O" and a letter of the alphabet or a number. Similarly, no space is allowed between letters of the alphabet and numbers.
- Up to four characters can be used.
- An alphabetic character can only be used in a program name if it begins with an alphabetic character. Although a program beginning with an alphabetic character can contain a number in it, one that begins with a number cannot contain an alphabetic character.
- A block which contains a program name must not contain other commands.
- A program name may not be used for a schedule program.
- The program name assigned to a subprogram must begin with address character "O", but this is not mandatory for main programs.
- Since program names are handled in units of characters, the following names are judged to be different program names.
    - O0123 and O123
    - O00 and O0
- Do not assign the same name to more than one program, otherwise it will not be possible to select the intended program.

3. Sequence Name

All blocks in a program are assigned a sequence name that begins with address character "N" followed by an alphanumeric sequence.
Functions such as a sequence search function, a sequence stop function and a branching function can be used for blocks assigned a sequence name.
A sequence name that contains only numbers is called a sequence number.

Sequence Name Designation

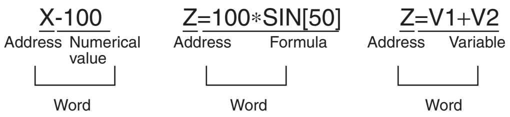
- Enter letters of the alphabet (A to Z) or numbers (0 to 9) following address character "N".
- Up to four characters can be used.
- Both alphabetic characters and numbers may be used in a sequence name. If an alphabetic character is used in a sequence name, however, the sequence name must begin with an alphabetic character.
- A sequence name must be placed at the top of block. However, a block delete command may be placed preceding a sequence name.
- Sequence numbers may be specified in any order. They can be used however desired, provided there is no duplication of numbers.
- Since sequence names are handled in units of characters, the following names are judged to be different sequence names.
    - N0123 and N123
    - N00 and N0
- When a sequence name is used, place a space or a tab after the sequence name.

4. Program Format

4-1. Word Configuration

A word is defined as an address character followed by a group of numeric values, an expression, or a variable name. If a word consists of an expression or a variable, the address character must be followed by an equal sign "=".

Examples:



LE33013R0300300040001

- An address character is one of the alphabetic characters A through Z and defines the meaning of the entry specified following it. In addition, an extended address character, consisting of two alphabetic characters, may also be used.

- Refer to SECTION 13, 3-2. "Variables" for more information on variables.

4-2. Block Configuration

A group consisting of several words is called a block, and a block expresses a command. Blocks are delimited by an end of block code.

- The end of block code differs depending on the selected code system, ISO or EIA:
  ISO: "LF"
  EIA: "CR"

- A block may contain up to 158 characters.

4-3. Program

A program consists of several blocks.

4-4. Programmable Range of Address Characters

Address	Function	Programmable Range		Remarks
		Metric	Inch	
O	Program name	0000 to 9999	same as left	Alphabetic characters available
N	Sequence name	0000 to 9999	same as left	
G	Preparatory function	0 to 999	same as left	
X, Z	Coordinate values (linear axis)	±99999.999 mm	±9999.9999 inch	
C	Coordinate values (rotary axis)	±359.999 deg.	±359.999 deg.	
I, K	Coordinate values of center of arc Taper amount and depth of cut in fixed thread cutting cycle Shift amount in grooving cycle	±99999.999 mm	±9999.9999 inch	
D, U, W, H, L	Automatic programming commands	0 to 99999.999 mm	0 to 9999.9999 inch	
E		±99999.999 mm/rev	±9999.9999 inch/rev	
A, B		0 to 99999.999 deg.	0 to 9999.9999 deg.	
F	Feedrate per revolution	0.001 to 99999.999 mm/rev	0.0001 to 999.9999 inch/rev	
	Feedrate per minute	0.001 to 99999.999 mm/min	0.0001 to 9999.9999 inch/min	
	Dwell time period	0.01 to 9999.99 sec	same as left	
T	Tool number	6 digits 4 digits	same as left	6 digits (with nose R compensation) 4 digits (without nose R compensation)
S SB	Spindle speed M-tool speed	0 to 9999 0 to 9999	same as left	
M	Miscellaneous function	0 to 511	same as left	
QA	C-axis revolution	1 to 1999 (rev.)	same as left	
SA	C-axis speed	0.001 to 20.000 min^{-1}	same as left	

5. Mathematical Operation Functions

Mathematical operation functions are used to convey logical operations, arithmetic operations, and trigonometric functions. A table of the operation symbols is shown below. Operation functions can be used together with variables to control peripherals or to pass on the results of an operation.

Category	Operation	Operator	Remarks
Logical operation	Exclusive OR	EOR	0110 = 1010 EOR 1100 (See *3.)
	Logical OR	OR	1110 = 1010 OR 1100
	Logical AND	AND	1000 = 1010 AND 1100
	Negation	NOT	1010 = NOT 0101
Arithmetic operation	Addition	+	8 = 5 + 3
	Subtraction	-	2 = 5 - 3
	Multiplication	*	15 = 5 * 3
	Division	/ (slash)	3 = 15/5
Trigonometric functions, etc.	Sine	SIN	0.5 = SIN [30] (See *4.)
	Cosine	COS	0.5 = COS [60]
	Tangent	TAN	1 = TAN [45]
	Arctangent (1)	ATAN	45 = ATAN [1] (value range: -90 to 90)
	Arctangent (2)	ATAN2	30 = ATAN 2 [1,(Square root 3)] (See *1.)
	Square root	SQRT	4 = SQRT [16]
	Absolute value	ABS	3 = ABS [-3]
	Decimal to binary conversion	BIN	25 = BIN [$25] ($ represents a hexadecimal number.)
	Binary to decimal conversion	BCD	$25 = BCD [25]
	Integer implementation (rounding)	ROUND	128 = ROUND [1.2763 x 102]
	Integer implementation (truncation)	FIX	127 = FIX [1.2763 x 102]
	Integer implementation (raising)	FUP	128 = FUP [1.2763 x 102]
	Unit integer implementation (rounding)	DROUND	13.265 = DROUND [13.26462] (See *2.)
	Unit integer implementation (truncation)	DFIX	13.264 = DFIX [13.26462] (See *2.)
	Unit integer implementation (raising)	DFUP	13.265 = DFUP [13.26462] (See *2.)
	Remainder	MOD	2 = MOD [17, 5]
Brackets	Opening bracket	[	Determines the priority of an operation. (Operations inside the bracket are performed first.)
	Closing bracket	]	

*1. The value of ATAN2 [b, a] is an argument (range: -180 to 180) of the point that is expressed by coordinate values (a, b).

*2. In this example, the setting unit is mm.

*3. Blanks must be placed before and after the logical operation symbols (EOR, OR, AND, NOT).

*4. Numbers after function operation symbols (SIN, COS, TAN, etc.) must be enclosed in brackets "[ ]". ("a", "b", and "c" are used to indicate the contents of the corresponding bits.)

Logical Operations

"a", "b", and "c" represent corresponding bits.

- Exclusive OR (EOR) c = a ␣ EOR ␣ b

LE33013R0300300080001

If the two corresponding values agree, EOR outputs 0.
If the two values do not agree, EOR outputs 1.

a	b	c
0	0	0
0	1	1
1	0	1
1	1	0

- Logical OR (OR) c = a ␣ OR ␣ b

LE33013R0300300080002

If both corresponding values are 0, OR outputs 0.
If not, OR outputs 1.

a	b	c
0	0	0
0	1	1
1	0	1
1	1	1

- Logical AND (AND) c = a ␣ AND ␣ b

LE33013R0300300080003

If both corresponding values are 1, AND outputs 1.
If not, AND outputs 0.

a	b	c
0	0	0
0	1	0
1	0	0
1	1	1

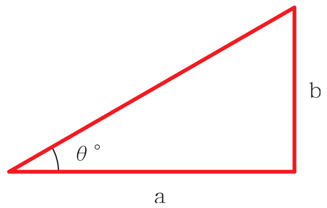
- Negation (NOT) b = NOT ␣ a

LE33013R0300300080004

NOT inverts the value (from 0 to 1, and 1 to 0).

a	b
0	1
1	0

- Arc tangent (1) (ATAN)



LE33013R0300300080005

θ = ATAN [b/a]
Arc tangent (2) (ATAN2)
θ = ATAN2 [b/a]

- Integer implementation (ROUND, FIX, FUP)
  Converts a specified value into an integer by rounding off, truncating, or raising the number at the first place to the right of the decimal point.
  (in units of microns)

6. Block Delete

[Function]
This function allows the operator to specify whether specific blocks are executed or ignored in automatic mode operation.
Blocks preceded by "/" are ignored during automatic mode operation if the BLOCK DELETE switch on the machine operation panel is set on. If the switch is off, the blocks are executed normally.
When the block skip function is activated, the entire block is ignored.

[Supplement]

- The slash "/" code must be placed at either the start of a block or immediately after a sequence name (number). If it is placed in another position in a block, it will cause an alarm.
- The slash "/" may not be contained in the program name block.
- Blocks which contain a "/" code are also subject to the sequence search function, regardless of the BLOCK DELETE switch position.
- The block delete function is not possible during SINGLE BLOCK mode. The succeeding block is executed, and then the operation stops.

7. Comment Function (CONTROL OUT/IN)

A program may be made easier to understand by using comments in parentheses.

- Comments must be parenthesized to distinguish them from general operation information.

- Comments are also subject to TV and TH checks.

Example:

N100 G00 X200 $\underbrace{\text{(FIRST STEP)}}_{\text{Comment}}$

LE33013R0300300100001

8. Program Storage Memory Capacity

The NC uses memory to store machining programs. The memory capacity is selectable depending on the size of the machining program. For execution, a program is transferred from the memory to the operation buffer (RAM). The capacity of the operation buffer is indicated by one program capacity.
If the size of the program to be executed is large, it is necessary to expand the one program capacity. The one program capacity can be selected from 320 m (1049.92 ft), 640 m (2099.84 ft.), 1280 m (4199.68 ft.), to expand program storage capacity.

9. Variable Limits

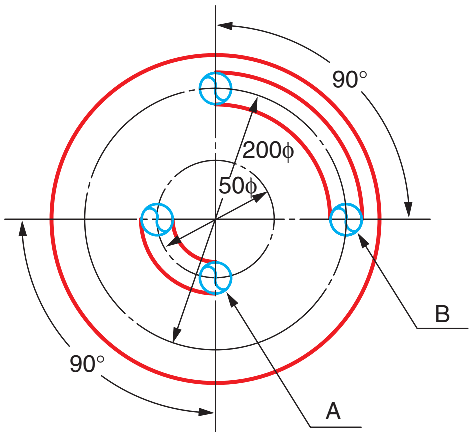
On execution of a command that specifies axis movement to a target point beyond the variable limit in the positive direction, the specified target point is replaced with the variable limit in the positive direction.
For commands specifying axis movement to a target point beyond the variable limit in the negative direction, axis movement is not executed and an alarm occurs.

10. Determining Feedrate for Cutting along C-Axis

10-1. Cutting by Controlling the C-axis Only

Although it is possible to machine a workpiece by controlling the C-axis, tool movement distance in unit time (one minute) differs according to the diameter of the position to be machined because the feedrate is specified in units of deg/min. This must be taken into consideration when making a program.
[Memo]
To match the unit of the C-axis feed command with the X- and/or Z-axis command, the feedrate command (F) should be calculated by converting 360 into 500 mm. This conversion should also be carried out when only a C-axis command is given.

Example:



Axis movement distance along slot A: ············$\pi \times 50/4 \fallingdotseq 39$ mm

Axis movement distance along slot B: ············$\pi \times 200/4 \fallingdotseq 156$ mm

Therefore, if cutting is carried out at a feedrate of 100 mm per minute, the feedrate (deg/min) of the C-axis is calculated as follows:

Along slot A(deg/min)············$100/39 \times 90 \fallingdotseq 230$

Along slot B(deg/min)············$100/156 \times 90 \fallingdotseq 58$

Convert the unit of feed from "deg/min" into "mm/min".

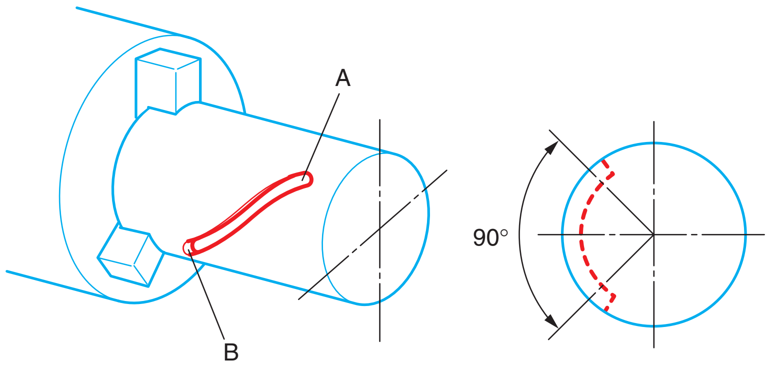
Slot A: (mm/min)············$230/360 \times 500 \fallingdotseq 320$ (F320)

Slot B: (mm/min)············$58/360 \times 500 \fallingdotseq 80$ (F80)

LE33013R0300300130001

10-2. Cutting by Controlling Both C-axis and Z-axis Simultaneously

Example:



Point A coordinate value	X = 80	
	Z = 100	
	C = 120	
Point B coordinate value	X = 80	
	Z = 50	
	C = 210	

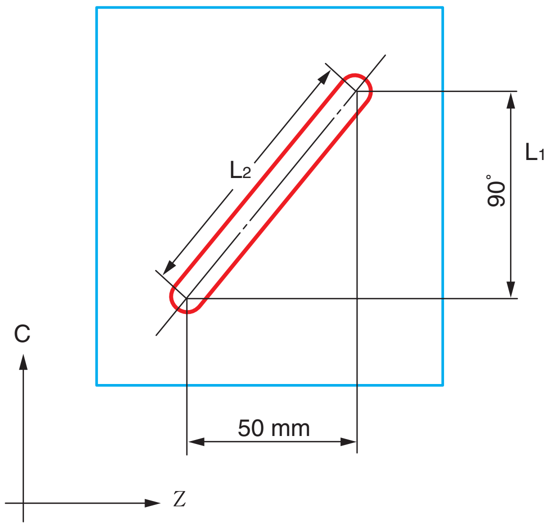
LE33013R0300300140001

When cutting the spiral from A to B with a two-flute end mill under the following cutting conditions, calculate the feedrate of C-axis as explained below:

Cutting conditions:	Feed per tooth	0.05 mm
	M-tool speed	400 min^{-1}

Procedure :

1 Calculate the distance between A and B.
A development of the diagram above is indicated below.



The distance, L_1, along the circumference is:

$$L_1 = 80 \times \pi \times \frac{90}{360} \fallingdotseq 63 \text{ (mm)}$$

The distance, L_2, between A and B is:

$$L_2 = \sqrt{63^2 + 50^2} \fallingdotseq 80 \text{ (mm)}$$

LE33013R0300300140002

2 Calculate the cutting time, T, on the basis of the cutting conditions indicated above to feed the axes along the slot.

$$T = \frac{L_2}{\text{(Feed per tooth) x (Number of teeth) x (min}^{-1}\text{)}}$$

$$= \frac{80}{0.05 \times 2 \times 400}$$

$$= 2 \text{ (min)}$$

LE33013R0300300140003

3 Inside the computer, the distance L3 between A and B is calculated in the following manner.

X-axis travel = 50 mm

C-axis travel = $90° \times \dfrac{500 \text{ mm}}{360°} = 125$ mm

(conversion based on 360° = 500 mm)

Therefore, the distance between A and B is calculated as below:

$$L_3 = \sqrt{50^2 + 125^2}$$

$$\fallingdotseq 135 \text{ (mm)}$$

LE33013R0300300140004

4 The feedrate to be specified in the program is approximately calculated as below:

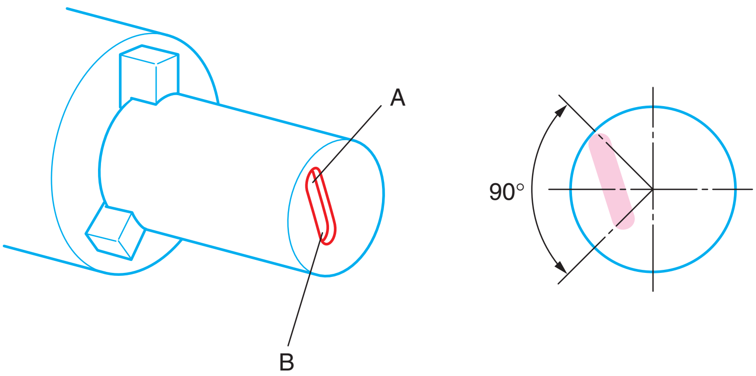
$$F = \frac{L_3}{T} = \frac{135}{2} = 67.5$$

LE33013R0300300140005

Specify F67.5 in the program.

10-3. Cutting by Controlling Both C-axis and X-axis Simultaneously

Example:



Point A coordinate value X = 80
Z = 100
C = 120

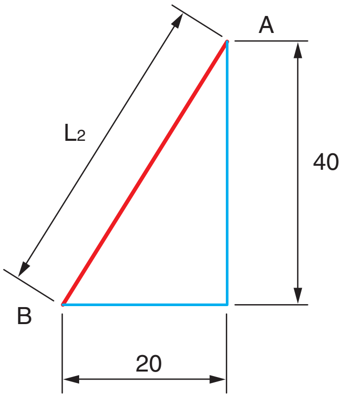
Point B coordinate value X = 40
Z = 100
C = 210

LE33013R0300300150001

- The cutting conditions are the same as used in "Cutting by Controlling Both C-axis and Z-axis Simultaneously".

Procedure :

1 Calculate the distance between A and B.



$$L_2 = \sqrt{40^2+20^2}$$
$$\fallingdotseq 44.7 \text{ mm}$$

LE33013R0300300150002

2 Calculate the cutting time, T, on the basis of the cutting conditions indicated above to feed the axes along the slot.

$$T = \frac{L_2}{(\text{Feed per tooth}) \times (\text{Number of teeth}) \times (\text{min}^{-1})}$$
$$= \frac{44.7}{0.05 \times 2 \times 400}$$
$$\fallingdotseq 1.12 \text{ min}$$

LE33013R0300300150003

3 Inside the computer, the distance L3 between A and B is calculated in the following manner.

X-axis travel = 40 mm

C-axis travel = $90° \times \dfrac{500 \text{ mm}}{360°}$ =125 mm

(conversion based on 360° = 500 mm)

Therefore, the distance between A and B is calculated as below:

$$L_3 = \sqrt{40^2 + 125^2}$$
$$\fallingdotseq 131.2 \text{ mm}$$

LE33013R0300300150004

4 The feedrate to be specified in the program is approximately calculated as below:

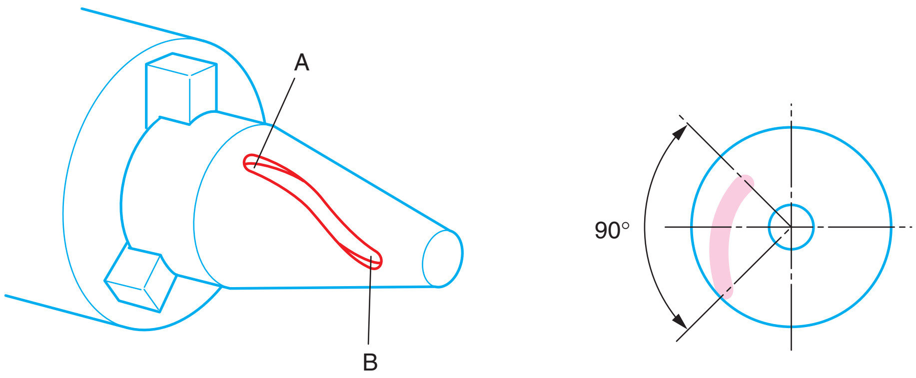
$$F = \frac{L_3}{T} = \frac{131.2}{1.12} = 117$$

LE33013R0300300150005

Specify F117 in the program.

10-4. Cutting by Simultaneous 3-axis Control of X-, Z-, and C-axis

Example:



Point A coordinate value	X = 80	Point B coordinate value	X = 40
	Z = 50		Z = 100
	C = 120		C = 210

LE33013R0300300160001

- When cutting a slot on a cone as indicated above, simultaneous 3-axis control of the X-, Z-, and C-axis becomes necessary. The feedrate to be programmed should be calculated in the following manner. Note that the example below assumes the same cutting conditions as in 11-2. "Cutting by Controlling Both C-axis and X-axis Simultaneously".

Procedure :

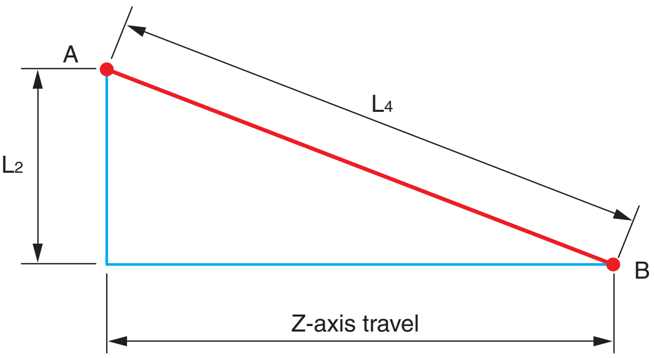
1 First, consider the development of the slot on the C-axis and X-axis. In this case, calculation of the feedrate is possible in the same manner as in "Cutting by Controlling Both C-axis and X-axis Simultaneously" .
The C and X-axis travel component, L2, is:

$$L_3 = \sqrt{40^2 + 20^2}$$

$$\fallingdotseq 44.7 \text{ mm}$$

LE33013R0300300160002

2 Calculate the actual distance between A and B from L2 calculated in (1).



$$L_4 = \sqrt{44.7^2 + 50^2}$$
$$\fallingdotseq 67.1$$

LE33013R0300300160003

3 Calculate the cutting time T for distance L4:

$$T = \frac{L_4}{(\text{Feed per tooth}) \times (\text{Number of teeth}) \times (\text{min}^{-1})}$$

$$= \frac{67.1}{0.05 \times 2 \times 400}$$

$$\fallingdotseq 1.68 \text{ min}$$

LE33013R0300300160004

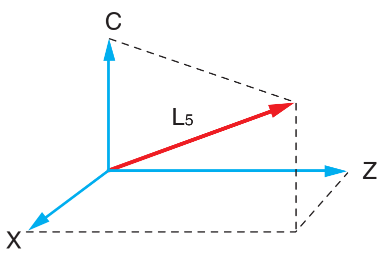
4 Inside the computer, distance L5 between A and B is calculated in the following manner.

X-axis travel = 40 mm

Z-axis travel = 50 mm

C-axis travel = $90 \times \frac{500 \text{ mm}}{365} = 125$ mm

(conversion based on 360 = 500 mm)



$$L_5 = \sqrt{40^2+50^2+125^2}$$
$$\fallingdotseq 140.4 \text{ mm}$$

LE33013R0300300160005

5 The feedrate to be specified in the program is approximately calculated as below:

$$F = \frac{L_5}{T} = \frac{140.4}{1.68} \fallingdotseq 83.6$$

LE33013R0300300160006

Specify F83.6 in the program.

SECTION 2 COORDINATE SYSTEMS AND COMMANDS

1. Coordinate Systems

1-1. Coordinate Systems and Values

To move the tool to a target position, the reference coordinate system must be set first to define the target position, and the target position is defined by coordinate values in the set coordinate system. There are the three types of coordinate system indicated below. A program coordinate system is used for programming.

- Encoder coordinate system
- Machine coordinate system
- Program coordinate system

1-2. Encoder Coordinate System

An encoder is used to detect the position of a numerically controlled axis. The encoder coordinate system is established based on the position data output by the encoder.
The position data directly output from the encoder is not displayed on the screen, and this coordinate system may be disregarded in daily operation.

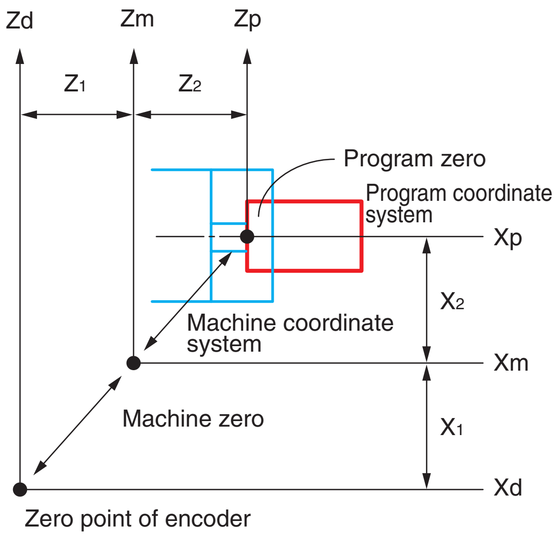
1-3. Machine Coordinate System

The reference point in the machine is referred to as the machine zero and the coordinate system which has its origin at the machine zero is called the machine coordinate system. The machine zero is set for each individual machine using system parameters and it is not necessary to change the setting after the installation of the machine.
If "0" is set for the encoder zero point offset (system parameter), the machine coordinate system agrees with the encoder coordinate system.

1-4. Program Coordinate System

The coordinate system used as the reference for program commands is called the program coordinate system.
The position of the origin of the program coordinate system varies according to the kind of workpieces to be machined and the origin is set at the required position by setting the zero offset data. The program coordinate system used for machining a specific kind of workpiece is thus defined based on the set origin.

Although the origin of a program coordinate system (program zero) can be set at any position, it is usually set on the centerline of a workpiece for the X-axis and at the left end face of workpiece for the Z-axis.



Xd, Zd : Output value of position encoder
(0: Zero point of position encoder)

Xm, Zm : Coordinate values in the machine coordinate system
(0: Machine zero)

Xp, Zp : Coordinate values in the program coordinate system
(0: Program zero)

X1, Z1 : Offset amount of position encoder

X1, Z1 : Offset amount of position encoder

LE33013R0300400040001

2. Coordinate Commands

2-1. Controlled Axis

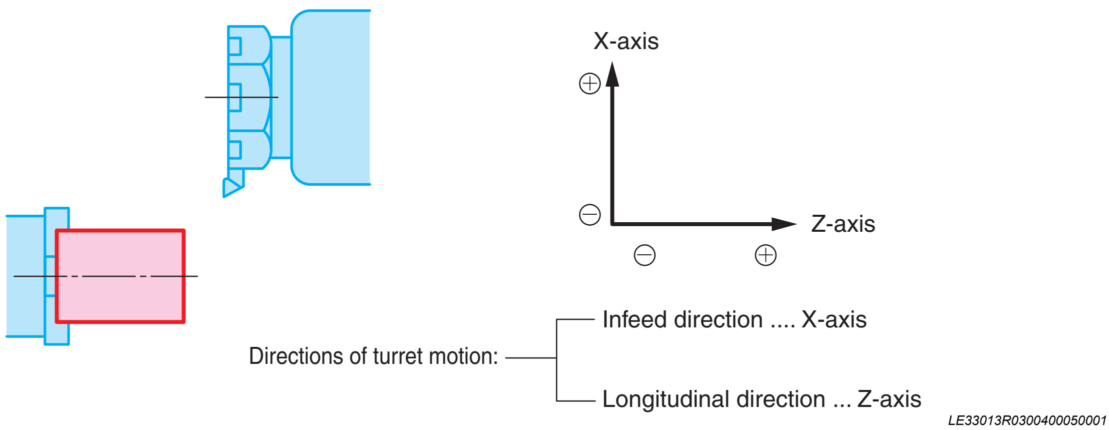
- The following table lists the addresses necessary for axis control.

	Address	Contents
Linear axis	X	Controlled axis in the direction parallel to the workpiece end face
	Z	Controlled axis in the direction parallel to the workpiece longitudinal direction.
Rotary axis	C	Rotary axis in a plane orthogonal to Z-axis

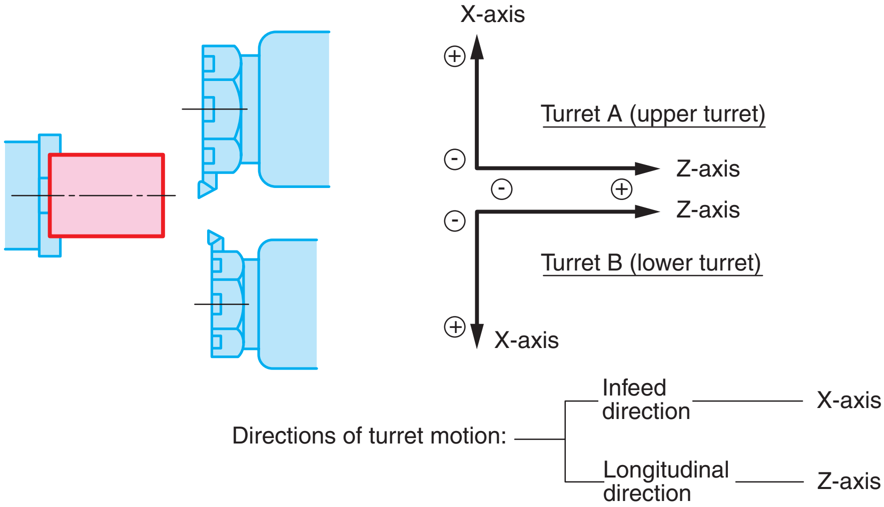
- A command used to move an axis consists of an axis address, a direction of movement, and a target point.
  For the designation of a target point, two different methods are available: absolute commands and incremental commands. With absolute commands, the target point is specified using the coordinate values in the program coordinate system and with incremental commands the target point is defined by relative movement distance in reference to the actual position.
  For details of absolute and incremental commands, refer to "Absolute and Incremental commands".

- The basic coordinate system is a right-hand orthogonal coordinate system that is fixed on a workpiece.

Single-saddle NC lathe

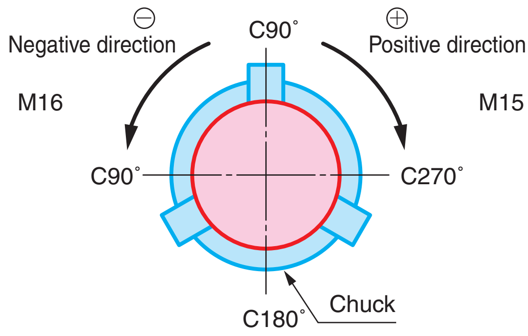


Two-saddle NC lathe



LE33013R0300400050002

C-axis coordinate system



(Viewed from tailstock)

LE33013R0300400050003

Rightward rotation is defined as positive direction of C-axis movement and is commanded by M15. M16 is used to specify C-axis movement in the negative direction.

2-2. Commands in Inch System

If the inch/metric switchable specification is selected, it is possible to specify dimensions in the inch unit system. Even if dimensions are specified in the inch system values in a part program, the NC processes the data on the basis of metric system values. The unit system to be selected for data input is determined according to the setting of an NC optional parameter (UNIT). The actual unit system for data input can be checked on the NC optional parameter (UNIT) screen.

[Supplement]

In the conversion from the inch system data to the metric system data, used for internal processing by the NC, real data values below the minimum input unit are rounded off. Integer data values are truncated.

2-3. Position of Decimal Point

It is possible to select the unit system of the place of a decimal point. Units of the data available with the control are shown below and the unit to be employed can be selected by entering a proper parameter data. Once the unit system of the command data is established, it applies to all numerical data to be entered, such as MDI operation and zero offset data.

2-3-1. Metric System

- 1 μm
- 10 μm
- 1 mm

2-3-2. Inch System (Inch/metric switchable specification):

- 1/10000 inch
- 1 inch

Unit Data Table (Value for data "1")

Dimension	Metric System			Inch System	
	1 μm	10 μm	1 mm	1/10000 inch	1 inch
Length: X, Z, I, K, D, H, L, U, W	0.001 (mm)	0.1 (mm)	1 (mm)	0.0001 (inch)	1 (inch)
Feed (/rev): F, E	0.001 (mm/rev)	0.01 (mm/rev)	1 (mm/rev)	0.0001 (inch/ rev)	1 (inch/rev)
Feed (/min): A, B, C	0.1 (mm/min)	1 (mm/min)	1 (mm/min)	0.01 (inch/min)	1 (inch/min)
Angle: A, B, C	0.001 (°)	0.01 (°)	1 (°)	0.001 (°)	1 (°)
Time: F, E	0.01 (sec)	0.1 (sec)	1 (sec)	0.01 (sec)	1 (sec)
Spindle min^{-1}: S	1 (min^{-1})	1 (min^{-1})	1 (min^{-1})	1 (min^{-1})	1 (min^{-1})
Surface speed: S	1 (m/min)	1 (m/min)	1 (m/min)	1 (feet/min)	1 (feet/min)

Example 1: 1 mm unit system

Commanding:

- 0.001 mm movement of X-axis — X0.001
- 10 mm movement of X-axis — X10
- 100.00 mm movement of X-axis — X100.01
- Feedrate of 0.23456 mm/rev. — F0.23456
  The following commands are all handed as X1 mm:
  X1
  X1.0
  X1.00
  X1.000

Example 2: 10 mm unit system

Commanding:

- 0.001 mm movement of X-axis — X0.1
- 10 mm movement of X-axis — X1000
- 100.010 mm movement of X-axis — X10001
- Feedrate of 0.23456 mm/rev. — F23.456

Example 3: 1 mm unit system

Commanding:

- 0.001 mm movement of X-axis — X0.1
- 10 mm movement of X-axis — X10000
- 100.010 mm movement of X-axis — X100010

- Feedrate of 0.23456 mm/rev. F234.56

[Supplement]

For F words, numerical data smaller than the selected unit system is effective if it consists of up to eight digits.

F1.2345678 ··········Acceptable
F100.000001·········Alarm (9 digits)

LE33013R0300400090001

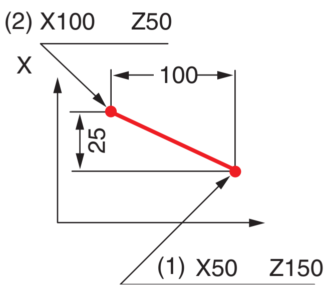
2-4. Absolute and Incremental Commands (G90, G91)

The amount of axis movement can be expressed by either absolute commands or incremental commands.

(1) Absolute commands
Designated with G90
Commanded values are coordinate values in the program coordinate system.
When the control is reset, it is in the G90 mode.

(2) Incremental commands
Designated with G91
Commanded values are the travel from the actual position to the target position.

Example:
(Positioning from point (1) to point (2)):



Absolute
G00 X50 Z150 (1)
X100 Z50 (2)
Incremental
G00 X50 Z150 (1)
*G91 X50 Z-100 (2)
*Designate dimensional differences between points (2) and (1).

LE33013R0300400100001

[Supplement]

1) In incremental programming, the X word should be expressed as a diameter. 2) It is not permissible to specify both G90 and G91 in the same block.

2-5. Diametric and Radial Commands

In a turning operation, the workpiece is rotated while being is machined. Due to the nature of the turning operation, the tool cuts a circle with a radius equivalent to the distance from the center of rotation to the tool nose position. In a program, X-axis commands specify the diameter of the circle to be cut. If a command of "X100" is specified, for example, the actual position data displayed on the screen is "100" and the workpiece is machined to a cylinder of 100-mm diameter.

In compound operations, commands in the X-axis direction are specified as diametric values too, although this type of operation is not a turning operation. In the coordinate conversion mode, however, the radial values (actual length in an orthogonal coordinate system) must be specified for both X- and Y-axis commands.

SECTION 3 MATH FUNCTIONS

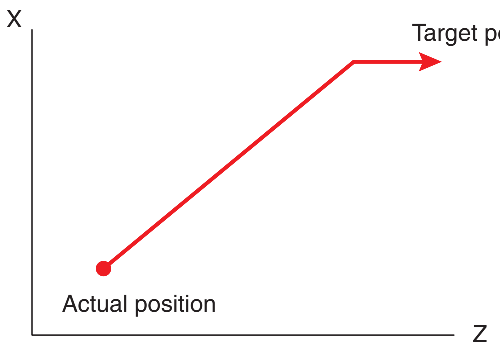
1. Positioning (G00)

[Function]
Each axis moves independently from the actual position to the target position at its own rapid feedrate. At the start and end of axis movement, it is automatically accelerated and decelerated.
[Programming format]
G00 X__ Z__ C__

X/Z/C : Indicates the target position for positioning operation.

[Details]

- In G00 mode positioning, execution of the commands in the next block begins only after the positioning at the target position given in the current block is completed.
- Non-linear interpolation mode:
  The axes move independently of each other at a rapid feedrate. Therefore, the resultant tool path is not always a straight line.



LE33013R0300500010001

[Supplement]

The rapid feedrates of each axis are set by the machine specifications.

2. Linear Interpolation (G01)

[Function]
The G01 command specifies the axes to move directly from the current position to the specified coordinate values at the specified feedrate.
[Programming format]
G01 X__ Z__ C__ F__

X, Z, C : Target point (end point)
F : Feedrate.
The specified value remains effective until updated by another value.

[Supplement]

1) The feedrate becomes zero when the NC is reset.
2) The feedrate for each axis is indicated below. (Calculate feedrate for X and Z-axes as incremental values.)
   G01 XxZzFf
   Calculation of feedrates:

   X-axis feedrate $FX = \frac{x}{L} f$

   Z-axis feedrate $FZ = \frac{z}{L} f$

   where $L_3 = \sqrt{x^2+z^2}$

   x, z, f: Command values specified in a program

3. Circular Interpolation (G02, G03)

[Function]
Circular interpolation can be used to generate a cutting path which follows an arc.
[Programming format]

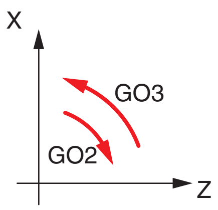
$$\begin{array}{l} \text{G02} \\ \text{(G03)} \end{array} \quad \text{X\_\_} \quad \text{Z\_\_} \quad \left\{ \begin{array}{c} \text{L\_\_} \\ \text{I\_\_ K\_\_} \end{array} \right\} \quad \text{F\_\_}$$

LE33013R0300500030001

G02	:	Direction of rotation	:	Sets clockwise rotation
G03	:	Direction of rotation	:	Sets counterclockwise rotation
X, Z	:	G90 mode	:	Set the end point in the program coordinate system
X, Z	:	G91 mode	:	Sets the end point in reference to the starting point (values should include signs)
I, K	:	Set the distance of the center of the arc from the starting point (values should include signs)		
L	:	Sets the radius of the arc		
F	:	Sets the feedrate		

[Details]

- The two directions of rotation, clockwise and counterclockwise, are defined when viewing the Z-X plane from the positive direction of the axis orthogonal to the plane in the right-hand orthogonal coordinate system.

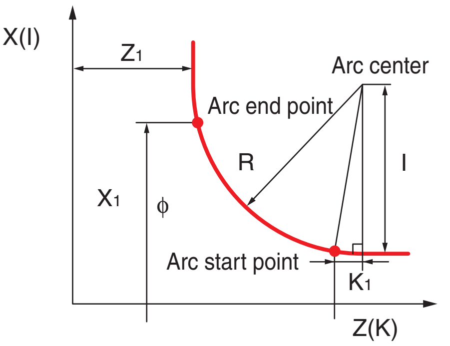


LE33013R0300500030002

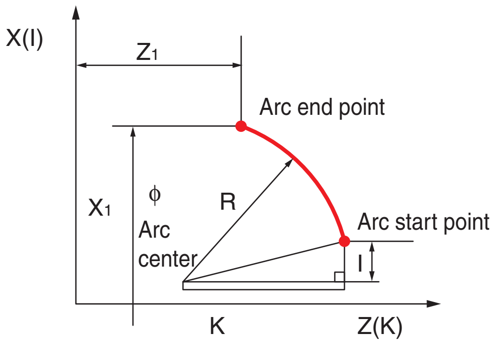
- The end point of an arc is defined as an absolute value or an incremental value depending on the G90/G91 selection.

- The center of an arc is expressed by I and K, which correspond to X and Z respectively. That is, I expresses the X coordinate value and K the Z coordinate value of the center of the arc in reference to the starting point of the arc.

For I and K, signed incremental values are used regardless of the mode, G90 or G91.



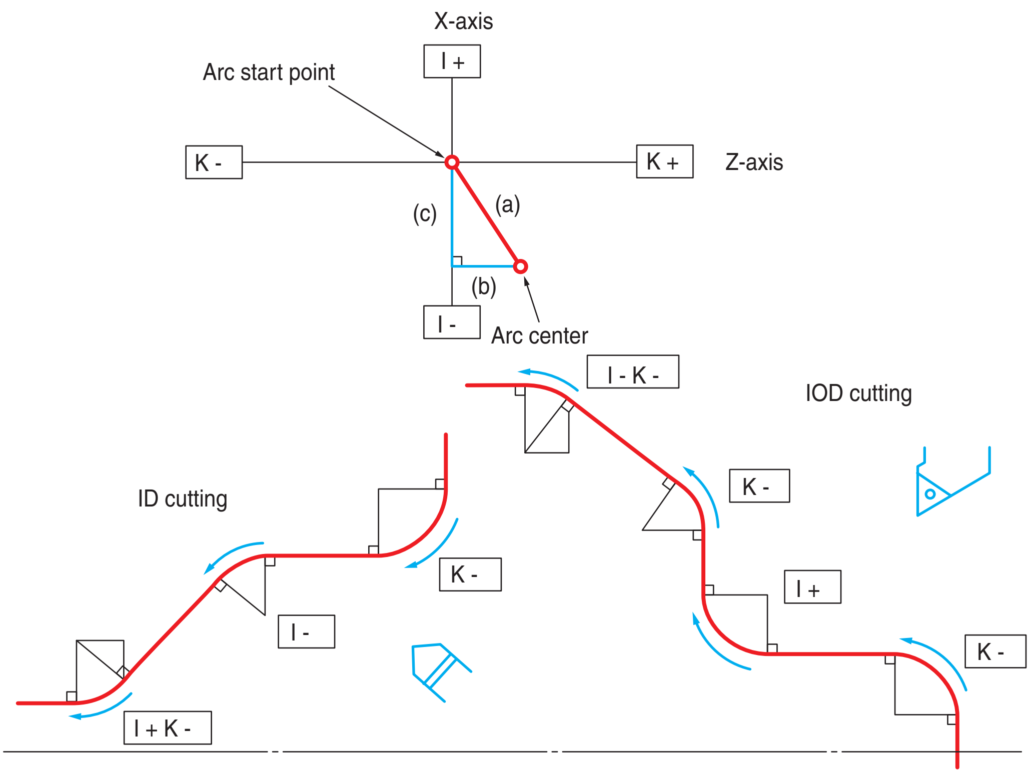
G02: Both I and K values are positive
Z_1, X_1 indicate the coordinate values of the arc end point.



G03: Both I and K values are positive
Z_1, X_1 indicate the coordinate values of the arc end point

LE33013R0300500030003

Determining Sign and Numeric Value of I and K Words:
See the figure below. Assume the coordinate system has its origin at the arc start point. Draw a right-angled triangle taking the segment connecting the arc center and arc start point as the hypotenuse. The length of side (b), parallel to the Z-axis, is the value of the K word and that of side (c), parallel to the X-axis, is the value of the I word.
Concerning the sign of these words, when side (b) lies in the positive direction of the assumed coordinate system, it is taken as a positive value and when it lies in the negative direction, it is negative. The sign of I words is determined in a similar way. That is, when side (c) lies in the positive direction of the coordinate system, the I word has a positive value and when it lies in the negative direction, the I word has a negative value.



LE33013R0300500030004

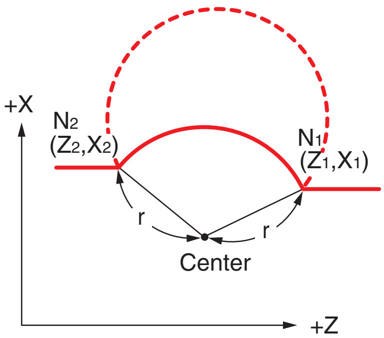
- Direct Radius Command
  It is possible to execute circular interpolation by specifying the X and Z coordinate values of the target point and the radius of the arc instead of using I and K commands.

[Supplement]

- The G code used to call circular interpolation is G02 or G03, as when using I and K.
- The radius of the arc is expressed by an L word which must have a positive value.
- A block containing an L word without a K or I word is an arc radius command.
- When expressing an arc by its radius, the commands must contain both X and Z words.
- If either of them is omitted, an alarm results.
- If an L word is specified in a block containing I and/or K word, an alarm results.
- If the distance from the current position to the target point (end point) is larger than two times the specified radius, an alarm results since circular interpolation cannot be performed.
- In direct arc command programming, one arc command yields two arcs; one with central angle less than 180, and another larger than 180. The arc with central angle less than 180 is selected.
  To obtain the arc whose central angle is greater than 180, specify "CALRG" in the block commanding circular interpolation.
- The direct radius command programming is effective in:
  LAP
  Tool nose radius compensation mode
  Subprograms
- Incremental programming mode (G91)
  In direct radius command programming, the control automatically calculates the coordinates of the center of the arc, I and K, from the programmed radius L and the coordinates of the end point, X and Z, to perform circular interpolation.

The program for the example in the figure to the right is as follows.
Program:

N1 G01 X_1 Z_1 F_1
N2 G03 X_2 Z_2 Lr
With the commands above,
the arc indicated by a thick solid line is obtained.



To move the tool along the arc indicated by dashed lines, program as follows:
N1 G01 X_1 Z_1 F_1
N2 G03 CALRG X_2 Z_2 Lr

LE33013R0300500030005

- Feedrates
  The feedrate during circular interpolation is the feedrate component tangential to the arc.

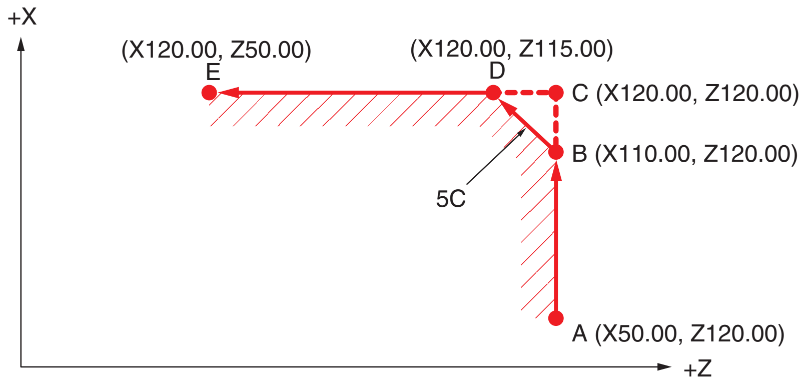
[Supplement]

1) If I or K is omitted, I0 or K0 applies.
2) I and K values should be specified as radii.
3) An arc extending into two or more quadrants can be specified by the commands in a single block.
4) If either X or Z is omitted, circular interpolation is possible within one quadrant.
5) An alarm will be activated if the difference in radius between the start and end point of an arc is greater than the value set for optional parameter (OTHER FUNCTION 1) No. 6 Allowable error in circular interpolation.

4. Automatic Chamfering

When cutting a workpiece, it is often necessary to chamfer a sharp edge (either straight-line chamfering (C-chamfering) or rounding). Although such chamfering can be accomplished using conventional interpolation commands (G01, G02, G03), the automatic chamfering function permits chamfering to be done with a simple program.
For chamfering at any required angle, the automatic any-angle chamfering function should be used.
To use the automatic chamfering function, set "1" for optional parameter (OTHER FUNCTION 1) Auto. any-angle chamfering. If the automatic any-angle chamfering function is required, set "any-angle chamfering" for this parameter.

4-1. C-chamfering (G75)



To cut the contour shown above along the points A, B, D and E, program as follows:
G75 G01 X120 L-5 FDD CR
after positioning the cutting tool at point A.
With the commands above, the cutting tool moves from point A to B and then to D, thus automatically chamfering the corner at 45 with a size of 5 mm.

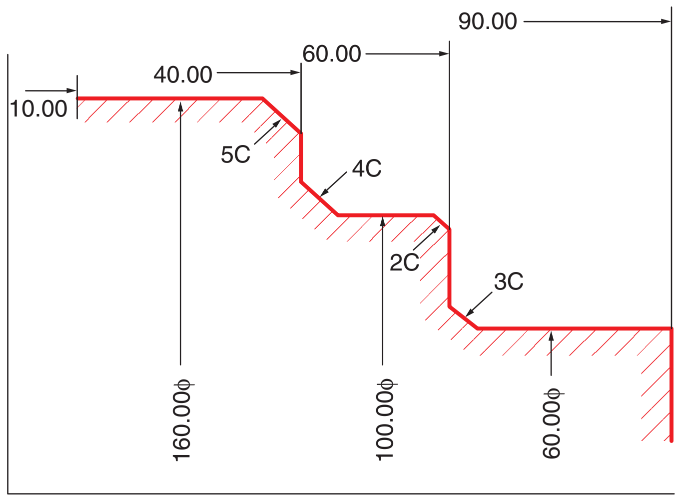
G75 : Specifies C-chamfering
X120 : X coordinate of Point C
L-5 : Size of chamfered face
The sign is determined by the direction of axis movement;
"+" when the Z-axis (X-axis) moves in the positive direction after X-axis (Z-axis) motion.
"-" when the Z-axis (X-axis) moves in the negative direction after X-axis (Z-axis) motion.

When the coordinates of point E are commanded, the cutting tool moves from Point D to Point E.
[Details]

- G75 is effective only in the G01 mode. If G75 is specified in another mode, it causes an alarm.
- G75 is non-modal and active only in the commanded block.
- If the axis movement dimension specified in the block calling for automatic chamfering (A - C in the figure above) is smaller than the absolute value of the L word (B - C in the figure above), an alarm results.
- If the axis movement dimensions specified in the block calling for automatic chamfering are zero both for X and Z, or if neither the X nor the Z value is zero in such a block, an alarm occurs. The block calling for the automatic chamfering mode can contain only one dimension word, either X or Z.

- The automatic chamfering program is effective in:
  LAP
  Tool nose radius compensation mode

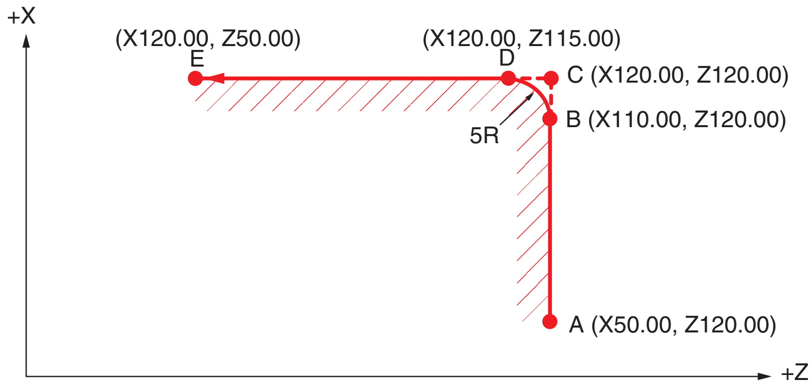
[Program example]




```
  :
  :
N101    G01   X60    Z92   F0.1
N102    G75          Z60   F0.05  L3
N103    G75   X100                L-2
N104    G75          Z40          L4
N105    G75   X160                L-5
N106                 Z10
  :
  :
```

LE33013R0300500050002

4-2. Rounding (G76)



LE33013R0300500060001

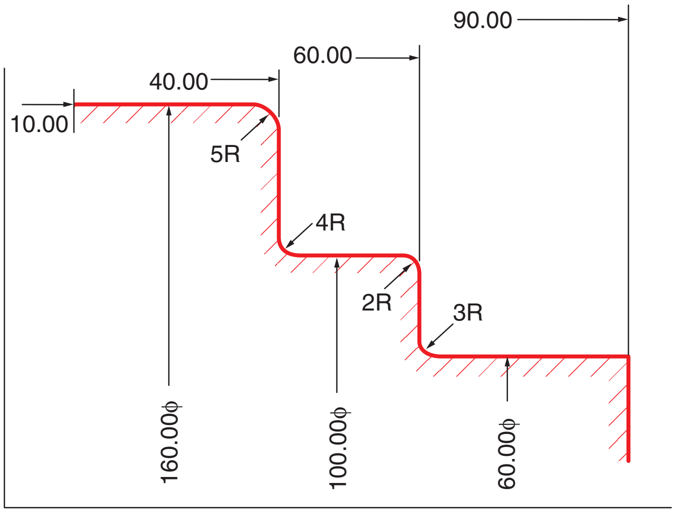
To cut the contour shown above along the points A, B, D and E, program as follows:
<u>G76</u> G01 <u>X120</u> <u>L-5</u> FDD CR
after positioning the cutting tool at point A.
With the commands above, the cutting tool moves from point A to B and then to D, thus automatically rounding the corner to a radius of 5 mm.

G76	:	Specifies rounding of a corner
X120	:	X coordinate of Point C
L-5	:	Radius of rounding circle The sign is determined by the direction of axis movement; "+" when the Z-axis (X-axis) moves in the positive direction after the X-axis (Z-axis) motion. "-" when the Z-axis (X-axis) moves in the negative direction after the X-axis (Z-axis) motion.

When the coordinates of point E are commanded, the cutting tool moves from point D to point E.
[Details]

- G76 is effective only in the G01 mode. If G76 is specified in a mode other than G01, an alarm occurs.
- G76 is non-modal and active only in the commanded block.
- The rounding describes a 1/4 circle with the radius specified by an L word.
- If the axis movement dimension specified in the block calling for automatic chamfering (A - C in the figure above) is smaller than the absolute value of the L word (B - C in the figure above), an alarm results.
- If the axis movement dimensions specified in the block calling for automatic chamfering are zero both for X and Z, or if neither X nor Z value is zero in such a block, an alarm occurs.
  The block calling for automatic chamfering mode can contain only one dimension word, either X or Z.
- The automatic chamfering program is effective in:
  LAP
  Tool nose radius compensation mode

[Program Example]



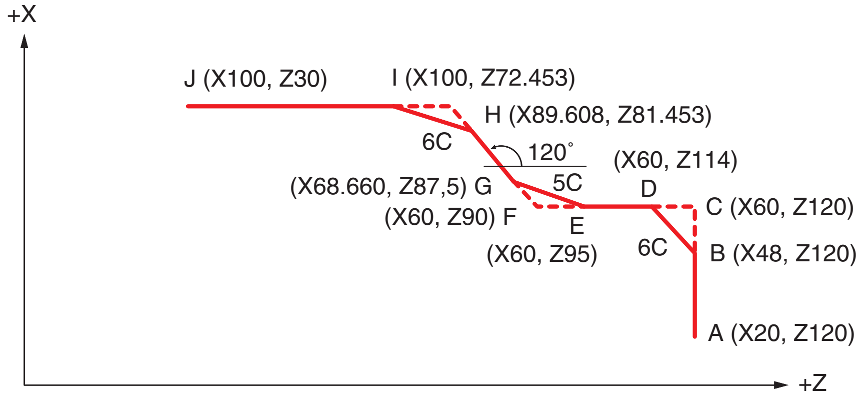
N101	G01	X60	Z92	F0.1	
N102	G76		Z60	F0.05	L3
N103	G76	X100			L-2
N104	G76		Z40		L4
N105	G76	X160			L-5
N106			Z10		

LE33013R0300500060002

4-3. Automatic Any-Angle Chamfering

When cutting a workpiece, it is often necessary to chamfer the sharp (C-chamfer or R-chamfer) corners and edges. If chamfering is required on edges having an angle other than 90°, programming chamfering using G01, G02 and G03 commands is not easy. This automatic chamfering function can program chamfering easily.
[Programming Examples]

(1) C-Chamfering (G75)




```
                        :
                        :
N100  G00        X20   Z120
N110  G75  G01   X60         L6      FΔΔΔ
N120  G75              Z90   L5
N130  G75  A120  X100        L6
N140                   Z30
                        :
                        :
```

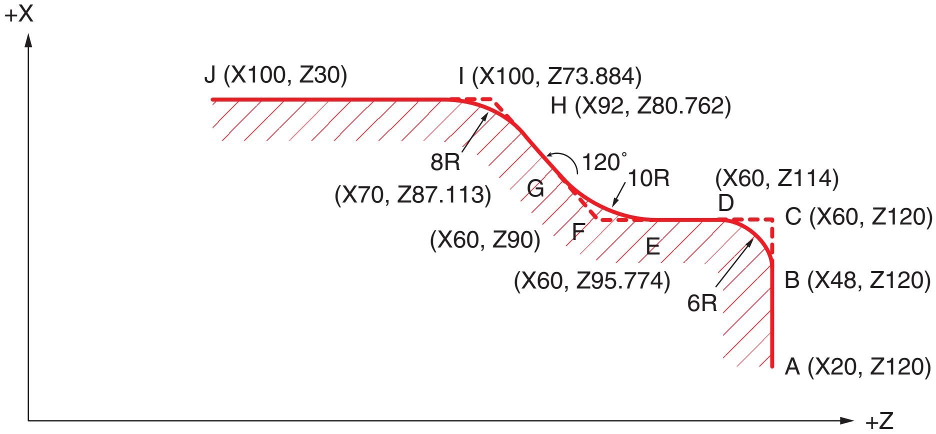
LE33013R0300500070001

With the program above, the cutting tool moves from point A to point J in the sequence A, B, D, E, G, H, I and J, thus accomplishing chamfering of B-D, E-G and H-I.

[Supplement]

Angle commands (A) are designated in reference to the Z-axis.

(2) R-Chamfering (G76)




```
                     :
                     :
N100   G00        X20    Z120
N110   G76   G01  X60           L6     FΔΔ
N120   G76               Z90    L10
N130   G76   A120 X100          L8
N140                     Z30
                     :
                     :
```

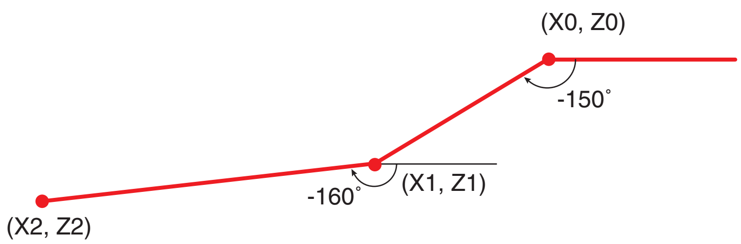
LE33013R0300500070002

With the program above, the cutting tool moves from point A to point J in the sequence A, B, D, E, G, H, I and J, thus accomplishing chamfering of B-D, E-G, and H-I.

[Supplement]

With the C-chamfer function, axis movements in the G00, G01, G34 and G35 modes can be designated by simply entering an angle command A without X and/or Z coordinate data.

Example:




```
G00   X0      Z0          CR
G01   A-150        FΔ     CR
      X2      Z2   A-160  CR
```

LE33013R0300500070003

(X1, Z1) should not be designated; it is automatically generated in the NC.

[Supplement]

1) Both G75 and G76 are effective only in the G01 mode and if they are designated in a mode other than G01an alarm occurs.
2) If the axis movement amount is smaller than the chamfering size, an alarm occurs.
3) Chamfering is possible only at corners between two lines. Chamfering at corners between two arcs, between a line and an arc, or between an arc and a line is impossible. If chamfering at such corners is attempted, an alarm occurs.
4) The chamfering command is effective both in the LAP and nose radius compensation mode.
5) If only an angle command A is designated in G00, G01, G34, or G35 mode operations, the next axis movement command must contain A, X and Z commands so that the end point of the line commanded can be defined. If these commands are not designated and the end point cannot be defined, then an alarm occurs.
6) If chamfering commands G75 and G76 are designated without axis movement commands X and Y or if they are designated only with an A command, the control reads the commands in the next sequence to calculate the point of intersection automatically. Therefore, if the next sequence does not contain adequate data for this calculation, an alarm occurs.

5. Torque Limit and Torque Skip Function

To transfer a workpiece from the first-process chuck to the second-process chuck with multi-process models*, the end face of the second-process chuck jaws must be pushed against the workpiece for stable workpiece seating. The torque limit command and the torque skip command are used to control the torque of the second-process chuck feed servomotor and to push the workpiece with the optimal thrust.
* Multi-process models include sub spindle models, opposing two-spindle models, etc.

5-1. Torque Limit Command (G29)

[Function]
Prior to workpiece transfer, designate the torque limit command to control the maximum torque of the second-process chuck feed servomotor.
[Program Format]
G29 P D __
(Designate an axis to be fed: Z or W, for D.)
[Details]

- The torque limit value is set as a percentage, taking the rated torque of the axis feed servomotor as 100%.
- The maximum torque limit value is set for optional parameter (OTHER FUNCTION 2).

5-2. Torque Limit Cancel Command (G28)

[Function]
The torque limit cancel command cancels the maximum torque limit designated with G29.
When this command is designated, the axis feed motor can output its maximum output torque.
[Programming format]
G28

5-3. Torque Skip Command (G22)

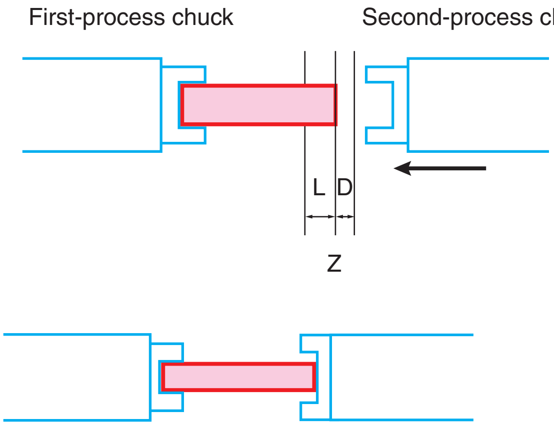
[Programming format]
G22 Z__ D__ L__ F__ PZ =__

- Z : Target point (mm)
- D : Distance between the target point and the approaching point as an incremental value (mm)
- L : Distance between the target point and the virtual approaching point as an incremental value (mm)
- F : Feedrate (mm/min or mm/rev)
- PZ : Preset torque value (%)

[Details]

- For the target point and the set torque value, designate the axis to be fed..
- An alarm (alarm A 1220) occurs if the preset torque value is not reached when the second-process chuck has moved to the virtual approaching point.
- Designate a value equal to or smaller than "2.5 m/min (8.20 fpm)" for F.
- Before setting a value for PZ, check the actual motor torque value** at axis feed at the feedrate designated by F, and set a value for PZ which is larger than the actual torque value by 10%.

  ** Check the RLOAD value displayed on the axis data page of the CHECK DATA screen.
  If the preset torque value is too small, it is reached during approaching motion, resulting in an occurrence of alarm 1219.



The explanation here is for a case in which a workpiece is transferred from the first-process chuck to the second-process chuck.

1. The second-process chuck approaches the workpiece at feedrate F.
2. The feedrate is reduced to 1/5 of F at the approaching point (Z - D) point).
3. The second-process chuck contacts the workpiece at target point Z.
   The servomotor is controlled so that the second-process chuck is kept pushed against the workpiece.
4. When the motor torque reaches the preset value, the NC recognizes workpiece seating to be complete, and the next program block is executed.

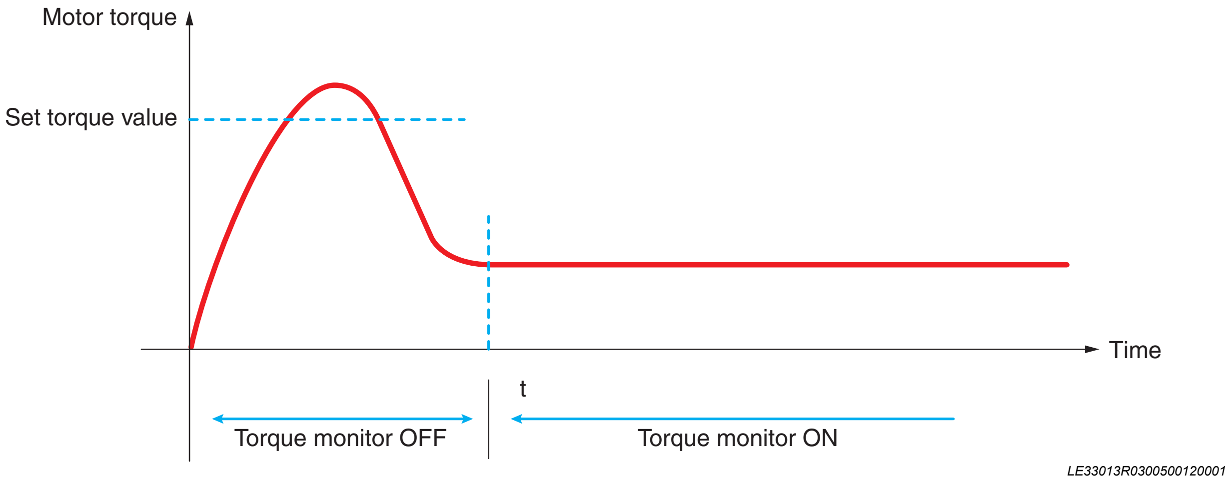
LE33013R0300500110001

Feedrate F → F/5

- Z : Target point
- D : Distance between the target point and the approaching point as an incremental value
- L : Distance between the target point and the virtual approaching point as an incremental value

5-4. Parameter Setting

(1) Torque skip torque monitoring delay time
If motor torque monitoring is started at the start of torque skip feed designated by G22, the preset torque value could, in some cases, be exceeded on starting up the motor.
To avoid this, set the torque monitoring delay time t for a parameter. Motor torque is not monitored for the time duration set for t.



Optional parameter (OTHER FUNCTION 2)

Setting unit : 10 (ms)
Setting range : 0 to 9999
Initial setting : 0

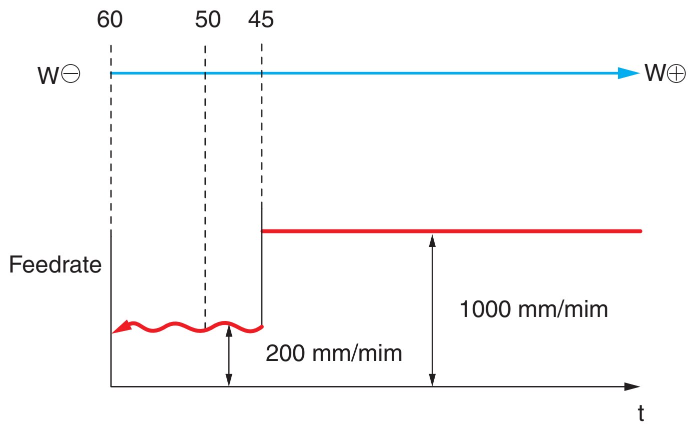
(2) Upper limit for torque skip torque limit
The upper limit for the P command value in the G29 block can be set.
Optional parameter (OTHER FUNCTION 2)

Setting unit : 1 (%)
Setting range : 1 to 100
Initial setting : 0

5-5. Program Example

This is a program example for transferring a workpiece to the sub spindle chuck.

```
                    :
                    :
G29 PW=30······Limits the maximum torque of the sub spindle feed motor
               (W-axis motor). (30 %)
G94 G22 W50 D5 L10 F1000 PW=25······Pushes the sub spindle chuck against
                                    the workpiece end face by torque skip
G29 PW=5······Lowers the W-axis motor torque.
M248······Sub spindle chuck close
M84······Main spindle chuck open
G28······Cancels W-axis torque limit.
G90 G00 W300······Returns the W-axis to the retract position at the rapid feedrate.
                    :
                    :
```



LE33013R0300500130001

SECTION 4 PREPARATORY FUNCTIONS

G codes are used to specify particular functions which are to be executed in individual blocks. Every G code consists of the address "G" plus a 3-digit number (00 to 399)

- Effective G Code Ranges

One-shot : A one-shot G code is effective only in a specified block and is automatically canceled when program execution moves to the next block.
Modal : A modal G code is effective until it is changed to another G code in the same group.

LE33013R0300600010001

- Special G Codes
  The mnemonic codes of subprogram calls (G101 through GI 10, for instance) and branch instructions are called special G codes. Every special G code must be specified at the beginning of a block, not part way through a block. Note, however, that a "/" (block delete) and a sequence name may be placed before a special G code.

1. Dwell (G04)

[Function]
If dwell is specified, execution of the next block is suspended for the specified length of time after the completion of the preceding block.
[Programming format]
G04 F__

F : Specify the length of time for which the execution of a program is suspended.
The unit of command values is determined by the selected programming unit system.
For details, refer to the optional parameter (unit system).
The maximum allowable length of a dwell period is 9999.99 seconds.

2. Zero Shift/Max. Spindle Speed Set (G50)

2-1. Zero Shift

[Function]
With the G50 code, zero offset value is automatically calculated and zero setting is carried out according to the calculated value.
This feature is effective when cutting a workpiece on which the same contour is repeated.
[Programming format]
G50 X__ Z__ C__

X/Z/C : Specify the coordinate value to be taken as the actual position data after zero shift.

[Details]
For the present X- and Z-axis position, the coordinate value specified following G50 are assigned.
[Program]

```
N004  G00  X0  Z0
N005  G50  X1  Z1
N006  G00  X2  Z2
```

LE33013R0300600030001

With the program above, the axes are positioned to the coordinate point (X0, Z0) by the commands in block N004 first. When the commands in N005 are executed, the coordinate system is re-established so that (X0, Z0), where the axes have been positioned, now has the coordinate values (X1, Z1) which are specified following G50.
This program shifts the origin of the coordinate system:
X = X0 - X1
Z = Z0 - Z1
Provided X0 = 100 mm and X1 = 200 mm, zero offset amount is calculated as;
100 - 200 = -100 mm
This amount can be checked on the screen.
Dimension words in sequences N006 and after that are all referenced to the origin newly established by the commands in N005.

[Supplement]

1) Axes not specified in the block containing G50 are not subject to zero offset.
2) G50 is non-modal and active only in the programmed block. (Zero offset is calculated only in the G50 block. All dimension words after that block are referenced to the shifted new origin.)
3) When the control is reset, all zero set data are cleared and the initial zero offset data become effective.
4) No tool offset number entry is allowed in the block containing the G50 code.

2-2. Max. Spindle Speed Set

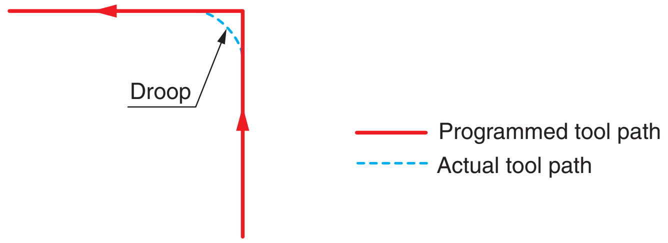
[Function]
Sometimes the spindle speed must be clamped at a certain speed due to the restrictions on the allowable speed of a chuck, influence of centrifugal force on workpiece gripping force, imbalance of a workpiece, or other factors. This feature allows a maximum spindle speed to be set in such cases.
[Programming format]
G50 S__

S : Specify the maximum spindle speed.

[Details]
Once set, the specified speed remains effective until another spindle speed is specified.

3. Droop Control (G64, G65)

[Function]
The axis movements of the machine are controlled by a servo system in which the axis moves to eliminate the lag (termed DIFF or droop) between the actual tool position and the commanded coordinate.
Due to existence of DIFF (servo error), the actual path does not precisely agree with the commanded tool path when cutting a sharp corner, as illustrated below:
The Droop Corner Control Function is provided to eliminate or reduce such path tracing error to acceptable amounts by stopping the generation of functions (pulses) at the corner until the DIFF reaches the preset permissible droop amount.



LE33013R0300600050001

[Programming format]

- Droop corner control OFF command
  G64
  (The control is placed in the G64 mode when G64 is turned ON.)

- Droop corner control ON command
  G65

[Details]

- With G65 presented, axis movement commands in G00, G01, G02, G03, G31, G32, G33, G34, and G35 mode are completed after the DIFF amount becomes smaller than the permissible droop amount.

- The permissible droop amount can be set within a range from 0 to 1.000 mm for a user parameter at the NC operation panel.

4. Feed Per Revolution (G95)

[Function]
Specify G95 to control tool movement (feedrate) in terms of "distance per spindle revolution" for turning operations.
[Programming format]
G95 F__

F : Specify movement distance per spindle revolution.
The unit of setting is determined according to the setting for the optional parameter (UNIT)

[Details]

- The allowable maximum feedrate depends on the machine specifications.
- On turning on the power, and after reset, the feed per revolution mode is selected.

5. Feed Per Minute (G94)

[Function]
Specify G94 to control tool movement (feedrate) in terms of "distance per minute" for turning operations.
[Programming format]
G94 F__

F : Specify tool movement distance per minute.
The unit of setting is determined according to the setting for the optional parameter (UNIT)

[Details]

- The allowable maximum feedrate depends on the machine specifications.

6. Constant Speed Control (G96/G97)

[Function]
When the constant speed cutting function is selected, cutting at a constant cutting speed is possible. This feature can reduce cutting time and also assure stable finish in end face cutting operations.

Constant Speed Cutting Command

[Programming format]
G96 S__

S : Set the cutting speed (setting unit: m/min)

Canceling Constant Speed Cutting

[Programming format]
G97 S__

S : Set the spindle speed to be used after canceling the constant speed cutting mode.

[Program Example]

↓

N OOO G96 S100·······All cutting following this block is executed at a cutting speed of 100 m/min.

↓

N OOO G97 S500·······After this block, cutting is carried out at a spindle speed of 500 min^{-1}.

↓

LE33013R0300600080001

[Supplement]

1) If the spindle speed exceeds the maximum or minimum speed allowed within the range selected by an M code while in the constant speed cutting mode, it is fixed at the allowed maximum or minimum speed automatically; the LIMIT indication light on the operation panel goes on.
2) If the X-axis is moved a large distance at the rapid traverse rate while in the constant speed cutting mode, for example from the turret indexing position toward the workpiece or vice versa, there will be sudden changes in the rotational speed which, depending on the chucking method, could be dangerous.
   Therefore, the constant speed cutting mode must be cancelled before commanding positioning of the cutting tool near the workpiece, return of the tool to the turret indexing position, or any other operation that causes large X-axis travel.
3) A block containing G96 or G97 must contain an S word.
4) Thread cutting programs cannot be executed in the G96 constant speed cutting mode.
5) To activate the constant speed cutting mode on turret B, specify G111 with G96. To restore the constant speed cutting mode to turret A, specify G110.
6) To execute the commands over two blocks continuously with control in the constant speed cutting mode without waiting for the spindle speed arrived signal, specify M61. To cancel this, specify M60.

SECTION 5 S, T, AND M FUNCTIONS

This section describes the S, SB, T, and M codes that specify the necessary machine operations other than axis movement commands.

S : Spindle speed
SB : Spindle speed of M-tool spindle
T : Tool number, tool offset number, tool nose radius compensation number
M : Miscellaneous function to control machine operation

One block can contain: one S code, one T code, and eight M codes.

1. S Functions (Spindle Functions)

[Function]
By specifying number following address S, spindle speed can be specified.
[Programming format]
S__
[Details]

- S command range: 0 to 65535
- If there is an S command and an axis move command in the same block, the S command is executed first and then the axis move command is executed.
- The S command will not be canceled when the NC is reset, however, it will be set to 0 when the power supply is turned off.
- To rotate the spindle, the S command must be specified in a block that precedes the block containing the spindle start command or in the same block.

[Supplement]

1) For a machine equipped with the transmission gears, the required gear range should be selected with the corresponding M code.
2) Spindle rotation (forward, reverse) and stop are specified by M codes.

2. SB Code Function

[Function]
M-tool spindle speed is specified using address SB.
[Programming format]
SB = __
f an address consisting of two or more characters is used, an equal symbol must be entered before a numeric value.

- SB command range: 0 to 65535
- M-tool spindle rotation (forward, reverse) and stop are specified by M codes.
- The SB command will not be canceled when the NC is reset, however, it will be set to 0 when the power supply is turned off.

- To rotate the M-tool spindle, the SB command must be specified in a block that precedes the block containing the M-tool spindle start command or in the same block.

[Supplement]

1) For the machine equipped with the transmission gears for driving the M-tool spindle, the required gear range should be selected by a corresponding M code.
2) M-tool spindle rotation (forward, reverse) and stop are specified by M codes.

3. T Functions (Tool Functions)

[Function]
By specifying a 4-digit number (NC without tool nose radius compensation function) or a 6-digit number (NC with tool nose radius compensation function) following address T, tool number, tool offset number, and tool nose radius compensation number are indicated.
[Programming format]

TOO∆∆□□

OO: Tool nose radius compensation number

∆∆ : Tool number (00 to 99, assuming maximum number of turret stations)

□□: Tool offset number

LE33013R0300700040001

The setting ranges for nose radius compensation numbers and tool compensation numbers are as follows:

(1) For offset 32-set specification

- Tool offset number: 00 to 32
- Tool nose radius compensation number: 00 to 32
(if tool nose radius compensation function is supported.)

(2) For offset 64-set specification

- Tool offset number: 00 to 64
- Tool nose radius compensation number: 00 to 64
(if tool nose radius compensation function is supported.)

(3) For offset 96-set specification

- Tool offset number: 00 to 96
- Tool nose radius compensation number: 00 to 96
(if tool nose radius compensation function is supported.)

[Details]
If there is a T command and an axis move command in the same block, the T command is executed first and then the axis move command is executed.

[Supplement]

The construction of the turret and its direction of rotation (forward, reverse, shorter-path) vary according to the machine specifications.

4. M Functions (Auxiliary Functions)

[Function]
The M codes are used for miscellaneous ON/OFF control and sequence control of the machine operation such as spindle start/stop and operation stop at the end of program. The programmable range for M codes is from 0 to 511.
[Examples of M codes]
The M codes listed below are processed as special functions.
For details on those M codes not listed here, refer to APPENDIX 3. "List of M Codes".

(1) M00 (program stop)
After the execution of M00, the program stops. If the NC is started in this program stop state, the program restarts.

(2) M01 (optional stop)
When M01 is executed when the optional stop switch on the machine operation panel is ON, the program stops. If the NC is started in this optional stop state, the program restarts.

(3) M02, M30 (end of program)
These M codes indicate the end of a program.
When M02 or M30 is executed, the main program ends and reset processing is executed. The program is rewound its start. (In the case of a schedule program, execution of M02 or M30 in the main program does not reset the NC.)

(4) M03, M04, M05 (spindle CW, CCW, stop)
These M codes control spindle rotation and stop; spindle CW (M03), spindle CCW (M04), and spindle stop (M05).

(5) M12, M13, M14 (rotary tool CW, CCW, stop)
These M codes control rotary tool rotation and stop for the turning center; rotary tool stop (M12), rotary tool CW (M13), rotary tool CCW (M14).

(6) M15, M16 (C-axis positioning direction)
These M codes control the C-axis rotation direction for positioning for the turning center; C-axis positioning in the positive direction (M15), C-axis positioning in the negative direction (M16).

(7) M19 (spindle orientation)
This controls spindle orientation.

(8) M20, M21 (tailstock barrier ON, OFF)
These M codes set and cancel the tailstock barrier which generates an alarm if the tool enters the area defined by the barrier; tailstock barrier ON (M21), tailstock barrier OFF (M20).

(9) M22, M23 (chamfering ON, OFF for thread cutting)
These M codes set and cancel chamfering for thread cutting; chamfering ON (M23), chamfering OFF (M22).

(10) M24, M25 (chuck barrier ON, OFF)
These M codes set and cancel the chuck barrier which generates an alarm if the tool enters the area defined by the barrier; chuck barrier ON (M25), chuck barrier OFF (M24).

(11) M26, M27 (thread pitch axis X-axis, Z-axis)
These M codes specify the effective thread pitch axis for conventional thread cutting cycles; X-axis pitch command (M27), Z-axis pitch command (M26).

(12) M32, M33, M34 (thread cutting mode; straight, zigzag, straight (reversed))
These M codes are used to specify the thread cutting mode in the compound fixed cycle and LAP; M32 for infeed along one side of the thread face to be cut (straight), M33 for zigzag infeed, and M34 for straight infeed along the opposite thread face from the one in the M32 mode (straight (reversed)).

(13) M40, M41, M42, M43, M44 (spindle drive gear range; neutral, gear 1, gear 2, gear 3, gear 4)
These M codes are used to select the spindle drive gear range; neutral (M40), gear 1 (M41), gear 2 (M42), gear 3 (M43), and gear 4 (M44).

(14) M48, M49 (spindle speed override ignore)
When the spindle speed override ignore function is valid, the spindle speed override rate is fixed at 100% regardless of the setting of the spindle override switch. The spindle speed override ignore function is canceled by specifying the cancel M code, resetting the CNC, or changing the operation mode.
< M codes >

Spindle speed override ignore............................ M49

Spindle speed override ignore cancel................M48

LE33013R0300700050001

(15) M55, M56 (tailstock spindle retract, advance)
These M codes specify tailstock retract/advance operation.

(16) M60, M61 (fixed surface speed arrival ignore OFF, ON)
These M codes are used to specify whether or not a program with constant surface speed control is executed continuously without waiting for attainment of the specified surface speed; M61 specifies advance to the next block without waiting for attainment of the specified surface speed, and M60 specifies advance to the next block only after attainment of the specified surface speed.

(17) M63 (spindle rotation answer signal ignore)
The M codes relating to spindle control (M03, M04, M05, M19, M40 - M44) and S command are executed at the same time with axis move commands specified in the same block.

(18) M73, M74, M75 (thread cutting pattern 1, 2, 3)
In multi-machining fixed cycle and thread cutting cycle in LAP, the cutting pattern (infeed pattern) is specified by these M codes. M73 for pattern 1, M74 for pattern 2, and M75 for pattern 3.

(19) M83, M84 (chuck clamp, unclamp)
Regardless of the chuck clamp direction (I.D. or O.D.), the M code used to specify the clamping of a workpiece is always M83.

(20) M85 (no return to the start point after the completion of LAP roughing cycle)
In LAP4, a roughing cycle is called by G85 or G86. When this M code is specified, the cutting tool does not return to the reference point of the cycle after the completion of the called roughing cycle, and the next block is executed continuously.

(21) M86, M87 (turret clockwise rotation ON, OFF)
These M codes are used to specify whether or not the turret rotation direction is fixed in the clockwise direction; M86 specifies turret clockwise rotation ON, and M87 specifies turret clockwise rotation OFF.

(22) M109, M110 (C-axis connection ON, OFF)
These M codes are used to select the spindle control mode for the multiple-process machining specification models. By specifying M110, the spindle is controlled in the C-axis control mode and by specifying M109, the control mode is returned to the spindle control mode. Note that M110 must be specified in a block without other commands.

(23) M124, M125 (STM time-over check ON, OFF)
These M codes are used to determine whether or not an alarm is generated if the counted STM execution cycle time exceeds the parameter-set time; M124 specifies that the alarm is generated, and M125 specifies that the alarm is not generated.

(24) M136 (shape definition for compound fixed cycle)
This M code is used to specify the shape for the compound fixed cycles provided for the multiple-process specification models. After the execution of the compound fixed cycle, the cutting tool returns to the start point of rapid traverse.

(25) M140 (tapping cycle rotary tool fixed speed arrived answer signal ignore)
This M code is used to ignore the tapping cycle rotary tool fixed speed arrived answer signal; by specifying this M code, the timing difference between the output of rotary tool fixed speed arrived answer signal and the start of cutting feed can be zeroed. Note that this M code is available with the multiple-process specification models.

(26) M141, M146, M147 (C-axis clamp used/not-used selection, C-axis unclamp, C-axis clamp)
For a compound fixed cycle carried out under light load on multiple-process specification models, it is not necessary to clamp the C-axis to carry out cutting. In such a case, M141 is used to select the "C-axis clamp is not used" state, thereby reducing cutting time.
M146 and M147 are used to control C-axis clamping and unclamping; M146 for C-axis clamp and M147 for C-axis unclamp.

(27) M156, M157 (center work interlock ON, OFF)
When center work is selected, operation is possible only when the tailstock spindle is at the predetermined position. For chuck work, the tailstock spindle must be at the retract end position. These M codes are used to cancel the interlock function.

[Supplement]

- When the power supply is turned off or after the NC is reset, the NC is in the M156 state.
- The state selected by these M codes is effective only for MDI and automatic operation modes.

(28) M160, M161 (feedrate override fixed at 100% OFF, ON)
These M codes are used to specify whether or not the setting of the feedrate override dial, when other than 100%, is valid; in the M161 mode, if the setting of the feedrate override dial on the machine operation panel is in other than 100%, the setting is ignored and the feedrate commands are executed assuming a setting of 100%, and in the M160 mode, the setting of the feedrate override dial is valid.

(29) M162, M163 (rotary tool spindle override fixed at 100% OFF, ON)
These M codes are used to specify whether or not the setting of the rotary tool spindle speed override dial, when other than 100%, is valid; in the M163 mode, if the setting of the rotary tool spindle speed override dial on the machine operation panel is in other than 100%, the setting is ignored and the rotary tool spindle speed commands are executed assuming the setting of 100%, and in the M162 mode, the setting of the rotary tool spindle speed override dial is valid.

(30) M164, M165 (slide hold and single block ignore OFF, ON)
These M codes are used to specify whether or not the slide hold ON and single block ON statuses, set by the switches on the machine operation panel, are valid; in the M165 mode, if the slide hold or single block function is set ON with the corresponding switch on the machine operation panel, these functions are made invalid, and in the M166 mode, if the slide hold or single block function is set ON by the corresponding switch on the machine operation panel, these functions are made valid.

(31) M166, M167 (tailstock spindle advance/retract interlock during spindle rotation ON, OFF)
To ensure safety, the tailstock spindle cannot normally be advanced or retracted while the spindle is rotating. However, tailstock spindle operation is permitted even while the spindle is rotating by turning OFF the interlock.

[Supplement]

- When the power supply is turned off or after the NC is reset, the NC is in the M166 state.
- The state selected by these M codes is effective only for MDI and automatic operation modes.

(32) M184, M185 (chuck open/close interlock ON, OFF)
To ensure safety, the chuck cannot normally be opened or closed while the spindle is rotating. However, chuck open/close operation is permitted even while the spindle is rotating by turning OFF the interlock.

[Supplement]

- When the power supply is turned off or after the NC is reset, the NC is in the M184 state.
- The state selected by these M codes is effective only for MDI and automatic operation modes.
- The state selected by these M codes is effective only when the door is closed.
- The chuck interlock OFF state is effective for chuck clamp/unclamp operation specified by M codes or external commands and it is not effective for the operation using the foot pedal and pushbutton switches.

(33) M193, M194 (thread cutting phase matching control OFF, ON)
In the M194 mode, the phase offset amount at the thread cutting start point is calculated and compensation is carried out at the start and end points. After the completion of the thread cutting cycle, the M194 mode must be canceled by specifying M193 in a block without other commands.

(34) M195, M196 (thread cutting phase matching move amount valid OFF, ON)
By specifying M196 in the block preceding the block which contains the commands to stop a program for thread cutting phase matching, the amount of manual axis movement done in phase matching is stored. M196 must be specified in a block without other commands.
After the completion of manual axis movement for phase matching, the M196 mode must be canceled by specifying M195 in a block without other commands.

(35) M197 (clearing thread cutting phase matching amount)
This M code is used to clear the amount which is stored as the manual axis movement amount for phase matching.

(36) M211, M212, M213, M214 (key-way cycle cutting mode; uni-directional, zigzag, specified cutting amount, equally-divided cutting amount)
M211 and M212 are used to specify the cutting direction in the key-way cutting cycle; uni-directional cutting (M211) and zigzag cutting (M212).
M213 and M214 are used to specify the infeed pattern; M213 specifies the specified cutting amount and M214 specifies the equally-divided cutting amount.

(37) M241, M242 (rotary tool spindle speed range, LOW, HIGH)
These M codes are used to select the spindle speed range of the rotary tool spindle for the multiple-process specification models; low-speed range (M241), high-speed range (M242).

5. M-tool Spindle Commands

5-1. Programming Format

```
 %
N001    G00     X1000     Z1000                T∆∆□□
N002                                           M110
N003    G094    X∆∆∆      Z∆∆∆      C∆∆∆       M15(M16)  SB=∆∆∆∆
N004    G01     X(Z)∆∆∆      F∆                M147  M13  (M14)

                Program block for rotary tool maching

N100            X(Z)∆∆∆
N101    G00     X1000     Z1000                M146
N102                                           M109
N103                                           M02
```

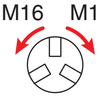
LE33013R0300700060001

[Details]

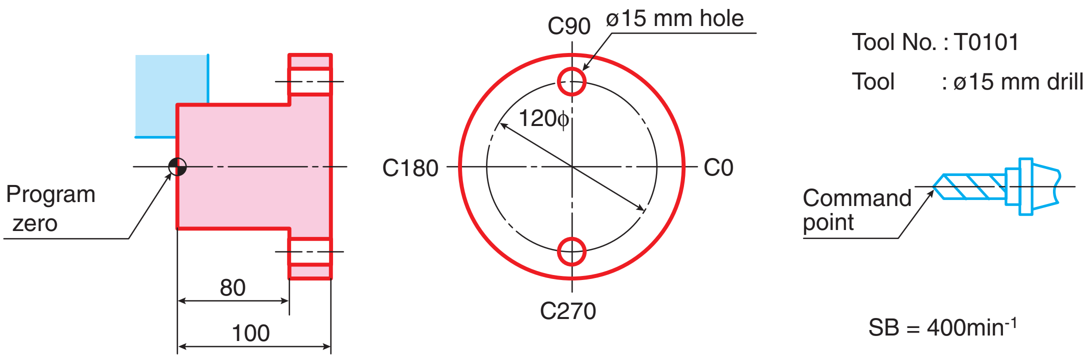
- M110 must be programmed in a block without other commands.
- It is advisable to limit the direction of rotation of the C-axis to either of the two directions, M15 or M16, for better positioning accuracy.
- M110 and M147 cannot be reset or canceled even when the control system is reset. To cancel them, specify M109 and M146, respectively.
- If commands relating to M-tools are specified while the C-axis is not engaged, an alarm occurs. An alarm does not occur if the M-tool spindle interlock (optional) is designated.

5-2. M Codes Used for C-axis Operation

The following codes are necessary for programming C-axis movements.

Code	Details	
M110	Used to designate the spindle to be controlled in the C-axis control mode. When programming C-axis commands, first specify M110 in a block without other commands.	
M109	Used for switchover from the C-axis control mode to the spindle control mode.	
M147	Used to clamp the C-axis.	
M146	Used to unclamp the C-axis. The control system automatically selects the M146 mode when the power is turned on. Program M146 before starting C-axis rotation.	
M141	C-axis clamp ineffective (compound fixed cycle mode)	
M15	Used to rotate the C-axis in the positive direction.	 Chuck end face
M16	Used to rotate the C-axis in the negative direction.	
QA =	Used to specify the number of C-axis revolutions. For example, QA=5 rotates C-axis five times.	

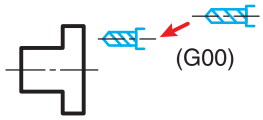
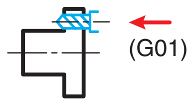
* When the NC is reset, it is placed in the M15 mode.
[Example of Program]



LE33013R0300700070002

To drill two 15 mm dia. holes, create a program as indicated below:

Continued from turning operation program

N099	G00	X1000	Z1000		M05	
N100					M01	
N101					M110	Designates the spindle as the C-axis.
N102				M15		Indexes C-axis in the positive direction.
N103	G94	X120	Z102	C90	T0101 SB=400	The spindle indexes at the 90° position in positive direction and the drill is positioned near the workpiece surface at the rapid feedrate.  Feedrate in mm/min mode is selected.
N104	G01		Z75	F40	M13 M147	The drill starts rotation at 400 min^{-1}{rpm} in the leftward direction. After thespindle is clamped, 15 mm dia. hole is drilled at a feedrate of 40 mm/min. 
N105	G00		Z102			The drill returns to the commanded point at the rapid feedrate.
N106				C270	M146	The spindle is indexed at the 270° position after it is unclamped.
N107	G01		Z75		M147	The second hole is drilled after the spindle is clamped.
N108	G00		Z102			
N109	G095	X1000	Z1000		M12 M146	The M-tool stops and the turret returns to the turret index position.
N110					M109	
N111					M02	

LE33013R0300700070003

- Calculate the feedrate (mm/min) for drilling with the equation below:
  Feedrate (mm/min) = Tool speed (rpm) x Feedrate (mm/rev)
  Therefore, when the tool speed is 400 min-1{rpm} and the feedrate is 0.1 mm/rev, the feedrate (mm/min) is calculated as:
  F = 400 x 0.1 = 40 mm/min

- When an end mill is used, its feedrate (mm/min) is calculated with the following equation:
  Feedrate (mm/min) = Tool speed (rpm)
  Feed (mm/blade)
  Number of end mill blades

- Assuming an end mill with four blades (flutes) is used at 300 min-1{rpm} and a feedrate of 0.05 mm/blade, the feedrate (mm/min) is
  F = 300 x 0.05 x 4 = 60 mm/min

6. STM Time Over Check Function

The duration of S, T, M cycle time is measured and if the measured time exceeds the parameter-set cycle time, an alarm occurs.

6-1. Check ON Conditions

- The check function is set effective or ineffective according to the setting for a machine parameter.

- The check function is turned on and off using the following M codes.

  M124 : STM time over check start

  M125 : STM time over check end

6-2. S, T, M Cycle Time Setting

Set, for the machine parameter, the allowable limit of cycle time when executing an S, T, and M codes.

- Parameter setting

  Units : 0.1 seconds

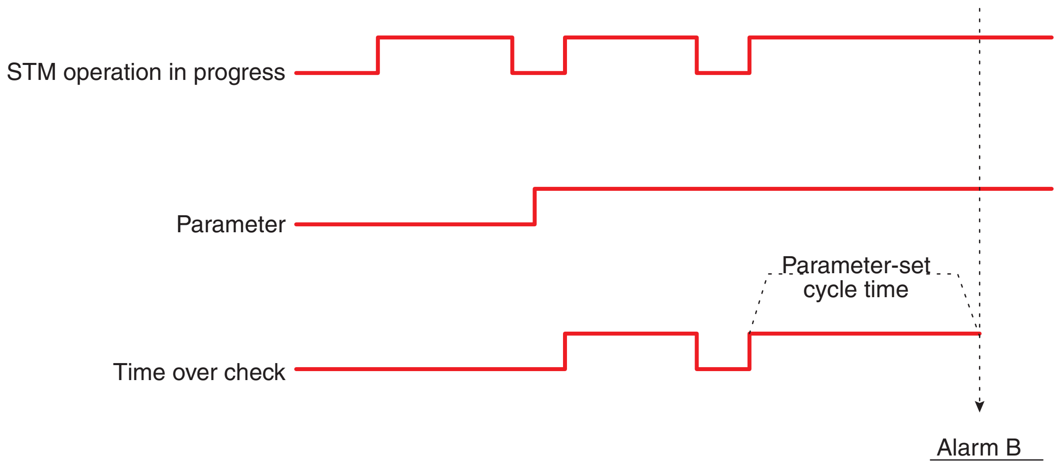
  Maximum setting : 600 seconds

6-3. Timing Chart Example

(1) Parameter setting

Parameter: ON STM time over check start

Parameter: OFF STM time over check end

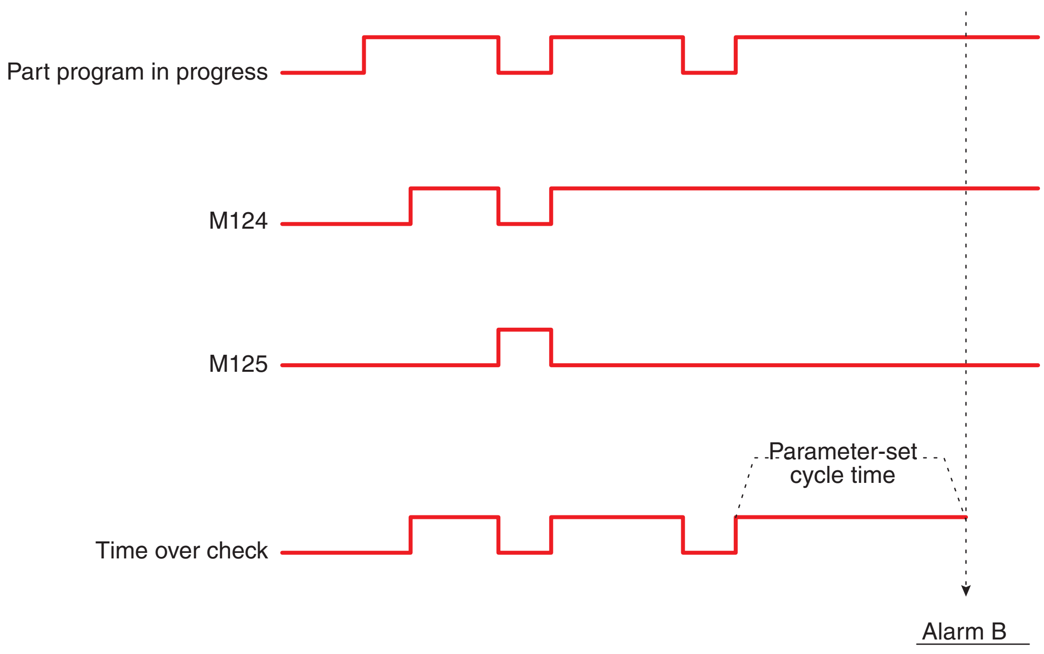


LE33013R0300700110001

(2) M Codes

M124 : STM time over check start

M125 : STM time over check end



LE33013R0300700110002

SECTION 6 OFFSET FUNCTION

1. Tool Nose Radius Compensation Function (G40, G41, G42)

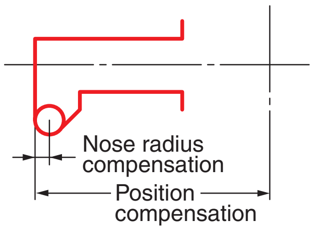
1-1. General Description

The tool tip point radius of most cutting tools used in turning operation is the cause of inconsistencies between the designated tool paths and the actually finished workpiece contour. With the tool radius compensation function, such geometric error is automatically compensated for by simple programming.

1-2. Tool Nose Radius Compensation for Turning Operations

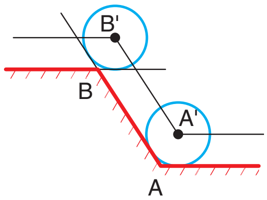
Tool Offset and Nose Radius Compensation

In turning operations, various types and different shapes of tools are used to finish one workpiece. ID cutting tools, OD cutting tools, rough cut tools, finish cut tools, drills, etc. Accordingly, the tool nose radius compensation function has to be activated simultaneously with the tool offset function.



LE33013R0300800020001

Tool Nose Radius Compensation at Discontinuous Point



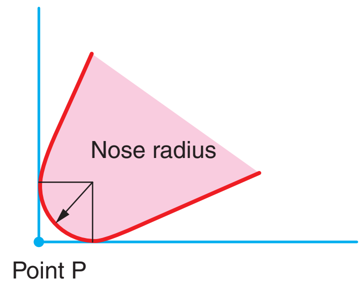
LE33013R0300800020002

Point A in the figure above constitutes a discontinuous point and an angle less than 180°. By using the tool nose radius compensation function, the tool path shown above can be generated by simply entering the coordinates of points A and B.

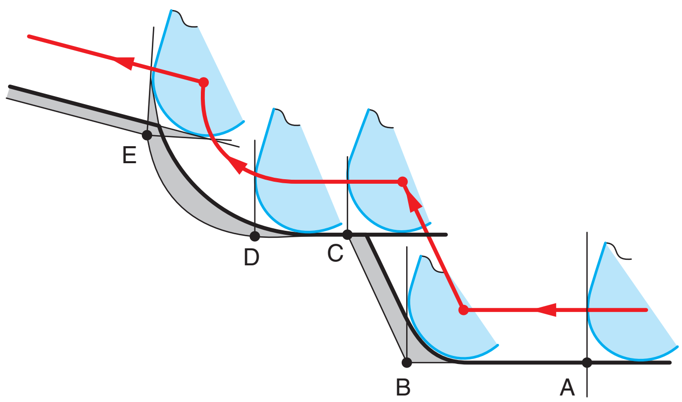
1-3. Compensation Operation

Geometrical Cutting Error due to Tool Nose Radius

If cutting along paths A-B-C-D-E in the figure below is intended but the tool nose radius compensation function is not activated, the shaded portions will remain uncut and cause geometrical errors. This is because the tool setting is made to locate the imaginary cutting point P at the datum point and trace the programmed path as controlled by NC commands. However, the actual cutting tip point is not precisely located on that datum point because of the tool nose radius and this produces geometrical errors.
The tool nose radius compensation function automatically compensates for the inconsistency between the designated and actual tool paths caused by the tool nose radius (see the figure below).



Tool Setting Point

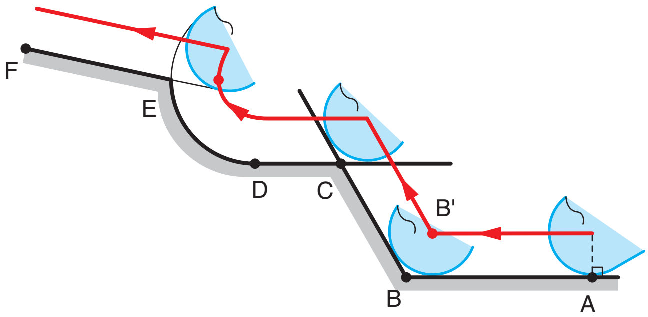


Tool Path and Resulting Error Without Tool Nose Radius Compensation

LE33013R0300800030001

Compensation Movement

With the tool nose radius compensation function activated, the error in the tool path described in (1) is compensated for as shown below to finish the workpiece to the dimensions specified in a program.

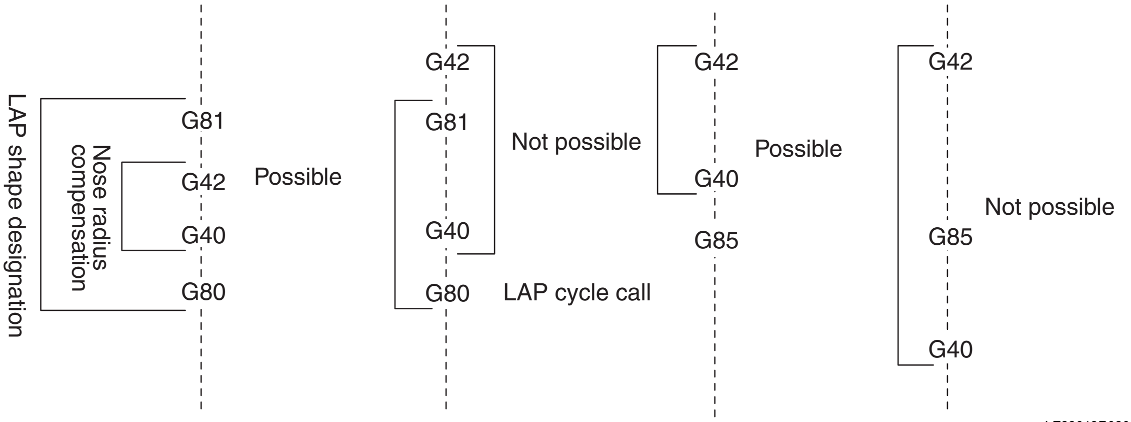


Tool Path with Tool Nose Radius Compensation

LE33013R0300800030002

Nose radius compensation during LAP mode

To use the tool nose radius compensation function in the LAP mode, programs for the respective turrets must contain the tool nose radius compensation programs independently as shown below.



LE33013R0300800030003

1-4. Nose Radius Compensation Commands (G, T Codes)

The programming commands - G and T codes, used to activate the tool nose radius compensation function, are detailed in this section.

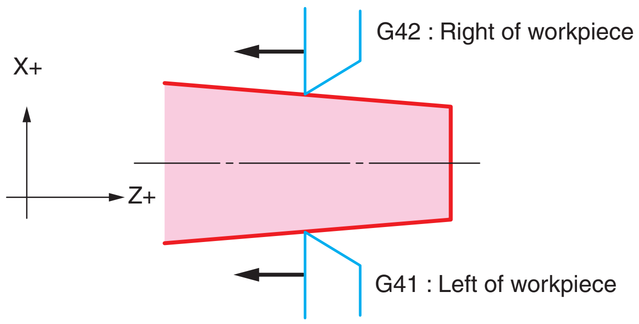
G Codes

G40 : Used to cancel the tool nose radius compensation mode.

G41 : Tool nose radius compensation - Left
Used when the tool moves on the left side of the workpiece.

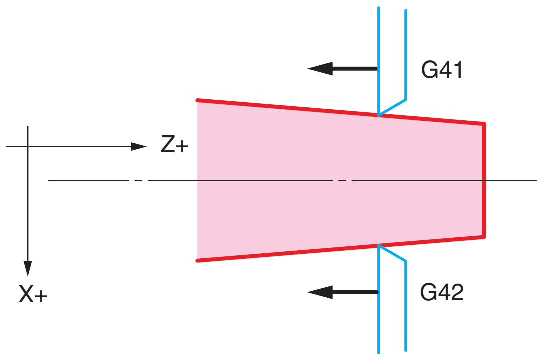
G42 : Tool nose radius compensation - Right
Used when the tool moves on the right side of the workpiece.

The term indicating the side of the workpiece, right or left, is determined according to the direction in which the tool is advancing.



LE33013R0300800040001

Since G41 and G42 codes are selected to agree with the coordinate system (right-hand system) the machine employs, they should be selected as below for lathes which have a coordinate system in which the positive direction of the X-axis is directed toward the operator.



LE33013R0300800040002

T Codes

Six numerical characters following address character "T" specify the nose radius compensation number, tool number, and tool offset number.

TOOΔΔ□□

OO: Tool nose radius compensation number

ΔΔ: Tool number

□□: Tool offset number

LE33013R0300800040003

[Supplement]

To change the tool offset during the execution of tool nose radius compensation, designate the tool nose radius compensation number and the tool number.

Example:

G01	Xa	Za		T010101······1)
G03	Xb	Zb	K	
G01		Zd		T110111······2)
G03	Xd	Zd	I	

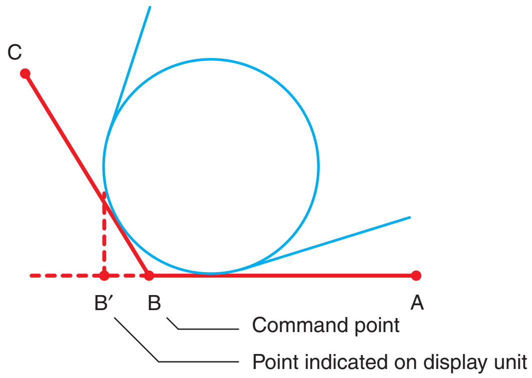
LE33013R0300800040004

Entry of only the tool offset No. (T01 or T11) in G code command (1) or (2) will cancel the nose radius compensation amount.

1-5. Data Display

The screen display during nose radius compensation is described here.

(1) Actual Position
Actual position data is displayed on the screen as with the conventional control system. However, the data displayed on the screen may be different from the programmed data because of the tool nose radius compensation.

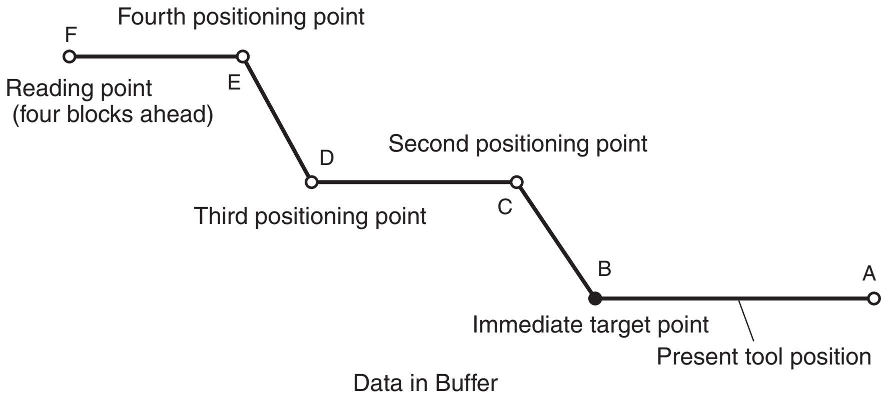


LE33013R0300800050001

(2) Alarm Display
If an alarm relating to the tool nose radius compensation function occurs, the ALARM light under STATUS DISPLAY goes on and the screen displays the message indicating the alarm contents.

1-6. Buffer Operation

The NC usually operates in the 3-buffer mode. While the positioning command from point A to point B is being executed, the positioning point data of points C, D and E are read and stored in the buffer. This is called the 3-buffer function.
When the tool nose radius function is activated, the target point E is calculated from straight lines DE and EF. This means that the data in the block four blocks ahead the current target point are read if the tool nose radius compensation function is active.

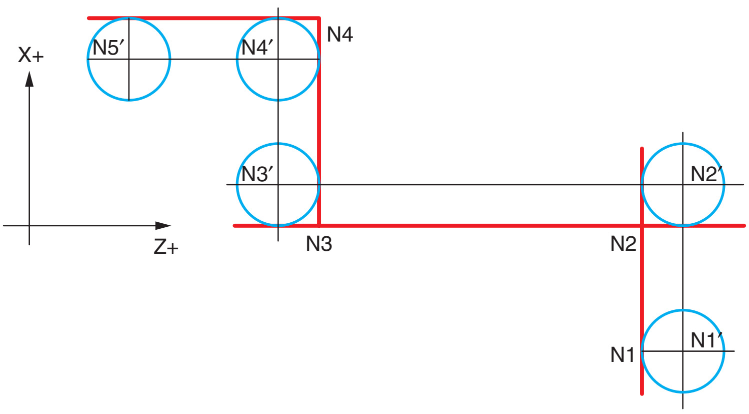


LE33013R0300800060001

1-7. Path of Tool Nose "R" Center in Tool Nose Radius Compensation Mode

To execute the motion shown below in the following program in the tool nose radius compensation mode, the path of the tool nose R center is obtained as follows:

N1	G42	X1	Z1
N2		X2	Z2
N3		X3	Z3
N4	G41	X4	Z4
N5		X5	Z5



LE33013R0300800070001

(1) To obtain point N2’ when the center of the tool nose R is at point N1’, proceed as follows:

- Draw a straight line parallel to the direction of tool advance, N1 - N2, offset in the specified direction, (to the right since G42 is specified), by the tool nose radius compensation amount. This yields the straight line passing N1’ and N2’.

- Draw a straight line parallel to the direction of tool advance, N2 - N3, offset in the specified direction, (to the right of or above N2 - N3 since G42 dominates the compensation mode) by the tool nose radius compensation amount. This yields the straight line passing N2’ and N3’.

- The nose R center for the commanded point N2’ is the point of intersection of these two straight lines.
  The center of the tool nose radius advances from point N1’ to N2’.

(2) To obtain point N3’:

- Draw a straight line parallel to the direction of tool advance, N2 - N3, offset in the specified direction, (to the right of or above N2 - N3 since G42 dominates the compensation mode), by the tool nose radius compensation amount. This yields the straight line passing N2’ and N3’.

- Draw a straight line parallel to the direction of tool advance, N3 - N4, offset in the specified direction, (to the left since G41 is specified), by the tool nose radius compensation amount. This yields the straight line passing N3’ and N4’.

- The nose R center for commanded point N3 is the point of intersection of these two straight lines.
  The center of the tool nose radius advances from point N2’ to point N3’.

(3) To obtain point N4’:
Follow the same procedure indicated above using points N3, N4 and N5.

1-8. Tool Nose Radius Compensation Programming

1-8-1. G41 and G42

The G41 and G42 codes are used to call out the tool nose radius compensation mode. Since the uses of these G codes are often confused in programming a part, this section deals with their particular differences.

G41 : This tool nose radius compensation code is used when the cutting tool moves on the left side of the workpiece in relation to its direction of advance.

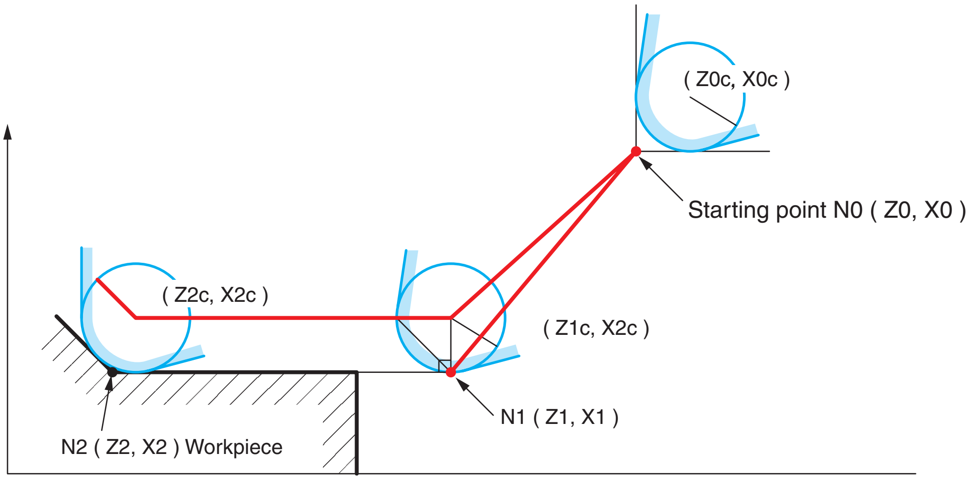
G42 : This tool nose radius compensation code is used when the cutting tool moves on the right side of the workpiece in terms of its direction of advance.

1-8-2. Behavior on Entering Tool Nose Radius Compensation Mode

N0	G00	X0	Z0	
N1	G42	X1	Z1	TOOOOOO
N2	G01	X2	Z2	

LE33013R0300800090001

The following example uses the program above to perform OD cuts with an OD turning tool.

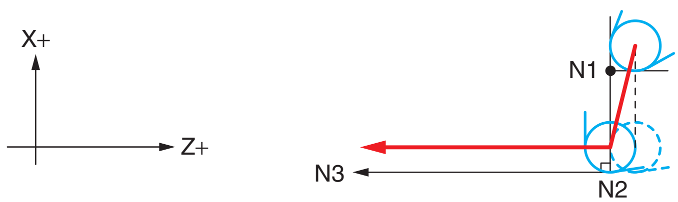


LE33013R0300800090002

Without the tool nose radius compensation function, positioning is performed so that the tool tip reference point is located exactly at the programmed coordinates. At the start up of the tool nose radius compensation mode activated by either G41 or G42, positioning is carried out so that the tool tip circle contacts the segment passing the programmed coordinates in the block containing G41 or G42 and those in the next block. This motion of the axes is called "Start-Up".

- At the start up of the tool nose radius compensation mode, both X- and Z-axis may move even if the block contains only one dimension word, either X or Z.

N1	G00	X100	Z100	S1000	T010101	M3
N2	G42	X80				
N3	G01		Z50	F0.2		

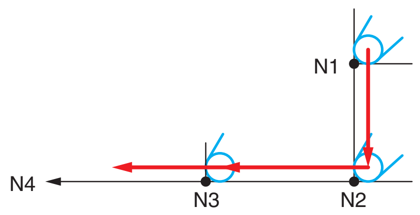


LE33013R0300800090003

Although the programmer might expect the axis movement indicated by broken lines because the N2 block contains only an X word, the actual tool path generated at the start up of the tool nose radius compensation mode is as shown by solid lines.

- Example of an ideal program for entry into the compensation mode:

```
N1   G00   X100  Z100   S1000   T010101   M3
N2         X80
N3   G42         Z90
N4   G01         Z50    F0.2
      :           :
```

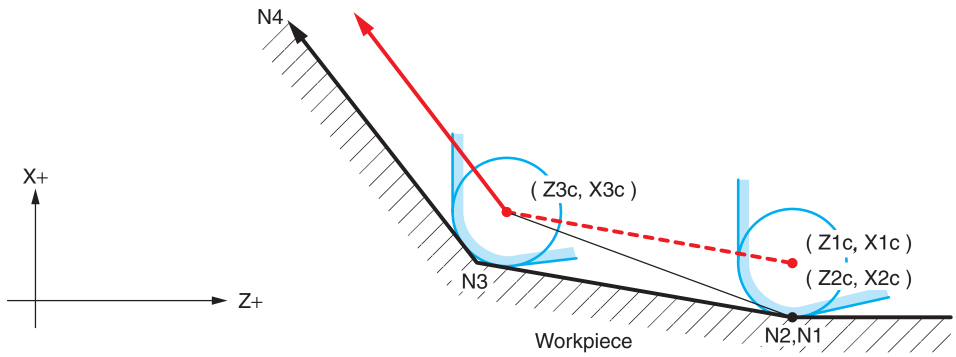


LE33013R0300800090004

In this program, the G42 block contains only a Z word, and points N2, N3 and N4 are all positioned on the same straight line.

- Either G00 or G01 must dominate the operation mode when entering into the tool nose radius compensation mode. Otherwise, an alarm will occur.

- When neither an X nor Z word is presented at the start up of the tool nose radius compensation mode, or when the point where the axes are presently located is specified in the start-up block, positioning is executed so that tool tip circle comes in contact with the segment passing through the designated coordinates and the coordinates in the next sequence. The tool nose radius compensation motion is activated from the following sequence.




```
N1   G00   X100  Z100  F0.2  S1000   T010101   M3
N2   G42
N3         X60   Z80
N4         X100  Z50
```

LE33013R0300800090005

With the program above, the tool tip circle is positioned so that it comes into contact with segments N2N3 and N3N4. That is, the blocks of commands after N3 sequence are all executed in the tool nose radius compensation mode.

- If the same point as in the start-up block is specified in the succeeding block, an alarm will result if the successive two blocks after that do not have dimension words, X and Z.
  Faulty program example 1:

```
N1   G01   X50    Z100   F0.2   S1000    T010101    M3
N2   G42
N3         X50    Z100
N4         X60    Z80
N5         X100   Z50
```

LE33013R0300800090006

Since sequence N3 designates a point identical to the one designated in the start-up sequence N2, an alarm occurs.
Faulty program example 2:

```
N1   G01   X50   Z100   F0.2   S500    T010101    M3
N2   G42
N3                             S1000
N4                             M08
N5         X50   Z100
N6         X60   Z80
```

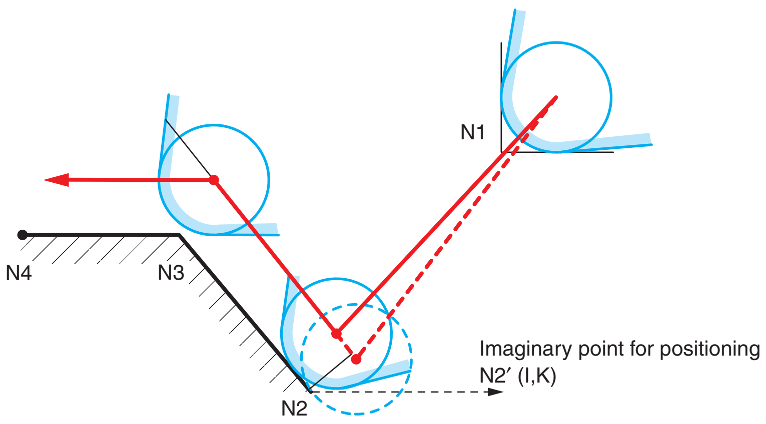
LE33013R0300800090007

Since sequences N3 and N4, the successive two sequences after the start-up of the tool nose radius compensation mode, do not contain X and Z axis movement commands, an alarm occurs.

- I and K command with G41 and G42

  In the block containing G41 and G42, by entering I and K words that specify the imaginary point, along with X and Z words that specify the nose radius compensation start-up, unnecessary axis motion required in conventional start-up program is eliminated.

N1	G00	X100	Z100	F0.2	S1000	T010101	M3
N2	G42	X60	Z80	K20			
N3	G01	X80	Z65				
N4			Z50				



LE33013R0300800090008

If block N2 containing G42 had no I and K words, positioning of the cutting tool by the commands in block N2 would be executed so that the tool nose radius comes into contact with line N2-N3 at designated point N2 and then moves to N3.
Addition of I and K words in block N2 positions the cutting tool to the point where the tool nose R is brought into contact with straight line N2-N3 and imaginary straight line N2-N2’ when the commands in block N2 are executed. Execution of the commands in block N3 brings the cutting tool to the programmed point N3 where the tool nose radius compensation is not active.

[Supplement]

- I and K words should be commanded in incremental values. In this case the dimensions are referenced to point N2.
- When only either I or K is provided without the other, the control interprets the word to have the value "0". Therefore, KO in the above program can be omitted.

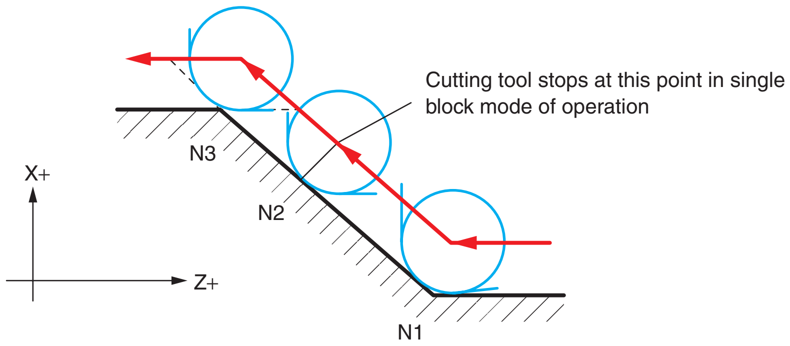
1-8-3. Behavior in Tool Nose Radius Compensation Mode

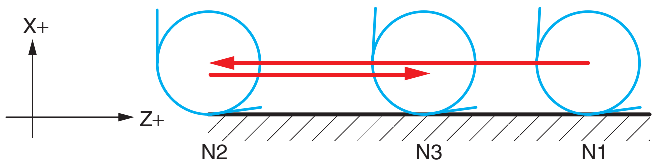
The tool nose radius compensation function provides the means to automatically compensate for the tool nose radius in continuous cutting.
Since such compensation is performed automatically, there are some restrictions in programming when the tool nose radius compensation function is used.

Straight line to straight line cutting

- Midpoint on a straight line
  When specifying a midpoint on a straight line, the point should be commanded carefully.
  When point N2 in the figure below is located on line N1 - N3, the cutting tool is positioned so that the tool tip circle comes into contact with line N1 - N3 at point N2.

- Returning along a straight line
  Such axis movement causes no problem when the program is written without using the tool nose radius compensation function.
  However, when this function is used the axis movements must be programmed carefully.
  Program Example:

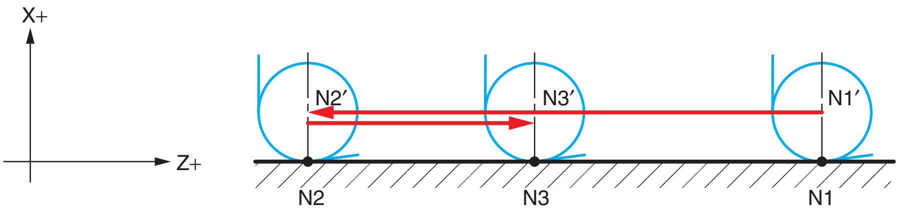




N1	G42	G01	X1	Z1
N2			X2	Z2
N3	G41		X3	Z3

LE33013R0300800100001

In this example points N2 and N3 are commanded while the cutting tool is at point N1.
When the cutting tool advances from point N1 to point N2, G42 is designated since the cutting tool moves on the right side of the workpiece with respect to the direction of tool advance. However, in the return motion of the tool from point N2 to point N3, the cutting tool is on the left side of the workpiece with respect to the direction of tool advance. Therefore, G41 is specified instead of G42.



LE33013R0300800100002

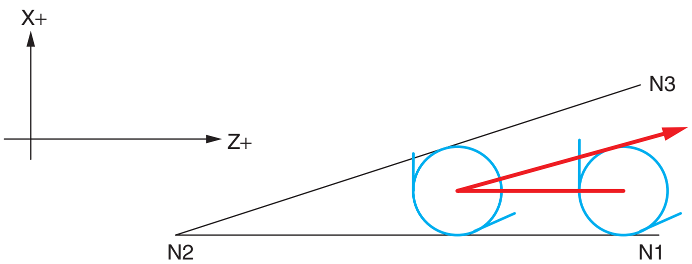
The axis movements above are possible by the special processing for the tool nose radius compensation function. Let’s consider the operation in this program in the light of section 1-7. "Path of Tool Nose "R" Center in Tool Nose Radius Compensation Mode."

a. The center of the tool nose R (N2’) at point N2 is obtained as follows:

- The line parallel to the straight line N1 - N2 is obtained, with an upward offset (G42) by the tool nose radius amount effective at N1.
- The line parallel to the straight line N2 - N3 is obtained, with an upward offset (G41) by the tool nose radius amount effective at N2.
- The center of the tool nose R is obtained as the point of intersection of the two straight lines obtained in steps in 1) and 2). However, since those two lines are parallel to each other, no point of intersection is obtained in this case. For such case, the control has a special processing feature in which the positioning is carried out so that the tool nose R comes into contact with point N2. Therefore, the path of the tool nose R center, when the cutting tool advances from point N1 to point N2, is obtained as N1’ - N2’.

b. The center of the tool nose R (N3’) at point N3 is obtained in the same manner as in 1).
In this way, the program on the previous page can return the cutting tool along the same straight line with the tool nose radius compensation function active.
If any of these three points is not precisely located on the same straight line, the tool path will be shifted considerably from the expected path.

- Two lines making an acute angle
In the figure below, although positioning from N1 to N2 is intended, the cutting tool cannot reach point N2. This is because it can move only up to the point where the tool nose R comes into contact with line N2 - N3.

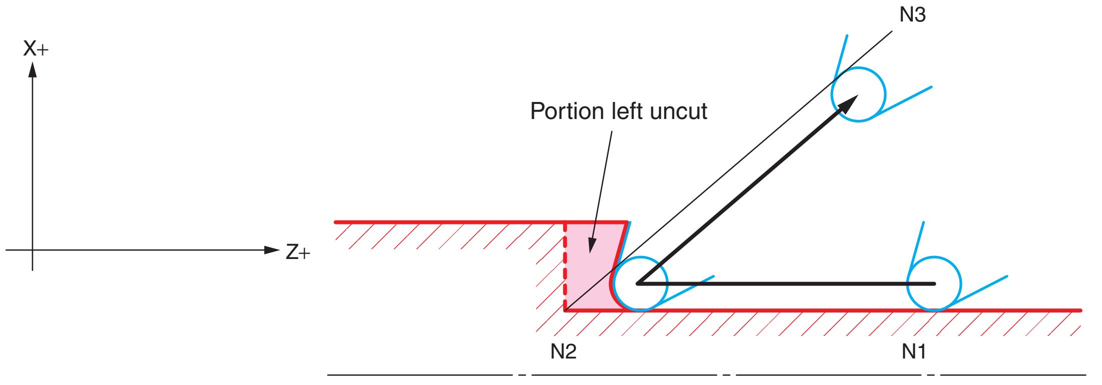


LE33013R0300800100003

This example illustrates a case where programmers are apt to be confused. Another example is provided below.

Example of faulty program 1 (completion of cutting):

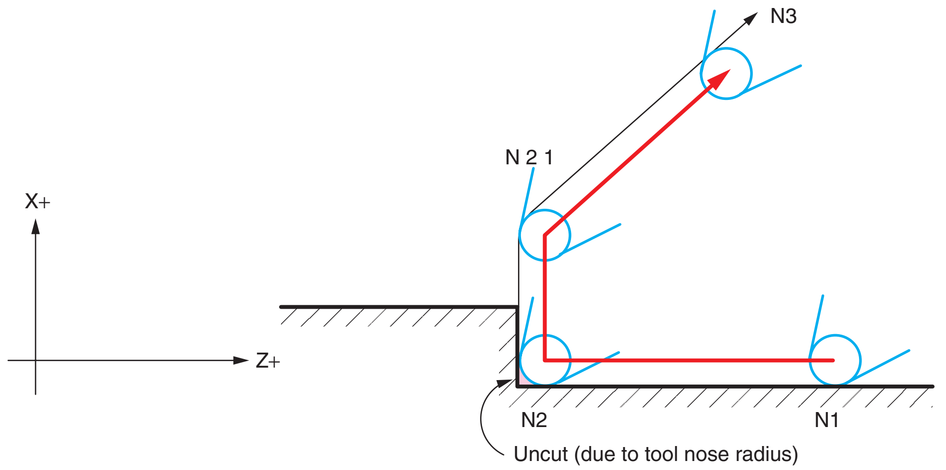
```
N1   G42   G01   X100   Z100   F0.2    S1000    T010101    M3
N2                      Z50
N3   G00         X300   Z300   M05
```



LE33013R0300800100004

With the program above, the programmer expected to cut up to point N2, (i.e., up to Z50) allowing a slight uncut portion on the sharp corner due to tool nose R. Contrary to this intention, however, the cutting tool leaves a considerable uncut section since it stops before reaching the desired point.
To improve such a program, enter one more point in the program as shown below:
Example of improved program 1:

```
N1   G42   G01   X100   Z100   F0.2    S1000    T010101    M3
N2                      Z50
N21              X104 ................... [ > 100 + 4 x (nose R) ]
N3   G00   X300         Z300                    M05
```

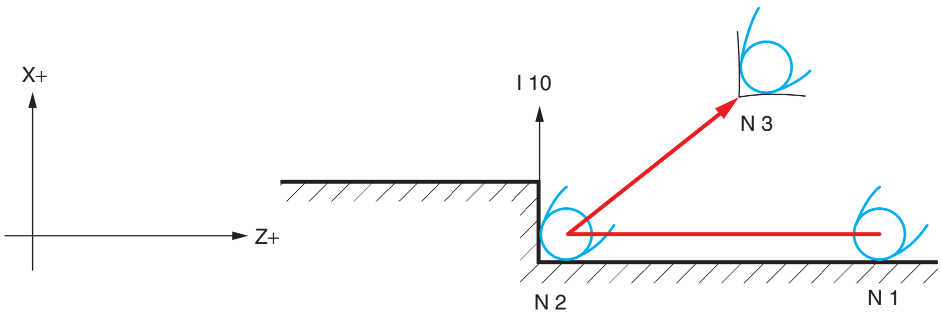


LE33013R0300800100005

The improved program generates the tool path shown above, and almost all the cutting can be accomplished as expected except for a slight uncut section due to the tool nose R.
To relieve the tool along X-axis in the positive direction in the N21 block, an X word must have a value larger than four times the nose R. This is because a distance twice the nose R is

necessary for the tool tip circle to fit in. In addition, because X words are expressed as diameters, the X word data has to be doubled. That is, the numerical value in such an X word must be larger than four times the tool nose R.
If a value smaller than the required amount is used, it might cause the cutting tool to move in the opposite direction toward point N21 and cut into the N1 - N2 surface.
Example of improved program 2 (using G40):

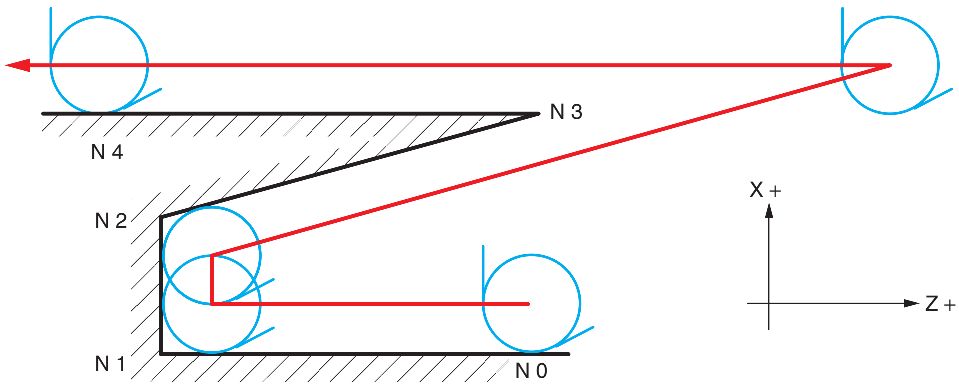
```
N1   G42  G01  X100  Z100  F0.2  S1000  T010101  M03
N2                   Z50
N3   G40  G00  X300  Z300  I10   M05
```



LE33013R0300800100006

The G40 command in N3 cancels the tool nose radius function. At point N2, the cutting tool moves so that the tool nose R contacts the line N1 - N2 and the vector I10 extending from point N2.

- Two lines making an obtuse angle
  Consider the case where the cutting tool is fed along the path N0 - N1 - N2 - N3 - N4 in the figure below.
  Angle N2N3N4 is an acute angle and the cutting tool moves along the line outside of that angle. Therefore, the cutting tool is moved to a point some distance from the workpiece at point N3.
  When preparing a program in which cutting similar to this contour is required, it is necessary to check the safety of tool motion and ensure that the tool does not strike against obstacles when moving to such a distant point.



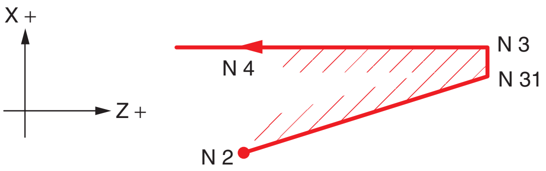
LE33013R0300800100007

Example program for the path above:

N0	G42	G00	X100	Z300		S1500	T010101	M03
N1	G01			Z100	F0.2			
N2			X104 .... [> 100 + 4 × (nose R)]					
N3	G00		X200	Z300				
N4	G01			Z50	S1000			

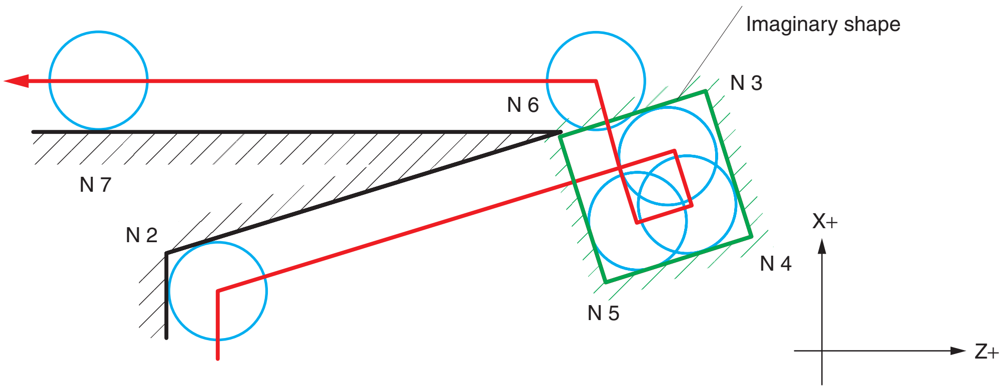
LE33013R0300800100008

It is advantageous to improve the program and eliminate a positioning sequence to a distant point through commands in the N3 block.
If N2N3N4 were not a sharp angle, such a problem would not occur. To eliminate sharp angles from the required contour, one possible solution is to interpose a short straight line N3 - N31.



LE33013R0300800100009

In some cases, such a modification is not possible. In these cases, to cut a sharp angle without positioning the cutting tool at a distant point, follow the steps detailed below.
Example of Improved Program:

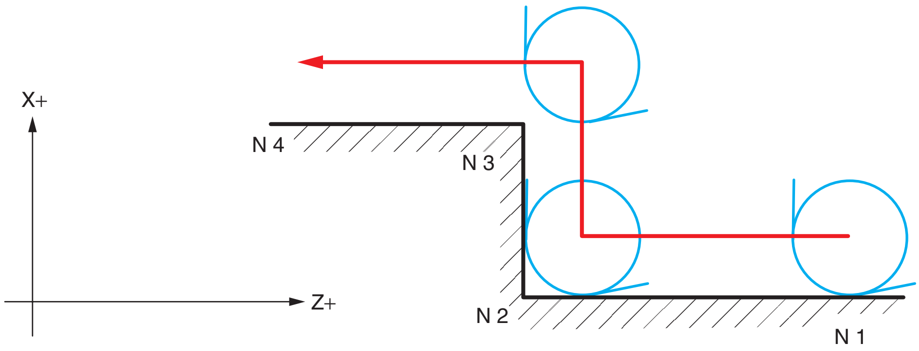


N0	G42	G00	X100	Z300		S1500	T010101	M03
N1	G01			Z100	F0.2			
N2			X104					
N3	G00		X200.48	Z301				
N4			X198.48	Z301.24	F1			
N5			X198	Z300.24				
N6	G01		X200	Z300				
N7				Z50	F0.2	S1000		

LE33013R0300800100010

In this improved program, the cutting tool moves along the imaginary square N3N4N5N6. This permits the operator to estimate the departure of the cutting tool from the programmed contour. Note that one side of the imaginary square must be longer than twice the nose radius.

- Two lines forming a right angle



N1	G42	G01	X100	Z100	F0.2	S1000	T010101	M03
N2				Z60				
N3			X150					
N4				Z20				

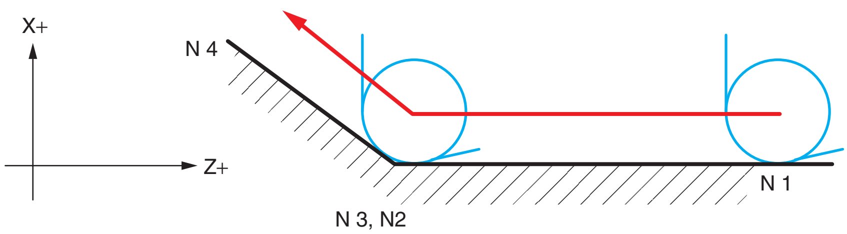
LE33013R0300800100011

There are no particular problems in this case.

- Command of identical point

a. If a block without axis movement commands is programmed during the tool nose radius compensation mode, the path of the tool nose R is the same as the one generated when there is no such block.

N1	G42	G01	X50	Z100	F0.2	S1000	T010101	M03
N2				Z80				
N3								
N4			X60	Z70				M08



LE33013R0300800100012

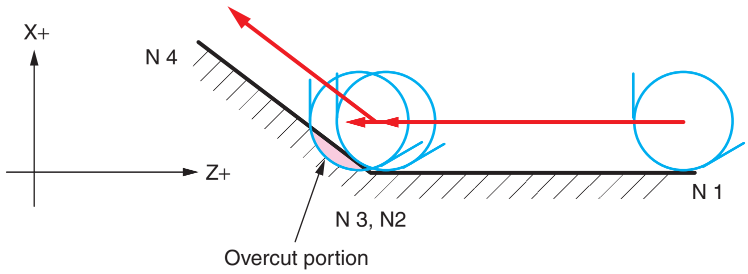
b. When two or more blocks without axis movement commands are programmed, or when the same point as commanded in the preceding sequence is repeatedly commanded during the tool nose radius compensation mode:
In this case, an axis motion that brings the tool nose R into contact with the programmed contour at the programmed coordinate point takes place. When the block of commands containing dimension words, X and/or Z, is read, the cutting tool returns to the correct compensated position.

Program 1:

N1	G42	G01	X50	Z100	F0.2	S1000	T010101	M04
N2				Z80				
N3				Z80				M08
N4			X60	Z70				

LE33013R0300800100013

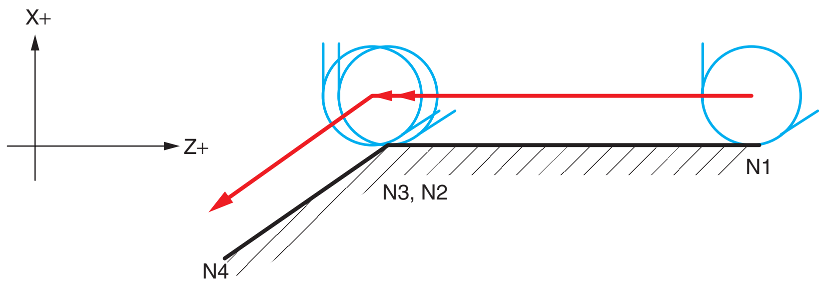
A program like this might cause overcutting as shown below:



LE33013R0300800100014

Depending on the contour to be cut, the unexpected motion may not result in overcut, as in program 2.
Program 2:

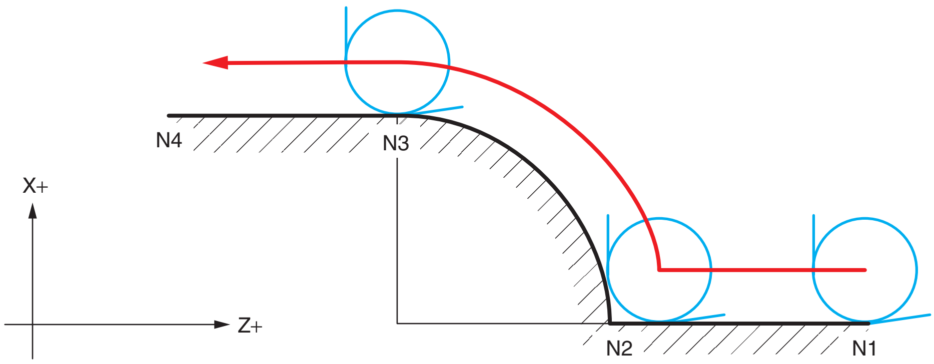
N1	G42	G01	X50	Z100	F0.2	S1000	T010101	M04
N2				Z80				
N3				Z80				
N4			X40	Z70				M08



LE33013R0300800100015

Straight line to arc cutting (arc to straight line cutting)

- Arc within one quadrant
  In a program where the cutting tool moves continuously from a straight line to an arc, the movement of the cutting tool is handled in the same way as in a case where the movement is from a straight line to a straight line.



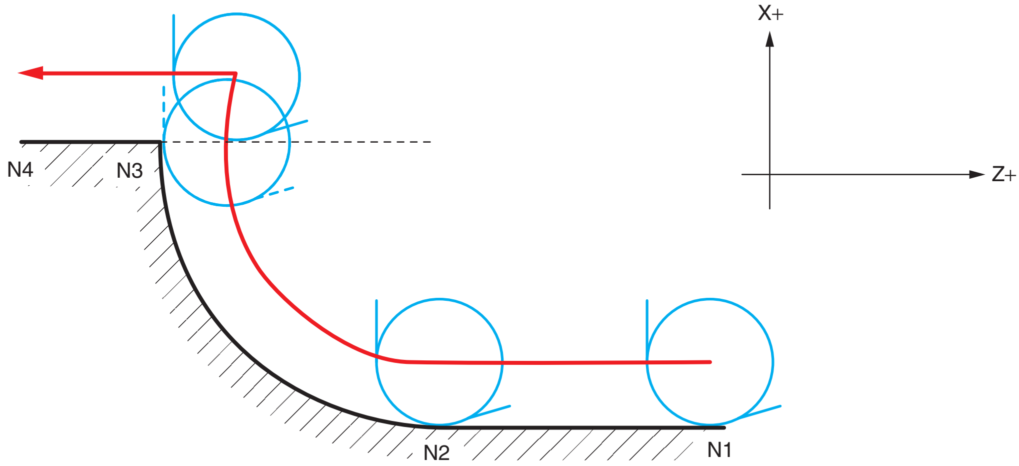
N1	G42	G01	X100	Z100	F0.2	S1000	T010101	M04
N2				Z80				
N3	G03		X140	Z60	K – 20			
N4	G01			Z40				

LE33013R0300800100016

The tool position at point N2 is determined so that the tool nose R comes into contact with both line N1 - N2 and arc N2 - N3. At point N3, the cutting tool is positioned in a similar way - the tool nose R makes contact at point N3.
When the cutting tool moves from point N3 to point N4, the cutting mode changes from circular interpolation to linear interpolation. If discontinuity at point N3 results during the tool path calculation, an alarm is displayed and machine operation is stopped.

- Arc in two quadrants

a. Case where the arc radius is greater than "2 x nose R":

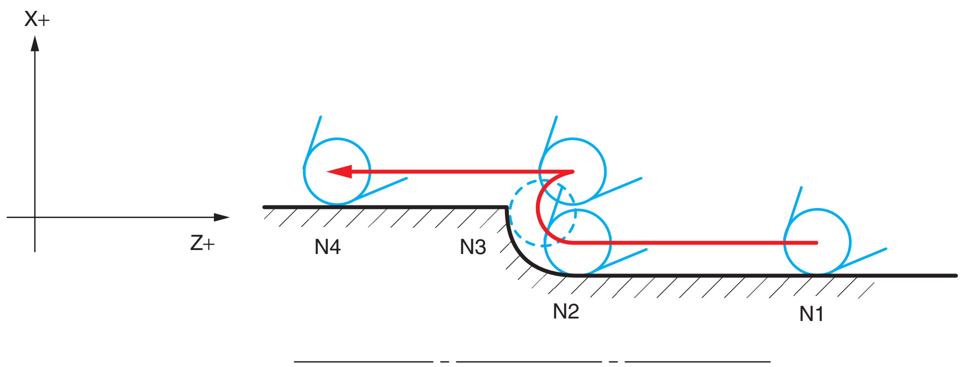


N1	G42	G01	X100	Z100	F0.2	S1000	T010101	M04
N2				Z80				
N3	G02		X140	Z60	I20			
N4	G01			Z40				

LE33013R0300800100017

The tool position determined by the commands in the N2 block is the point where the tool nose R comes into contact with line N1 - N2 at point N2. In the N3 sequence, the cutting tool is positioned so that it comes into contact with both the extension of straight line N2 - N3 and the extension of arc N3 - N4.

b. Case where the arc radius is equal to "2 x nose R":



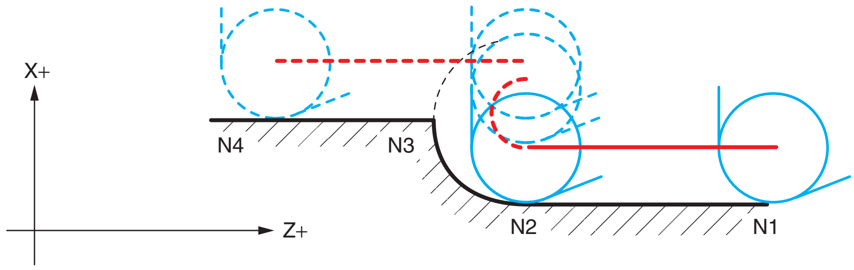
Tool nose R = 0.8 mm

N1	G42	G01	X100	Z100	F0.2	S1000	T010101	M04
N2				Z80				
N3	G02		X103.2	Z78.4	I1.6			
N4	G01			Z40				

LE33013R0300800100018

When the radius of the programmed arc equals twice the tool nose R, the cutting tool is located at the point where the tool nose R comes into contact with both the extension of arc N2 - N3 and the extension of straight line N3 - N4, after the execution of the commands in N3 block (see the figure in "1)" above). That is, the cutting tool is positioned right above point N2, as shown in the figure directly above.

c. Case where the arc radius is less than "2 x nose R" (impossible):



N1	G42	G01	X100	Z100	F0.2	S1000	T010101	M04
N2				Z80				
N3	G02		X102	Z79	I1			
N4	G01			Z40				

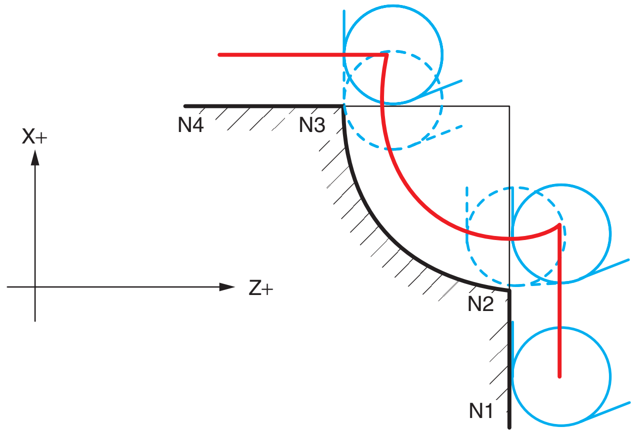
LE33013R0300800100019

The commands in block N3 specify positioning of the cutting tool at the point where the tool nose R comes into contact with both the extension of arc N2-N3 and the extension of straight line N3-N4; however, such a point cannot be obtained. Therefore, when the control executes the commands in block N3, an alarm occurs and the machine stops.
In this kind of case, cutting using the tool nose radius compensation function is not possible.

[Supplement]

When cutting inside an arc, the programming must satisfy the following condition: $R \ 2 \times R_N$ (where R: arc radius, R_N: nose R)

- Arc in three quadrants



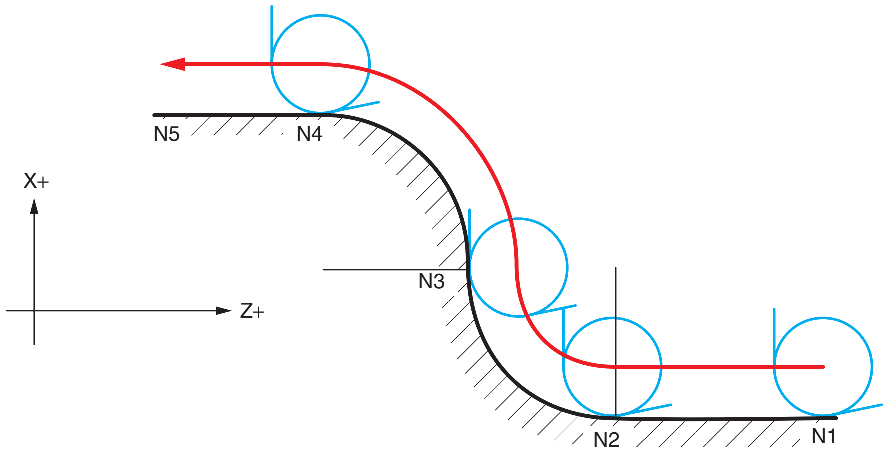
N1	G42	G01	X100	Z100	F0.2	S1000	T010101	M04
N2			X120					
N3	G02		X160	Z80	I20			
N4	G01			Z60				

LE33013R0300800100020

Positioning by the commands in block N2 is to the point where the tool nose R comes into contact with both the extension of straight line N1 - N2 and the extension of arc N2 - N3.
Other axis motions of the cutting tool are identical to those for cutting an arc in two quadrants.

Arc to arc cutting

Arc to arc cutting can be programmed in the same manner as straight line to arc cutting.
The tool path is generated so that the tool nose R is brought into contact with each arc or its extension.
If the tool path becomes discontinuous in the process of path calculation due to an error, the machine stops with an alarm displayed on the screen.
Other motions of the cutting tool are as explained in (2), "Straight line to arc cutting".



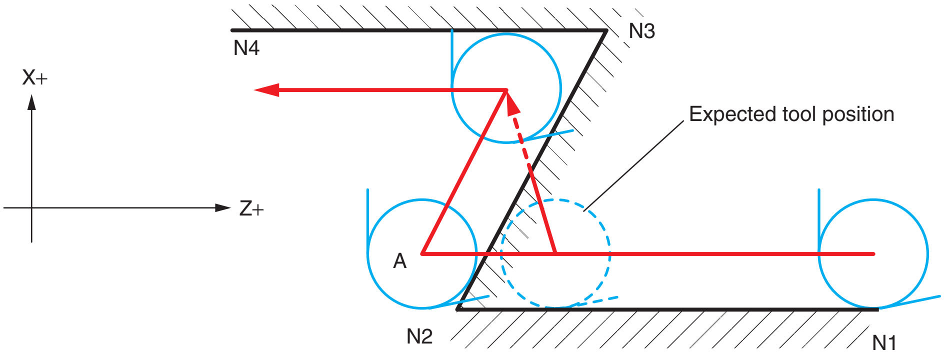
N1	G42	G01	X100	Z100	F0.2	S1000	T010101	M04
N2				Z80				
N3	G02		X140	Z60	I20			
N4	G03		X180	Z40	K – 20			
N5	G01			Z20				

LE33013R0300800100021

Switching from G41 to G42 or from G42 to G41

Before switching the tool nose radius compensation mode from G41 to G42 or from G42 to G41, it is advisable to cancel the compensation mode by specifying G40.
If a switch-over is to be done with the compensation mode active, carefully check the movement of the cutting tool resulting from the switch-over.

- Switch-over in straight line to straight line cutting
  Program Example:



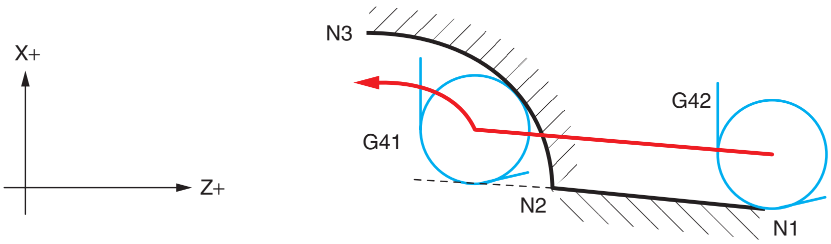
N1	G42	G00	X_1	Z_1	T
N2	G01		X_2	Z_2	F
N3	G41	G00	X_3	Z_3	
N4			X_4	Z_4	

LE33013R0300800100022

The motion of the cutting tool generated by the above program is as follows:
Commands in blocks, N1 and N2 are governed by G42 and those in blocks N3 and later are governed by G41. To position the cutting tool at point N2, the tool nose R center lies to the right side of straight line N1 - N2 since block N2 is in the G42 mode. As for block N3, the tool nose R center lies to the left side of straight line N2 - N3 since block N3 is in the G41 mode. As a result, the cutting tool is positioned at point A as shown above.
Positioning in block N2 is carried out at the left side of straight line N2 - N3.

- Switch-over in straight line to arc cutting
  The concept is the same as for straight line to straight line cutting.

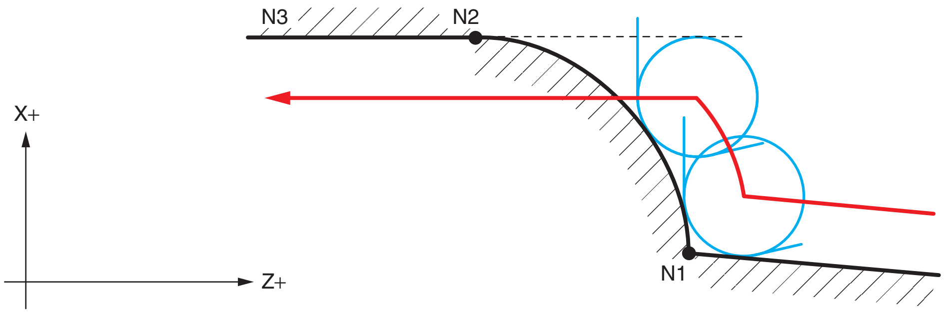
N1	G42	G01	X_1	Z_1	F_1	T
N2			X_2	Z_2		
N3	G41	G03	X_3	Z_3	I_3	K_3



LE33013R0300800100023

- Switch-over in arc to straight line cutting
  Again, the concept is the same as for straight line to straight line cutting.

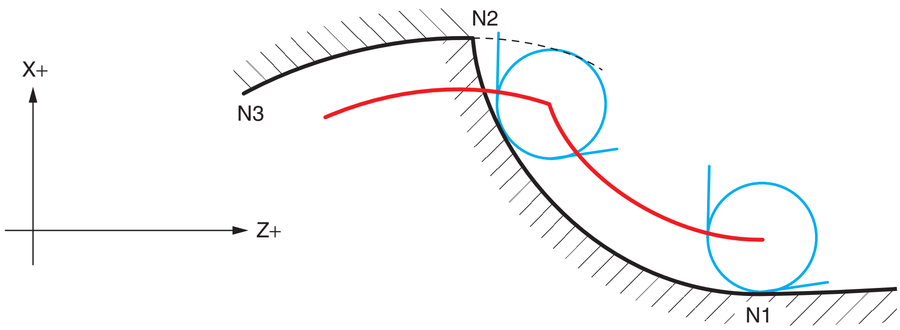
N1	G42	G01	X_1	Z_1	F_1	T
N2	G03		X_2	Z_2	I_2	K_2
N3	G41	G01	X_3	Z_3		



LE33013R0300800100024

- Switch-over in arc to arc cutting
  Once again, the concept is the same as for straight line to straight line cutting.

N1	G42	G01	X_1	Z_1		F_1		T
N2	G02		X_2	Z_2	I_2		Z_2	
N3	G41		X_3	Z_3	I_3		Z_3	

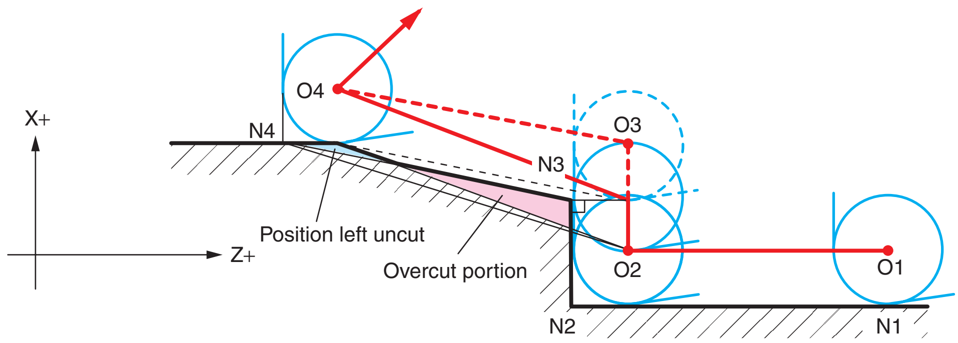


LE33013R0300800100025

1-8-4. Behavior on Cancelation of the Tool Nose Radius Compensation Mode

G40 given with X- or Z-axis motion command

To cancel the tool nose radius compensation mode, the G40 code is used. It is essential to understand the cutting tool movements that result from the cancelation of the compensation mode in order to avoid unexpected trouble.
In the tool nose radius compensation mode, the tool path is generated so that the tool nose R is always in contact with the programmed contour, but the axis position is controlled so that the tool tip reference point traces the programmed contour when the tool nose radius compensation mode is not active. Therefore, under- or over-cut often results when entering into or when canceling the tool nose radius compensation mode.



LE33013R0300800110001

Cutting a contour comprising straight line segments as illustrated above is programmed as shown below if the tool nose radius compensation mode is not active.

```
N1   G01   X100   Z100   F0.2   S1000   T010101   M03
N2                Z60
N3         X120
N4         X130   Z20
N5   G00   X300   Z300
```

LE33013R0300800110002

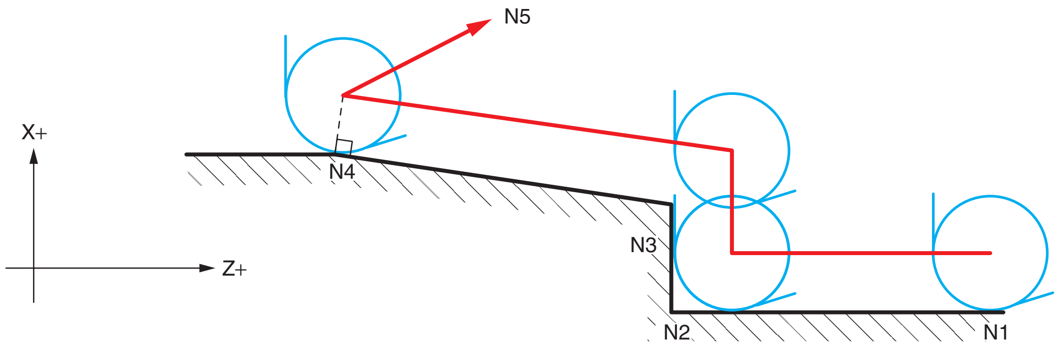
With the commands above, the cutting tool moves along the path indicated by broken lines. That is, for designated point N3 the tool center is positioned at point O3, and at point O4 for designated point N4.
The uncut part parallel to straight line N3 - N4 is left. Therefore the tool nose radius compensation function can be effectively used to cut such a contour accurately. See the programs on the following pages.

- When the tool nose R compensation cancel command is designated:

```
N1   G42   G01   X100   Z100   F0.2   S1000   T010101   M03
N2                      Z60
N3               X120
N4   G40         X130   Z20
N5   G00         X300   Z300
```

LE33013R0300800110003

The tool path generated in the above program is shown by solid lines.
Positioning fort programmed point N3 is carried out at the point where the tool nose R comes into contact with point N3, and that for programmed point N4 is carried out at point O4; the same point reached by the program in which the tool nose radius compensation function is not activated.
Therefore, the uncut part will be near point N4 while the section near point N3 is overcut.
Improved program:

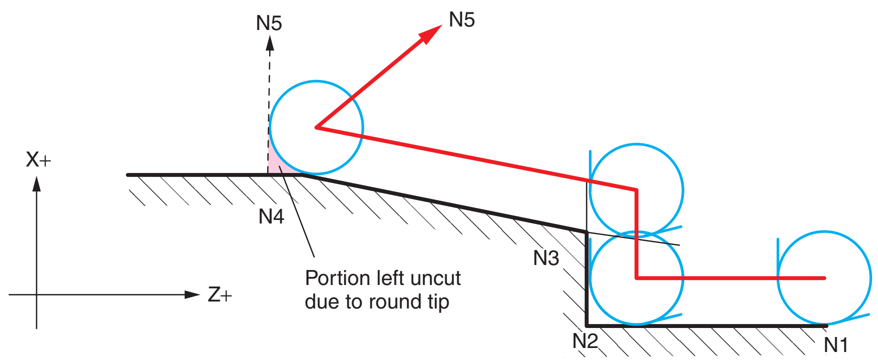


N1	G42	G01	X100	Z100	F0.2	S1000	T010101	M03
N2				Z60				
N3			X120					
N4			X130	Z20				
N5	G40	G00	X300	Z300				

LE33013R0300800110004

To cut the exact contour up to Point N4, the G40 command which cancels the tool nose radius compensation mode is specified in block N5.
Although the program yields almost the expected contour, the tool nose R goes beyond the designated point N4 along Z-axis since it comes into contact with line N3 - N4 at point N4. When this kind of overtravel causes no interference or overcutting, there are no problems.

- Eliminating possible overcutting along Z-axis, see the program below:




```
N1    G42   G01   X100   Z100   F0.2   S1000     T010101     M03
N2                       Z60
N3                X120
N4                X130   Z20
N5    G40   G00   X300   Z300   I10
```

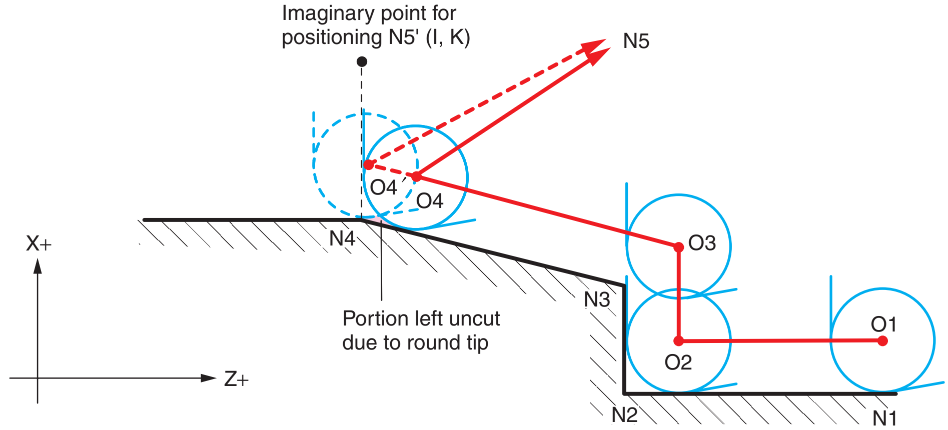
LE33013R0300800110005

I and K words specified in the G40 block allow the tool to move to the point where the tool nose R is brought into contact with both line N3 - N4 and line N4 - N5.

I and K command with G40

In the block containing G40, by entering I and K words that specify the imaginary point along with X and Z words that specify the point where nose radius compensation is canceled, unnecessary axis motion required in conventional canceling program is eliminated.

```
N1    G42   G01   X100   Z100   F0.2   S1000     T010101     M03
N2                       Z60
N3                X120
N4                X130   Z20
N5    G40   G00   X300   Z300   I10    K0
```



LE33013R0300800110006

If block N5 containing G40 has no I and K words, positioning of the cutting tool by the commands in block N4 is executed so that the tool nose R comes into contact with line N3 - N4 at designated point N4 and then moves along the path indicated by broken lines toward point N5.
Addition of I and K words in block N5 positions the cutting tool to the point where the tool nose R is brought into contact with straight line N3 - N4 and imaginary straight line N4 - N5' when the commands in block N4 are executed. Execution of the commands in block N5 brings the cutting tool to the programmed point N5 where tool nose radius compensation is not active.

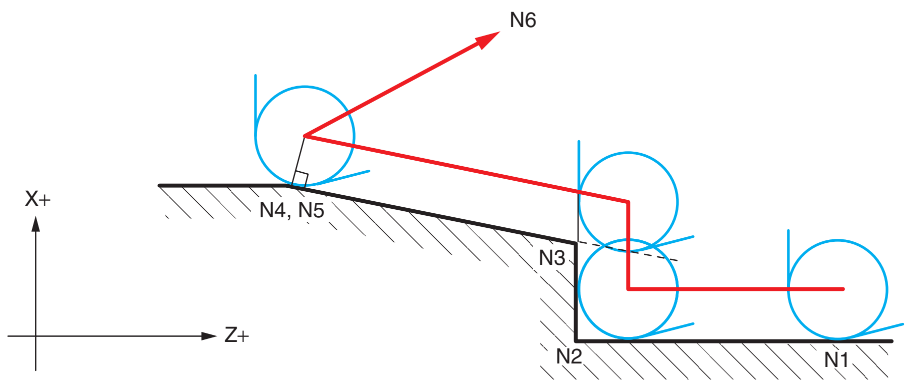
[Supplement]

- I and K words should be commanded as incremental values. In this case the dimensions are referenced to point N4.
- When either I or K only is specified without the other, the control interprets the word to have the value "0". Therefore, K0 in the above program can be omitted.

Independent G40

When the G40 code is programmed without other commands in the same block, positioning is carried out at the point where the tool nose R comes into contact with the point specified in the previous block since the G40 block has no X and Z words which call for axis movement.

```
N1   G42   G01   X100   Z100   F0.2   S1000    T010101    M03
N2               Z60
N3               X120
N4               X130   Z20
N5   G40
N6   G00         X300   Z300
```



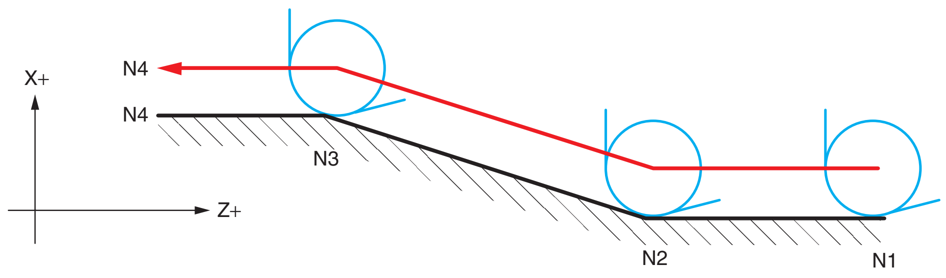
LE33013R0300800110007

When the tool nose radius compensation mode is canceled (G40), the mode of operation must be either G00 or G01. If not, an alarm occurs.

1-8-5. Relieving Tool to Change "S" or "M" Code during Cutting

The tool nose radius compensation function is designed to automatically compensate the tool nose radius in a continuous cutting program; with the dimensions of the workpiece programmed, compensation is automatically applied to finish the part to the programmed dimensions. However, such a powerful function requires careful programming when continuous cutting is interrupted to change S and/or M commands.
This section deals with some programming examples in which the programmer experienced unexpected results by relieving the cutting tool during cutting on a continuous path.

Original contour and associated program (program 1):



LE33013R0300800120001

Program 1:

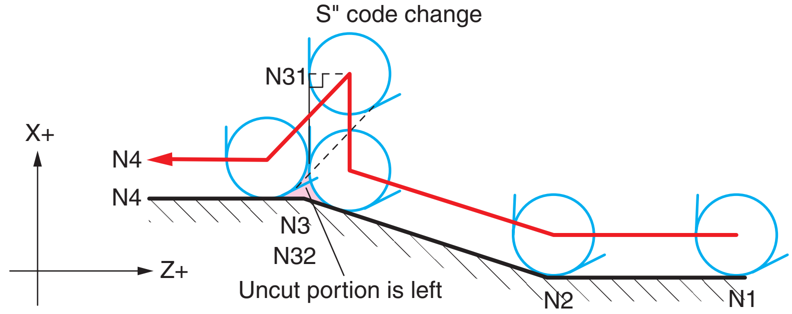
N1	G42	G01	X100	Z100	F0.2	S1500	T010101	M03
N2				Z80				
N3			X120	Z40		S1000		
N4				Z20				

LE33013R0300800120002

The original contour comprises: straight line - slope - straight line.

Program 2

The contour is the same as in program 1, but the cutting tool is relieved at point N3 in the +X direction to change the spindle speed, then continuous cutting is intended.



LE33013R0300800120003

Program 2:

N1	G42	G01	X100	Z100	F0.2	S1500	T010101	M03
N2				Z80				
N3			X120	Z40				
N31	G00		X124					
N32	G01		X120			S1000		
N4				Z20				

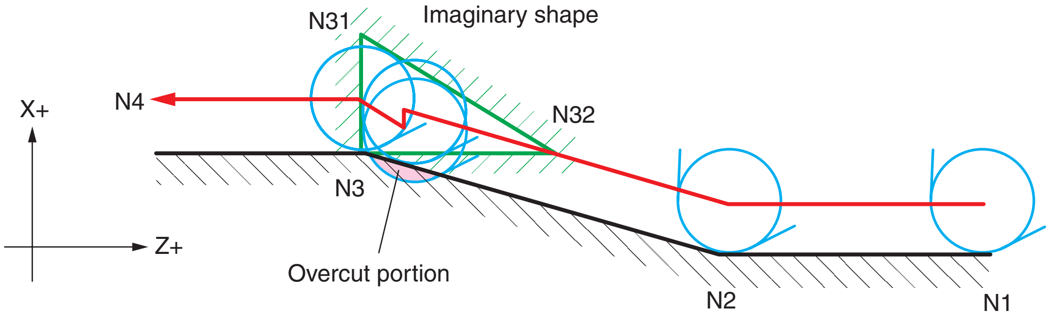
LE33013R0300800120004

In program 2, the cutting tool is positioned at a point where the tool nose R is in contact with line N3 - N31 at point N31 when the commands in block N31 are executed since the three designated points

N3, N31 and N32 lie on the same straight line. From N3 to N31, the positioning is on the right hand side of the line. Commands in block N32 position the cutting tool at the point where the tool nose R is brought into contact with straight lines N31 - N32 and N3 - N4 on the right side of the direction of tool advance. This causes the cutting tool to move not only in the X-axis direction but also in the Z-axis direction although block N32 contains only an X word.
Such cutting tool movements leave an uncut portion as shown above.

Program 3

In this program, an attempt is made to eliminate the uncut portion caused by program 2.



LE33013R0300800120005

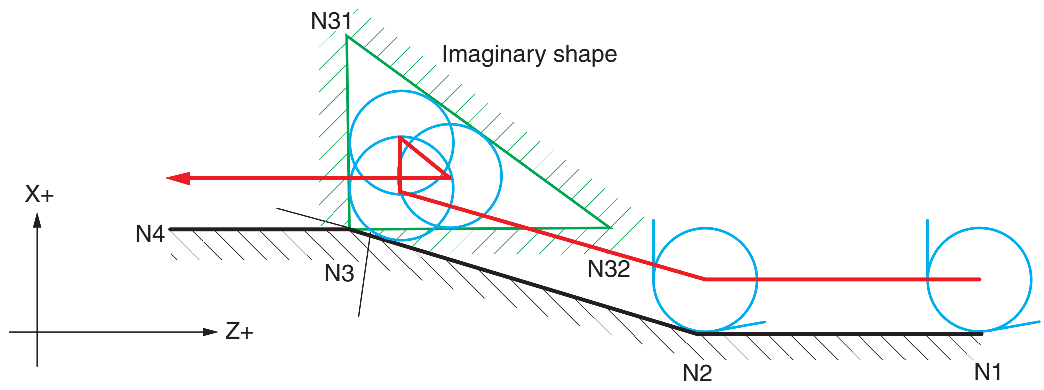
Program 3:

N1	G42	G01	X100	Z100	F0.2	S1500	T010101	M03
N2				Z80				
N3			X120	Z40				
N31	G00		X124					
N32			X120	Z42		S1000		
N4		G01		Z20				

LE33013R0300800120006

When the control feeds the cutting tool from point N2 to point N3, it reads the position data of point N31 as well as those of point N3. This permits the tool nose R to be positioned at the point where it is in contact with the two straight lines N2 - N3 and N3 - N31.
After that, positioning is carried out at the point where the tool nose R comes into contact with the two straight lines N3 - N31 and N31 - N32, when positioning is performed with the commands in block N31. This moves the cutting tool in the -X direction although the commands in that block specify tool movement in the +X direction. This is due to the positioning in block N3, where the tool nose R goes beyond side N31 - N32.
Similarly, positioning of the cutting tool in block N32 is carried out at the point where the tool nose R comes into contact with both straight lines N31 - N32 and N32 - N4. This also causes the cutting tool to move in the direction opposite to the programmed direction. The result is overcutting.

Program 4



LE33013R0300800120007

Program 4:
In this program, a tool looping similar to that performed in program 3 is executed with the numeral values modified to avoid overcutting.

```
N1   G42  G01  X100  Z100  F0.2  S1500   T010101   M03
N2                   Z80
N3              X120 Z40
N31  G00        X126
N32             X120 Z43         S1000
N4   G01             Z20
```

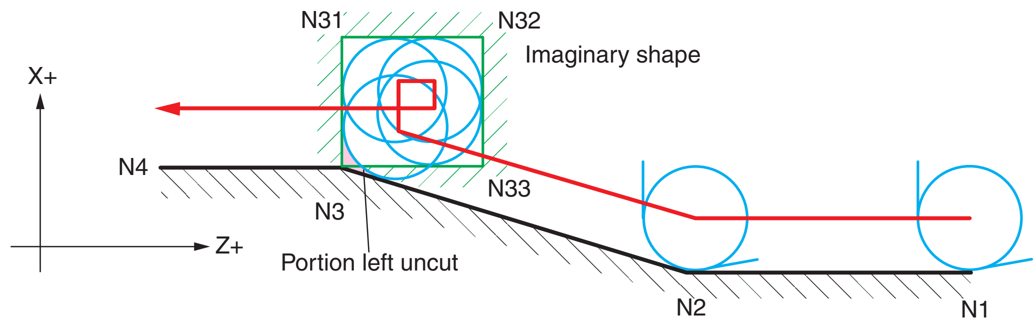
LE33013R0300800120008

This program almost yields the expected finish. However, there are still latent problems, such as:

- overcutting is caused depending on the size of the tool nose R
- the length of side N31 - N32 cannot be readily found.
  These problems are solved by looping the tool path along a square as explained next.

Program 5

Program 5 solves the problems encountered with program 4.



LE33013R0300800120009

Program 5:

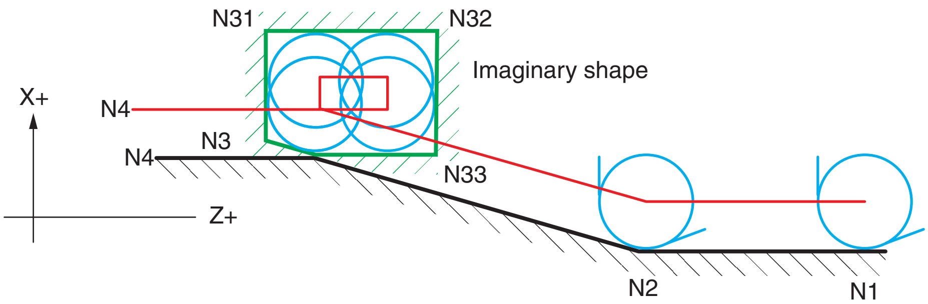
N1	G42	G01	X100	Z100	F0.2	T010101	M03
N2				Z80			
N3			X120	Z40			
N31	G00		X124······(> 120 + 4 × (nose R))				
N32				Z42	S1000······(> 40 + 2 × (nose R))		
N33	G01		X120				
N4				Z20			

LE33013R0300800120010

In this looping path, the tool nose R moves inside the programmed rectangle, N3 - N31 - N32 - N33. Therefore, axis behavior can be easily expected if only these respective sides are longer than twice the tool nose R (four times on the X-axis).
Since this program still leaves an uncut portion, it can be further improved as indicated in program 6.

Program 6

In this program, point N3 is shifted in the -Z direction by an amount equivalent to the tool nose R to eliminate the uncut part seen in Program 5. This program gives a fully satisfactory result.



LE33013R0300800120011

Program 6:

N1	G42	G01	X100	Z100	F0.2	S1500	T010101	M03
N2				Z80				
N3			X120.5	Z39				
N31	G00		X124·············(> 120.5 + 4 × (nose R))					
N32				Z42	S1000··········(> 39 + 2 × (nose R))			
N33	G01		X120					
N4				Z20				

LE33013R0300800120012

Programs 1 through 5 will provide some clues to complete the program for the intended cutting.
As an imaginary shape for tool path looping, select a rectangle or polygon but not a triangle. Triangles are apt to lead to unexpected tool movements.

[Supplement]

1) If either the X- or Z-axis exceed its soft-limit, a "Limit Alarm" results.
2) During the tool nose radius compensation mode, commands that do not cause axis motion, although dimension words are present, (zero offset by G code for instance, or thread cutting fixed cycle (G31, G32 and G33)) cannot be specified.
3) To activate the tool nose radius compensation mode from LAP mode operation, designate G41 or G42 in the block preceded by the one containing G81 or G82 in which the cutting dimensions in the LAP are specified. In LAP mode operation, tool nose radius compensation is active both in rough and finish cut cycles.
   Be sure to enter G40, which cancels the tool nose radius compensation mode, before specifying the end of LAP contour designation code G80.
4) While in the tool nose radius compensation mode, the same point should not be commanded repeatedly. However, one block that does not contain axis motion commands can be programmed; the control is designed to accept such block.
5) At the start up of tool nose radius compensation mode, the control starts execution of the commands after it read in the commands in the successive two blocks. Therefore, pressing the CYCLE START button in the MDI mode after entering the commands for one block cannot start machine operation.
6) Incremental commands (G91) can be provided in the tool nose radius compensation mode.

2. Cutter Radius Compensation Function

2-1. Overview

This function automatically offsets the tool paths to generate the required shape in multi-processing just by programming the final shape.
Using this function, cutters of different diameters can be used to machine workpieces of the same shape without modifying the program.

2-2. Programming

Designation of offset plane (G17, G18, G119)

[Function]
Changeover among the X-Z plane (nose R compensation), the X-Y plane (cutter radius compensation on contour generation machining plane (face)), and C-X-Z plane (cutter radius compensation on contour generation machining plane (side)) is possible by designating the appropriate G code.
[Programming format]

G17 : X-Y plane
(cutter radius compensation on contour generation machining plane, face)

G18 : X-Z plane (nose R compensation)

G119 : C-X-Z plane
(cutter radius compensation on contour generation machining plane, side)

[Details]

- G17 and G119 are effective only while the C-axis is joined.
- When the C-axis control cancel command (M109) is executed, the X-Z plane (G18) is automatically selected.
- When the power is turned on or the control is reset, the X-Z plane (G18) is selected.

Cutter radius compensation function ON/OFF (G40, G41, G42)

[Function]
Turn the cutter radius compensation function on and off by means of G codes.
[Programming format]

G40 : Cutter radius compensation function OFF

G41 : Cutter radius compensation function, left
(viewed in the direction of tool advance, the tool is positioned at the left side of the workpiece)

G42 : Cutter radius compensation function, right
(viewed in the direction of tool advance, the tool is positioned at the right side of the workpiece)

[Details]

- In the G17 plane, compensation can be activated in the following G code modes.
  G00, G01, G101, G102, G103
- In the G119 plane, compensation can be activated in the following G code modes.
  G00, G01, G132, G133

Cutter radius compensation values

[Function]
The cutter radius compensation values are designated using a 6-digit T command.

T O O Δ Δ □□
O O : Tool nose radius compensation number
Δ Δ : Tool number
□□ : Tool offset number

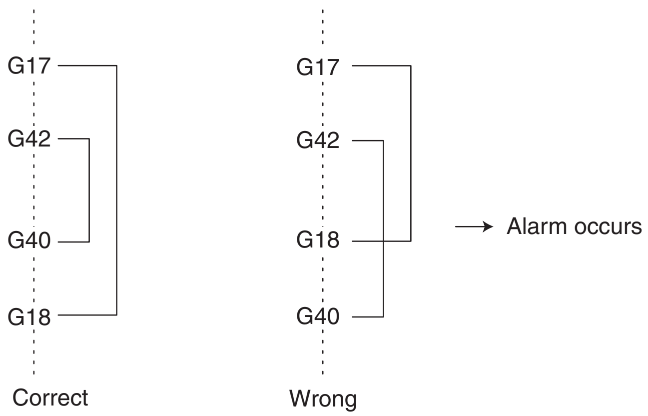
LE33013R0300800140001

[Details]

- Set the cutter radius compensation value in advance at the nose R column in the tool data setting screen.
- Set the same value for both X and Z. If different values are set, the value having larger absolute value takes effect.
- The nose R pattern number is effective only in the G18 (nose R compensation) mode. In the G17 and G119 (cutter radius compensation) modes, it is ignored.

Designation of cutter radius compensation plane and turning on/off the function

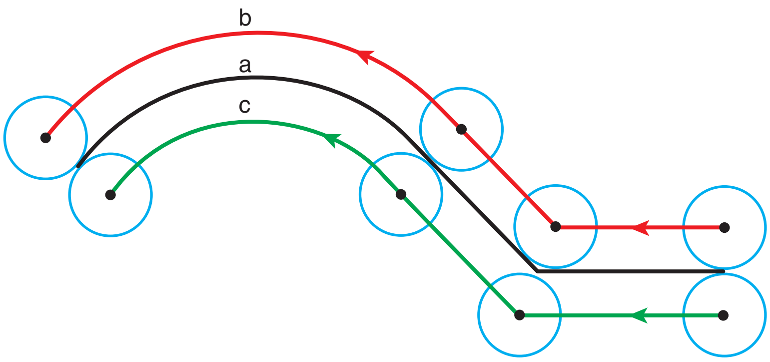
- Before calling the cutter radius compensation function (G41, G42), designate the plane (G17, G18, G119).
- When switching the cutter radius compensation direction (G41, G42), cancel the cutter radius compensation function first by designating G40 before calling the other G code.
- To change the compensation plane, cancel the cutter radius compensation function by designating G40. If G17, G18 or G119 is designated in the G41 or G42 mode, an alarm occurs.



LE33013R0300800140002

2-3. Operations

Tool motion in the G17 and G119 modes with the cutter radius compensation function active, is illustrated below.



LE33013R0300800150001

a : Programmed path (final shape)

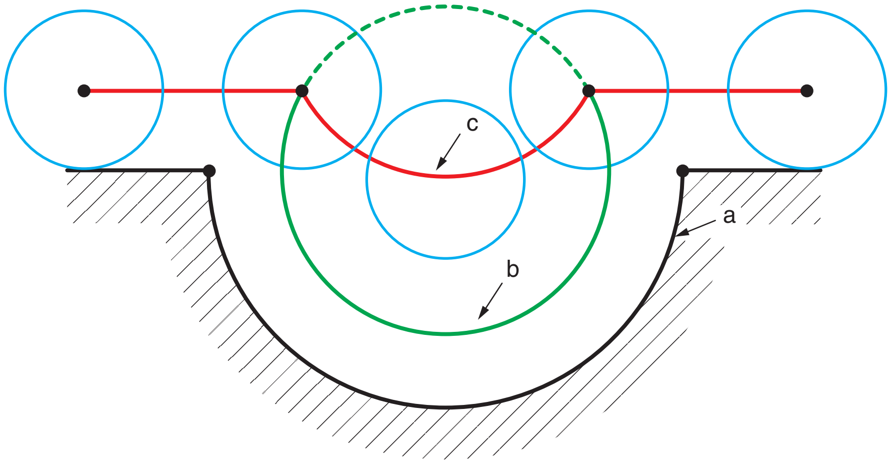
b : Tool path in the G42 mode

c : Tool path in the G41 mode

In the cutter radius compensation OFF (G40) state, the cutter center moves along the path "a".

[Supplement]

- If the tool paths calculated in the G102 or G103 mode with the cutter radius compensation active, create an arc having a center angle of greater than 180°, the arc which has the center angle of "360° - obtained angle" is selected. See the figure below. This is because the contour generation function selects the arc with a center angle of less than 180° from the two possible arcs satisfying the designated arc definition.

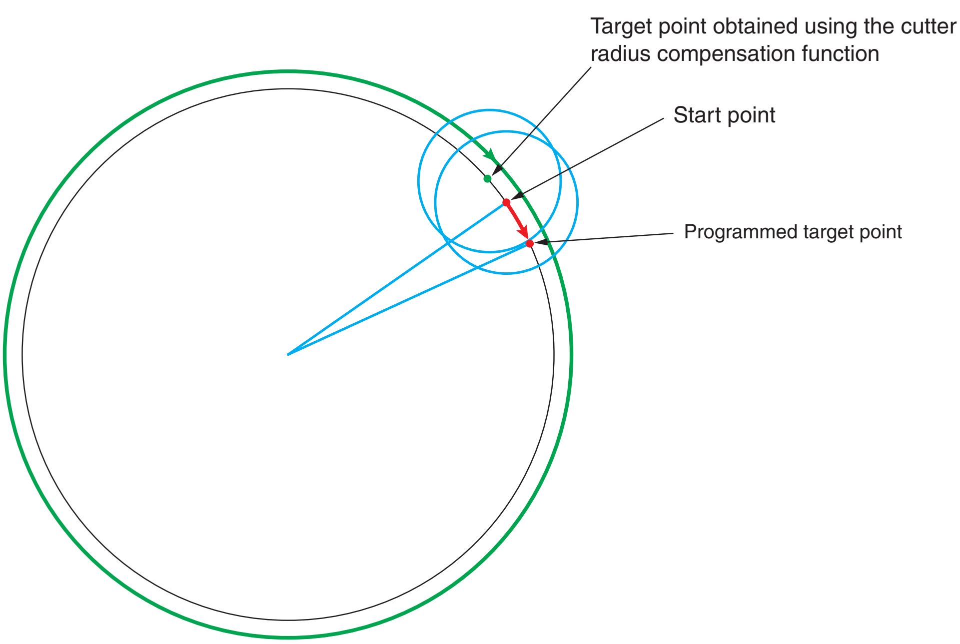


a : Programmed tool paths

b : Tool paths obtained using the cutter radius compensation function (> 180°)

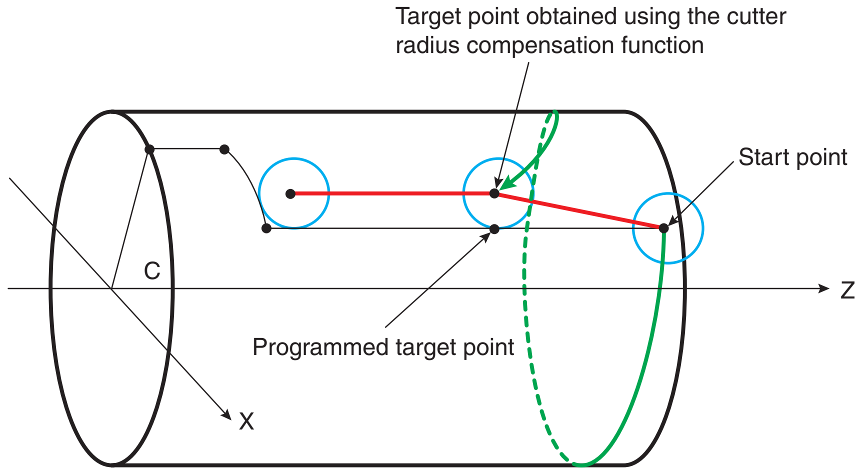
c : Actual tool paths (<180°)

- In the G00 and G01 modes, if the C-axis motion amount is less than the radius of the cutter, the C-axis might make a full circle when the cutter radius compensation function is activated for such a command. See the figure below.



Cutter Radius Compensation for Contour Generation (Face) (1/2)

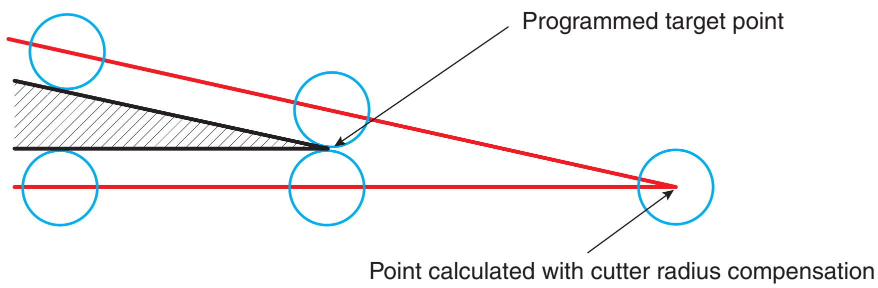
[Supplement]



Cutter Radius Compensation for Contour Generation (Side) (2/2)

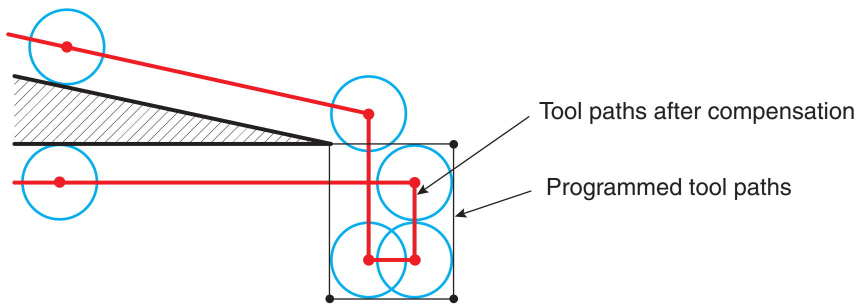
In the G00 and G01 modes, the direction of rotation follows the designated command (M15, M16). In the G101, G102, and G103 modes, the direction of rotation is automatically determined by the control. Therefore, if M15 is designated because the programmed target point is in the M15 direction viewing from the start point, there may be cases in which the target point calculated using the cutter radius compensation function comes to lie in the M16 direction. As the result, the C-axis makes virtually a full circle.
If such a problem occurs, designate the cutter radius compensation function in a different block, or change the target point.

- If the cutter radius compensation function is made active for tool paths which run outside the acute angle shape, there may be cases in which the calculated target point lies far from the programmed target point. See the figure below.



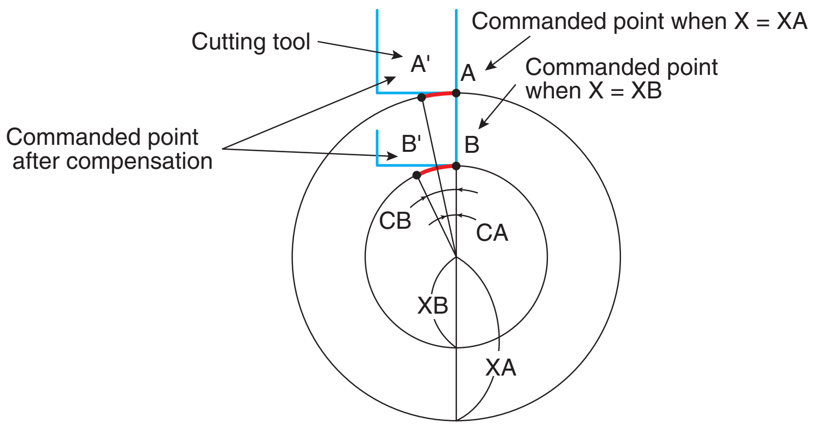
[Supplement]

To avoid such a problem, it is necessary to change the program as shown in the figure below.



Example of Programmed Escape

- An alarm occurs if the X position is changed during the cutter radius compensation on the G119 plane (C-X-Z plane). This is because the compensation plane is changed when the X value is changed on the G119 plane, thus making it impossible to guarantee the compensation value. See the figure below.
  The commanded points change between A and B according to the X value (XA, XB) even when the Z-C commands are the same. As illustrated below, the actual compensation values will vary at points A and B even when a tool of the same diameter is used.



Machining (Side) Viewed from Front
Cutter Radius Compensation for Contour Generation

SECTION 7 FIXED CYCLES

1. Fixed Cycle Functions

Using G31, G32, G33, G34, and G35, it is possible to cut a variety of threads - straight thread, taper thread, thread on an end face, and variable lead thread.

2. Fixed Thread Cutting Cycles

For details on writing thread cutting programs, refer to "Precautions when Programming Thread Cutting Cycles".

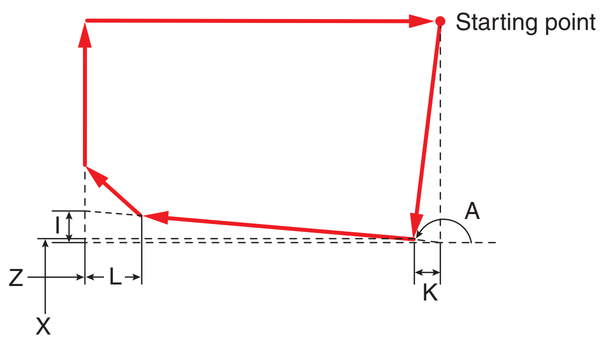
2-1. Fixed Thread Cutting Cycle: Longitudinal (G31, G33)

[Programming format]

$$\begin{array}{l} \text{G33 X\_\_ Z\_\_} \begin{Bmatrix} \text{I\_\_} \\ \text{A\_\_} \end{Bmatrix} \text{(E\_\_) F\_\_ (K\_\_)(L\_\_)(J\_\_)(C\_\_)} \\ \text{(G31)} \end{array}$$

LE33013R0300900030001

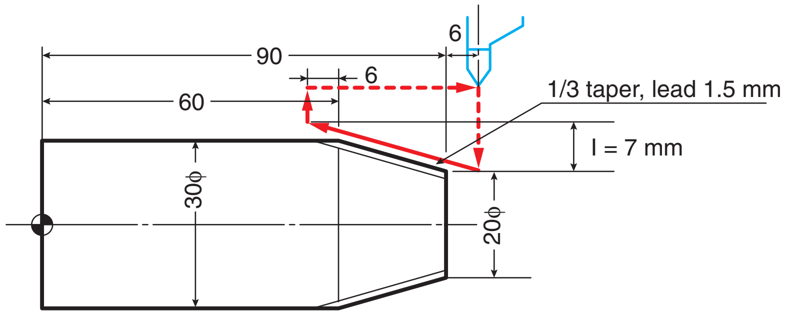
G33 : Fixed thread cutting cycle (longitudinal) command
G31 : Fixed thread cutting cycle command
X : Thread diameter for each thread cutting cycle
Z : Coordinate value of thread end point in Z-axis direction
F : Thread lead (F/J if a J word is provided.)
I : Difference in radius between start and end of taper
A : Taper angle (Taper is specified by either an I or A word.)
E : Z-axis shift amount of thread cutting starting point
(When no K word is specified, the control assumes K=0.)
L : Chamfering distance
(When no L word is specified, the control assumes L=one lead at thread cutting start)
L word is effective in the thread chamfering ON (M23) mode.
J : Number of threads within a distance specified by the F word
(When no J word is specified, the control assumes J=1.)
C : Phase difference for multi-thread thread cutting
(when no C word is specified, the control assumes C=0.)



LE33013R0300900030002

A thread cutting cycle can be programmed in the same manner with G31 as with G33.

Example Program: Constant Lead Taper Thread



N001 G00 X40 Z96	Positioning to the thread cutting starting point, X = 40 mm (in dia.) and Z = 96 mm, at a rapid feedrate.
N002 G33 X17 Z54 I7 F1.5	• With an I word in the G33 block, the taper thread cutting cycle indicated in the drawing above is carried out. • X, Z and F words are determined in the same manner as when cutting a straight thread. • The value I in this example can be calculated according to the following equation: [(96 - 54) × 1/3]/2 = 7 Note that the I word specifies a difference in radius.
N003 X16.5	Program X dimension (as a diameter) for each succeeding thread cutting pass.
N004 X16.2	
N005 X16.05	

Thread lead is commanded as a lead along the Z-axis.

LE33013R0300900030003

[Supplement]

1) The sign of an I word determines increasing or decreasing taper ("+" for increasing taper and "-" for decreasing taper). The plus sign (+) may be omitted.
2) The difference in radius between the starting point and end point is expressed by an I word.

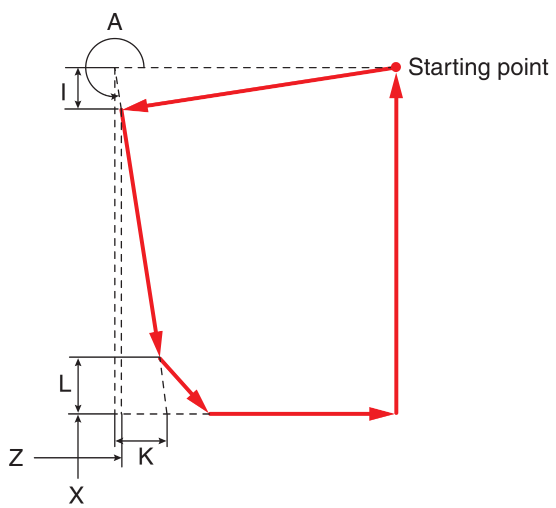
2-2. Fixed Thread Cutting Cycle: End Face (G32)

[Programming format]

G32X__ Z__ $\left\{\begin{matrix} K\_\_ \\ A\_\_ \end{matrix}\right\}$ (E___)(I__)(L__)(J__)F__(C__)

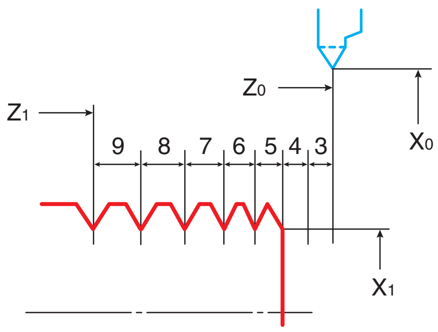
LE33013R0300900040001

X	:	Coordinate value of thread end point in X-axis direction
Z	:	Coordinate value of thread cutting pass in Z-axis direction
F	:	Thread lead (F/J if a J word specified.)
K	:	Difference between starting point and end point for taper thread cutting (When no K word is specified, the control assumes K=0.) "+" increasing taper "-" decreasing taper
A	:	Taper angle referenced to axis parallel to Z-axis (Taper is specified by either an I or A word.)
E	:	Lead variation per lead in cutting variable lead thread (When no E word is specified, the control assumes E=0.)
I	:	X-axis shift amount of thread cutting starting point (When no I word is specified, the control assumes I=0.)
L	:	Chamfering distance (When no L word is specified, the control assumes L=one lead at thread cutting starting)
J	:	Number of threads within a distance specified by F word (When no J word is specified, the control assumes J=1.)
C	:	Phase difference for multi-thread thread cutting (C=0 if omitted.)



LE33013R0300900040002

Example Program: Variable Lead Thread



N001	G00	X_0	Z_0			Positioning to thread cutting starting point X0, Z0 at a rapid feedrate.
N002	G33	X_1	Z_1	E1	F2.5	• With an E word in G33 block, variable lead thread cutting cycle is performed along the paths indicated in the drawing above. • X and Z words can be determined in the same manner as when cutting a straight thread. • The value of an E word is used to specify the lead variation rate per pitch. If it is 1 mm, set "E1". • The F word specifies the first lead employed in starting the thread cutting cycle. F2.5 in this example.
N003		X_2				Program Z dimension for each succeeding thread cutting pass.
N004		X_3				

LE33013R0300900040003

[Supplement]

The sign of an E word expresses increasing or decreasing lead.

- Increasing lead: E+
- Decreasing lead: E-

The "+" sign may be omitted.

[Supplement]

When determining the F word value, use the following equation:

$$D = n \times (F_0 \pm \frac{n \times E}{2})$$

where,
D : displacement after "n" revolutions, (mm)
n : number of revolutions required for displacement D, min^{-1}{rpm}
F_0 : thread lead at start of thread cutting cycle
E : lead variation amount per revolution
$\pm$: increasing or decreasing lead
"+" : increasing lead
"–" : decreasing lead

Using the equation above, the F value can be calculated as follows:

$$D = n \times (F_0 + \frac{nE}{2})$$

$$42 = 7\ (F_0 + \frac{7 \times 1}{2})$$

$$F_0 = 2.5$$

3. Non-Fixed Thread Cutting Cycle (G34, G35)

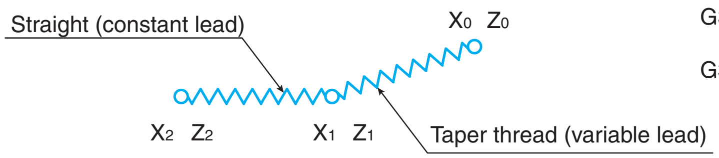
[Function]
Used for a variety of special thread cutting, such as parts combining a straight thread with a taper thread, or a variable lead thread and straight thread.
[Programming format]
G34 X__ Z__ (E__) F__ (C__) (J__)
G35

F : Thread lead
E : Lead variation amount
C : Thread cutting phase difference (if not specified, the control assumes C=0)
J : Number of threads within the specified lead F (if not specified, the control assumes J=1)

[Details]

- Thread cutting is performed from the actual position to the position X, Z with a lead of F.
- If a lead variation amount, E, is specified, the thread cutting start position increases (G34) or decreases (G35) by the amount specified for E every pitch.

Example:



G00	X_0	Z_0		
G34	X_1	Z_1	E	F
G34		Z_2	E_0	

LE33013R0300900050001

[Supplement]

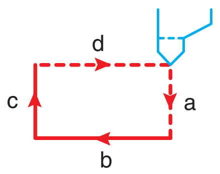
1) The lead for the straight thread in the example above is the one at the start point (Z0) of the variable lead thread. If the straight thread is required to be cut with the lead obtained at point Z1, specify an F word again.
2) To specify the thread lead along the axis parallel to the X-axis, command M27. As M27 is effective only within one block, specify M27 for individual blocks when selecting the thread lead along the X-axis. For thread lead along the Z-axis, specify M26.

4. Precautions when Programming Thread Cutting Cycles

Observe the following points when programming thread cutting cycles:

- The G codes commanding thread cutting (G31 to G35) cannot be designated in the G96 (constant peripheral speed cutting ON) mode.

- Motion of Thread Cutting Tool
  In thread cutting cycles called for by G31, G32 and G33, tool paths a and d are executed at a rapid feedrate, b at the feedrate specified by an F word, and c at the rate determined by parameter setting.



LE33013R0300900060001

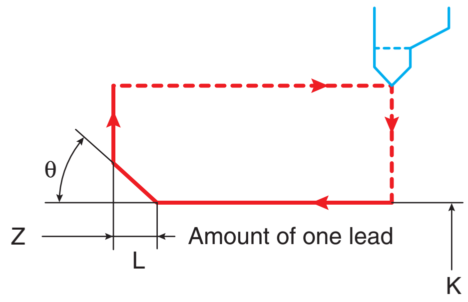
- Lead of Taper Thread
  The lead of a taper thread is parallel to the Z-axis in the G31 and G33 mode, and parallel to the X-axis in the G32 mode.
  In the G34 and G35 modes, an M code is used to designate the direction of thread lead:
  M26: Cancelation of M27, parallel to Z-axis
  M27: Parallel to X-axis
  If no M code is specified in the G34 or G35 mode, the control assumes M26, parallel to the Z-axis.

- Number of Thread Cutting Passes
  Determine the number of thread cutting passes to complete the thread according the workpiece material, thread lead, etc.

- Spindle Speed Change During Thread Cutting Cycle
  If the spindle speed is changed during a thread cutting cycle, it will shift the starting point of the cycle, damaging the thread being cut.
  Therefore, never change spindle speed during thread cutting cycle.

- Feedrate Override
  The feedrate override dial is inoperative during a thread cutting cycle.

- Chamfering
  Chamfering in thread cutting to produce a thread vanish cone can be programmed by commanding M23, if required. To cancel this mode, command M22.
  M22: Chamfering OFF
  M23: Chamfering ON



Equal to one lead when no L command is given

LE33013R0300900060002

The feedrate used for chamfering in the X-axis direction is set at Feedrate of chamfering in thread cycle of optional parameter (OTHER FUNCTION 1).

$$\text{Feedrate on the X-axis (mm/min)} = \left\{ \frac{\text{Parameter setting } (\mu)}{10^3} \times \frac{60 \times 10^3 \text{ (msec)}}{8 \text{ (msec)}} \right\}$$

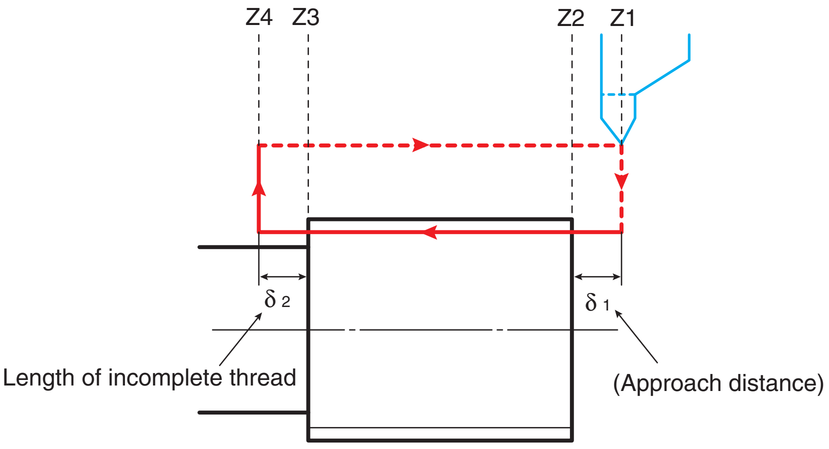
LE33013R0300900060003

Therefore, the chamfering angle θ is determined by the feed in the Z-axis direction (designated in the thread cutting program) and the feed in the X-axis direction.
Example chamfering program:

N001	G00	X40	Z80	M23
N002	G33	X29	Z50	F1.5
N003		X28.5		
N001		X28.2		
N001		X28.05		
N001	G00	X1	Z1	M22

LE33013R0300900060004

- Extra Length in Thread Cutting Program
  Since a certain length of incomplete thread is usually produced near the start and end point of the cut, it is necessary to add appropriate amounts δ1 and δ2 at the start and end of the thread to be cut in order to cut the proper thread shape.



LE33013R0300900060005

Values δ_1 and δ_2 vary depending on the cutting conditions. Generally, these values must satisfy the following equations:

$\delta_2 > K \times N \times P$
$\delta_1 > K \times N \times P$
where,
N: spindle speed
P: lead
K: machine-model-dependent constant

LE33013R0300900060006

The values of constant K for individual models are indicated below:

Model	**K[X10^{-3}]**		**Model**	**K[X10^{-3}]**
LB200	0.48		MACTURN250	1.17
LB200MY	0.64		MACTURN350	1.39
LB250	0.48		MACTURN550	1.60
LB300	0.53		MULTUS-B300	1.07
LB300MY	0.64		LVT300	0.75
CAPTAIN L370	0.53		LVT400	1.07
LB400	0.64		LT200	0.75
CAPTAIN L470	0.64		LT300	1.28
LB35II	1.28		LCS250	0.96
LB45II	0.96		LAW-2S	1.28
LB45IIMY	1.17		(Priority given to smooth cut surface)	2.56
LU35	1.28		LAW-F	0.85
LU45	1.28		(Priority given to smooth cut surface)	2.56
LU300	0.53		LFS10	0.85
LU300MY	0.64		LF150G	0.64
LU400	0.64		LOC650 (LC50)	0.96

Example:
For the LU300, with a peripheral speed of 100 m/min, a 10 mm diameter and a thread lead of 1.5 mm, the spindle speed and feedrate are calculated as follows.

Spindle speed $N = \frac{100 \times 10^3}{10\pi} = 3183$ (rev/min)

Feedrate $N \times P = 3183 \times 1.5 = 4775$ (mm/min)

The incomplete thread length can be calculated as follows:
$\delta 1 = \delta 2 = 0.53 \times 10^{-3} \times 4775 = 2.53$ (mm)

Therefore, δ1 and δ2 should be 2.53 (mm) or longer.

LE33013R0300900060007

- Restrictions on Cutting Speed
In a thread cutting cycle, the following restrictions apply to the relationship between spindle speed and thread lead:

Programmable thread lead	0.001 to 1000.000 mm		
Spindle speed:	X-axis :	Max. feedrate of X-axis	$> N \times P$
	Z-axis :	Max. feedrate of Z-axis	$> N \times P$

where,
N: spindle speed
P: lead

LE33013R0300900060008

[Supplement]

1) The same restrictions apply in G01 linear interpolation mode operation.
2) The maximum feedrates vary according to the machine specifications.

- Inch System Thread
When cutting inch threads, the metric lead converted from the desired inch lead is used in programming. To cut an accurate inch thread with the converted metric thread lead value, either enter 8 digits below the programmable increment, 1 µm, or use a J word in combination with an F word.
Example: To cut an inch thread of 11 threads per inch (25.4/11 8 2.309091)

G34	X	Z	F25.4 J11	(1 mm unit system)
G34	X	Z	F230.9091	(10 µm unit system)
G34	X	Z	F2.309091	(1 mm unit system)
G34	X	Z	F2309.091	(1 µm unit system)

- Feed Hold During Thread Cutting Cycle
This function is effective while an the Z (X) axis is moving in the G33 (G32) mode. Pressing the SLIDE HOLD pushbutton during the thread cutting cycle stops axis movement immediately, breaking the thread being cut and damaging the workpiece. This function is provided to prevent such trouble.
Activate this function to check the dimensions and shape of the threads being cut and also to check the tip point of the thread cutting tool.

[Operation]

- When the SLIDE HOLD pushbutton is pressed during a thread cutting cycle:

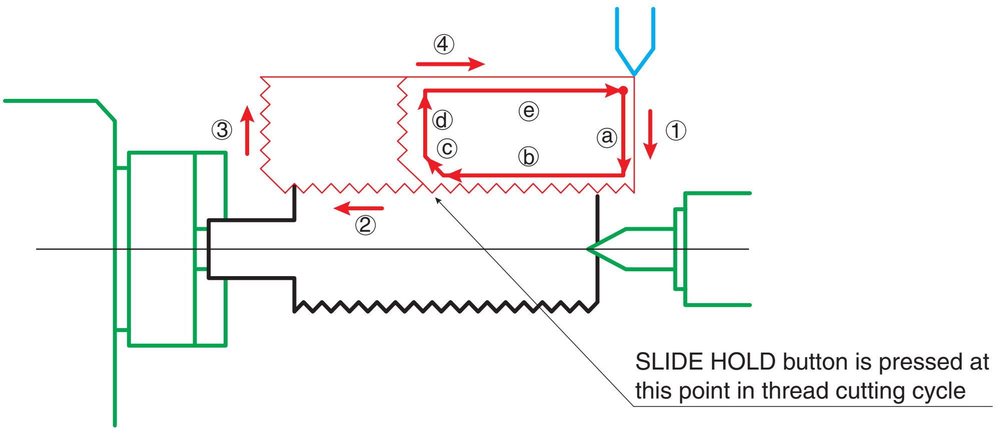
  a. Chamfering equivalent to one lead length or the length specified by an L command is performed.

  b. The X-axis returns to the thread cutting cycle starting point.

  c. The Z-axis returns to the thread cutting cycle starting point.

  d. The control is in cycle stop mode waiting for pressing of the CYCLE START button.

- When the CYCLE START button is pressed:
  The interrupted thread cutting cycle is continued.
  This interruption operation can be repeated as many times as necessary in the same thread cutting cycle. When the SLIDE HOLD button is pressed while the axes are moving along path (1) or (4) (see figure) where thread cutting is not executed, axis movement stops immediately. Pressing the CYCLE START button after that resumes the thread cutting cycle.
  If the SLIDE HOLD button is pressed while the axis is moving along path (3), axis movement stops after it reaches the end point of path (3).

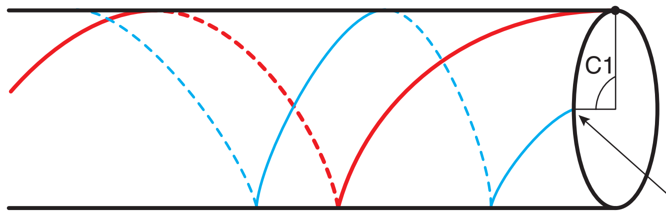


Normal thread cutting cycle : (1) → (2) → (3) → (4)
Cycle after slide hold : (a) → (b) → (c) → (d) → (e)

LE33013R0300900060009

- Designation of Phase Difference (Angle) for Multi-thread Thread Cutting
  Multi-thread thread can be programmed easily by designating the thread cutting start point.
  For G33 cycle:

```
G33  X1  Z1  F  }
     X2         }  First thread
     X3         }
G33  X1  Z1  F  }
     X2  C2     }  Second thread
     X3  C3     }
```



LE33013R0300900060010

Thread cutting is carried out by shifting the thread phase by the amount (angle) specified by the C command.
For G32 cycle:

```
G32  X1  Z1  F   C1
         Z2  C1
         Z3  C1
```

LE33013R0300900060011

For G34, G35 cycle:

```
G34  X1  Z1  E1  C
G34      Z2  E2
```

LE33013R0300900060012

The C command is ignored except in the first sequence

[Supplement]

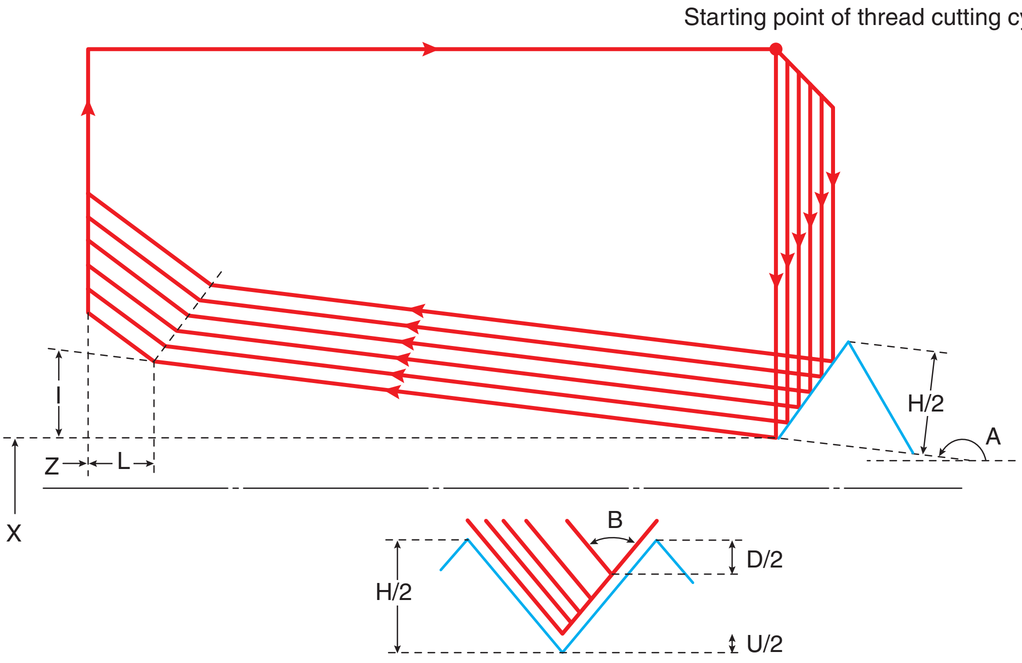
1) Programmable range for C command: 0 - 359.999
2) In the G32 and G33 cycles, the C command value designated in the first block remains effective for the subsequent blocks.
3) In the G32 and G33 cycles, if a C command value differing from the value designated in the first block is designated in a subsequent block, this C command value is ignored.
4) In the G34 and G35 cycles, a C command value can be designated only in the first sequence block; a C command value designated in the second and subsequent sequence blocks is not acceptable.

For multi-thread thread cutting operation, refer to "Thread Cutting Compound Cycle (G71/G72)".

5. Thread Cutting Compound Cycle (G71/G72)

5-1. Longitudinal Thread Cutting Cycle (G71)

[Function]
In G71 mode thread cutting cycle as shown below is performed:



LE33013R0300900070001

[Programming format]

G71X__ Z__ {A__ / I__} B__D__U__H__L__E__F__J__M__Q

LE33013R0300900070002

X : Final diameter of thread

Z : Z coordinate of end point of thread

A : Taper angle

I : Difference in radius between starting point and end point for taper thread
(expressed as a radius)
For taper thread, use either an A or I word.

B : Infeed angle
(0° x B < 180°; 0 if no designation. Normally it is equal to the cutter tip point angle.)

D : Depth of cut in the first thread cutting cycle
(Expressed as a diameter)

U : Finishing allowance
(Expressed as a diameter; no finishing cycle is performed if no U word is designated.)

H : Thread height
(Expressed as a diameter)

L : Chamfering distance in final thread cutting cycle
(Effective in M23 mode; if no L word is designated in the M23 mode, L is assumed to be the distance equivalent to one lead.)

E : Lead variation rate per lead for variable lead thread

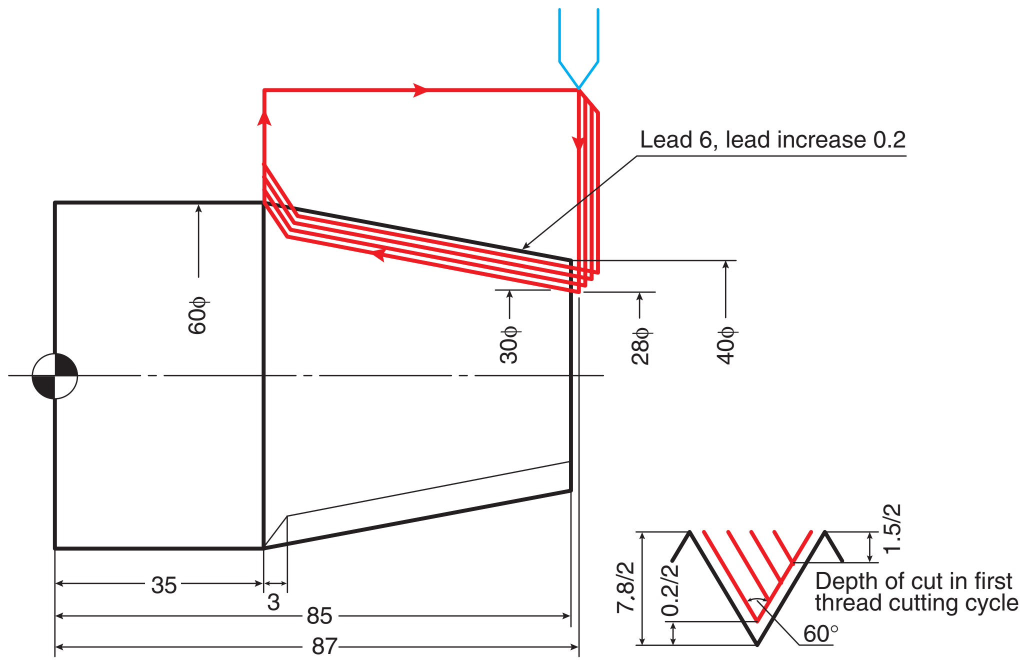
F : Thread lead (F/J if a J word specified.)

J : Number of threads within a distance specified by F word
(When no J word is designated, the control assumes J=1.)

M : Used to select thread cutting pattern and mode of infeed.
(For details, refer to "M Code Specifying Thread Cutting Mode and Infeed Pattern")

Q : The number of threads for multi-thread thread cutting (refer to "Multi-thread Thread Cutting Function in Compound Fixed Thread Cutting Cycle".)

5-2. Program Example for Longitudinal Thread Cutting Compound Fixed Cycle (G71)



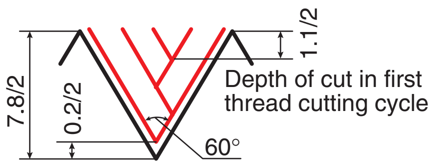
LE33013R0300900080001

Example 1:
Using M32 (one-face cutting mode) and M75 (infeed pattern 3):

```
N001  G71  X28  Z35  I11  B60  D1.5  U0.2  H7.8  L3  E0.2  F6
$                                    M23   M32   M75
```

LE33013R0300900080002

Example 2:
Using M33 (zigzag cutting mode) and M74 (infeed pattern 2)

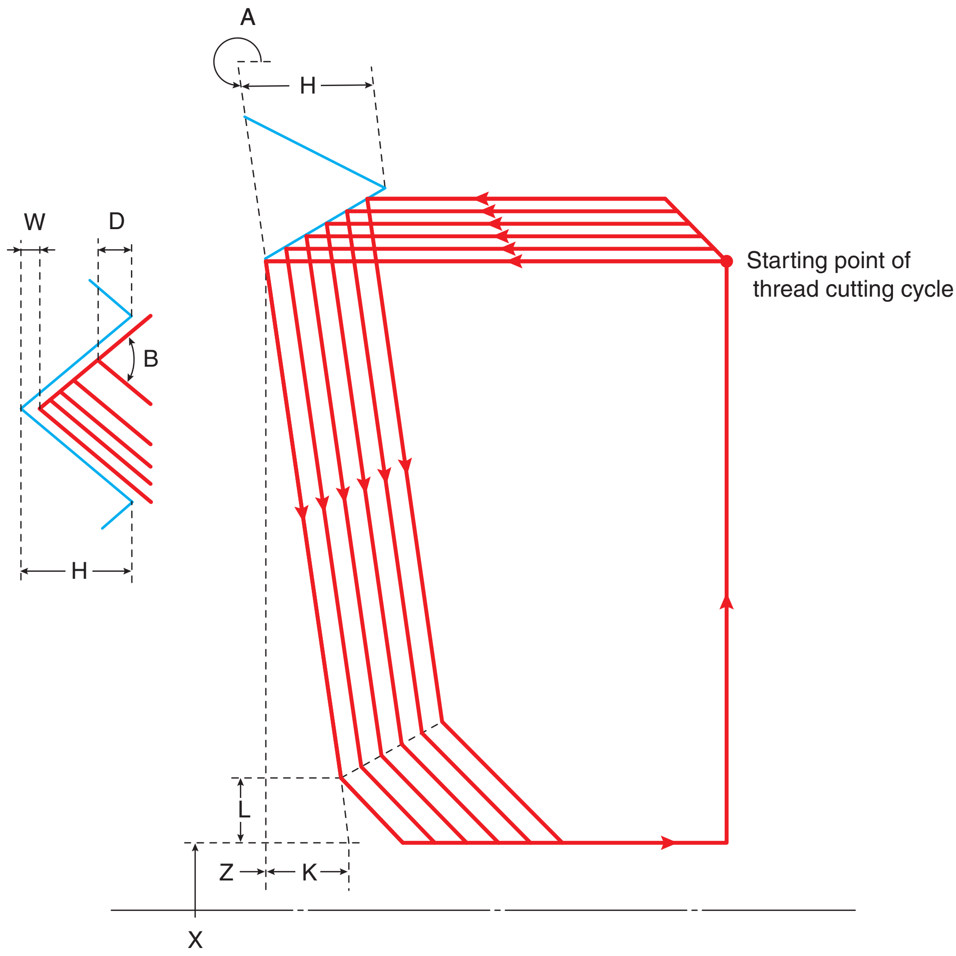


N0001 G71 X28 Z35 I11 B60 D1.1 U0.2 H7.8 L3 E0.2 F6
$ M23 M32 M75

LE33013R0300900080003

5-3. Transverse Thread Cutting Compound Fixed Cycle (G72)

[Function]
In the transverse thread cutting compound fixed cycle, the thread cutting cycle shown below is performed.



LE33013R0300900090001

[Programming format]

G72 X__ Z__ { A__ / K__ } B__D__ W__H__L__E__F__J__M__Q__

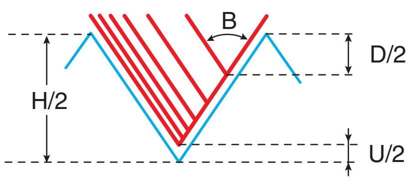
LE33013R0300900090002

- X : X coordinate of end point of thread
- Z : Z dimension of final thread cutting cycle
- A : Taper angle
- K : Distance between starting point and end point for taper thread
  For taper thread, use either an A or K word.
- B : Infeed angle(0° x B < 180°; 0 if no B command is designated.)
- D : Depth of cut in the first thread cutting cycle
- W : Finishing allowance
  (No finishing cycle is performed if no W word is designated.)
- H : Thread height
- L : Chamfering distance in final thread cutting cycle
  (Effective in the M23 mode; if no L word is designated in the M23 mode, L is assumed to be the distance equivalent to one lead.)
- E : Lead variation per lead in cutting variable lead thread
  (When no E word is specified, the control assumes E=0.)
- F : Thread lead (F/J if a J word specified.)
- J : Number of threads within a distance specified by F word
  (When no J word is specified, the control assumes J=1.)
- M : Used to select thread cutting pattern and mode of infeed.
  (For details, refer to "M Code Specifying Thread Cutting Mode and Infeed Pattern")
- Q : The number of threads for multi-thread thread cutting (refer to "Multi-thread Thread Cutting Function in Compound Fixed Thread Cutting Cycle")

5-4. M Code Specifying Thread Cutting Mode and Infeed Pattern

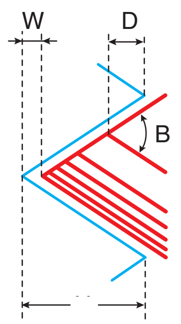
The tool angle, B, thread height, H, depth of cut in first thread cutting cycle, D, and stock removal, W, are indicated below for thread cutting in the longitudinal direction and in the transverse direction.

Thread Cutting in Longitudinal Direction



B : Tool angle
H : Thread height
D : Depth of cut in first thread cutting cycle
U(W) : Stock removal

Thread Cutting in Transverse Direction



LE33013R0300900100001

5-4-1. M Codes Specifying Thread Cutting Mode

The thread cutting mode is specified with an M code. The correspondence between modes and M codes is as follows:

M32 : Straight infeed along thread face (on left face)

M33 : Zigzag infeed

M34 : Straight infeed along thread face (on right face)

When none of these M codes is specified, the control automatically selects the M32 mode.
If tool angle B is 0°, the cutting tool is fed straight independent of the designated cutting mode.

5-4-2. M Codes Specifying the Infeed Pattern

The infeed pattern is specified with an M code. The correspondence between patterns and M codes is as follows:

M73: Infeed pattern 1

The amount of infeed is D (diameter value) in each thread cutting cycle up to the point D mm away from the position "H - U (W)". After that point is reached, the infeed amount is changed to D/2, D/4, D/8 and D/8, leaving stock removal U (W) if specified. The infeed amount in the finishing cycle is the specified amount U (W).
When no U (W)word is specified, the finishing cycle is not performed.

M74: Infeed pattern 2

The amount of infeed is D (in diameter) in each thread cutting cycle until the position "H - U (W)" is reached. After that, the finishing cycle is carried out with an infeed amount of U (W). If no U (W) word is specified, the finishing cycle is not performed.

M75: Infeed patterns 3 and 4

In each thread cutting path of the thread cutting cycle, depth of cut is determined so that metal removal rate is optimum. Infeed pattern 3 or pattern 4 can be selected by the setting at Infeed pattern in the M75 mode of optional parameter (OTHER FUNCTION 1).

Infeed pattern 3

- When M32, M34 is designated
  $D^2 \geq \{H^2 - (H - U\ (W))^2\}$
  Each thread cutting path in the cycle is determined by the cutting point which is explained as the depth from the workpiece OD; the first path is created at cutting point "D", the second path at cutting point "2D", and the "n"th path at cutting point "nD" until the path reaches the cutting point of "H - U (W)". Finally, the cutting tool is fed by "U (W)" to carry out the finishing cycle. The finishing cycle is not carried out if U (W) is not designated in the program.
  $D^2 < \{H^2 - (H - U\ (W))^2\}$
  In each thread cutting path, assume the thread cutting point d1 (D) and metal removal volume S_1 for the first path, d_2 and S_2 for the second path, and d_2 and S_n for the "n"th path, then cutting points d_2 to dn are determined so that S_2 to S_n will be the most appropriate metal removal volume to provide high cutting accuracy while minimizing the number of paths. This cycle is repeated until the cutting point of "H - U (W)" is reached. Finally, the cutting tool is fed by "U (W)" to carry out the finishing cycle. The finishing cycle is not carried out if U (W) is not designated in the program.

- When M33 is designated
  $D^2 \geq \{H^2 - (H - U (W))^2\}$
  The thread cutting cycle is repeated with the cutting point at each even numbered thread cutting path being "D" until the cutting point of "H - U (W)" is reached. In each odd numbered tool paths, the cutting point is calculated as;

$$\frac{1}{2}\left(\sqrt{n+1}\,D + \sqrt{n-1}\,D\right)$$

LE33013R0300900150001

Finally, the cutting tool is fed by "U (W)" to carry out the finishing cycle. The finishing cycle is not carried out if U (W) is not designated in the program.
$D^2 < \{H^2 - (H - U (W))^2\}$
In each thread cutting path, assume the thread cutting point d_1 (D) and metal removal volume S_1 for the first path, d_2 and S_2 for the second path, and d_n and S_n for the "n"th path, then cutting points d_2 to d_n (n = even number) for the even numbered paths are determined so that S_2 to S_n (n = even number) will be the most appropriate metal removal volume to provide high cutting accuracy while minimizing the number of paths. For the odd numbered paths, the cutting point is determined by
$d_n = 1/2 (d_{n-1} + d_{n+1})$ (d = odd number).
This cycle is repeated until the cutting point of "H - U (W)" is reached. Finally, the cutting tool is fed by "U (W)" to carry out the finishing cycle. The finishing cycle is not carried out if U (W) is not designated in the program.

Infeed pattern 4

- When M32, M34 is designated
  The following pattern is created regardless of the values of H, D, and U(W).
  In each thread cutting path, assume the thread cutting point d_1 (D) and metal removal volume S_1 for the first path, d_2 and S_2 for the second path, and dn and Sn for the "n"th path, then cutting points d_2 to d_n are determined so that S_2 to S_n will be the most appropriate metal removal volume to provide high cutting accuracy while minimizing the number of paths. This cycle is repeated until the cutting point of "H - U (W)" is reached. Finally, the cutting tool is fed by "U (W)" to carry out the finishing cycle. The finishing cycle is not carried out if U (W) is not designated in the program.

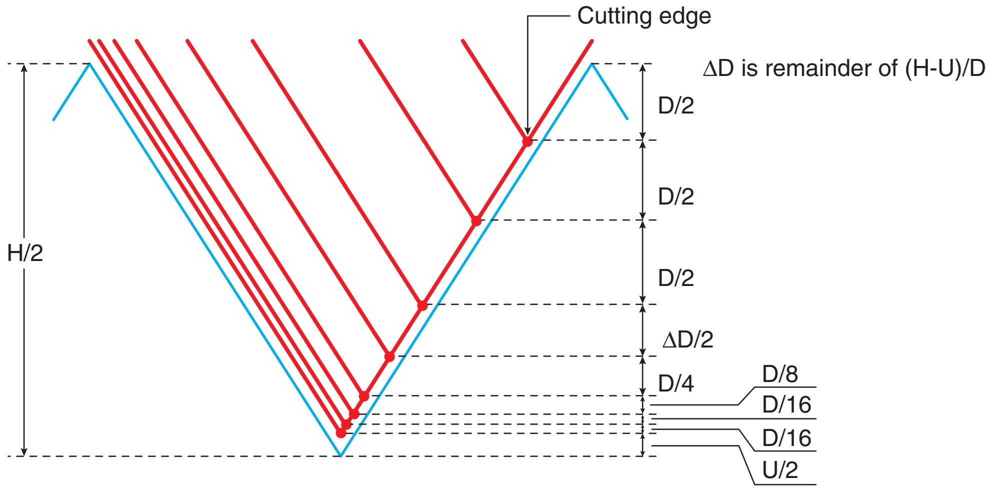
- When M33 is designated
  In each thread cutting path, assume the thread cutting point d_1 (D) and metal removal volume S_1 for the first path, d_2 and S_2 for the second path, and dn and d_n for the "n"th path, then cutting points d_2 to d_n (n = even number) for the even numbered paths are determined so that S_2 to S_n (n = even number) will be the most appropriate metal removal volume to provide high cutting accuracy while minimizing the number of paths. For the odd numbered paths, the cutting point is determined by
  $d_n = 1/2 (d_{n-1} + d_{n+1})$ (d = odd number).
  This cycle is repeated until the cutting point of "H - U (W)" is reached. Finally, the cutting tool is fed by "U (W)" to carry out finishing cycle. The finishing cycle is not carried out if U (W) is not designated in the program.

[Supplement]

Since X commands are specified as diameter values, the actual infeed amount is "D/2".
When no infeed-pattern-designating M code is programmed, the control automatically selects M73.
By combining the M codes designating cutting mode and infeed pattern, ten types of thread cutting cycle each are available for longitudinal thread cutting and transverse thread cutting.

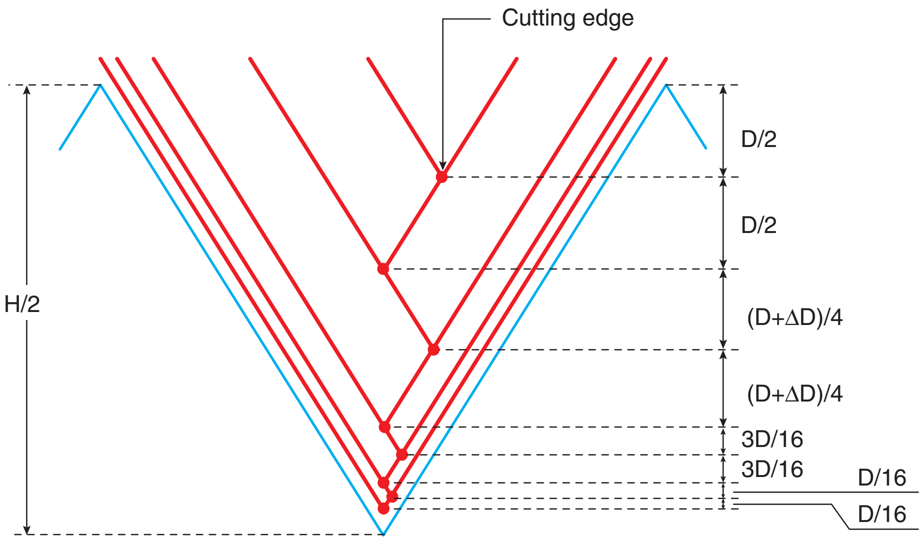
5-4-3. Longitudinal Thread Cutting Cycles

- M32 + M73 Mode



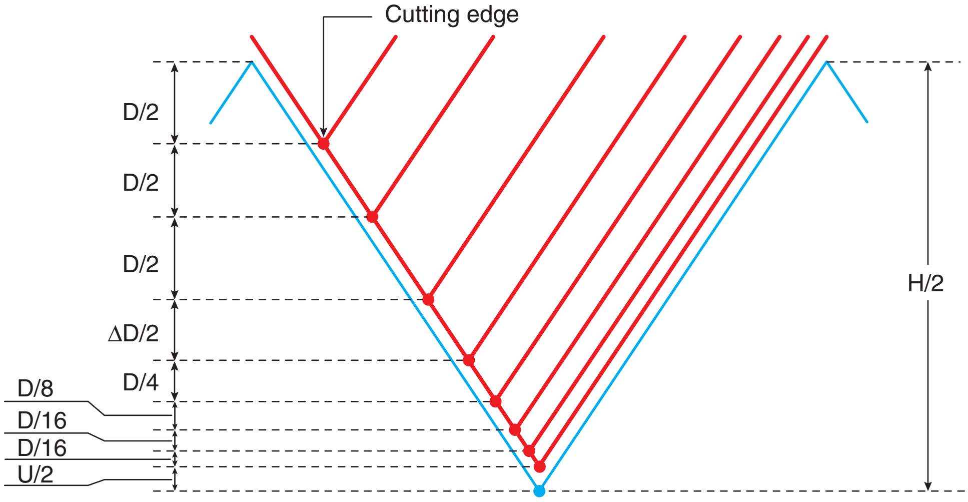
LE33013R0300900160001

- M33 + M73 Mode



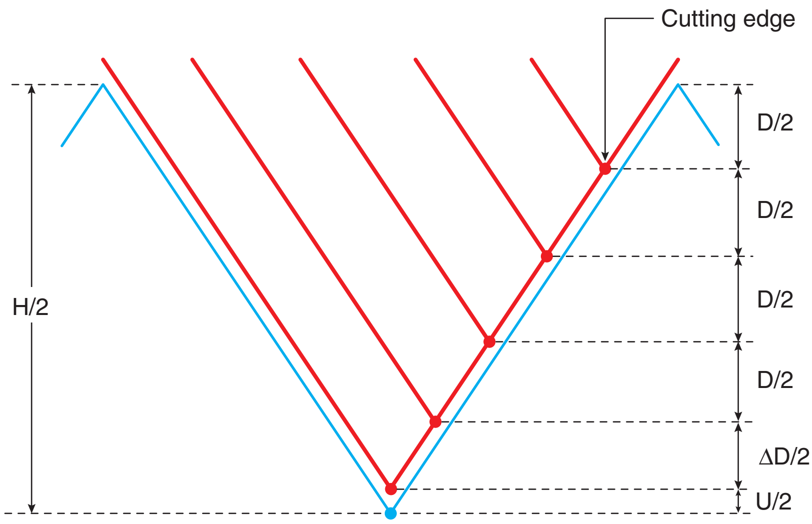
LE33013R0300900160002

- M34 + M73 Mode



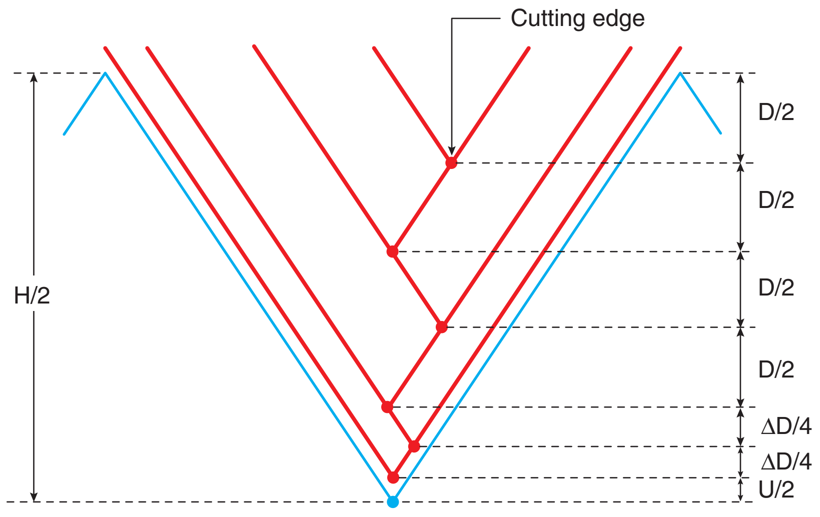
LE33013R0300900160003

- M32 + M74 Mode



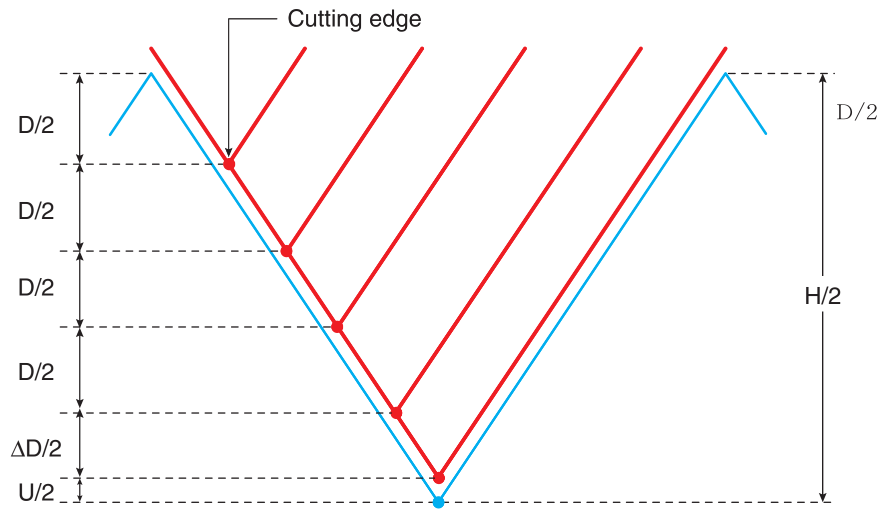
LE33013R0300900160004

- M33 + M74 Mode



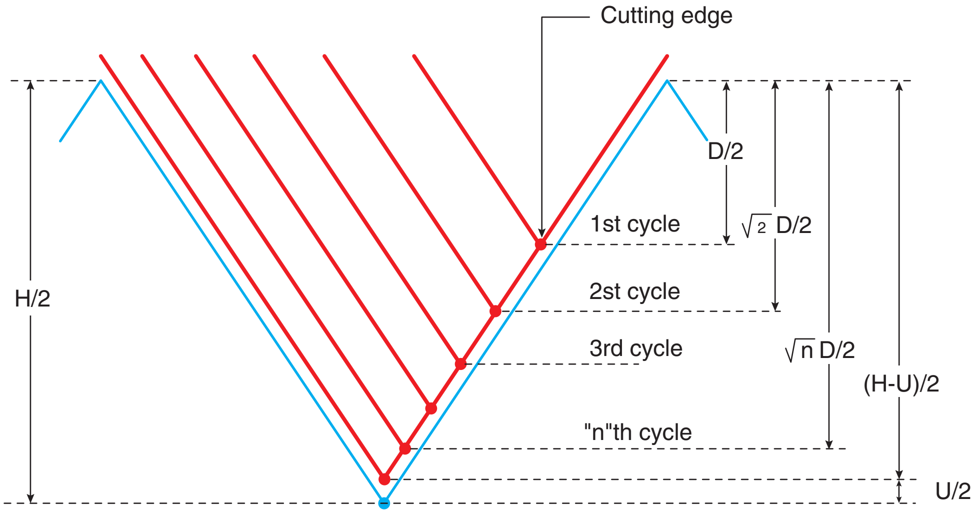
LE33013R0300900160005

- M34 + M74 Mode



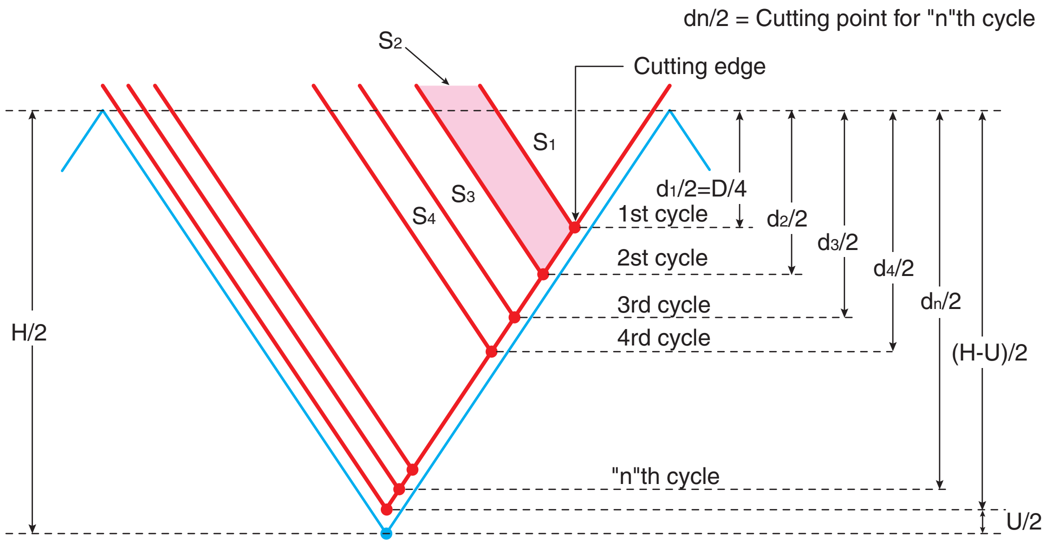
LE33013R0300900160006

- M32 + M75 Mode (infeed pattern 3 $D^2 \geq \{H^2 - (H - U\ (W))^2\}$)



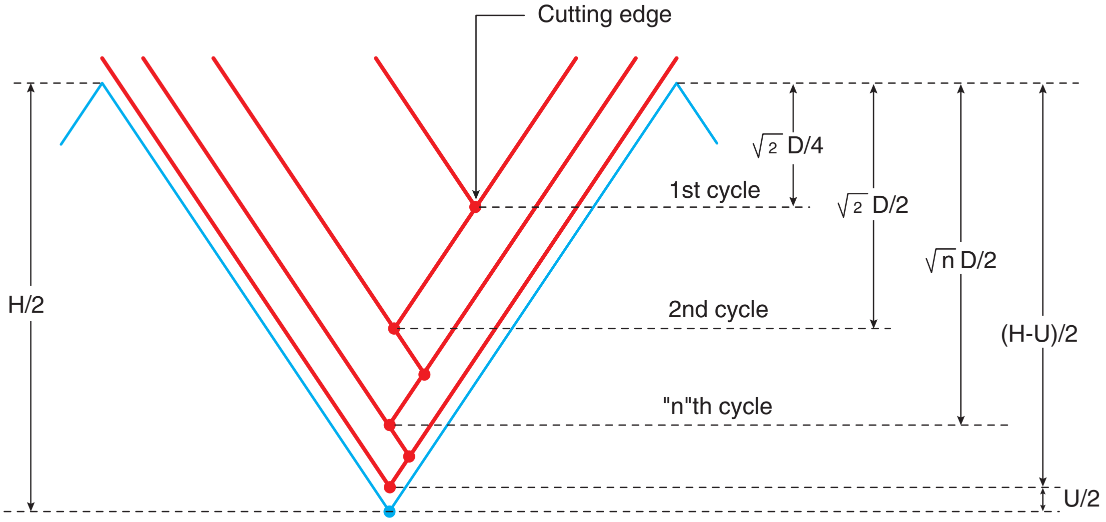
LE33013R0300900160007

- M32 + M75 Mode (infeed pattern 3 $D^2 < \{H^2 - (H - U\ (W))^2\}$ or infeed pattern 4)



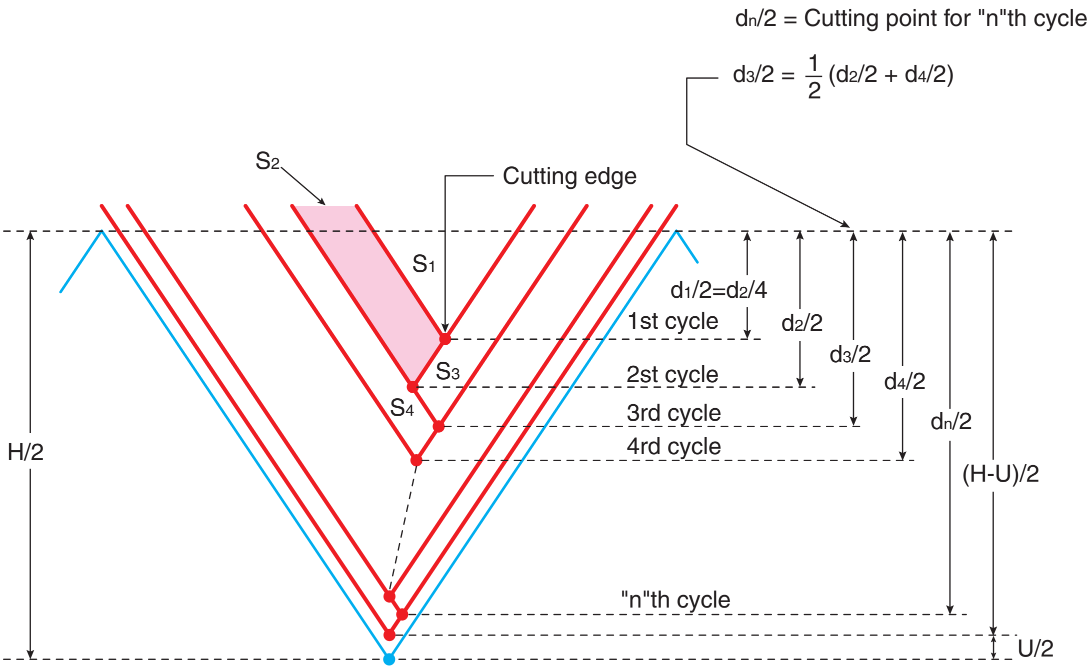
LE33013R0300900160008

- M33 + M75 Mode (infeed pattern 3 $D^2 \geq \{H^2 - (H - U\ (W))^2\}$)



LE33013R0300900160009

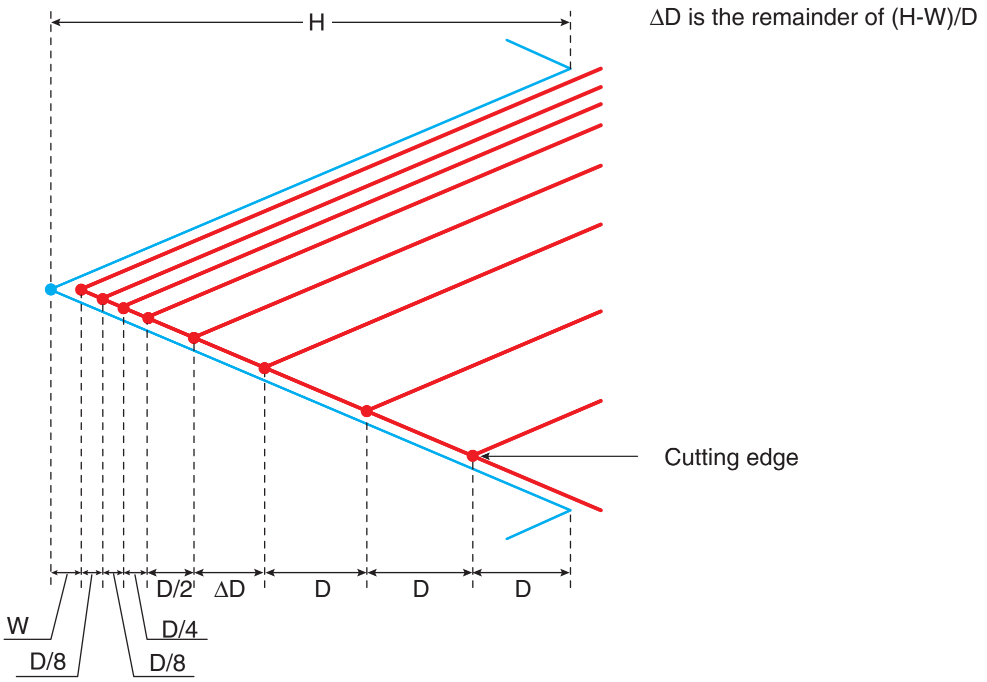
- M33 + M75 Mode (infeed pattern 3 $D^2 - \{H^2 - (H - U\ (W))^2\}$ or infeed pattern 4)



LE33013R0300900160010

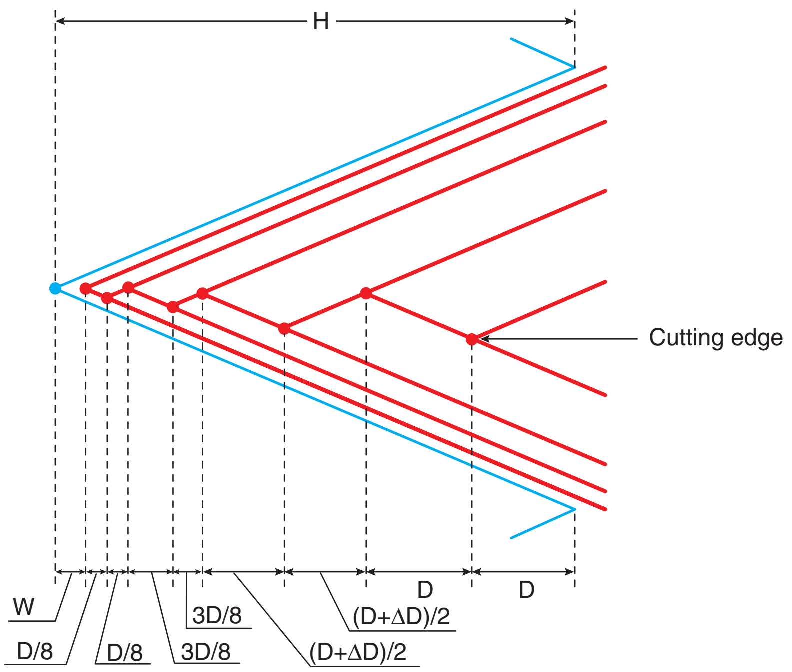
5-4-4. Transverse Thread Cutting Cycles

- M32 + M73 Mode



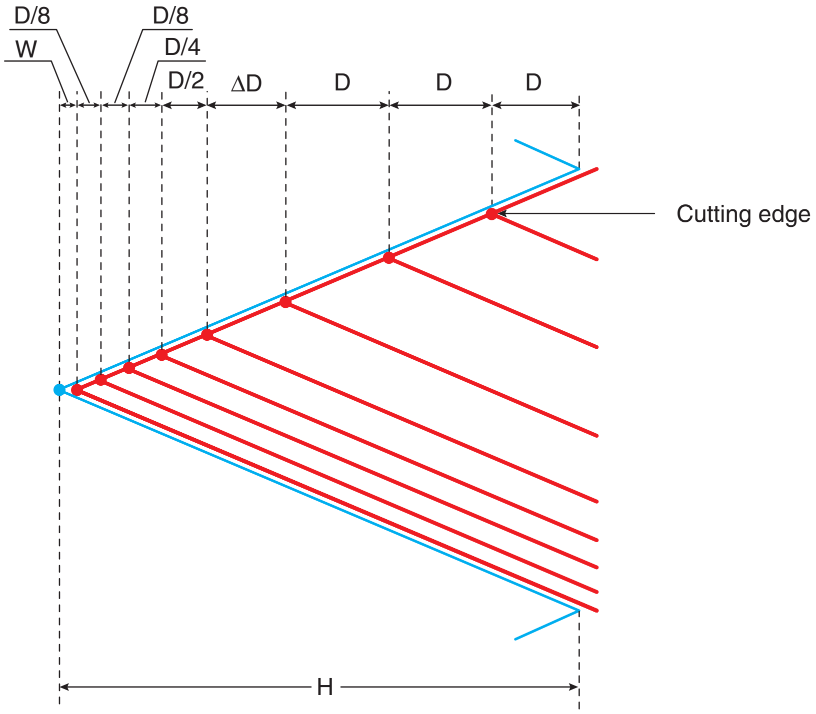
LE33013R0300900170001

- M33 + M73 Mode



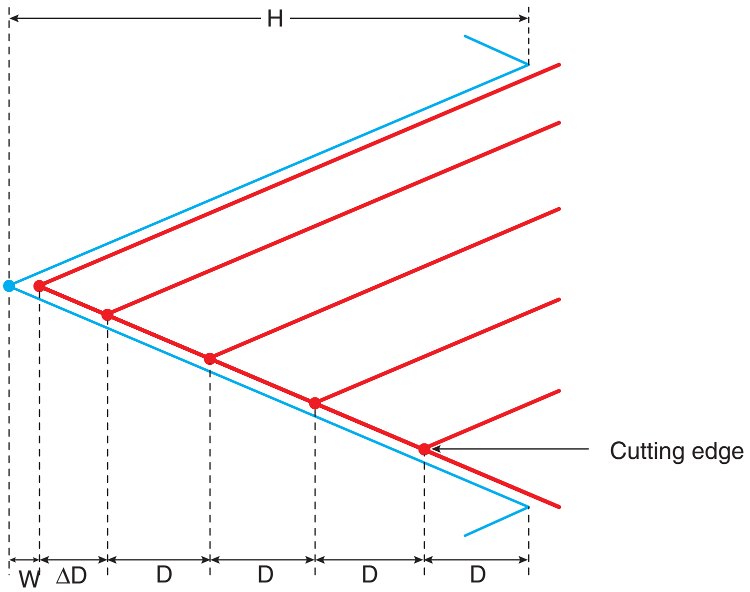
LE33013R0300900170002

- M34 + M73 Mode



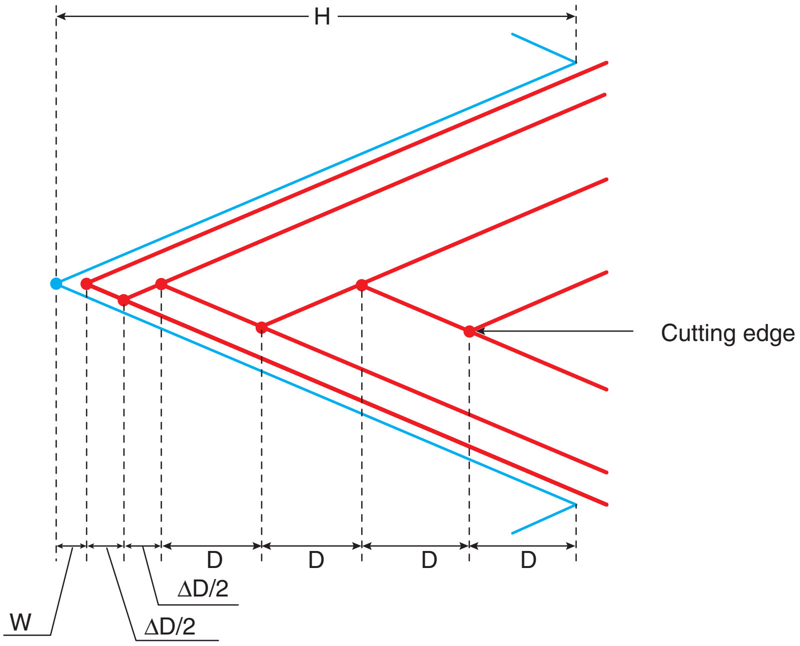
LE33013R0300900170003

- M32 + M74 Mode



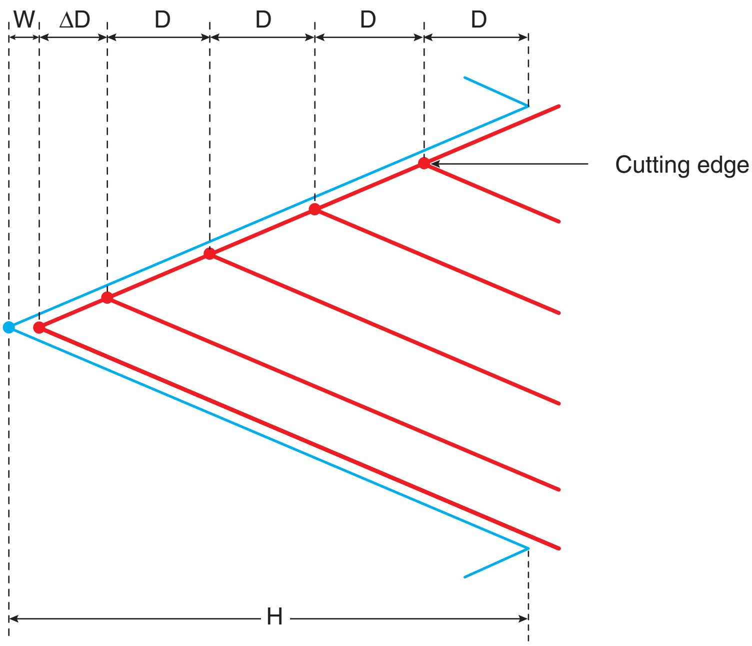
LE33013R0300900170004

- M33 + M74 Mode



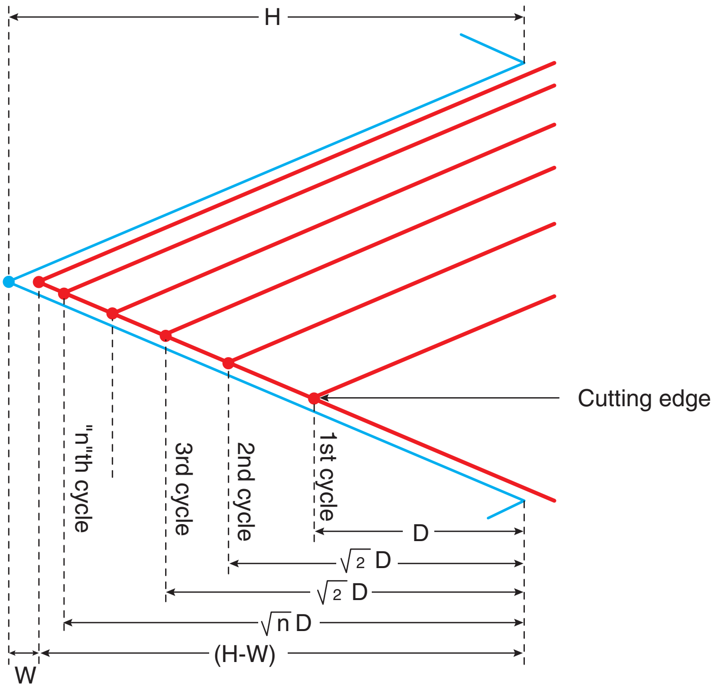
LE33013R0300900170005

- M34 + M74 Mode



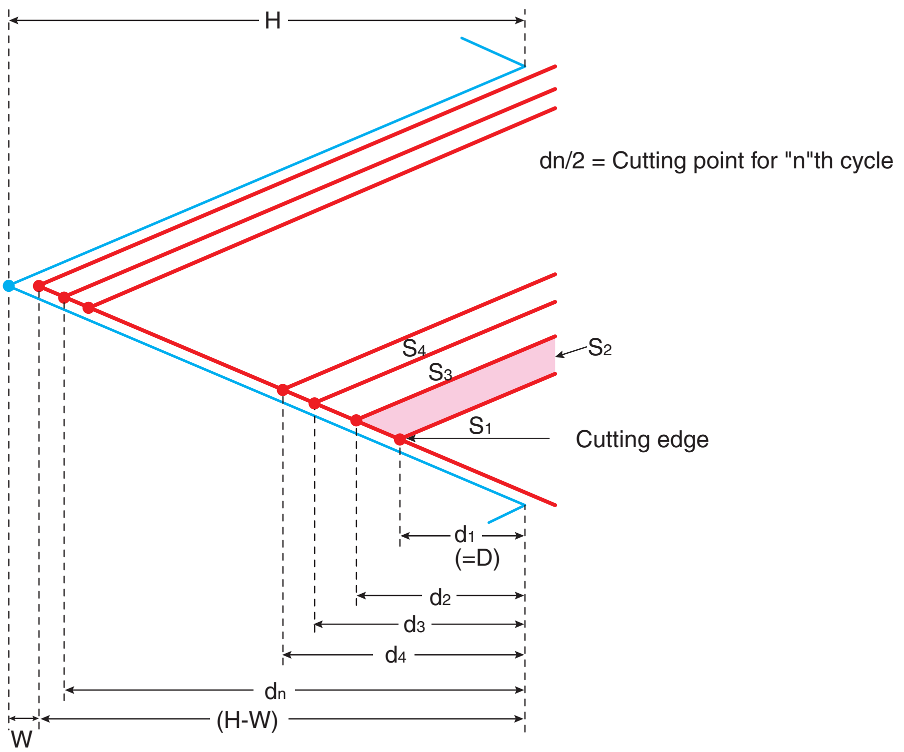
LE33013R0300900170006

- M32 + M75 Mode (infeed pattern 3 $D^2 \geq \{H^2 - (H - U\ (W))^2\}$)



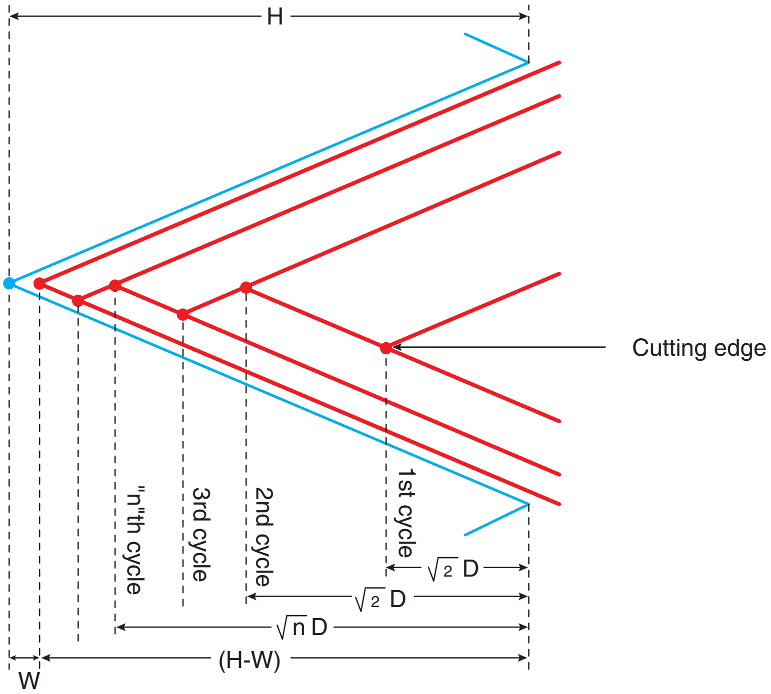
LE33013R0300900170007

- M32 + M75 Mode (infeed pattern 3 $D^2 < \{H^2 - (H - U\ (W))^2\}$ or infeed pattern 4)



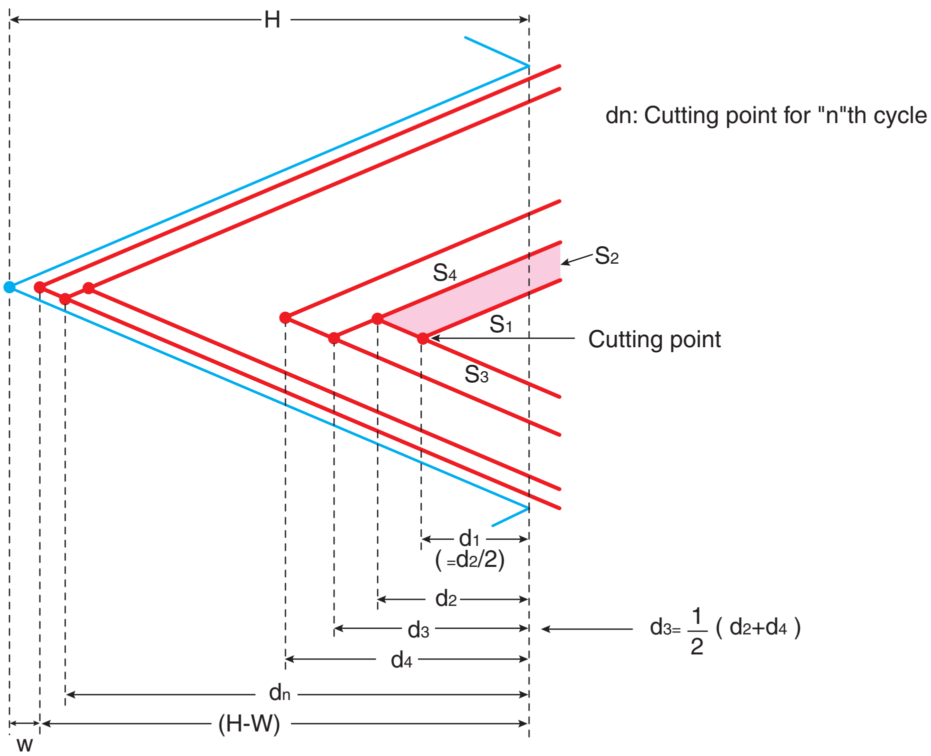
LE33013R0300900170008

- M33 + M75 Mode (infeed pattern 3 $D^2 \geq \{H^2 - (H - U (W))^2\}$)



LE33013R0300900170009

- M33 + M75 Mode (infeed pattern 3 $D^2 < \{H^2 - (H - U (W))^2\}$ or infeed pattern 4)

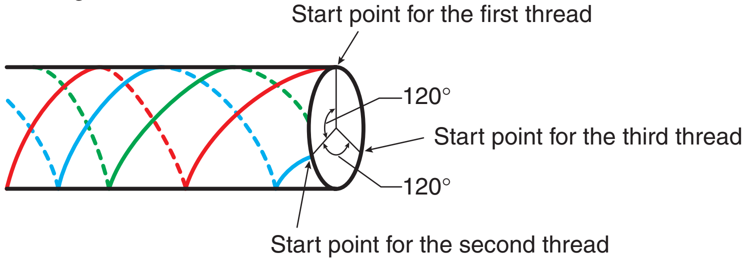


LE33013R0300900170010

5-5. Multi-thread Thread Cutting Function in Compound Fixed Thread Cutting Cycle

In the thread cutting cycle called by G32, G33, etc., a multi-thread thread cutting cycle is designated by designating the phase difference with a C command. In the compound fixed thread cutting cycle, multi-thread cutting can be designated by simply designating the number of threads with a Q command. The phase difference is automatically calculated.

Assuming Q = 3:



Example of Machining Loci

LE33013R0300900180001

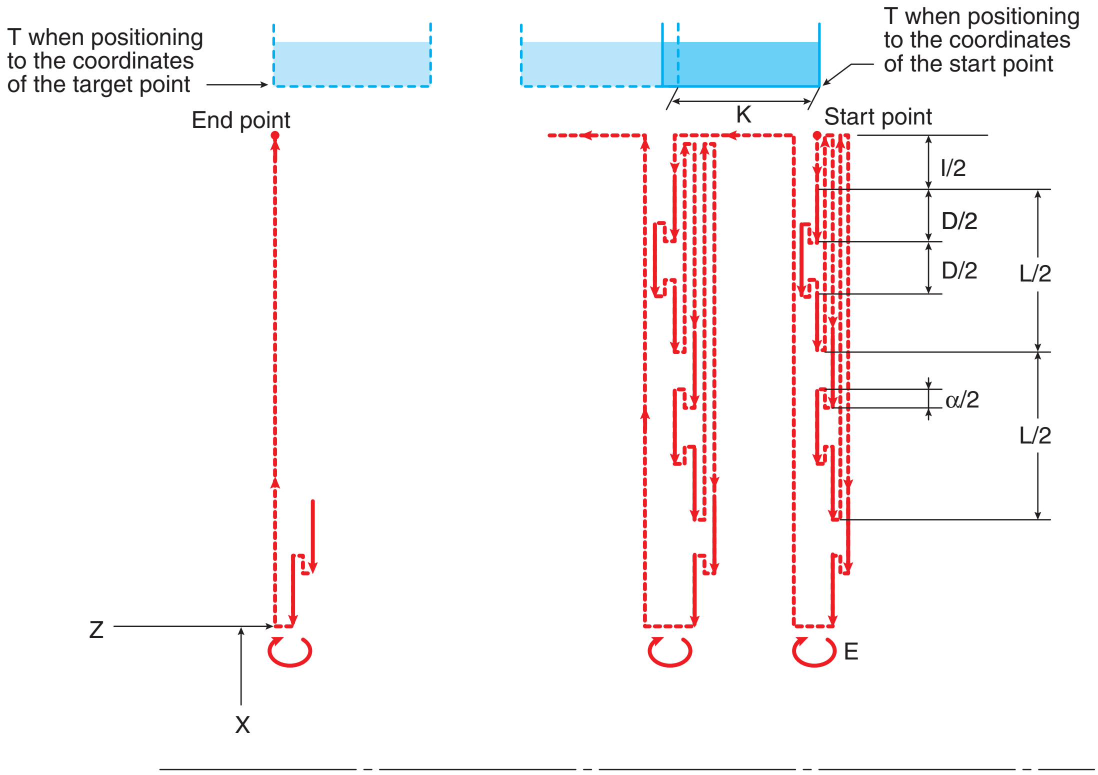
[Details]

- Command range: 0 to 9999
- If the Q command is omitted, the control assumes Q = 1.
- In a multi-thread thread cutting cycle, cutting is carried out in the order of 1st, 2nd ... "n"th thread. Then, cutting is repeated in the order of 1st, 2nd ... "n"th thread with different infeed amounts.

6. Grooving/Drilling Compound Fixed Cycle

6-1. Longitudinal Grooving Fixed Cycle (G73)

[Function]
In the G73 mode, a grooving cycle is performed as shown below.



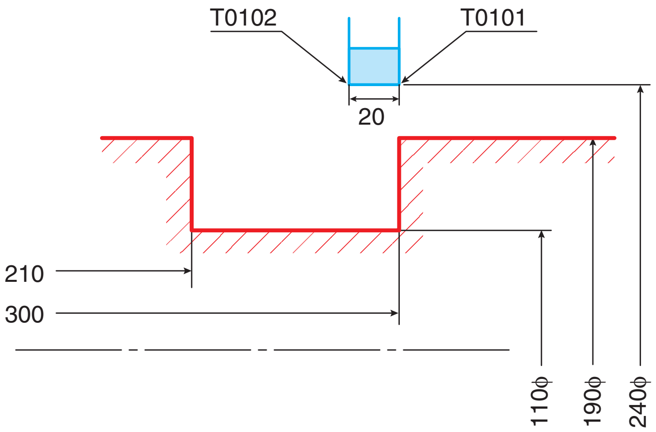
LE33013R0300900190001

[Programming format]
G73 X__ Z__ I__ K__ D__ L__ F__ E__ T__

- X : X coordinate of target point
- Z : Z coordinate of target point
- I : Shift amount in X-axis direction (as a diameter; if no I word is specified the control assumes I = 0)
- K : Shift amount in Z-axis direction (if no K word is specified the control assumes K=0)
- D : Depth of cut (infeed amount)
- L : Total infeed amount for tool withdrawal motion (as a diameter; tool sequence is not performed when L word is not specified.)
- DA : Retraction amount "a" is specified. When no DA word is specified, the amount set with the optional parameter (long word) No. 7 is used as the retraction amount. This applies both in the G94 and G95 modes.
- E : Duration of dwell motion when target point on X-axis is reached (Command unit is the same as for an F word in the G04 mode.) If no E word is specified, this sequence is not performed.

T : Tool offset number determining the tool offset amount when target point on the Z-axis is reached.
(If no T word is specified, the tool offset number selected on positioning to the starting point of the grooving cycle is selected. The T command after this block is the one designated when positioning to the starting point is performed.)

6-2. Example Program for Longitudinal Grooving Compound Fixed Cycle (G73)



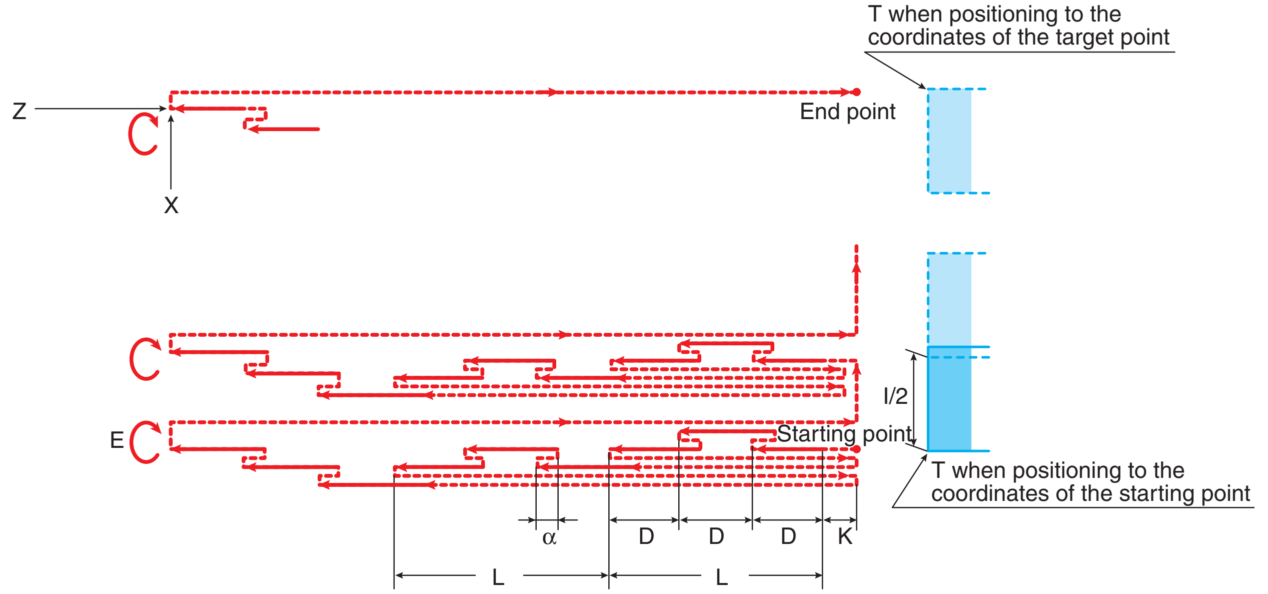
N0001	G00	X1000	Z1000	S300	T0101	M03	M42		
N0002		X240	Z300						
N0003	G73	X110	Z210	I45	K18	D20	E0.2	F0.3	T0102

* : (Z-axis tool offset amount of #2) - (Z-axis tool offset amount of #1) = 20

LE33013R0300900200001

6-3. Transverse Grooving/Drilling Fixed Cycle (G74)

In the G74 mode, a grooving cycle is performed as shown below.



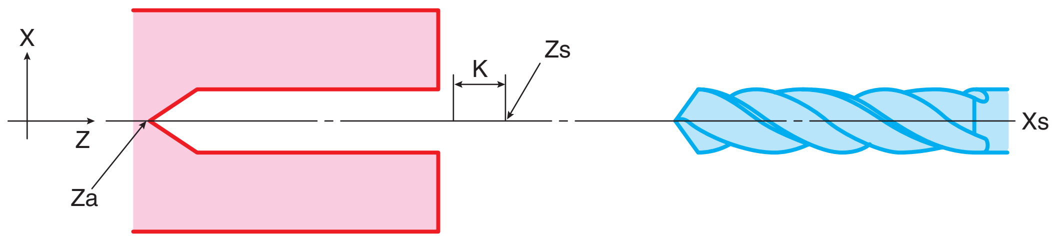
LE33013R0300900210001

[Programming format]
G74 X__ Z__ I__ K__ D__ L__ F__ E__ T__

- X : X coordinate of target point
- Z : Z coordinate of target point
- I : Shift amount in X-axis direction (as a diameter; if no I word is specified, the control assumes I = 0)
- K : Shift amount in Z-axis direction (if no K word is specified, the control assumes K = 0)
- D : Depth of cut (infeed amount)
- L : Total infeed amount for tool withdrawal motion (The sequence is not performed when no L word is specified.)
- DA : Retraction amount "a" is specified. When no DA word is specified, the amount set at Pecking amount in grooving and drill cycle of optional parameter (OTHER FUNCTION 1) is used as the retraction amount. This applies both in the G94 and G95 modes.
- E : Duration of dwell when target point on Z-axis is reached (Command unit is the same as an F word in G04 mode.) If no E word is specified, this sequence is not performed.
- T : Tool offset number determining the tool offset amount when target point on X-axis is reached. (If no T word is specified, the tool offset number selected on positioning to the starting point of the grooving cycle is selected. The T command after this block is the one designated when positioning to the starting point is performed.)

6-4. Example Program for Transverse Grooving/Drilling Fixed Cycle (G74)

Example: Drill cycle program



N0001	G00		S	T	M			
N0002		Xs	Zs					
N0003	G74	Xs	Za	K	D	L	E	F

LE33013R0300900220001

[Supplement]

A Z coordinate must always be specified in the G74 block.

6-5. Axis Movements in Grooving/Drilling Compound Fixed Cycle

(1) The axis moves the amount specified by "I (K)" at a rapid traverse rate along the X (or Z) axis from the cycle starting point.

(2) After the axis has been infed by the amount "D", it retracts by the amount "DA" at a rapid traverse rate.
This peck-feeding cycle is repeated until the programmed target point in the infeed axis direction is reached.

(3) When an L word is specified in the program, the infeed axis returns to the cycle start point each time total infeed amount in the repeated peck feeding cycles reaches "L".

(4) When the target point in the infeed axis direction is reached, dwell motion is activated for the duration commanded in an E word. If no E word is specified, dwell motion is not performed.
After that, the axis returns to the cycle starting point level, and then a shift is executed in another axis direction by the commanded amount "K" or "I" at a rapid traverse rate.

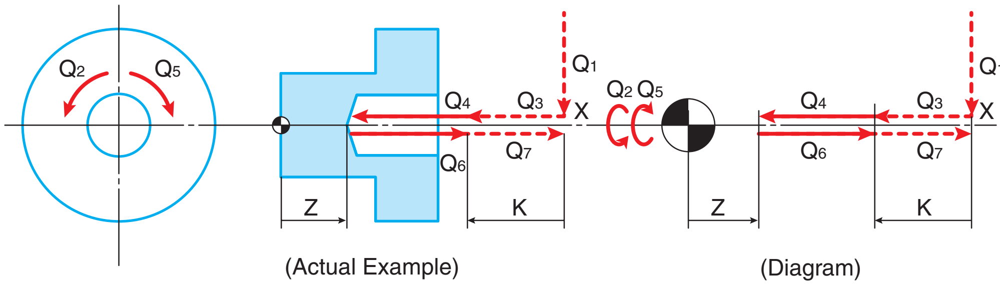
(5) This completes one grooving cycle. The steps (1) through (4) are repeated to machine the desired groove.

(6) When the offset tool position (offset number specified in the same block) reaches or goes beyond the target point in the X or Z axis direction during repetition of a grooving cycle with shift, the target point of the shift operation is taken as the final target point of the cycle; the final grooving cycle is performed at that position. When the axis reaches the target depth in the final grooving cycle, the axes return to the starting point of the compound fixed cycle.

7. Tapping Compound Fixed Cycle

7-1. Right-hand Tapping Cycle (G77)

[Function]
The compound cycle called out by G77 executes a tapping cycle like the one illustrated below.



(Actual Example) (Diagram)

LE33013R0300900240001

[Programming format]
G77 X__ Z__ K__ F__

G77 : G code to call out tapping compound fixed cycle.
Specify this G code immediately after a sequence number (name).
X : X coordinate of tapping cycle start point (target point)
Z : Z coordinate of tapping cycle end point (target point)
K : Rapid axis feedrate for axis feed from the cycle start point to the cutting start point
F : Feedrate

Axis movements:

Q_1 : The X-axis is positioned at the specified positioning target point (cycle start point) at a rapid feedrate. In this positioning cycle, no Z-axis movement occurs and thus the turret must be positioned at a point where it will not interfere with the workpiece during this positioning before calling out the G77 cycle.

Q_2 : The spindle rotates clockwise at the speed applying before the G77 cycle is called. Therefore, the required spindle speed must be specified before calling the G77 cycle. If this compound fixed cycle is called without designating a spindle speed, axis infeed does not occur since the spindle does not rotate and thus the cycle is halted.

Q_3 : The Z-axis is positioned at a position designated by a K word at a rapid feedrate.

Q_4 : Tapping is performed from the point reached in Q_3 to the depth specified by a Z word at a specified feedrate (F).

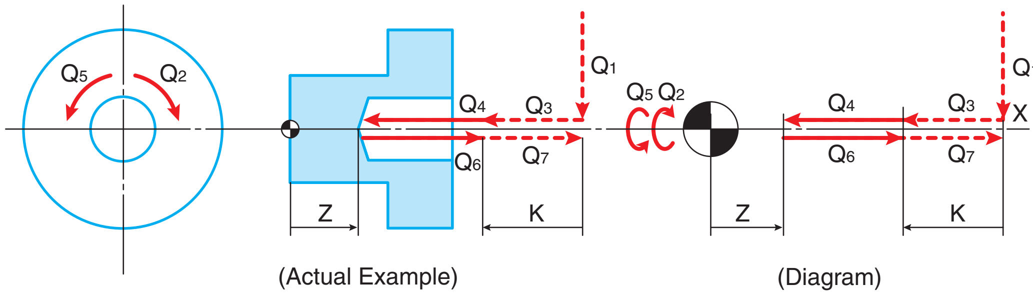
Q_5 : The spindle stops once and then starts in the reverse direction at the same speed as used in infeeding.

Q_6 : The Z-axis retracts to a point reached in the Q_4 cycle at a cutting feedrate.

Q_7 : The Z-axis retracts to a point reached in the Q_3 cycle at a rapid feedrate.

7-2. Left-hand Tapping Cycle (G78)

[Function]
The compound cycle called out by G78 executes a tapping cycle like the one illustrated below.



(Actual Example) (Diagram)

LE33013R0300900250001

[Programming format]
G78 X__ Z__ K__ F__

- G78 : G code to call out tapping compound fixed cycle.
  Specify this G code immediately after a sequence number (name).
- X : X coordinate of tapping cycle start point (target point)
- Z : Z coordinate of tapping cycle end point (target point)
- K : Rapid axis feedrate for axis feed from the cycle start point to the cutting start point
- F : Feedrate

Axis movements:

- Q_1 : The X-axis is positioned at the specified positioning target point (cycle start point) at a rapid feedrate. In this positioning cycle, no Z-axis movement occurs and thus the turret must be positioned at a point where it will not interfere with the workpiece during this positioning before calling out the G78 cycle.
- Q_2 : The spindle rotates counterclockwise at the speed applying before the G77 cycle is called. Therefore, the required spindle speed must be specified before calling the G78 cycle. If this compound fixed cycle is called without designating a spindle speed, axis infeed does not occur since the spindle does not rotate and thus the cycle is halted.
- Q_3 : The Z-axis is positioned at a position designated by a K word at a rapid feedrate.
- Q_4 : Tapping is performed from the point reached in Q_3 to the depth specified by a Z word at a specified feedrate (F).
- Q_5 : The spindle stops once and then starts in the forward direction at the same speed as used in infeeding.
- Q_6 : The Z-axis retracts to a point reached in the Q_4 cycle at a cutting feedrate.
- Q_7 : The Z-axis retracts to a point reached in the Q_3 cycle at a rapid feedrate.

[Supplement]

- While the tapping compound cycle is being executed, the feedrate override is fixed at 100%.
- Even when the SLIDE HOLD button is pressed during the execution of the tapping compound fixed cycle, the slide hold function is ignored. The single block function is also ignored even when the SINGLE BLOCK switch has been turned on.
- After the execution of the tapping compound cycle (G77, G78), the spindle stops and the stop state remains in effect. When cutting is to be conducted continuously, specify the spindle start command before progressing to the subsequent operation.

8. Compound Fixed Cycles

8-1. List of Compound Fixed Cycle Commands

Code	Cycle Name	Programming Format	Remarks
G181	Drilling Cycle (With repeat function)	G181, X, Z, C, R, I(K), F, Q, E	Used for drilling operation.
G182	Boring Cycle (With repeat function)	G182, X, Z, C, R, I(K), F, Q, E	Used for boring operation carried out with a boring bar or a similar tool.
G183	Deep Hole Drilling Cycle (With repeat function)	G183, X, Z, C, R, I(K), F, Q, D, E, L	Permits cutting chips to be broken while drilling a deep hole.
G184	Tapping Cycle (With repeat function)	G184, X, Z, C, R, I(K), F, Q, E	Used for tapping operation.
G185	Thread Cutting Cycle (Longitudinal) (Without repeat function)	G185, X, Z, C, I, K, F, SA=	Used for longitudinal thread cutting operation.
G186	Thread Cutting Cycle (Transverse) (Without repeat function)	G186, X, Z, C, I, K, F, SA=	Used for transverse thread cutting operation on end face.
G187	Straight Thread Cutting Cycle (Longitudinal) (Without repeat function)	G187, X, Z, C, I, K, F, SA=	Used for continuous longitudinal thread cutting operation.
G188	Straight Thread Cutting Cycle (Transverse) (Without repeat function)	G188, X, Z, C, I, K, F, SA=	Used for continuous transverse thread cutting operation on end face.
G189	Reaming/Boring Cycle (With repeat function)	G189, X, Z, C, R, I(K), F, Q, E	Used for reaming operation.
G190	Key Way Cutting Cycle (With repeat function)	G190, X, Z, C, I(K), D, U(W), E, F, Q, M211 (M212), M213 (M214)	Used for key way cutting.
G178	Synchronized tapping-forward (With repeat function)	G178, X, Z, C, R, I(K), F, D, J, Q, M141, M136	Used for tapping using the rigid tapper
G179	Synchronized tapping-reverse (With repeat function)	G179, X, Z, C, R, I(K), F, D, J, Q, M141, M136	Used for tapping using the rigid tapper
G180	Cancel of Fixed Cycle	G180	Used to cancel a fixed cycle mode presently selected. G180 must be programmed in a block without other commands.

[Supplement]

1) In the G185, G186, G187, and G188 fixed cycle modes, feedrates can be programmed only in the G95 (mm/rev) mode. In this case, an F command indicates the feed per C-axis revolution.
2) In the modes G181 through G184, G189, and G190, feedrates can be programmed only in the G94 (mm/min) mode. Feedrate commands in units of mm/rev are not accepted.
3) In G181 through G184, G189, and G190 modes, the control judges the cutting direction on the basis of the programmed I and K words: I for cutting in the X-axis direction and K for cutting in the Z-axis direction.
4) The "SA =" command is effective only in modes G185 through G188.

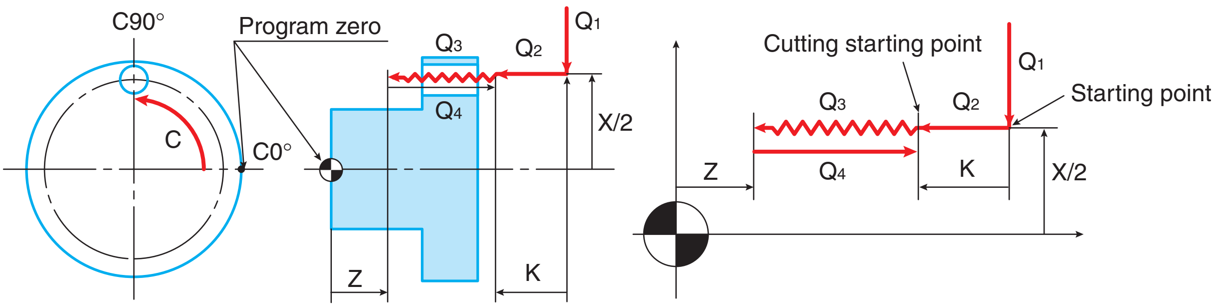
8-2. Basic Axis Motions

This section describes the basic axis motions in each cycle. For details on address characters and M codes, refer to sections 10-3 and 10-4 respectively.

8-2-1. G181, G182, G183, G184, G178, G179 and G189 modes

In these modes, the following cycle is carried out in a single block of commands.

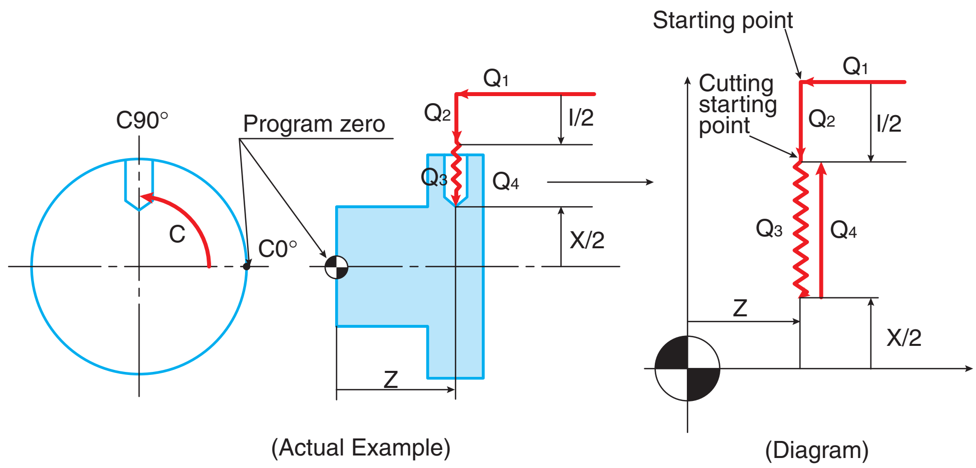
Face Machining (With K command)



(Actual Example) (Diagram)

LE33013R0300900280001

Side Machining (With I command)



LE33013R0300900280002

	Face Machining (With K command)	Side Machining (With I command)
Q1	Positioning of X- and C-axis at the rapid feedrate	Positioning of Z- and C-axis at the rapid feedrate
Q2	Positioning of Z-axis at the point "Q1 - K" at the rapid feedrate	Positioning of X-axis at the point "Q1 - I" at the rapid feedrate
Q3	Cutting along Z-axis from point Q2 to the commanded point Z	Cutting along X-axis from point Q2 to the commanded point X
Q4	Z-axis returns to the point where cutting started (Q3) at either a specified feedrate or the rapid feedrate depending on the called fixed cycle mode.	X-axis returns to the point where cutting started (Q3) at either a specified feedrate or the rapid feedrate depending on the called fixed cycle mode.

Axis movement sub cycles Q3 and Q4 are repeated each time a C command is given or according to the commanded Q word.

[Supplement]

1) For K or I commands, only positive values are allowed. If a negative value is specified, an alarm occurs.
2) The axis feed direction is determined automatically. The axis is then fed by amount K or I.
3) In the Q3 cycle, the end point of cutting may be specified by an R command.

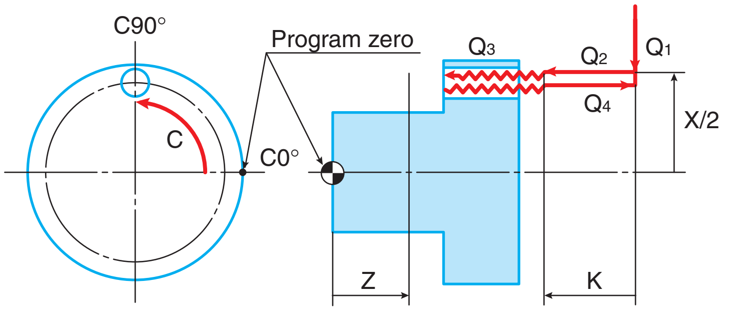
- Return point designation for the fixed cycle
  In the Q4 cycle, the axis is returned to the cutting start point after the completion of cutting. However, this return point may be changed to the cycle start point by changing the setting at Multi cycle return point of optional parameter (MULTIPLE MACHINING) or the M code specified in a part program.

  M code : Designation of shape in compound fixed cycle
  M136 By specifying this M code, it is possible to return the axis to the start point (rapid feed start point) after the completion of a Q4 cycle, as in the case when "1" is set for the optional parameter.

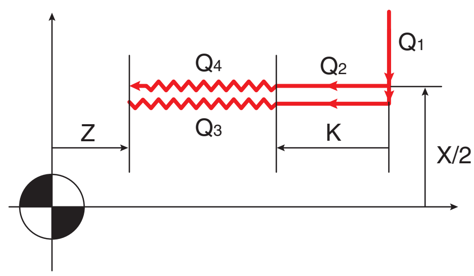
This M code is cleared by the reset operation and it is effective only in the specified block.
An M code is given priority over the optional parameter setting.
When no M code is designated, the optional parameter setting becomes effective.

- The basic axis motions of the tapping cycle (G184), synchronized tapping cycle (G178/G179), and milling/boring cycle (G189) are shown below when the setting for the optional parameter indicated above is "1" or if an M136 command exists in the program.

Face Machining (With K command)



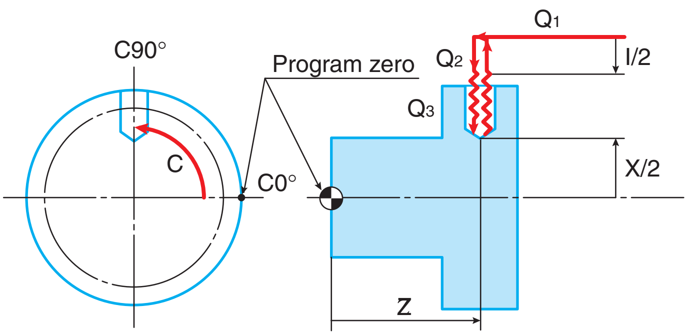
(Actual Example)



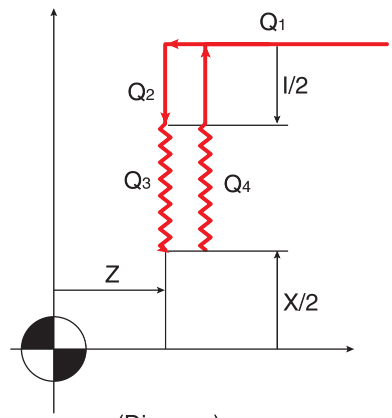
(Diagram)

LE33013R0300900280003

Side Machining (With I command)



(Actual Example)



(Diagram)

LE33013R0300900280004

	Face Machining (With K command)	Side Machining (With I command)
Q_1	Positioning of X- and C-axis at the rapid feedrate	Positioning of Z- and C-axis at the rapid feedrate
Q_2	Positioning of Z-axis at the point "Q1 - K" at the rapid feedrate	Positioning of X-axis at the point "Q_1 - I" at the rapid feedrate
Q_3	Cutting along Z-axis from point Q_2 to the commanded point Z	Cutting along X-axis from point Q_2 to the commanded point X
Q_4	X-axis returns to Q_3 where cutting started at a cutting feedrate and then to Q_2 at the rapid feedrate.	X-axis returns to Q_3 where cutting started at a cutting feedrate and then to Q_2 at the rapid feedrate.

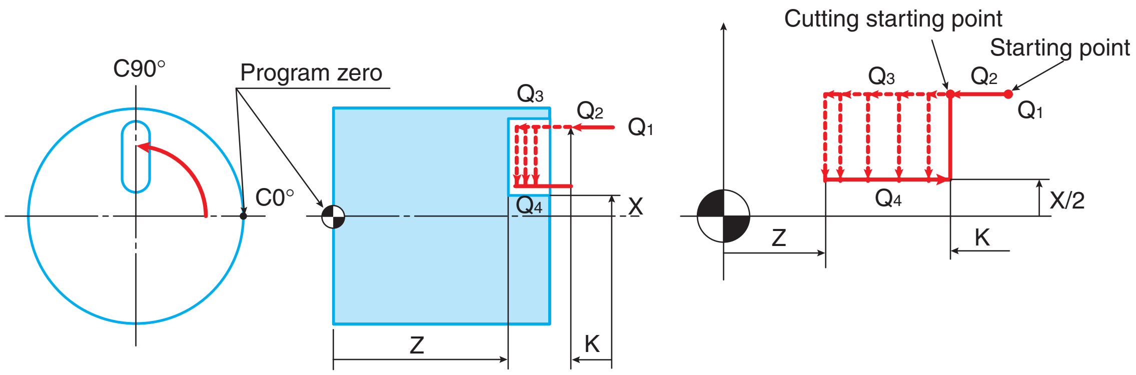
- C-axis clamp effective/ineffective command
  When the workpiece is cut using a small-diameter drill in the compound fixed cycle, or when the material to be cut is soft, the C-axis does not need to be clamped during cutting.
  When M141 (C-axis clamp ineffective) is designated, C-axis clamp motion is eliminated, resulting in a reduced cycle time.
  M169 is only effective within one block.

- Ignoring the M-tool constant speed rotation answer for M140 (tapping cycle)
  In the tapping cycle, cutting feed starts after receiving the M-tool constant speed rotation answer. Because of this, a time lag occurs between the start of tool rotation and the start of cutting feed. Normally, the time lag is adjusted by a mechanism in the tapping unit. If the time lag cannot be adjusted, designate M140 (ignoring the M-tool constant speed rotation answer). The M-tool constant speed rotation answer is ignored.

8-2-2. G190 mode

In this mode, the following cycle is carried out in a single block of commands.

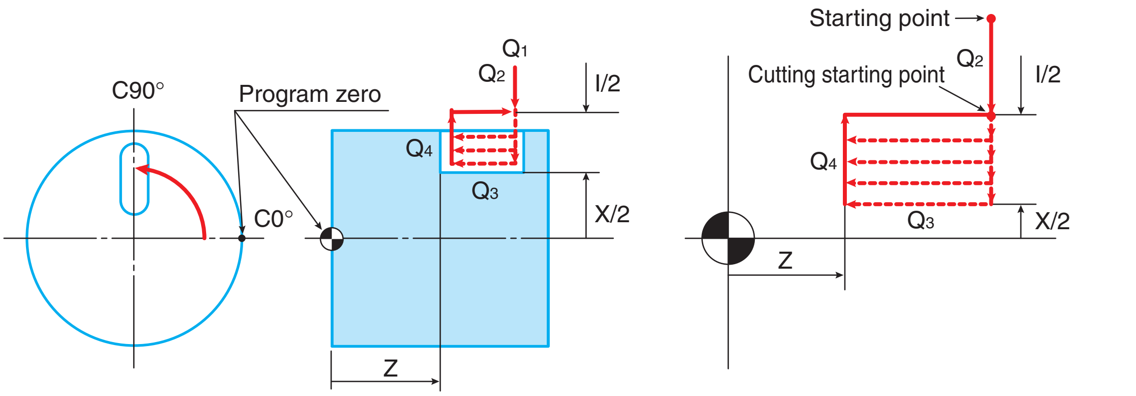
Face Machining (With K command)



(Actual Example) (Diagram)

LE33013R0300900290001

Side Machining (With I command)



(Actual Example) (Diagram)

LE33013R0300900290002

	Face Machining (With K command)	Side Machining (With I command)
Q_1	Positioning of C-axis at the rapid feedrate	Positioning of C-axis at the rapid feedrate
Q_2	Positioning of Z-axis at the point "Q_1 - K" at the rapid feedrate	Positioning of X-axis at the point "Q_1 - I" at the rapid feedrate
Q_3	Cutting along Z-axis from point Q_2 to the commanded point Z	Cutting along X-axis from point Q_2 to the commanded point X
Q_4	Z-axis returns to the Q_3 cycle start point at the rapid feedrate. The cycle is repeated until Z-axis reaches the programmed Z level.	X-axis returns to the Q_3 cycle start point at the rapid feedrate. The cycle is repeated until X-axis reaches the programmed X level.

Axis movement sub cycles Q_3 and Q_4 are repeated each time a C command is given or according to the commanded Q word.

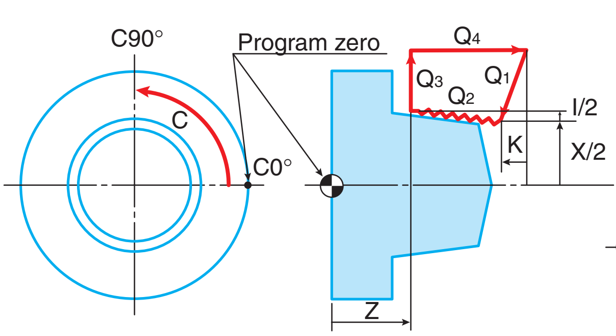
[Supplement]

1) For K or I commands, only positive values are allowed. If a negative value is specified, an alarm occurs.
2) The axis feed direction is determined automatically. The axis is then fed by amount K or I.

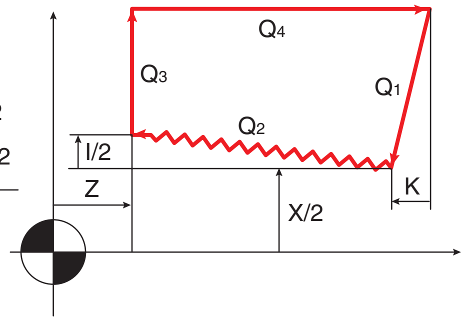
8-2-3. G185, G186, G187, and G188 modes

In these modes, the following cycle is carried out in a single block of commands.

Longitudinal Thread Cutting (G185 and G187)



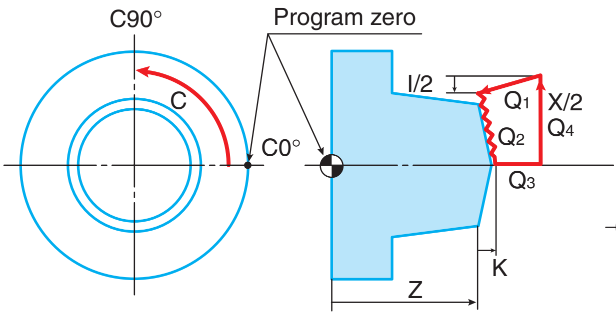
(Actual Example)



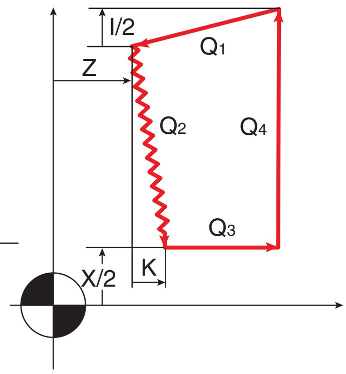
(Diagram)

LE33013R0300900300001

Transverse Thread Cutting (G186 and G188)



(Actual Example)



(Diagram)

LE33013R0300900300002

	Longitudinal Thread Cutting (G185 and G187)	**Transverse Thread Cutting (G186 and G188)**
Q_1	Positioning of X-, Z- (Z ± K) and C-axis at the rapid feedrate	Positioning of X- (X ± I), Z- and C-axis at the rapid feedrate
Q_2	Cutting along X- (X ± I), Z- and C-axis	Cutting along X-, Z- (Z ± K) and C-axis
Q_3	Positioning of X-axis at the starting point of sub cycle Q_1 at the rapid feedrate	Positioning of Z-axis to the starting point of sub cycle Q_1 at the rapid feedrate
Q_4	Positioning of Z-axis at the starting point of sub cycle Q_1 at the rapid feedrate	Positioning of X-axis to the starting point of sub cycle Q_1 at the rapid feedrate

In G187 or G188 mode operation, only sub cycles Q_1 and Q_2 are carried out.

8-3. Address Characters

X : For cutting on an end face and longitudinal thread cutting, "X" indicates the X-coordinate of the cycle starting point.
For cutting on an OD and transverse thread cutting as well as key way cutting, "X" indicates the X-coordinate of the end point of the cycle.

Z : For cutting on an end face and longitudinal thread cutting as well as key way cutting, "Z" indicates the Z-coordinate of the end point of the cycle.
For cutting on an OD and transverse thread cutting, "Z" indicates the Z-coordinate of the starting point of the cycle.

C : C-axis indexing angle

I : Shift amount in the G00 mode for cutting on an OD, cutting starting point in transverse thread cutting cycle, end point of taper thread in longitudinal thread cutting cycle

J : Number of threads

K : Shift amount in the G00 mode for cutting on an end face, cutting starting point in longitudinal thread cutting cycle, end point of taper thread cutting in transverse thread cutting cycle

F : Cutting feedrate

D : Depth of cut per peck feed in deep-hole drilling and key way cutting
Start position of tapping with M-tool spindle in synchronized tapping.

E : Duration of dwell motion at the end point in drilling, boring and tapping cycle (omissible)
Infeed amount in key way cutting

L : Axis relieving amount in deep-hole drilling cycle

U : Finish allowance in side key way cutting

W : Finish allowance in face key way cutting

SA = : Programmable only in multiple-fixed cycle of G185 through G188 (thread cutting cycles).
C-axis rotation speed command.
This SA= command is programmed to obtain the axis movement amount of the C-axis in G185 through G188 thread cutting cycles.

R : Infeed amount for drilling cycleSpecify the distance from the cutting starting point. The sign of the R command indicates the direction of cutting.
An R command in the X-axis direction should be given as a diametral value.

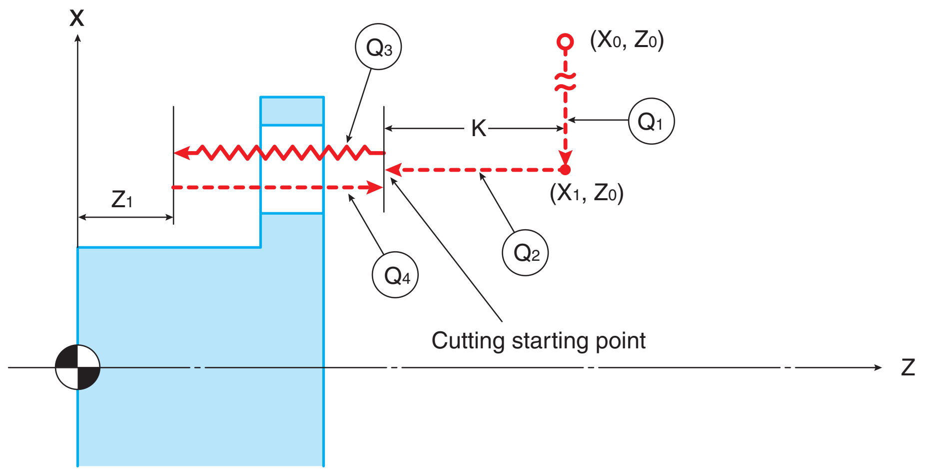
Q : The number of holes (equally spaced) to be machined using the multiple-fixed cycle repeat function

8-4. M Codes

- For designating key way cutting direction
  M211: One-directional cutting
  M212: Zigzag cutting

- For designating key way cutting method
  M213: Designated infeed
  M214: Equal infeed

8-5. Drilling Cycle (G181)



LE33013R0300900330001

[Program format]

		:				
N100	G0	X0	Z0			
N101	G94				SB=	
N102	G181	X1	Z1	C	K	F
N103	G180					
		:				

LE33013R0300900330002

Cycle operation

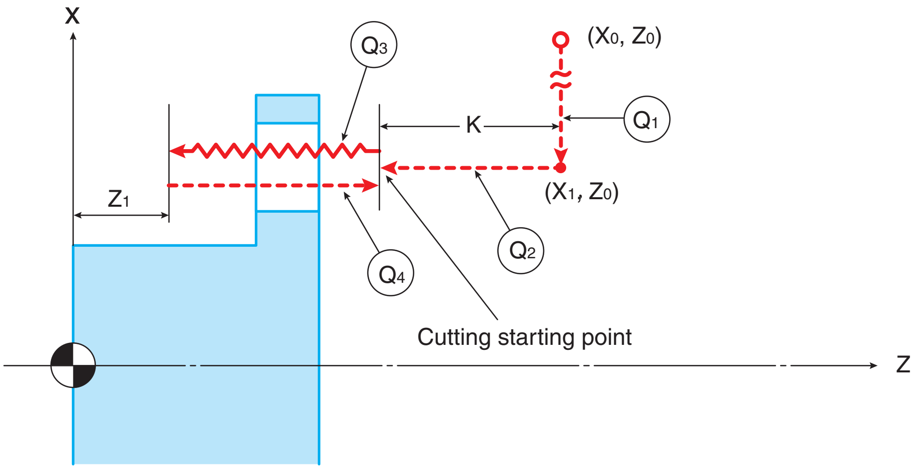
Q_1 : The axes are positioned in the G00 mode at the point specified by (X_1, Z_0), and the C command value. After the completion of positioning, the M-tool spindle starts rotating in the forward direction.

Q_2 : The Z-axis is positioned at a point "-K" from Z_0. After the completion of positioning, the C-axis is clamped.

Q_3 : Cutting is performed up to Z_1 in the G01 mode.

Q_4 : The axes are positioned at the cutting starting point in the G00 mode. After the completion of positioning, the C-axis is unclamped.

8-6. Boring Cycle (G182)



LE33013R0300900340001

[Program format]

		:					
N100	G00	X_0	Z_0				
N101	G94			SB=			
N102	G182	X_1	Z_1	C	K	F	E
N103	G180						
		:					

LE33013R0300900340002

Cycle operation

Q_1 : The axes are positioned in the G00 mode at the point specified by (X_1, Z_0) and the C command value. After the completion of positioning, the M-tool spindle starts rotating in the forward direction.

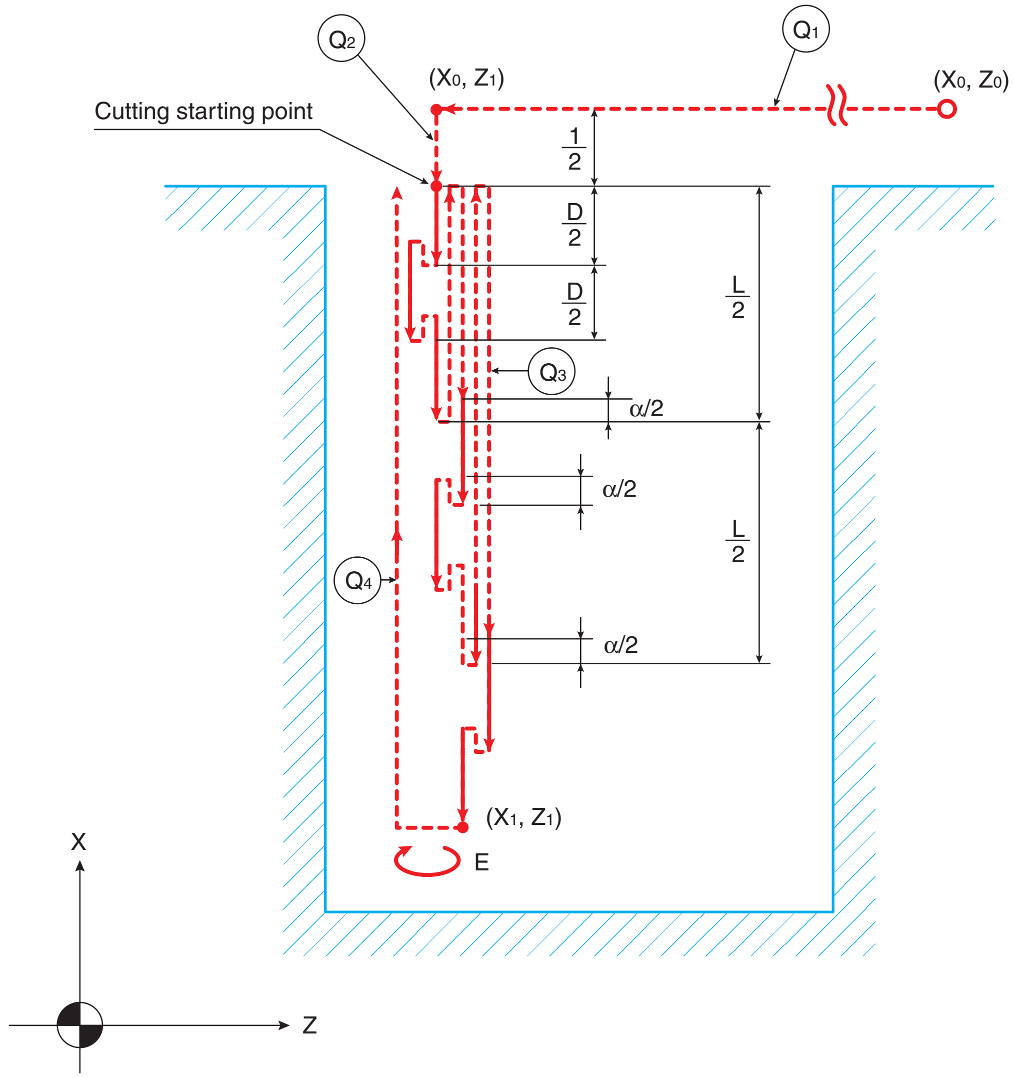
Q_2 : The Z-axis is positioned at a point "-K" from Z_0. After the completion of positioning, the C-axis is clamped.

Q_3 : Cutting is performed up to Z_1 in the G01mode. After the completion of axis movement (cutting), the axis dwells for the time specified by "E"(omissible).
After the completion of the dwell command, the M-tool spindle stops rotating.

Q_4 : The axes are positioned at the cutting starting point in the G00 mode after the M-tool spindle has been stopped. After the completion of positioning, the C-axis is unclamped and the M-tool spindle rotates in the forward direction.

An E command in the Q_3 cycle should be programmed in the same manner as an F command in the G04 mode.

8-7. Deep Hole Drilling Cycle (G183)



LE33013R0300900350001

[Program format]

		:							
N100	G00	X_0	Z_0						
N101	G94			SB=					
N102	G183	X_1	Z_1	C	I	F	E	D	L
N103	G180								
		:							

LE33013R0300900350002

Cycle operation

Q_1 : The axes are positioned in the G00 mode to the point specified by (X_0, Z_1) and the C command value. After the completion of positioning, the M-tool spindle starts rotating in the forward direction.

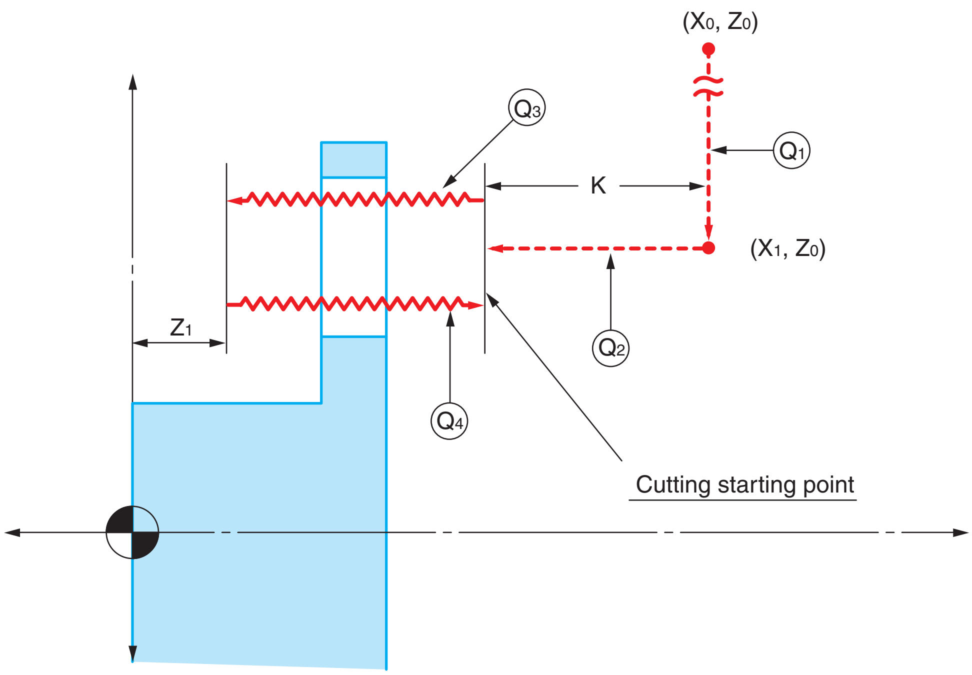
Q_2 : The X-axis is positioned at a point "-I" from X_0. After the completion of positioning, the C-axis is clamped.

Q_3 : A drilling cycle in step feed mode is carried out up to X_1.
"Step feed" means the axis movement illustrated in the diagram. That is, the axis is fed by "D" and then it retracts by "α" at the rapid feedrate. This infeed and rapid retraction cycle is repeated until the total infeed amount reaches "L", where the axis is returned up to the cutting starting point. The axis is then infed to the previous drilled depth and then the cycle indicated above is repeated up to the target point X_1.
At the bottom of the hole, the dwell function is activated for time duration "E" (omissible).

Q_4 : The axes are positioned at the cutting starting point in the G00 mode. After the completion of positioning, the C-axis is unclamped.

For "α", the value set at Pecking amount in drilling cycle of optional parameter (MULTIPLE MACHINING) is used.

8-8. Tapping Cycle (G184)



LE33013R0300900360001

[Program format]

		:					
N100	G00	X_0	Z_0				
N101	G94			SB=			
N102	G184	X_1	Z_1	C	K	F	E
N103	G180						
		:					

LE33013R0300900360002

Cycle operation

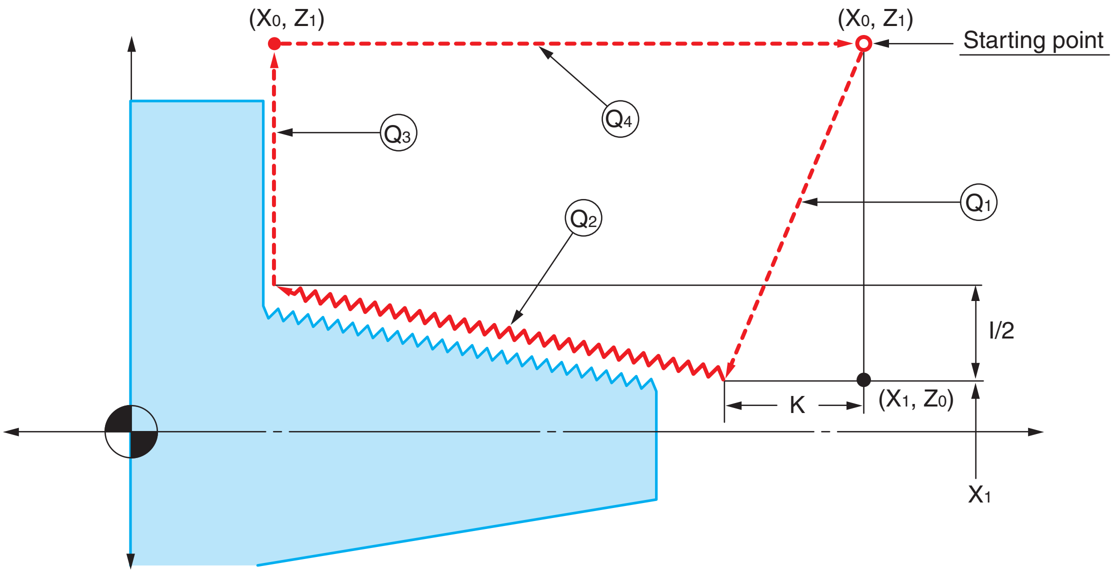
Q_1 : The axes are positioned in the G00 mode at the point specified by (X_1, Z_0) and the C command value. After the completion of positioning, the M-tool spindle starts rotating in the forward direction.

Q_2 : The Z-axis is positioned at a point "-K" from Z_0. After the completion of positioning, the C-axis is clamped.

Q_3 : Cutting is performed up to Z_1 in the G01 mode. After the completion of axis movement (cutting), the axis dwells for "E" (omissible).
After the completion of dwell command, the M-tool spindle stops and then reverses its rotating direction.

Q_4 : After the M-tool spindle has started to rotate in the reverse direction, the axis is fed up to the cutting starting point in the G01 mode.
After the axis has returned to the cutting starting point, the C-axis is clamped, and the M-tool spindle stops and rotates in the forward rotation.

8-9. Longitudinal Thread Cutting Cycle (G185)



LE33013R0300900370001

[Program format]

```
                    :
N100   G00   X0   Z0
N101   G95              SB=
N102   G185  X1   Z1    C   I   K   F   SA=
N103   G180
                    :
```

LE33013R0300900370002

Cycle operation

Q_1 : The axes are positioned in the G00 mode at the point specified by (X_1, Z_0 - K) and the C command value. After the completion of positioning, the M-tool spindle starts rotating in the forward direction.

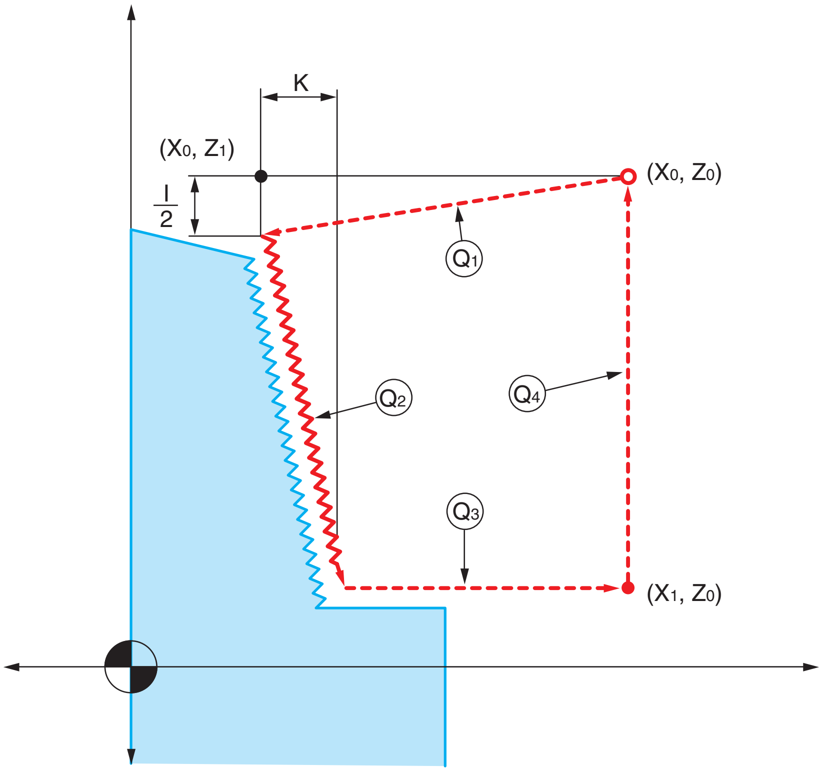
Q_2 : The C-axis starts rotation and the thread cutting cycle is carried out up to point (X_1 + I, Z_1) in the G01 mode. After the completion of thread cutting, the C-axis stops rotation.

Q_3 : The axes are positioned in the G00 mode at X_0

Q_4 : The axes are positioned in the G00 mode at the starting point.

In G185 thread cutting mode operation, cutting feed is synchronized with the rotation of the C-axis. Therefore, the F command must be equivalent to one pitch of the thread.

8-10. Transverse Thread Cutting Cycle (G186)



LE33013R0300900380001

[Program format]

		:					
N100	G00	X_0	Z_0				
N101	G95			SB=			
N102	G186	X_1	Z_1	C	I	K	SA=
N103	G180						
		:					

LE33013R0300900380002

Cycle operation

Q_1 : The axes are positioned in the G00 mode to the point specified by (X_0 - I, Z_1) and the C command value. After the completion of positioning, the M-tool spindle starts rotation in the forward direction.

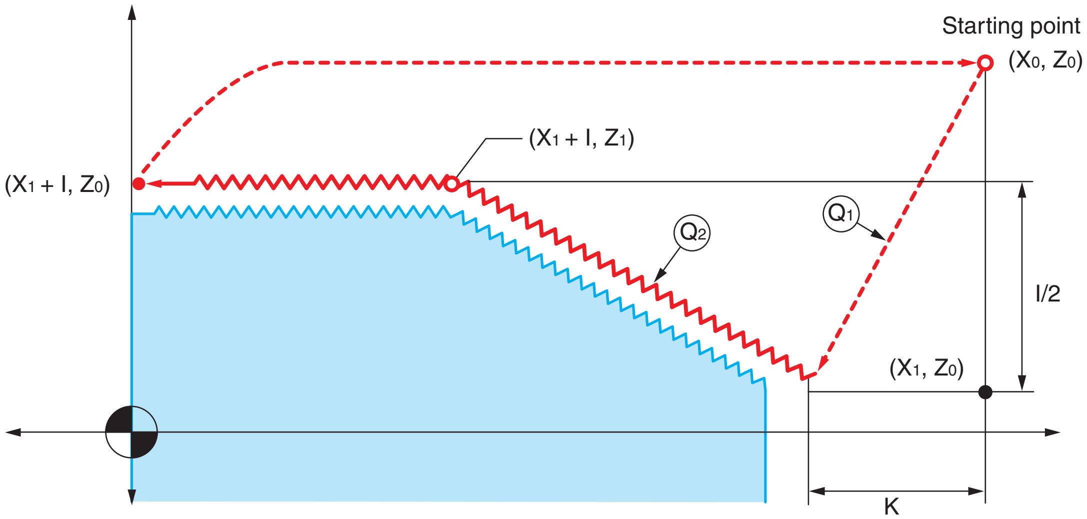
Q_2 : The C-axis starts rotation and the thread cutting cycle is carried out up to point (X_1, Z_1 + K) in the G01 mode. After the completion of thread cutting, the C-axis stops rotation.

Q_3 : The axes are positioned in the G00 mode at Z_0

Q_4 : The axes are positioning in G00 at the starting point

In G186 thread cutting mode operation, cutting feed is synchronized with the rotation of the C-axis. Therefore, the F command must be equivalent to one pitch of the thread.

8-11. Longitudinal Straight Thread Cutting (G187)



LE33013R0300900390001

[Program format]

```
                          :
N100    G00    X0      Z0
N101    G95                    SB=
N102    G187   X1      Z1      C    I    K    SA=
N103                   Z2                     SA=
N104    G180              :
```

LE33013R0300900390002

Since the G187 cycle contains only Q_1 and Q_2 cycles, repeated designation of G187 in succession as in the program above can cut threads continuously.

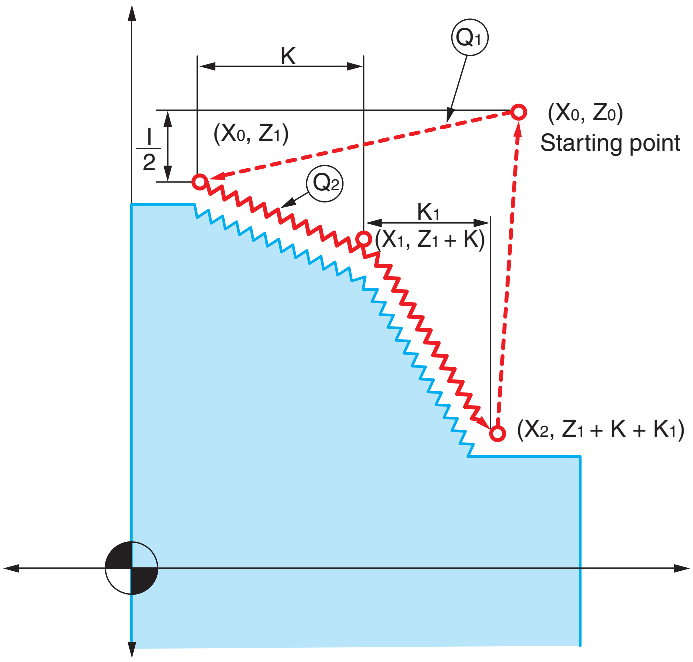
Cycle operation

Q_1 : The axes are positioned in the G00 mode to the point specified by X_1 and Z_0 - K. After the completion of positioning, the M-tool spindle starts rotation in the forward direction.

Q_2 : The C-axis starts rotation. The thread cutting cycle is carried out up to point (X_1 + I, Z) in the G01 mode.

The thread cutting cycle is carried out in accordance with the commands in sequence N103 up to the commanded target point (X_1 + I, Z_2). Then, the axes are returned to the starting point at the rapid feedrate by the command G180 (cancel) specified in the N104 sequence.

8-12. Transverse Straight Thread Cutting (G188)



LE33013R0300900400001

[Program format]

			:						
N100	G00	X_0	Z_0						
N101	G95			SB=					
N102	G188	X_1	Z_1	I	K	F	C	SA=
N103		X_2	Z_2		K1		C	SA=
N104	G180								
			:						

LE33013R0300900400002

Since the G188 cycle contains only Q_1 and Q_2 cycles, repeated designation of G188 in succession as in the program above can cut threads continuously.

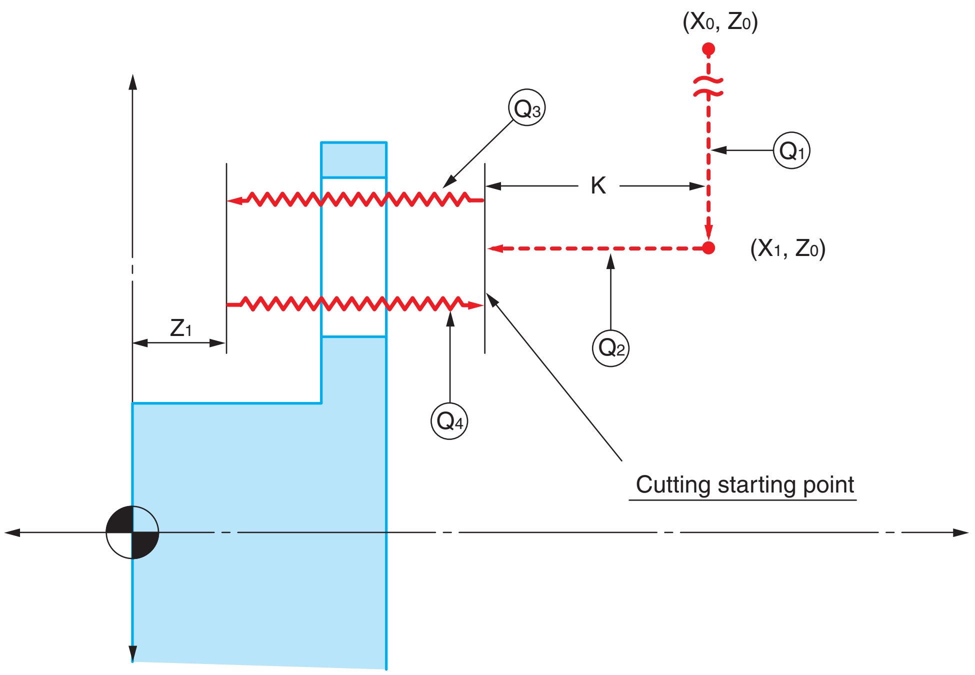
Cycle operation

Q_1 : The axes are positioned in the G00 mode to the point specified by X_0 - I and Z_1. After the completion of positioning, the M-tool spindle starts rotation in the forward direction.

Q_2 : The C-axis starts rotation and the thread cutting cycle is carried out up to point (X_1, Z_1 + K) in the G01 mode.

The thread cutting cycle is carried out in accordance with the commands in sequence N103 up to the commanded target point (X_1 + I, Z_2). Then, the axes are returned to the starting point at the rapid feedrate by the command G180 (cancel) specified in the N104 sequence.

8-13. Reaming/Boring Cycle (G189)



LE33013R0300900410001

[Program format]

			:					
N100	G00	X0	Z0					
N101	G94			SB=				
N102	G189	X1	Z1	C	K	F	Q	E
N103	G180							
			:					

LE33013R0300900410002

Cycle operation

Q_1 : The axes are positioned in the G00 mode to the point specified by (X_1, Z_0) and the C command value. After the completion of positioning, the M-tool spindle starts rotating in the forward direction.

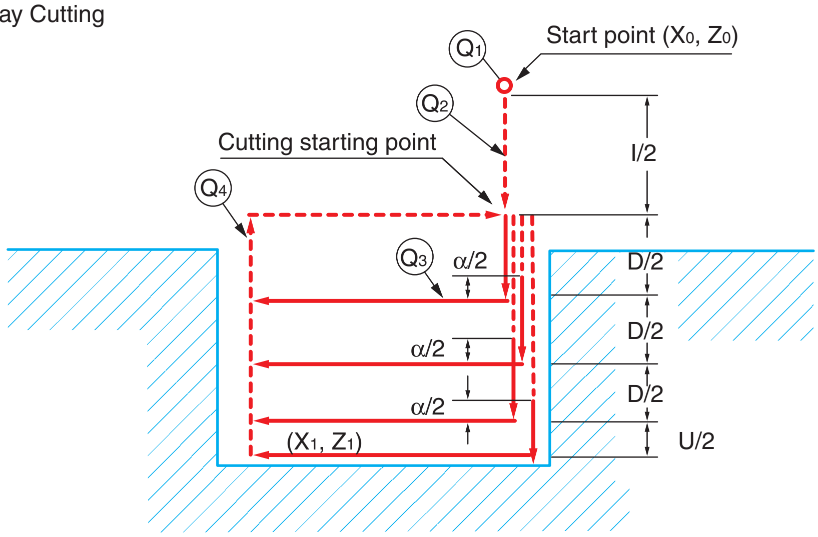
Q_2 : The Z-axis is positioned at a point "-K" from Z_0. After the completion of positioning, the C-axis is clamped (omissible).

Q_3 : Cutting is performed up to Z_1 in the G01 mode. After the completion of cutting, dwell for "E" is carried out (omissible).

Q_4 : Cutting is performed up to the cutting starting point in the G01 mode. After the completion of axis movement, the C-axis is unclamped.

8-14. Key Way Cutting (G190)

Side Key Way Cutting



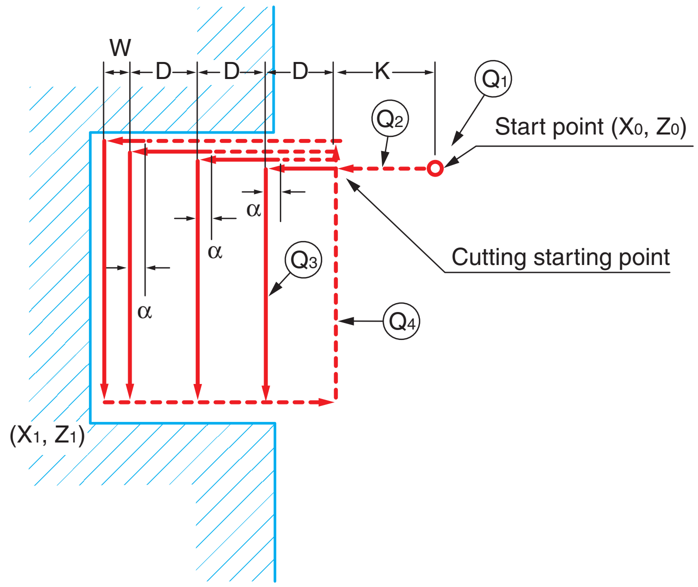
LE33013R0300900420001

[Program format]

		:				
N100	G00	X_0	Z_0			
N101	G94			SB=		
N102	G190	X_1	Z_1	C I D U E F	M211	M213
N103	G180					
		:				

LE33013R0300900420002

Face Key Way Cutting



LE33013R0300900420003

[Program format]

			:			
N100	G00	X0	Z0			
N101	G94			SB=		
N102	G190	X1	Z1	C K D W E F	M211	M213
N103	G180					
			:			

LE33013R0300900420004

Cycle operation

Q_1 : The X and Z axes are positioned at the designated position on the C-axis in the G00 mode. After the completion of positioning, the M-tool spindle starts rotating in the forward direction.

Q_2 : The X-axis (Z-axis for face key way cutting) is positioned at a point -I (-K for face key way cutting) from X_0 (Z_0 for face key way cutting) in the G00 mode. After the completion of positioning, the C-axis is clamped.

Q_3 : Key way cutting is carried out in the "one directional, designated infeed" mode. For the "one directional, designated infeed" mode, refer to "Key Way Cutting Modes" below.

Q_4 : The axes are positioned at the starting point in the G00 mode. After the completion of positioning, the C-axis is unclamped.

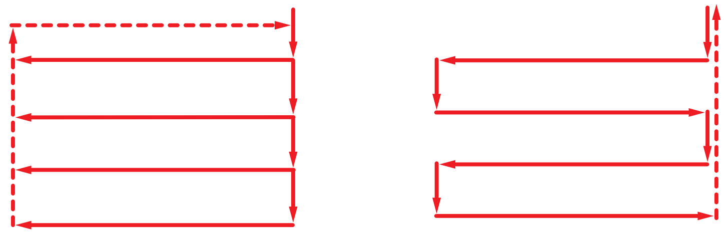
For "α", the value set at Pecking amount in drilling cycle of optional parameter (MULTIPLE MACHINING) is used.

Key Way Cutting Modes

In key way cutting cycles, it is possible to select the cutting direction and cutting method with M codes.

(1) Selection of cutting direction (M211, M212)

One-directional Cutting Mode (M211) | Zigzag Cutting Mode (M212)

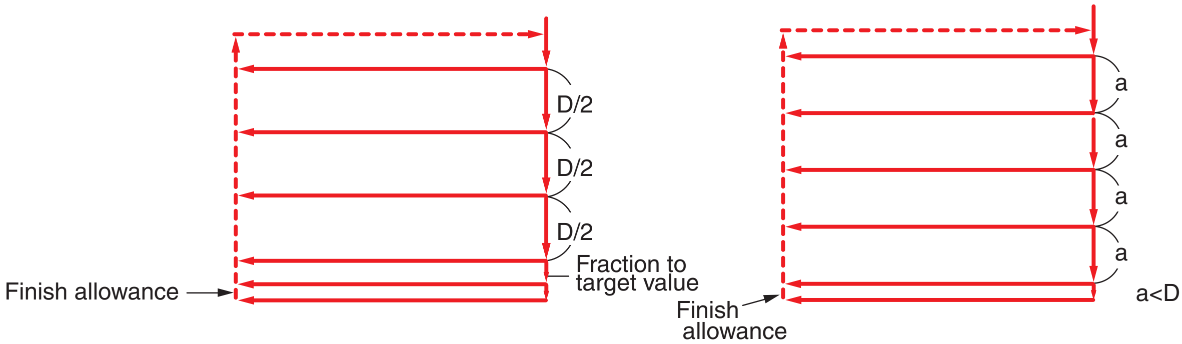


Cutting in one direction | Cutting direction changes along the cutting path

LE33013R0300900420005

(2) Selection of infeed mode (M213, M214)

Designated Infeed Mode (M213) | Equal Infeed Mode (M214)



The tool is infed by the designated amount "D"; in the final cutting path, the depth equivalent to the fraction is cut.

Infeed amount "a", obtained by dividing the total depth of cut to the target point into equal parts is determined so that "a" is not greater than "D".

LE33013R0300900420006

[Details]
In either cutting mode, the finish allowance U or W is left on the workpiece in the rough cutting cycle; this finish allowance is finally removed in the finish cutting cycle.

- Before starting fixed cycle mode operation, the C-axis must be placed in the unclamped (M146) state.
- In the G181 through G184, G189, G190, G178 and G179 modes, the first cycle is executed in the order Q_1, Q_2, Q_3, then Q_4. However, Q_3 and Q_4 are repeated after that, when a C or Q command is specified.
  The C-axis clamp and unclamp commands (M147 and M146) necessary for repeating sub cycles Q_3 and Q_4 are automatically generated.

- When the called fixed cycle mode is canceled, the control is in the M146 and M13 mode. Specify M147 and M12, if necessary.

- The block right after the one canceling the fixed cycle mode must contain both X- and Z-axis commands.

8-15. Synchronized Tapping Cycle

[Programming format]
G178 X__ Z__ C__ R__ K(I)__ F__ D__ J__ Q__ M141 M136
(G179)

G178	:	Forward tapping cycle command The forward tapping cycle can also be called by designating G184 instead of G178, but in this case "1" must be set in advance at G184 tapping mode of optional parameter (MULTIPLE MACHINING) (if the setting for this parameter is "0", G184 calls a conventional floating tap cycle).
G179	:	Reverse tapping cycle command
X, C	:	Tapping on the front surface (if not designated, tapping is performed at the position where the previous tapping cycle was carried out.)
Z, C	:	Tapping on the side surface
Z	:	Hole bottom level for tapping on the front surface
X	:	Hole bottom level for tapping on the side surface
R	:	Depth of cutThe total tap length, from the cutting start point, is designated as R. Cutting direction is indicated by a positive (+) or negative (-) sign preceeding the R. When tapping is carried out on the side surface along the X-axis, a diametrical value is designated for R. However, note that the R command and the hole bottom level command, Z (X), cannot be designated simultaneously.
K	:	Shift amount to the cutting start point for tapping in the front surface
I	:	Shift amount to the cutting start point for tapping in the side surface
F	:	Cutting feedrate Sets the feedrates for cutting feed during the cycle. Determine a value so that the M-tool spindle feed per revolution is equal to the thread pitch. Cutting feedrate F in the G94 and G95 modes is as follows, where P is the thread pitch (mm) and SB stands for the M-tool spindle speed (min^{-1}). G94 mode: <u>F = P x SB</u> (mm/min) G95 mode: <u>F = P</u> (mm/rev)

[Supplement]

In the synchronized tapping cycle, G95 (feed per revolution for the M-tool spindle) can be used.

D	:	Tapping start point of the M-tool spindle The position where the M-tool spindle starts tapping is designated as "D". The M-tool spindle is governed by "constant start point control". This is used when no new D command is assigned, therefore the M-tool spindle starts tapping at the position defined by the previously designated D command.

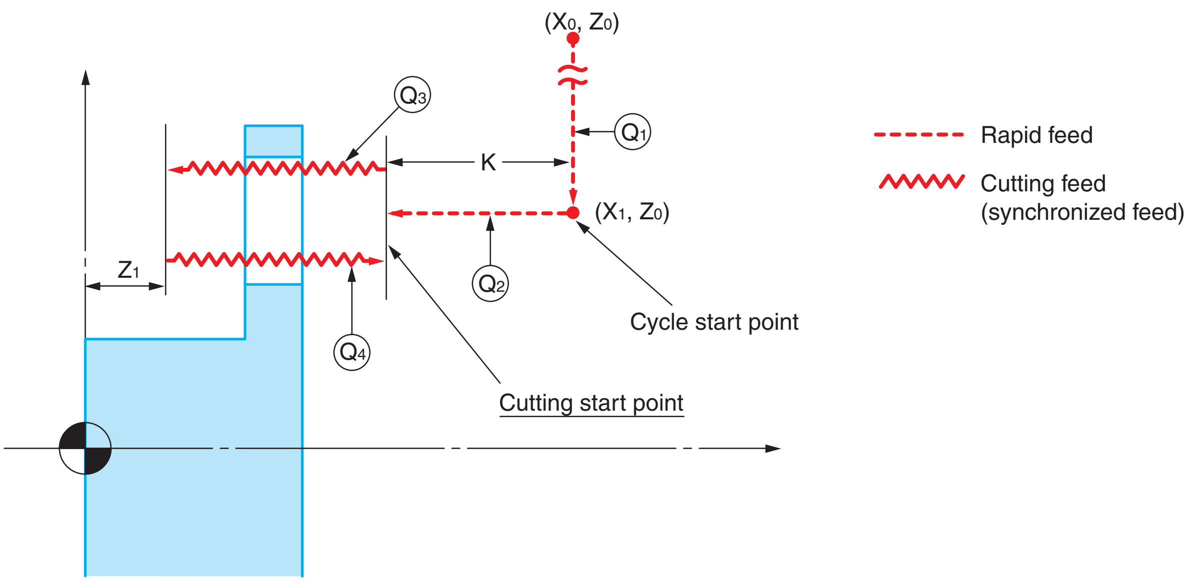
[Supplement]

When the one-point clutch specification is not selected for the M-tool spindle clutch, its start point is not guaranteed even when a D command is designated.

J : Number of threads
Whenever the number of threads per inch is specified with inch taps, it is convenient to use J as an indicator, to avoid confusion with metric measurements.

Q : Number of holes (for details, refer to "Repeat Function").

8-15-1. Forward Tapping Cycle (G178)



LE33013R0300900440001

[Programming format]

			:				
N100	G00	X0	Z0				
N101	G94				SB=		
N102	G178	X1	Z1	C	K	F	D
N103	G180						
			:				

LE33013R0300900440002

Cycle operation

Q_1 : The axes are positioned at the coordinates (X_1, C) at the rapid feedrate, and M-tool spindle rotation is stopped.

Q_2 : The axes are positioned at the cutting start point (Z_0, -K) at the rapid feedrate.With M-tool spindle one-point clutch models, the M-tool spindle oscillates after the completion of positioning at (Z_0, -K) to engage the clutch.
When a D command had been designated, the M-tool spindle is positioned at point D along the tapping rotation direction.

Q_3 : After the C-axis had been clamped, the M-tool spindle is synchronized with the Z-axis to point Z_1 while being rotated in the forward direction.
Axis motion is suspended at point Z_1 until the M-tool spindle and the Z-axis come within the droop.

Q_4 : The M-tool spindle is synchronized with the Z-axis at coordinates (Z_0, -K) while being rotated in the reverse direction.
Then, the C-axis is clamped.
When the return point of cutting is set at the cycle start point, the axes are positioned at point Z_0 at the rapid feedrate.

[Supplement]

1) During the execution of steps Q_3 and Q_4, the M-tool spindle override and feed axis override are set at 100%.
2) When slide hold is designated during the execution of the steps Q_3 and Q_4, the axes are moved as follows.
   Step Q_3
   The M-tool spindle and the feed axes are stopped at that position.
   The M-tool spindle is then returned to the cutting start point in synchronized cutting feed while it rotates in the direction opposite to that it had been rotating in, then enters the slide hold state.
   Step Q_4
   The axes are returned to the cutting start point in synchronized feed, and the slide hold state is established.
3) The dry run function is not effective during the execution of steps Q_3 and Q_4.
4) The M-tool spindle is stopped when the synchronized tapping cycle is complete.

8-15-2. Reverse tapping cycle (G179)

Axis motion is the same as in the forward tapping cycle but the M-tool spindle rotates in the reverse direction in steps Q_2, Q_3, and Q_4.

8-16. Repeat Function

When cutting equally spaced holes, the use of the repeat function simplifies programming

```
G183  X40  Z80  C0  I46  D10  E1  F40  Q6
G180
```

Specify the number of holes to be drilled.

LE33013R0300900460001

The repeat function allows repeated designation in two blocks.
Note that the repeat function is effective for G178, G179 and G181 through G184 and G189, G190 cycles.

- When no Q word is specified or Q_0 is specified, the control regards the designation as "Q_1".
- The fixed cycle command associated with the Q word is effective only in one block. Be sure to specify G180 in the block following such a block.

```
  G181  X100  Z150  C30  I46  E1  Q4
* |     X80         C90              |
  G180
```

* :The commands in this block cannot be given.

LE33013R0300900460002

- When a Q command is specified, positioning in intervals of 360/Q is executed automatically from the commanded C position.

8-17. Tool Relieving Command in Deep-hole Drilling Cycle for Chip Discharge.

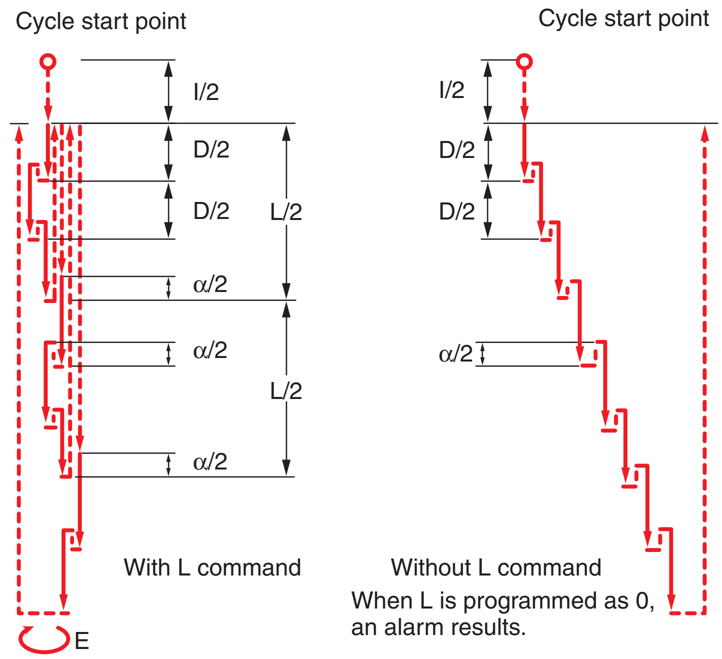
As explained before, in deep-hole drilling cycle (G183), the use of D and E codes executes step feed and breaks the chips in which the drill will otherwise tend to get entangled. In addition to this, it is possible to program drilling with discharge of chips outside the hole being drilled.

L: Infeed amount (as a diameter) for tool relieving motion
When an L** command is programmed in a deep-hole drilling cycle, peck feeding is repeated until the total infeed amount reaches "L" and then the drill returns to the cycle start point at the rapid traverse. The drill is then positioned to a point -α mm (*) away from (above) the depth of the previous cut at the rapid traverse rate; the programmed drilling cycles is repeated until the commanded X (Z) value is reached.
The value set at Pecking amount in drilling cycle of optional parameter (MULTIPLE MACHINING) is used for "α".

To execute a drilling with chip discharge command in a fixed cycle program, program a block of commands like that shown below.

```
G183  X40  Z80  C0  146  D10  E1  F40  L50
```

LE33013R0300900470001



LE33013R0300900470002

8-18. Drilling Depth Setting (Only for drilling cycles)

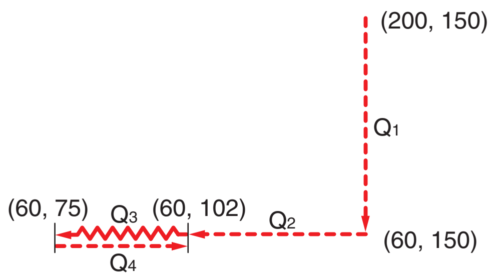
For the drilling cycles called by G178, G179, G181, G182, G183, G184, and G189, the drill hole depth may be specified by an R command (see below) from the position shifted to by I or K, instead of specifying the end point of the drilling cycles.

±R : Drilling hole depth in drilling cycleIn a drilling cycle, the drill hole depth (distance) is specified by an R command.
The R command in the X-axis direction must be specified as a diameter.
The use of an R command allows the drilling cycle to be programmed by reference to the hole depth instead of the end point of the cycle.

When using an R command in a fixed cycle program, program a block of commands like that shown below.

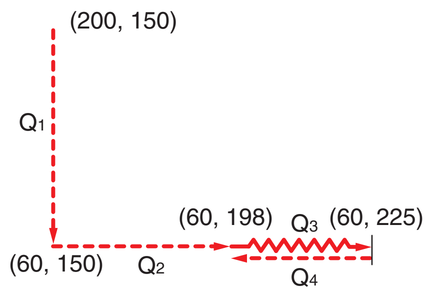
N102	G 94	X200	Z150	T0101	SB=400
N103	G181	X 60	R - 27	C0 K48	F40

LE33013R0300900480001



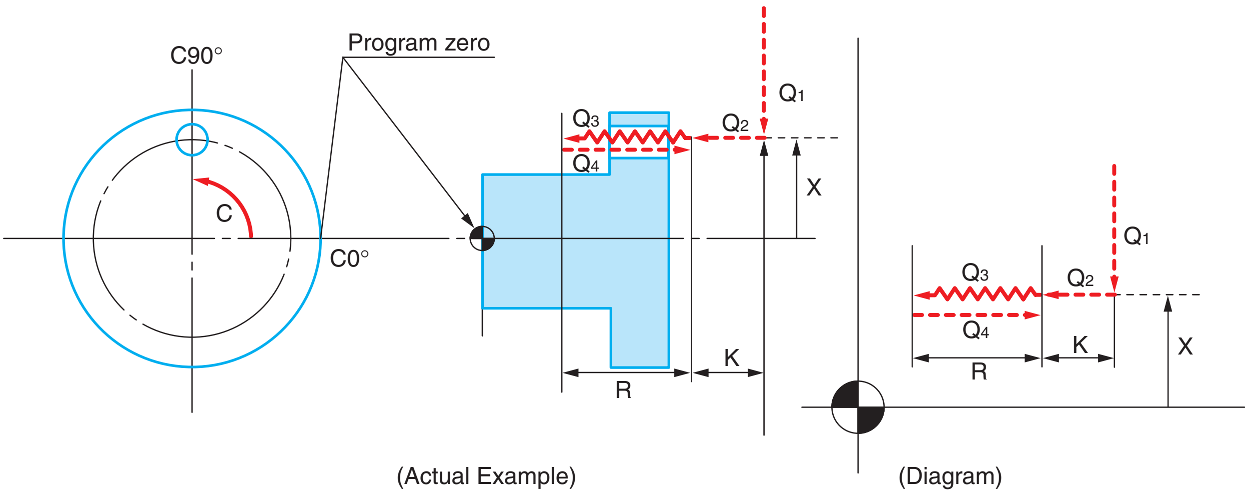
LE33013R0300900480002

The direction of drilling is determined by the plus or minus sign of the R command. If R27 were specified instead of R-27 in the program above, the direction of the drilling cycle would be as indicated below.



LE33013R0300900480003

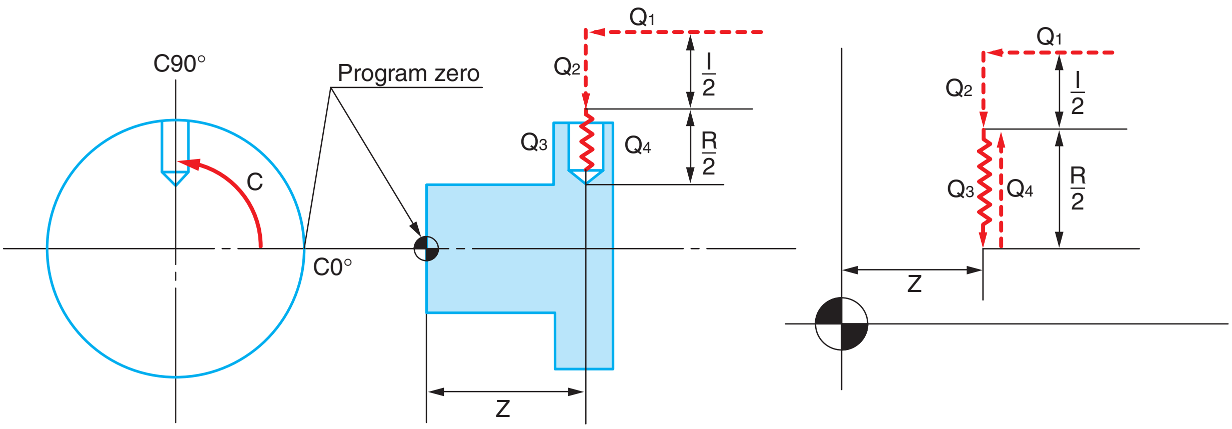
Face Machining (With K command)



(Actual Example) (Diagram)

LE33013R0300900480004

Side Machining (With I command)



(Actual Example) (Diagram)

LE33013R0300900480005

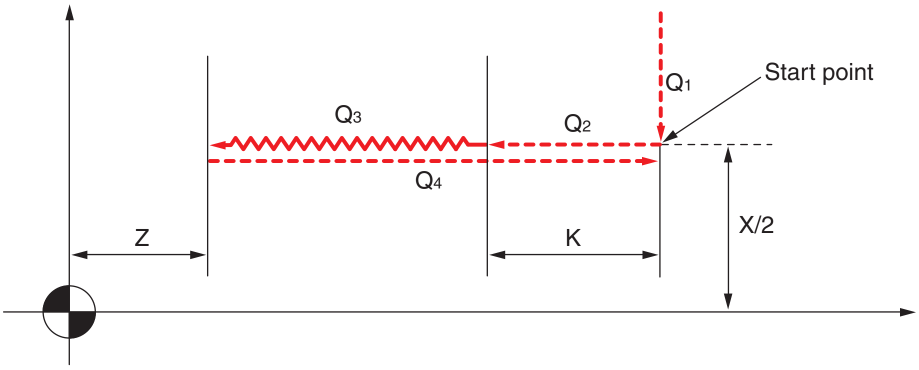
	Face Machining (With K command)	Side Machining (With I command)
Q_1	Positioning of X- and C-axis at the rapid feedrate	Positioning of Z- and C-axis at the rapid feedrate
Q_2	Positioning of Z-axis to the point defined by incremental amount "±K" from the present position at the rapid feedrate	Positioning of X-axis to the point defined by incremental amount "±I" from the present position at the rapid feedrate
Q_3	Cutting along Z-axis up to the commanded point Z	Cutting along X-axis up to the commanded point X
Q_4	Z-axis returns to the point where cutting started (Q_3) either at a specified feedrate or the rapid feedrate depending on the called fixed cycle mode.	X-axis returns to the point where cutting started (Q_3) either at a specified feedrate or the rapid feedrate depending on the called fixed cycle mode.

- For I and K commands, only a positive value is allowed. If a negative value is specified, an alarm occurs.
- The direction of axis feed is automatically determined. The axis is fed in the determined direction by the amount specified by the I or K command.
- With an R command, the direction of drilling is determined by the plus or minus sign. In the example above, the R value is negative.

8-19. Selection of Return Point

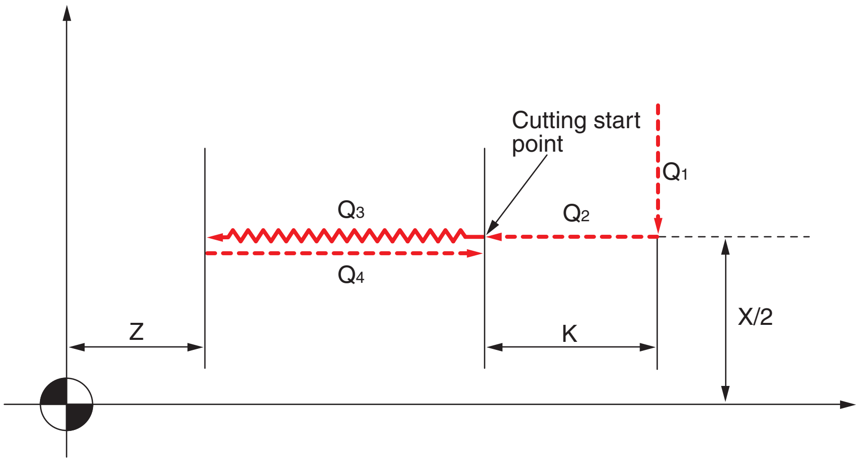
In the G178, G179, G181 through G184, G189 and G190 cycles, the return point after the completion of cutting can be selected by setting at Multi cycle return point of optional parameter (MULTIPLE MACHINING).

- When the setting at Multi cycle return point is "rapid feedrate start point":



LE33013R0300900490001

- When the setting at Multi cycle return point is "cutting start point":



LE33013R0300900490002

8-20. M-tool spindle Interlock Release Function (optional)

Usually, an attempt to rotate the M-tool spindle while the C-axis is not in the joined state causes an alarm. However, using the M-tool spindle interlock release M code in the optional operation time reduction function allows rotation of the M-tool even if the C-axis is not in the joined state.
M-tool spindle interlock M codes:

M152 : M-tool spindle interlock release cancel (interlock ON)
M153 : M-tool spindle interlock release (interlock OFF)

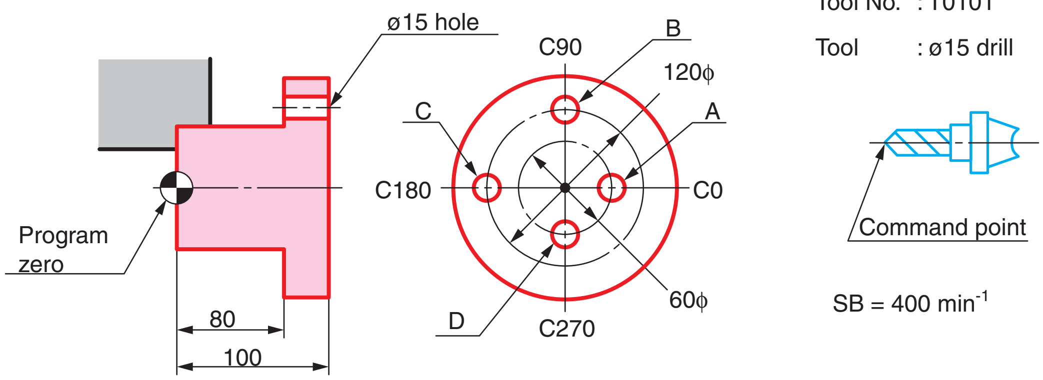
- When M153 is designated, M13 and M14 commands are effective regardless of the C-axis joining state.
- When the control is reset, M152 (interlock ON) is effective.
- When the power is turned on, M152 (interlock ON) is effective.

8-21. Other Remarks

- No incremental data can be specified during fixed cycle mode operation.
- After the programmed fixed cycle is completed, the C-axis is in the unclamped state and the M-tool is rotating (M146 and M13 modes). Specify M147 and M12, if necessary.
- If the pattern is repeated by the Q command in a fixed cycle, the Q_2 sequence is executed after the C-axis has rotated if the cycle start point is selected as the return point.

8-22. Program Examples

Example 1:



LE33013R0300900520001

When drilling the four 15 mm dia. holes shown above, program as below using G181 for the drilling cycle.

Continued from turning operation program							
N099	G00	X1000	X1000				M05
N100							M110
N101							M15
N102	G094	X200	Z150				T0101 SB=400
N103	G181	X60	Z75 (R-27)	C0	K48	F40	
N104		X120		C90			
N105				C180			
N106		X60		C270			
N107	G180						
N108	G00	X1000	Z1000				M146 M12
N109	G95						M109
N110							M02

Both X and Z axes are programmed.

The spindle (C-axis) indexes to the 0° position. After the drill is positioned at X60 at the rapid feedrate, it starts rotating in the leftward direction at 400 min^{-1}.

The drill is positioned at Z102 at the rapid feedrate.

The drill is fed to Z75 at 40 mm/min, thereby drilling hole A.

The drill is returned to Z102 at the rapid feedrate.

The spindle indexes to the 90° position. After the drill is positioned at X120 at the rapid feedrate, it drills hole B in th same manner as N103.

After the spindle indexes to the 180° position, the same drilling cycle is executed for hole C.

After the spindle indexes to the 270° position, the drill is positioned at X60 and then the same drilling cycle is executed for hole D.

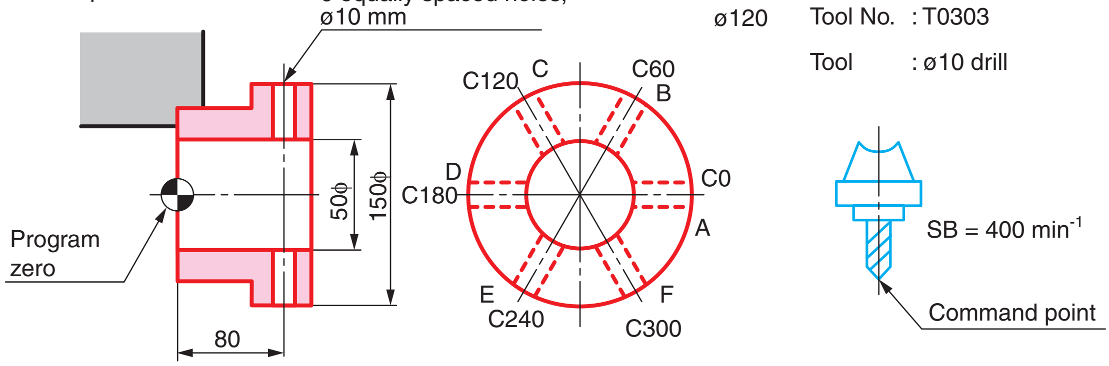
G180 cancels the fixed cycle mode.

LE33013R0300900520002

[Supplement]

- Tool rotation, and C-axis clamp and unclamp commands need not be designated in blocks N103 through N106 as they are generated automatically.
- In block N104, which calls out the drilling cycle at the second hole, program only the commands differing from those specified in the previous block N103. In blocks N105 and N106, the same programming concept applies.
- S, M and T codes must not be programmed during fixed cycle mode operation.

Example 2:



LE33013R0300900520003

When drilling the six equally spaced 10 mm dia. holes shown above, program as below using G183 for the deep hole drilling cycle.

Continued from turning operation program									
N099	G00	X1000	Z1000						M05
N100									M110
N101									M15
N102	G94	X200	Z100						T0303 SB=400
N103	G183	X40	Z80	C0	I46	D10	E1	F40	
N104				C60					
N105				C120					
N106				C180					
N107				C240					
N108				C300					
N109	G180								
N110	G00	X1000	Z1000						M12
N111	G95								M109
N112									M02

1. The spindle indexes to the 0 position. After the drill is positioned at X80 mm at the rapid feedrate, it starts rotating at 400 min^{-1}.
2. The drill is positioned at X154 at the rapid feedrate.
3. The drill is fed to X40 at the commanded feedrate 40 mm/min.
4. The drill is returned to X154 at the rapid feedrate. This completes drilling of hole A.

Drilling cycle at point B
Drilling cycle at point C
Drilling cycle at point D
Drilling cycle at point E
Drilling cycle at point F

LE33013R0300900520004

The deep-hole drilling cycle is executed in the peck feed mode.

D word : peck feed stroke (mm)

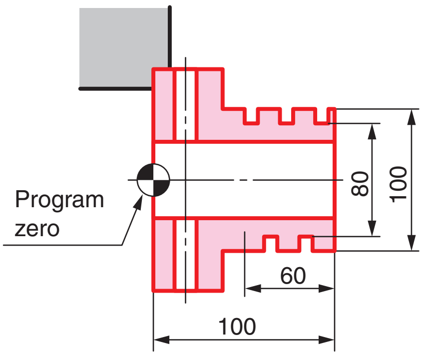
E word : duration of dwell motion (seconds)

In the program shown above, peck feed in 10 mm increments (diameter value) is repeated until the programmed depth is reached, where dwell motion is executed for one second.

[Supplement]

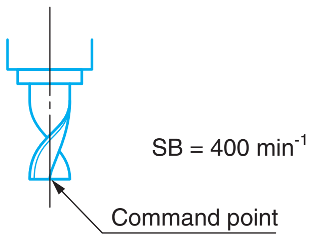
- Tool rotation, and C-axis clamp and unclamp commands in block N103 through N108 need not be designated as they are generated automatically.
- In block N102, which calls out drilling cycle on the second hole, program only the commands differing from those specified in the previous block N103. In blocks N105 and N106, the same programming concept applies.
- S, M, and T codes must not be programmed during fixed cycle mode operation.
- The value D of the peck feed stroke is specified in absolute value and is always preceded by a plus sign.
- The peck feed stroke D command is effective for deep-hole drilling cycle. The duration, E, of dwell motion is effective for deep-hole drilling, boring, and reaming cycles.

Example 3:



Tool No. : T0505

Tool : ø5 drill



LE33013R0300900520005

When cutting a thread having a width of 5 mm and 60 mm long as shown above, program as below using G185 to call out the thread cutting fixed cycle.

Continued from turning operation program

N099	G00	X1000	Z1000			M05
N100						M110
N101						M15
N102	G095	X110	Z120			T0505 SB=400
N103	G185	X95	Z60	C0	F10	SA12
N104		X90				
N105		X85				
N106		X80				
N107	G180					
N108	G00	X1000	Z1000			M12
N109	G95					M109
N110						M02

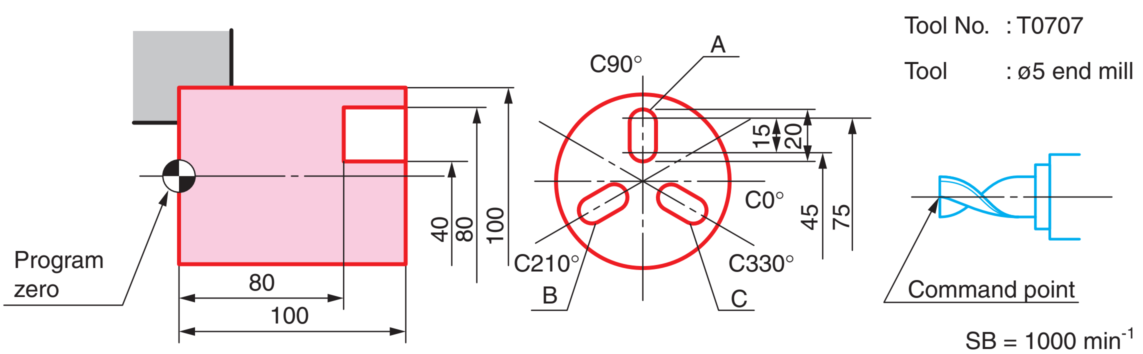
- The spindle indexes to the 0° position. After the end mill is positioned to X95 at the rapid feedrate, it starts rotating in the leftward direction at 400 min^{-1}.
- The end mill is fed to Z60 at the commanded feedrate, 10 mm/rev.
- The end mill is returned to X110 at the rapid feedrate.
- The end mill is returned to Z120 at the rapid feedrate.
- The spindle indexes to the 0° position. After the end mill is positioned at X90 at the rapid feedrate, the thread cutting cycle is executed.
- In the same way, the spindle indexes and thread cutting cycle is executed up to X85
- In the same way, the thread cutting cycle is executed at X80
- G180 cancels the fixed cycle mode.

LE33013R0300900520006

[Supplement]

- Command SA = 12 in the N103 block specifies the feedrate along the C-axis as 12 min^{-1}.
- The repeat function is not available for the thread cutting fixed cycles called by G185 through G188.
- In the thread cutting fixed cycle called for by G185 through G188, only the G95 feedrate mode is selectable. Programming G94 in these modes results in alarm. In this mode, the feedrate of the C-axis is referred to.
- In the thread cutting cycles called by G185 through G188, only constant pitch thread can be cut.
- In the G183 mode, an F code specifies thread pitch.

Example 4:



LE33013R0300900520007

When cutting a key way having a width of 5 mm and 20 mm long as shown above, program as below using G190 to call out the key way cutting fixed cycle.

Continued from turning operation program

N099	G00	X1000	Z1000							M05
N100										M110
N101										M15
N102	G94	X200	Z100							T0707 SB= 1000
N103		X75	Z120							
N104	G190	x45	Z80	C90	K15	D8	W0.2	E15	F30	M211 M213
N105				C210						
N106				C330						
N107	G180									
N108	G00	X1000	Z1000							M12
N109	G95									M109
N110										M02

1. The spindle is indexed to the 90° position. The end mill is positioned at the Z105 mm position at the rapid feedrate and the M-tool spindle starts rotating in the CCW direction at 1000 min^{-1}.
2. The end mill cuts to the Z97 mm position at 15 mm/min.
3. The end mill cuts to the X45 mm position at 30 mm/min.
4. The end mill returns to the Z105 mm position at the rapid feedrate.
5. The end mill returns to the X75 mm position at the rapid feedrate
6. The end mill cuts to a position 8 mm from the previous infeed position.
7. Steps 3) - 6) are repeated until the Z-axis reaches the 80 mm position.

Machining of groove B

Machining of groove C

LE33013R0300900520008

[Supplement]

- Tool rotation, and C-axis clamp and unclamp command in blocks N104 through N106 are unnecessary as they are generated automatically.
- In block N105, which calls out drilling cycle on the second hole, program only the commands differing from those provided in the previous block N104. In blocks N106 and N107, the same programming manner applies.
- S, M, and T codes must not be programmed during fixed cycle mode operation.

SECTION 8 LATHE AUTO-PROGRAMMING FUNCTION (LAP)

1. Overview

LAP (Lathe Auto-Programming) is a function to make full use of high-speed processing capability which characterizes the NC. With this function, the control automatically generates a tool path to produce the required part contour.
When this function is used, a program comprising the dimension data of the final contour to be finished, including rough cut conditions, is prepared as the Contour Definition Program; when it is called out with the cutting conditions specified, the control automatically generates the tool path for the rough cut cycles, and then finishes the workpiece to the programmed dimensions.
This feature permits the programmer to complete part programs by simply picking up the dimensions specified in an engineering drawing. It not only simplifies programming but also reduces programming time; it also makes the preparatory steps for programming easier, as well as the program check procedure.
Various cutting modes available with the LAP can cope with any intended type of cutting.
In addition to the features above, LAP4 (see note) is also available; with LAP4, a workpiece can be cut using the most efficient tool paths by simply entering the blank workpiece shape.

Features of LAP:

- No special programming language is needed. The same methods as used in conventional programming also apply for the LAP function.
- Programming time can be greatly reduced.
- Programming for rough cut cycles can be eliminated, and this simplifies manual calculation in programming.
- Change of cutting conditions, such as depth of cut and feedrate, is possible during a rough cut cycle.
- By entering the blank workpiece shape, unnecessary air-cutting tool paths can be eliminated to improve cutting efficiency (LAP4).
  Note: LAP4 has been developed by extending the functions of LAP3.

2. G Codes Used to Designate Cutting Mode (G80, G81, G82, G83)

There are five cutting modes available for the lathe automatic program (LAP) function:

AP Mode I : for bar turning
AP Mode II : for copy turning
AP Mode III : for thread cutting
AP Mode IV : for high-speed bar turning (LAP4 only)
AP Mode V : for bar copy turning (LAP4 only)

Note: "AP" is the abbreviation of "Auto Program".
For details on AP modes I through V and programs, refer to sections 10-1 to 10-6.
The correspondence between the cutting modes listed above and the G codes that can be used with LAP is as follows.

G85 : Used to call out bar rough turning cycle.
AP Mode I/AP Mode IV
G84 : Change of conditions for bar rough turning
G86 : Used to call out copy turning cycle.
AP Mode II/AP Mode V
G87 : Used to call out finish turning cycle.
G88 : Used to call out continuous thread cutting cycle.
AP Mode III

↔

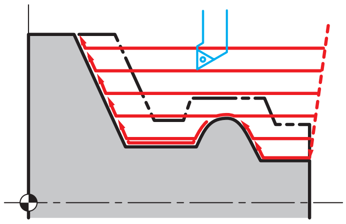
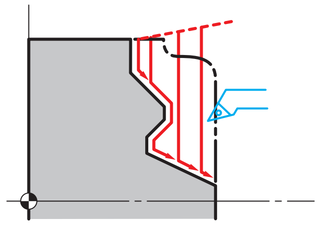
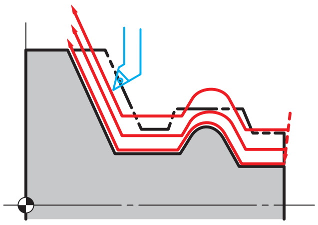
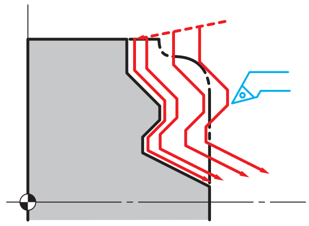
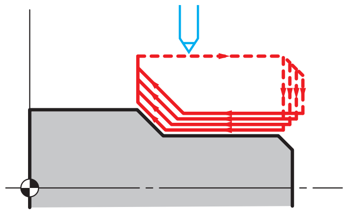
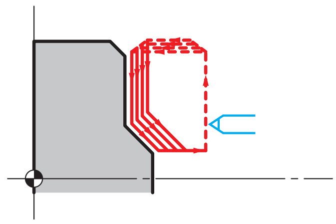
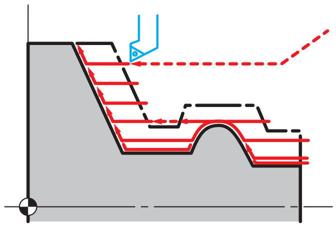
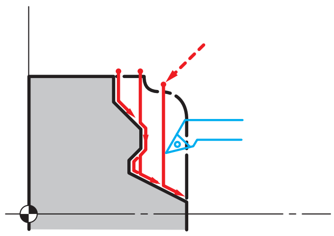
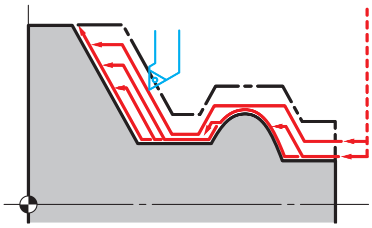
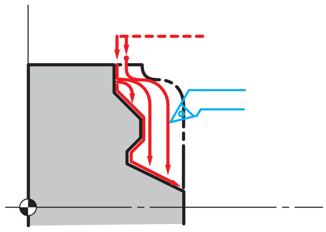
G83 : Start of blank shape definition (LAP4 only)
G81 : Start of longitudinal contour definition
G82 : Start of transverse (on end face) contour definition
G80 : End of contour definition

LE33013R0301000020001

Details of G85, G84, G86, G87 and G88 are given in sections 5. through 9. below.

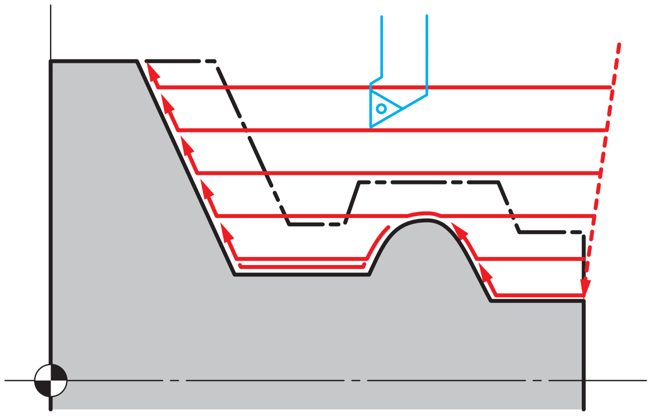
3. List of Cutting Modes

In LAP, longitudinal or transverse mode can be designated for each of the AP modes I through V. The modes that can be used with LAP are summarized in the table below.

			Longitudinal Mode	Transverse Mode
LAP 4	LAP 3	AP mode I	(1) 	(6) 
		AP mode II	(2) 	(7) 
		AP mode III	(3) 	(8) 
		AP mode IV	(4) 	(9) 
		AP mode V	(5) 	(10) 

* The numbers assigned to each mode here, (1) to (10), correspond to their descriptions on the following pages.

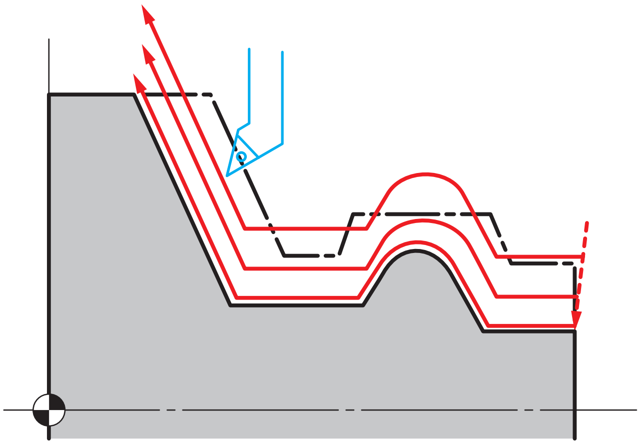
(1) AP Mode I, Longitudinal Cutting Mode (G85 + G81 + G80)



LE33013R0301000030011

Cutting is executed while shifting the cutting level by the depth of cut. A part program can be created by simply designating the finish contour data.

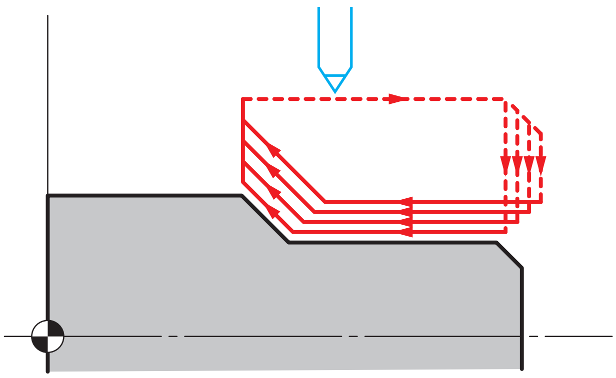
(2) AP Mode II, Longitudinal Cutting Mode (G86 + G81 + G80)



LE33013R0301000030012

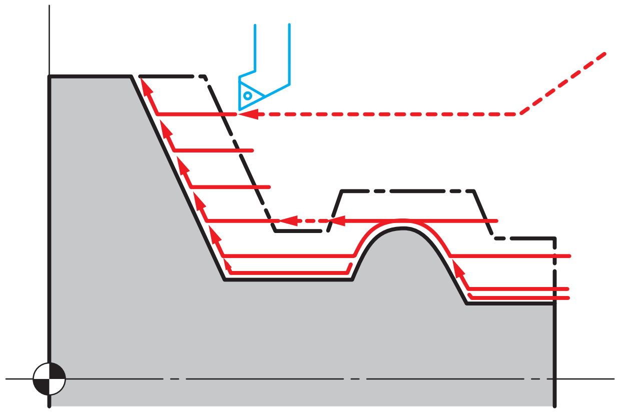
Cutting is executed along the finish contour. Cast-iron and forged workpieces can be cut at a higher speed than in AP Mode I since unnecessary tool motion is reduced.

(3) AP Mode III, Longitudinal Cutting Mode (G88 + G81 + G80)



LE33013R0301000030013

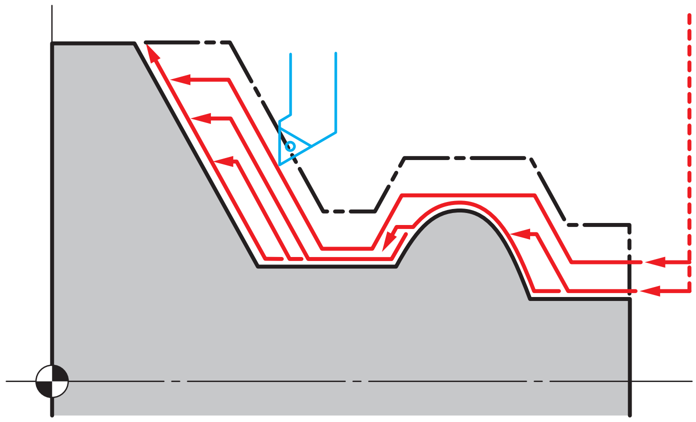
(4) AP Mode IV, Longitudinal Cutting Mode (G85 + G83 + G81 + G80) (LAP4 only)



LE33013R0301000030014

The area between the blank material shape and the finish contour is cut. The cutting tool moves at the rapid feedrate in other areas. The time required for cutting is the shortest possible.

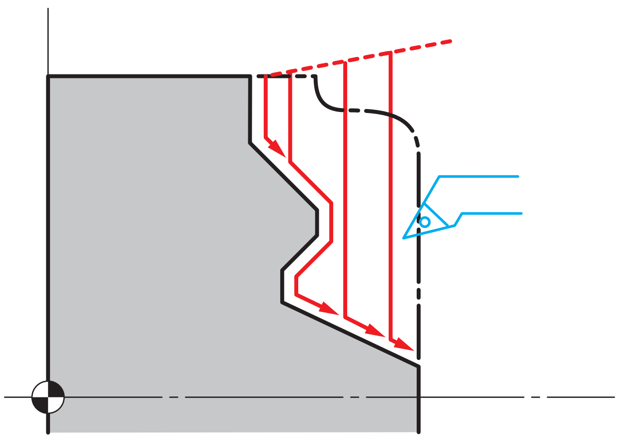
(5) AP Mode V, Longitudinal Cutting Mode (G86 + G83 + G81 + G80) (LAP4 only)



LE33013R0301000030015

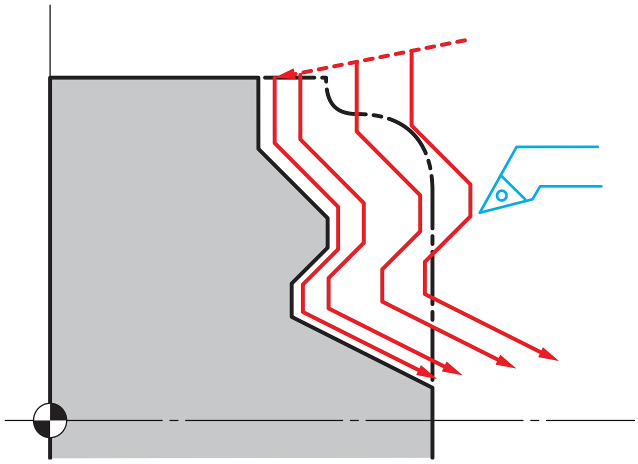
Cutting is carried out along the blank material shape. Since the number of tool impacts against the forged workpiece surface is small in this mode it is effective for untended night time operation where tool life is an important consideration.

(6) AP Mode I, Transverse Cutting Mode (G85 + G82 + G80)



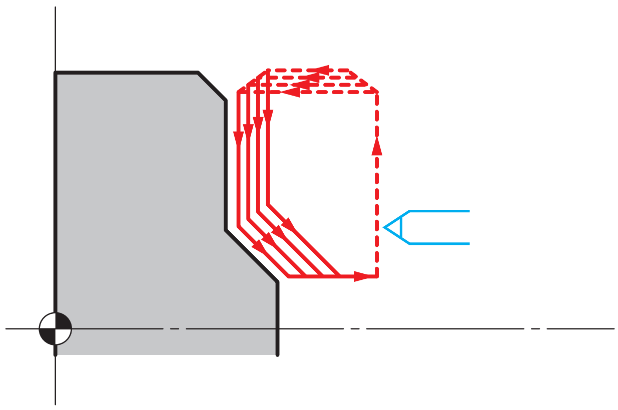
LE33013R0301000030016

(7) AP Mode II, Transverse Cutting Mode (G86 + G82 + G80)



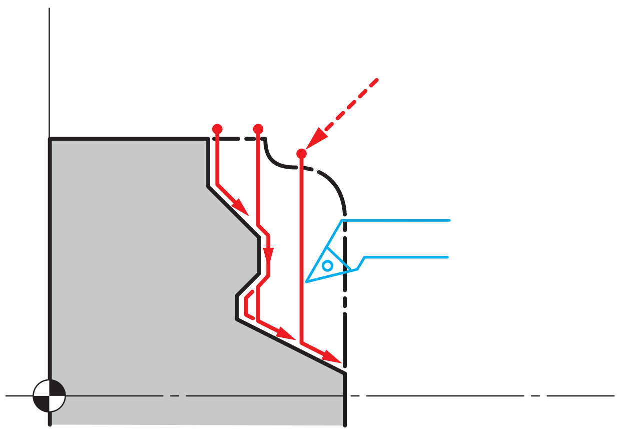
LE33013R0301000030017

(8) AP Mode III, Transverse Cutting Mode (G88 + G82 + G80)



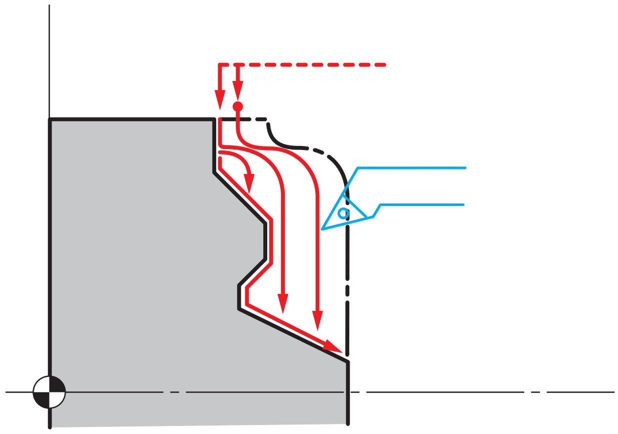
LE33013R0301000030018

(9) AP Mode IV, Transverse Cutting Mode (G85 + G83 + G82 + G80) (LAP4 only)



LE33013R0301000030019

(10) AP Mode V, Transverse Cutting Mode (G86 + G83 + G82 + G80) (LAP4 only)



LE33013R0301000030020

4. Code and Parameter Lists

The G codes, M codes, and parameters used with LAP are summarized below.

G Codes

G Code	Description
G80	End of contour definition
G81	Start of contour definition, longitudinal
G82	Start of contour definition, transverse
G83	Start of blank shape definition (LAP4 only)
G84	Change of rough turning conditions, bar turning
G85	Bar turning rough turning cycle
G86	Copy turning cycle
G87	Finish turning cycle
G88	Continuous thread cutting cycle

M Codes

M Code	Description
M32	Straight infeed along thread face (on left face) in G88
M33	Zigzag infeed in G88
M34	Straight infeed along thread face (on right face) in G88
M73	Infeed pattern 1 in G88
M74	Infeed pattern 2 in G88
M75	Infeed pattern 3 in G88
M85	No return to the cutting starting point after the completion of rough turning cycle (LAP4 only)

LAP-Related Parameters (1/2)

Parameter	Description	Default	Data Setting Range
D	Depth of cut in rough turning cycle	Alarm	D > 0
DA	Depth of cut after rough turning condition change point A	DA = D	DA > 0
DB	Depth of cut after rough turning condition change point B	DB = DA	DB > 0
FA	Feedrate after rough turning condition change point A	FA = F	FA > 0
FB	Feedrate after rough turning condition change point B	FB = FA	FB > 0
E	Feedrate in rough turning cycle along finish contour	F active at entry of LAP mode	E > 0
XA	X coordinate of rough turning condition change point A	No change of cutting conditions	$\lvert XA \rvert \leq 99999.999$

Parameter	Description	Default	Data Setting Range
XB	X coordinate of rough turning condition change point B	No change of cutting conditions at point B	$\lvert XB \rvert \le 99999.999$
ZA	Z coordinate of rough turning condition change point A	No change of cutting conditions	$\lvert ZA \rvert \le 99999.999$
ZB	Z coordinate of rough turning condition change point B	No change of cutting conditions at point B	$\lvert ZB \rvert \le 99999.999$

LAP-Related Parameters (2/2)

Parameter	Description	Default	Data Setting Range
U	Stock removal in X-axis direction for finish turning cycle	U = 0	$U \ge 0$
W	Stock removal in Z-axis direction for finish turning cycle	W = 0	$W \ge 0$
H	Thread height in G88 thread cutting cycle	Alarm	$H > 0$
B	Tip point angle of thread cutting tool in G88	B = 0	$0 \le B < 180°$

NC Parameters

Parameter	Contents	Initial Value
Optional parameter (OTHER FUNCTION 1)	Relieving amount in LAP-bar turning (0.001 mm)	100
	LAP clearance (0.001 mm) (LAP4 only)	2000
	Infeed pattern in thread cutting cycle	Infeed pattern 3

[Supplement]

- The following words must be specified as incremental values.
  D, DA, DB, U, W and H
- D, DA, DB, XA, XB, U and H words must be commanded as diameter values.
- In thread cutting cycles using the M73 pattern, "H - U" must be greater than or equal to D:
  $H - U \ge D$
- In the M74 and M75 patterns, it must be positive:
  $H - U \ge 0$
- When more than one alphabetic character is used in succession, the control interprets the expression as a variable. Therefore, it is necessary to use a delimiter for extended address characters:
  DA =, DB =, FA =, FB =, XA =, XB =, ZA = and ZB =

5. Bar Turning Cycle (G85)

[Program format]

N0103 G85 NAT01 ______ D F U W G84

Sequence number

Change of rough turning conditions

Stock removal in finish turning cycle, Z component

Stock removal in finish turning cycle, X component

Feedrate in rough turning cycle

Depth of cut in rough turning cycle

Enter either tab or space code.

Sequence name in the first block of contour defining blocks

G code calling out bar turning cycle
To be designated right after sequence number (name).

LE33013R0301000050001

[Function]
With the commands above, the control starts searching for the contour definition program beginning with the sequence name NAT01. After assigning the parameter data of D, F, U, W and G84 to NAT01, the control starts the bar turning cycle.

[Supplement]

- Do not designate an S, T, or M code in the G85 block.
- The D word is used to specify depth of cut in the rough turning cycle. When a G84 command indicating change of cutting conditions is designated, the D word is effective up to the point where the change is made, XA and ZA.
  A D word must be always be designated in the G85 block, with a value greater than "0". Illegal designation will cause an alarm.
- The F word is used to specify the feedrate in a rough turning cycle. When a G84 command indicating change of cutting conditions is designated, the F word is effective up to the point where the change is made, XA and ZA.
- If no F word is designated in the G85 block, the feedrate which was effective before the execution of the G85 block is effective.
  The F word must be positive. If not, an alarm occurs.
- When a U and/or W word is not designated, U and/or W is assumed to be "0".
  U and W words must be positive or zero. If not, an alarm occurs.

6. Change of Cutting Conditions in Bar Turning Cycle (G84)

[Program format]

```
N •••    G85    N •••
$        G84    XA = (ZA =)    DA =    FA =
$               XB = (ZB =)    DB =    FB =
```

Specifies the point where cutting conditions are changed.

Feedrate after cutting condition change point

Depth of cut after cutting condition change

Indicates that the commands are continuous. (Must be specified at the beginning of the block.)

LE33013R0301000060001

[Function]
These commands allow the cutting conditions to be changed from the desired point(s) during a rough turning cycle. If no change in cutting conditions is necessary, do not use them.
[Details]
These commands must be programmed in the block containing G85, which calls out the bar turning cycle. Since the number of characters in one line will be very large if these commands are specified in the same line, they are written in different lines preceded by "$", which indicates that the commands in these lines belong to the same block.

- G84 and commands following it must be designated after "N......G85 N......".
- For OD turning, the coordinate values of "LAP starting point", "rough turning condition change point A" and "rough turning condition change point B" must be designated so that they become smaller in this order. For ID turning, they must be designated so that they become larger in this order.
- If both cutting condition change points A and B exist when infeed D is executed, the depth of cut and the feedrate designated for XB = (ZB =) are effective.
- If the present position is before XA but the tool path will go beyond XA when a cutting cycle is performed with the depth of cut D from the present position, the cycle is performed with D designated; DA is designated when the present position is on XA.
- In longitudinal cutting, ZA = and ZB = commands must not be designated. In transverse cutting, XA = and XB = commands must not be designated, either.

7. Copy Turning Cycle (G86)

[Program format]

NO123 G86 NAT02 D F U W

- NO123: Sequence number
- G86: G code calling out copy turning cycle
  To be designated right after sequence number (name).
- NAT02: Sequence name in the first block of contour defining blocks
- (blank): Enter either tab or space code.
- D: Depth of cut
- F: Feedrate
- U: Stock removal in finish turning cycle, X component
- W: Stock removal in finish turning cycle, Z component

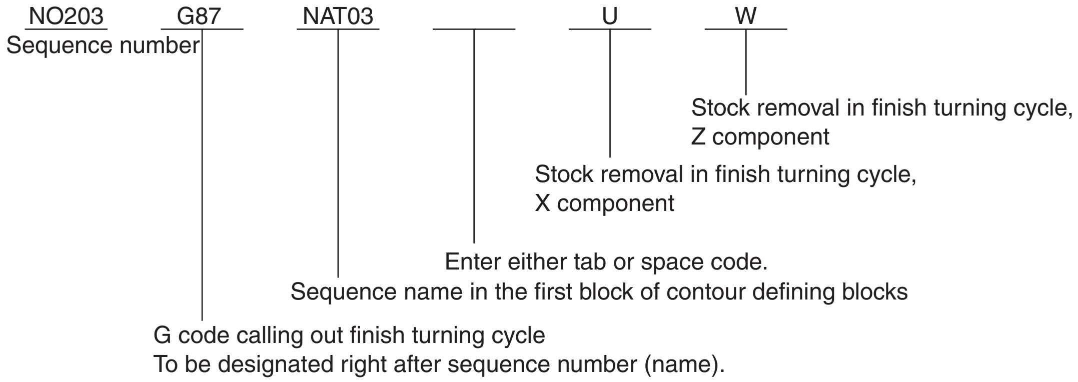
LE33013R0301000070001

[Function]
With the commands above, the control starts searching for the contour definition program beginning with the sequence name NAT02. After assigning parameter data of D, F, U and W to NAT02, the control starts the copy turning cycle.
[Details]

- Do not designate an S, T, or M code in the G86 block.
- The D word is used to specify depth of cut in each cycle and must be designated in the G86 block without fail.
  The D word value must be positive. If not, an alarm results.
- The F word specifies the feedrate for the blocks until an E word is designated in the contour definition program.
  If no F word is designated in the G86 block, the feedrate which was effective before the execution of the G86 block is effective.
  The F word must be positive. If not, an alarm occurs.
- When no U and/or W word is designated, U and/or W is assumed to be "0".
- U and W words must be positive or zero. If not, an alarm occurs.

8. Finish Turning Cycle (G87)

[Program format]



LE33013R0301000080001

[Function]
With the commands above, the control starts searching for the contour definition program beginning with the sequence name NAT03. After assigning the parameter data of U and W to NAT03, the control starts the finish turning cycle.
[Details]

- Do not designate an S, T, or M code in the G87 block.
- The feedrate designated in the contour definition program is the effective one.
  If no F word is designated in the contour definition program, the feedrate which was effective before this block becomes effective.
- When no U and/or W word is designated, U and/or W are/is assumed to be "0".
  U and W words must be positive or zero. If not, an alarm occurs.

9. Continuous Thread Cutting Cycle (G88)

[Program format]

N0143 G88 NAT04 ___ D H B U W M32 (M33, M34) M73 (M74, M75)

N0143: Sequence number

G88: G code calling for continuous thread cutting cycle
To be designated right after sequence number (name).

NAT04: Sequence name in the first block of contour defining blocks

___: Enter either tab or space code.

D: Depth of cut

H: Height of thread to be cut

B: Tip point angle of thread cutting tool

U: Stock removal in finishing cycle, X component

W: Stock removal in finishing cycle, Z component

M32 (M33, M34): Cutting mode

M73 (M74, M75): Cutting mode

LE33013R0301000090001

[Function]
With the commands above, the control starts searching for the contour definition program beginning with sequence name NAT04. After assigning the parameter data of D, H, B, U, W, M32 (M33, M34) and M73 (M74, M75) to NAT04, the control starts the thread cycle.
[Details]

- Do not designate an S, T, or M code in the G88 block.

- The D word is used to specify the depth of cut in the first thread cutting cycle. The depth of cut in each thread cutting cycle after that varies according to the selected infeed pattern (M73, M74, M75).
  A D word must be designated in the G85 block without fail with a value greater than 0. Illegal designation will cause an alarm.

- The H word must have a positive value and must be specified in the G88 block without fail. If the numerical data of the D word is not positive, or if it is omitted, an alarm occurs.
  The H value must be greater than the U and/or W value. If not, an alarm occurs.

- The B word specifying the tip point angle of thread cutting tool must have a value within the following range:
  $0 \leq B \leq 180°$
  When no B word is designated, it is assumed to be "0".

- M32, M33, and M34 are used to select the cutting mode.
  M32 : Straight infeed along thread face (on left face)
  M33 : Zigzag in feed in G88
  M34 : Straight infeed along thread (on right side)

  When none of M32, M33 and M34 is designated, the control selects M32.

- M73, M74 and M75 are used to select infeed pattern. When no such M code is present, the M73 pattern is automatically selected.

- In the M73 pattern, "H - U" must be greater than or equal to "D".
  H - U ≥ 0
  If not, an alarm occurs.

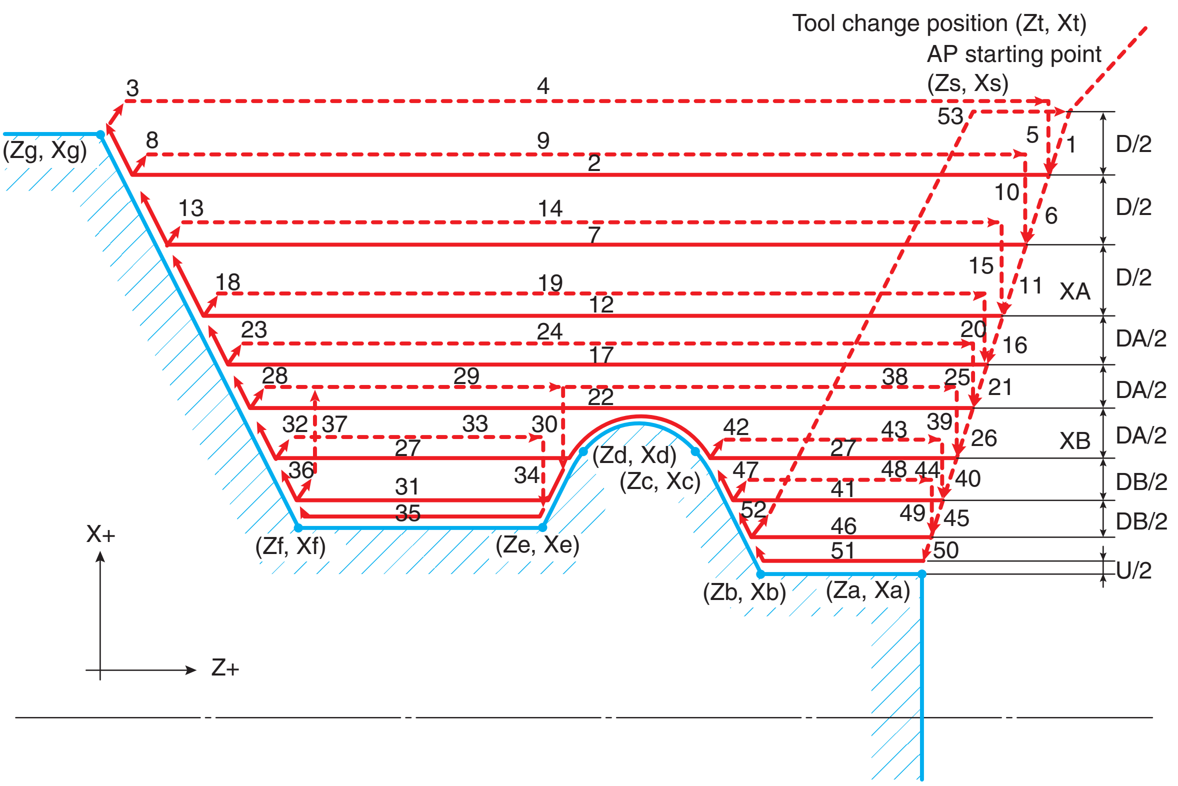
10. AP Modes

AP modes I through V are explained here. You are advised to refer also to the "precautions" in section 10-5-5.

10-1. AP Mode I (Bar Turning)

[Function]
In AP mode I, the area surrounded by the AP starting points and the contour defined by the contour definition program starting with G81 (or G82) is cut while shifting the cutting level by the depth of cut designated by D.
This mode is effective for normal turning, for example bar turning.
Since both rough turning and finish turning can be executed using the same contour definition program when stock removal is designated using the U (X-axis direction) or W (Z-axis direction) command, the program length can be reduced.

10-1-1. Tool Path and Program - Longitudinal Cutting



LE33013R0301000120001

Contour definition

NAT01	G81								······	Start of longitudinal contour definition G code
N0001	G00	Xa	Za							Finish contour definition blocks
N0002	G01	Xb	Zb			Fb	Sb	Eb		
N0003		Xc	Zc			:	:	:		
N0004	G03	Xd	Zd	Id	Kd	Fd	Sd	Ed		
N0005	G01	Xe	Ze			Fe	Se	Ee		
N0006		Xf	Zf			:	:	:		
N0007		Xg	Zg			Fg	Sg	Eg		
N0008	G80									

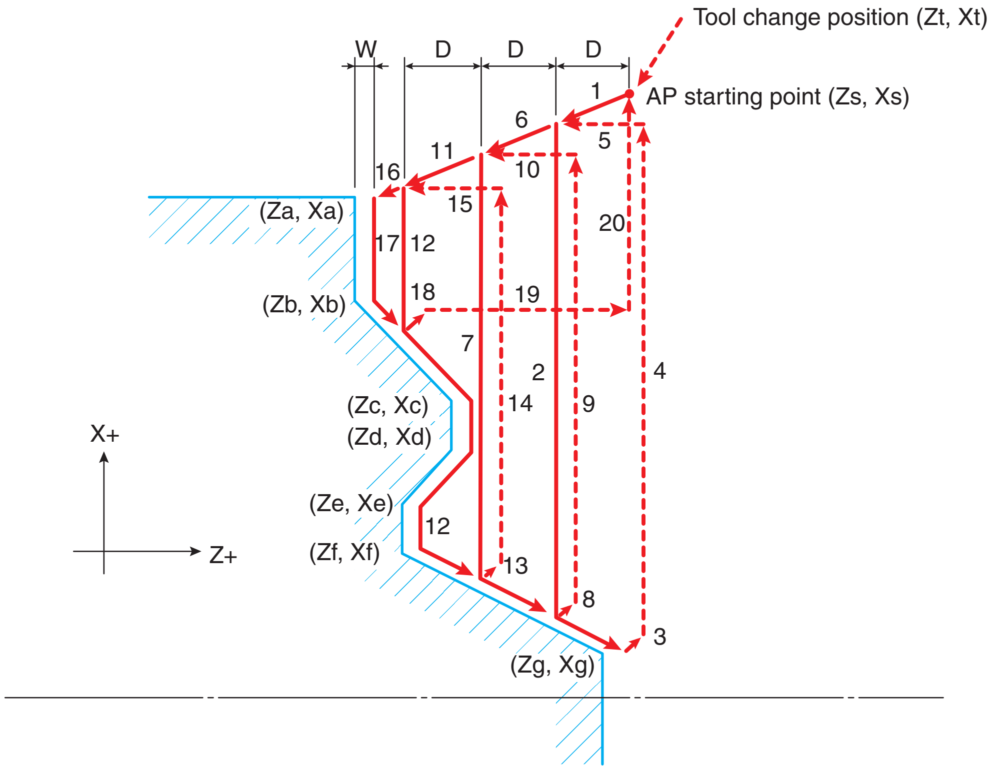
Rough Turning Cycle									······	End of contour definition G code
N0101	G00	Xt	Zt							Tool change position
N0102		Xs	Zs			STM				Starting point of AP, S, T, and M for rough turning cycle
N0103	G85	NAT01		D		U	W	M85	···	Calls for rough turning cycle
$	G84	XA =	DA =	FA =	F					Continued line: Cutting condition change point XA
$		XB =	DB =	FB =						Continued line: Cutting condition change point XB

Finish Turning Cycle

N0201	G00	Xt	Zt			Tool change position
N0202					STM	S, T, and M for finish turning cycle
N0203	G87	NAT01				Calls for finish turning cycle

LE33013R0301000120002

10-1-2. Tool Path and Program - Transverse Cutting



LE33013R0301000130001

Contour definition

NAT01	G82							Start of transverse contour definition G code
N0011	G01	Xa	Za					
N0012		Xb	Zb		Fb	Sb	Eb	Finish contour definition blocks
N0013		Xc	Zc		:	:	:	
N0014		Xd	Zd		Fd	Sd	Ed	
N0015		Xe	Ze		Fe	Se	Ee	
N0016		Xf	Zf		:	:	:	
N0017		Xg	Zg		Fg	Sg	Eg	
N0018	G80							End of contour definition G code

Rough Turning Cycle

		Xt	Zt					Tool change position
N0111	G00	Xs	Zs		STM			Starting point of AP, S, T, and M for rough turning cycle
N0112	G85	NAT10	D	F	U	W		Calls for rough turning cycle
N0113	G84	ZA =	DA =		FA =		M85	Continued line: Cutting condition change point ZA
$		ZB =	DB =		FB =			Continued line: Cutting condition change point ZB
$								

Finish Turning Cycle

N0211	G00	Xt	Zt		Tool change position
N0212				STM	S, T, and M for finish turning cycle
N0213	G87	NAT10			Calls for finish turning cycle

LE33013R0301000130002

10-1-3. Outline of Bar Turning Cycle

Rough turning cycle in the longitudinal direction (example A)

(1) The commands in block N0101 position the tool at the tool change point.

(2) With the commands in block N0102, the S, T, and M commands for the rough turning cycle are selected, then the axes are positioned at the LAP starting point.
When no S, T, or M command is designated in this block, those selected in the previous block(s) are effective.

(3) The NAT01 command in block N0103 causes the control to search for the program assigned the program name NAT01. A rough turning cycle in the bar turning mode is performed with this program.
The cutting conditions for the rough turning cycle are also specified in the same block.

D : Depth of cut
F : Feedrate
U : X component of stock removal in finish turning cycle
W : Z component of stock removal in finish turning cycle

To change the cutting conditions during the rough turning cycle, designate the following commands with G84.

XA : X coordinate of cutting condition change point A
DA : Depth of cut after point A
FA : Feedrate after point A

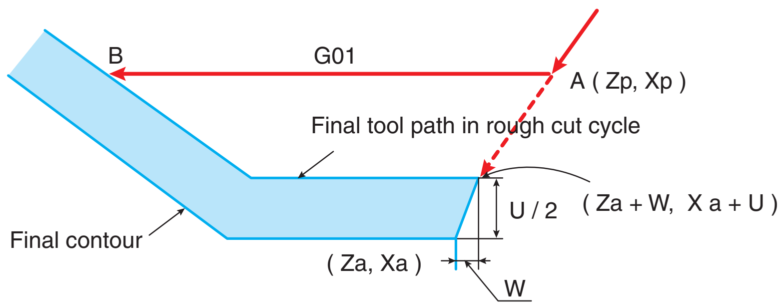
To change the cutting conditions again, designate the following commands.

XB : X coordinate of cutting condition change point B
DB : Depth of cut after point B
FB : Feedrate after point B

- Cutting condition change points must be programmed in the block containing G85. For clear programming, commands relating to such points are programmed in different lines, each line preceded by the $ character which indicates that the line is a continuation of the previous one.
- When no F word is designated in this block, the feedrate commanded last is effective.
- The point data of the cutting condition change points must become smaller in the following order: AP starting point, XA, then XB, when performing OD turning. For ID turning, they must become larger in this same order.

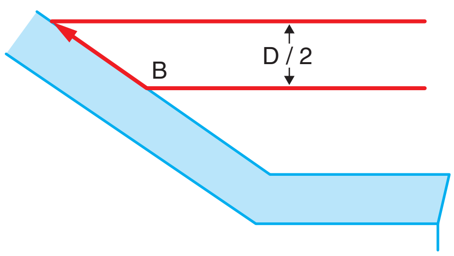
(4) Upon reaching the commands in block N0001, the control calculates the intersection point of two straight lines: the line parallel to the Z-axis running at "Xs-D/2" and the one passing through the two points (Xs, Zs) and (Xa + U, Za + W). Then, the axes are positioned at the calculated point A (Xp, Zp).
Positioning is performed at the rapid feedrate when G00 is designated in the first block of the contour definition blocks, and it is performed at a cutting feedrate when G01 is designated in the first block of the contour definition blocks.
Select the AP starting point (Xs, Zs) with respect to the coordinated point (Xa, Za) to meet the following requirements:
Xs < Xa for ID cutting
Xs > Xa for OD cutting
If the finish allowance U is made so large that "Xa + U" falls outside "Xs" with respect to the workpiece, an alarm results.

(5) Cutting is performed in the G01 mode up to point B where the straight line parallel to the Z-axis and passing through point A intersects the final contour of the rough turning cycle. The feedrate in this cutting cycle is the one selected by the F word when the rough turning cycle was called out.



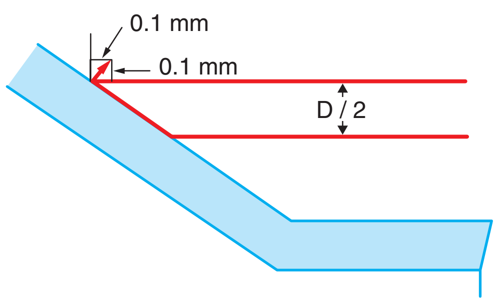
LE33013R0301000140001

(6) After point B has been reached, the final contour of the rough turning cycle is cut up to the point whose X coordinate is Xb + D. If G80, indicating the end of contour definition, is encountered before this point is reached, the final rough turning contour is cut up to the point specified in the block preceding the G80 block.
The feedrate in this cut is as specified by E, which is designated in a contour definition program. If no E word is designated in the corresponding contour definition program, the one designated last becomes effective. When an E word has not been specified, the feedrate specified when calling out the rough turning cycle becomes active.



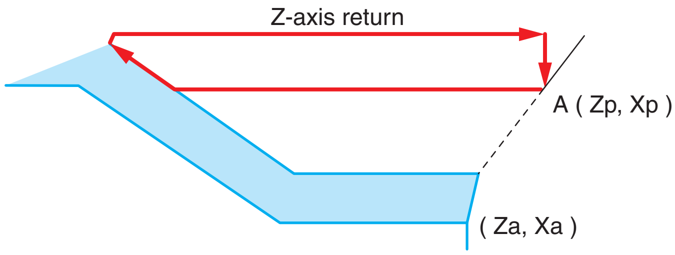
LE33013R0301000140002

(7) After the completion of the cutting explained in (6), the cutting tool is relieved from the workpiece in the direction opposite the infeed direction along the X-axis, and toward Zs along the Z-axis, by 0.1 mm on each axis (diameter value in the case of the X-axis).
The relief amount is set at Relieving amount in LAP-BAR turning of optional parameter (OTHER FUNCTION 1) in units of μ.



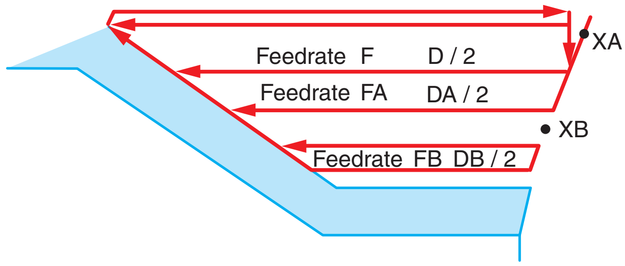
LE33013R0301000140003

(8) This completes the final rough turning cycle. The Z-axis returns to Zp as determined in step (4) at the rapid feedrate and then the X axis returns to Xp.



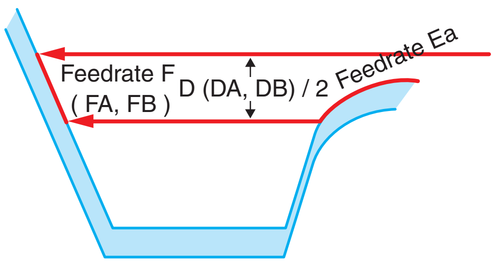
LE33013R0301000140004

(9) Steps (4) through (8) are repeated up to the cutting condition change point. After that point, cutting is continued with the depth of cut (D) and feedrate (F) changed.



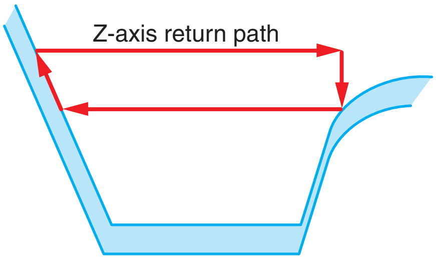
LE33013R0301000140005

(10) If the cutting in step (6) is along a descending slope, and the contour to be cut is below the cutting point (Xp), first the contour is cut until the programmed depth of cut is reached and then cutting is performed parallel to the Z-axis in the G01 mode up to the point where this path parallel to the Z-axis intersects the final rough turning contour. Cutting along the parallel line is performed at the feedrate specified by an F word (FA/FB).



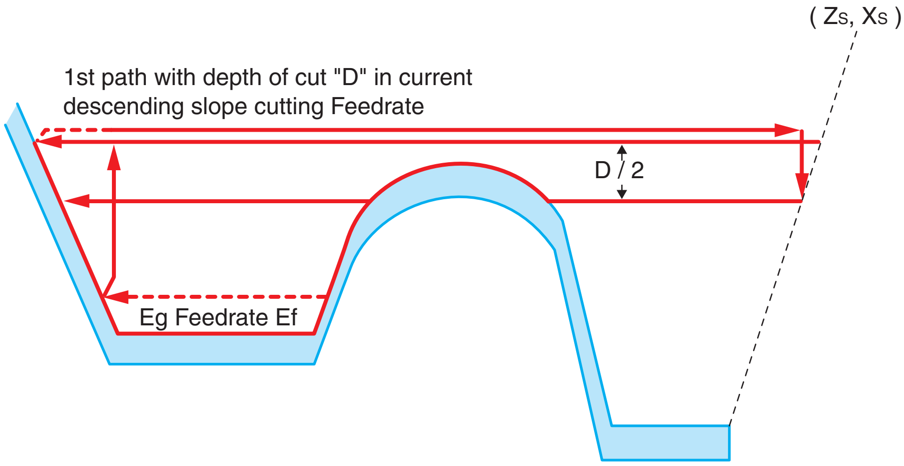
LE33013R0301000140006

(11) Subsequently, steps (6) and (7) are repeated. The Z-axis then returns to the point where cutting along the X-axis was started in the G01 mode in step (10). After the completion of Z-axis positioning, the X-axis is positioned at the point where the previous cutting cycle was started.



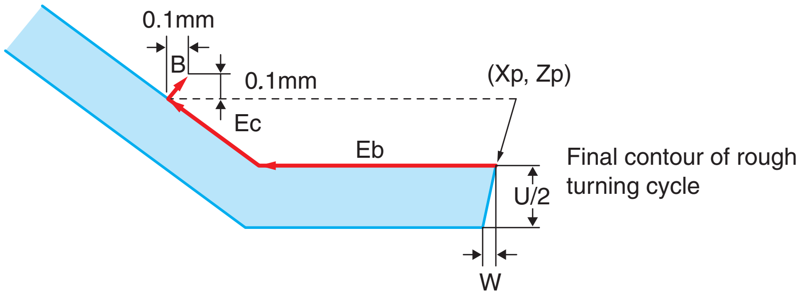
LE33013R0301000140007

(12) Steps (10) and (11) are repeated until the most recessed section along the X-axis has been cut. After that, both the X- and Z-axes retract by 0.1 mm (radius value for the X-axis), and the X-axis is positioned at the coordinate value for "first cutting level D along the descending slope + 0.2 mm". The Z-axis returns to the point which has the same coordinate value as the starting point D of the cutting cycle of the descending slope + 0.2 mm". The Z-axis returns to the point which has the same coordinate value as the starting point of the cutting cycle of the descending slope with depth of cut D. The X-axis is then positioned at that point.



LE33013R0301000140008

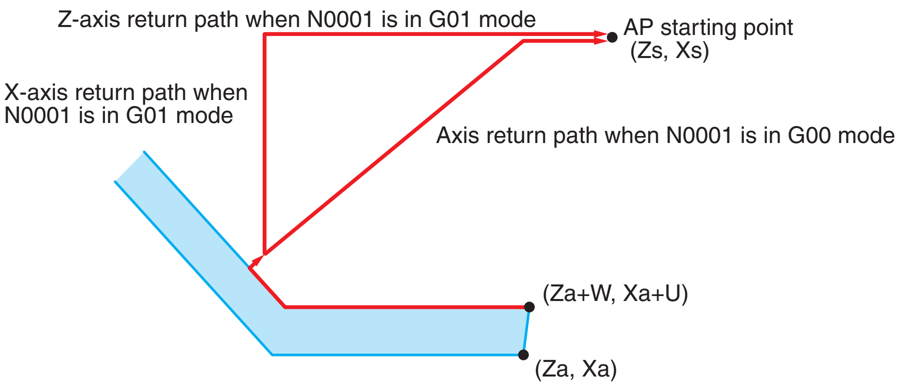
(13) The steps described above are repeated until the X-axis reaches the level where a tool path is generated below the "Xa + U" level. When this level is reached, the final rough turning is carried out along the contour up to point B.
The feedrate for cutting along the final rough cutting contour is the one specified by the E word.



LE33013R0301000140009

After the completion of the final rough turning step, the X- and Z-axes are relieved by 0.1 mm (diameter value for the X-axis). The relief amount is set at Relieving amount in LAP-BAR turning of optional parameter (OTHER FUNCTION 1).

(14) On completion of step (13), the axes return to the AP starting point (Xs, Zs).
There are two patterns of axis return motion:
The two axes return to the AP starting point simultaneously when G00 is designated in the first block of the contour definition program (the block following the one containing either G81 or G82).
When G01 is designated in the block indicated above, positioning on the X-axis is done first, then the Z-axis returns to the AP starting point.



LE33013R0301000140010

This completes the rough turning cycle.

Finish bar turning cycle - longitudinal cutting (example A)

(1) The commands in block N0201 position the axes at the tool change position.

(2) With the commands in block N0202, the S, T, and M commands for the finish turning cycle are selected.

(3) The NAT01 command in block N0203 causes the control to search for the program assigned the program name NAT01. The finish bar turning cycle is performed with this program.

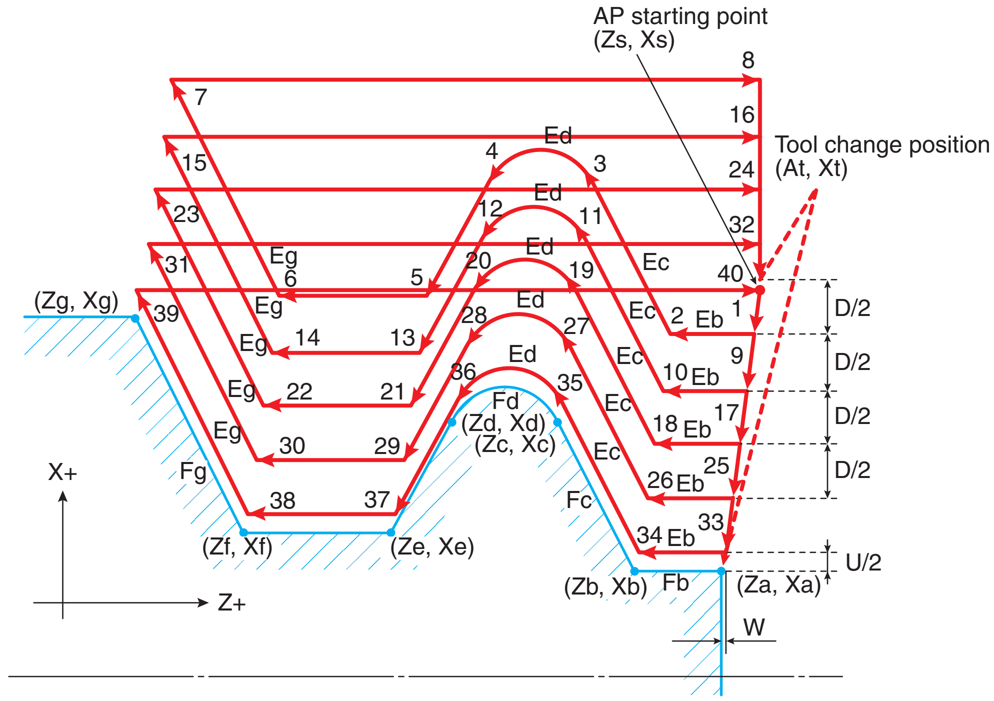
(4) The finish turning cycle is performed on the basis of the data designated in the contour definition program under the cutting conditions specified for the finish turning cycle.

(5) After the finish turning cycle is completed, the commands in the block following N0203 are executed.

10-2. AP Mode II (Copy Turning)

[Function]
In AP Mode II, the finish contour designated by the contour definition program is shifted in a parallel manner up to the AP starting point. Cutting is executed along the shifted finish contour while shifting the cutting level in increments of depth of cut D.
When this mode is used for cutting a workpiece with a constant cutting depth, for example cast iron or forged workpieces, the cutting speed can be higher than in AP Mode I.

10-2-1. Tool Path and Program - Longitudinal Cutting



LE33013R0301000160001

Contour Definition

NAT20	G81	....								Start of longitudinal contour definition G code
N0021	G01	Xa	Za							
N0022		Xb	Zb			Fb	Sb	Eb		
N0023		Xc	Zc			:	:	:		
N0024	G03	Xd	Zd	Id	Kd	Fd	Sd	Ed		Finish contour definition blocks
N0025	G01	Xe	Ze			Fe	Se	Ee		
N0026		Xf	Zf			:	:	:		
N0027		Xg	Zg			Fg	Sg	Eg		
N0028	G80	....								End of contour definition G code

Rough Turning Cycle

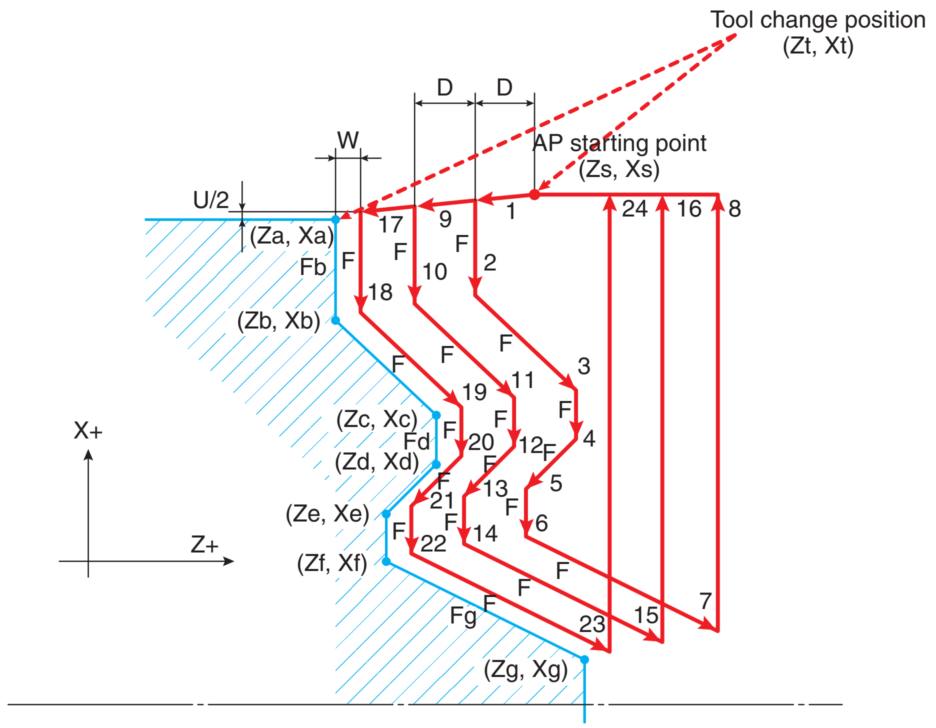
N0121	G00	Xt	Zt							Tool change position
N0122		Xs	Zs					STM		Starting point of AP, S, T, and M for rough turning cycle
N0123	G86	NAT20		D	F	U	W	M85···		Calls for rough turning cycle

Finish Turning Cycle

N0221	G00	Xt	Zt	....						Tool change position
N0222						STM				S, T, and M for finish turning cycle
N0223	G87	NAT20		....						Calls for finish turning cycle

LE33013R0301000160002

10-2-2. Tool Path and Program - Transverse Cutting



LE33013R0301000170001

Contour Definition

NAT30	G82	....							Start of transverse contour definition G code
N0031	G01	Xa	Za						
N0032		Xb	Zb	Fb	Sb	Eb			Finish contour definition blocks
N0033		Xc	Zc	:	:	:			
N0034		Xd	Zd	Fd	Sd	Ed			
N0035		Xe	Ze	Fe	Se	Ee			
N0036		Xf	Zf	:	:	:			
N0037		Xg	Zg	Fg	Sg	Eg			
N0038	G80	....							End of contour definition G code
Rough Turning Cycle									
N0131	G00	Xt	Zt						Tool change position
N0132		Xs	Zs			STM			Starting point of AP, S, T, and M for rough turning cycle
N0133	G86	NAT30		D	F	U	W	M85···	Calls for rough turning cycle
Finish Turning Cycle									
N0231	G00	Xt	Zt						Tool change position
N0232							STM		S, T, and M for finish turning cycle
N0233	G87	NAT30	....						Calls for finish turning cycle

LE33013R0301000170002

10-2-3. Outline of Copy Turning Cycle

Rough turning cycle in the longitudinal direction (example A)

(1) The commands in block N0121 position the axes at the tool change position.

(2) With the commands in block N0122, S, T, and M commands for the rough turning cycle are selected, then the axes are positioned at the AP starting point.
When no S, T, or M command is specified in this block, those selected in the preceding block(s) are effective.

(3) The NAT20 command in block N0103 causes the control to search for the program assigned the program name NAT20. A rough turning cycle in the copy turning mode is performed with this program.
The cutting conditions for the rough turning cycle are also specified in the same block.

D : Depth of cut
U : X component of stock removal in finish turning cycle
W : Z component of stock removal in finish turning cycle

Also program an F word if required. When no F word is designated in the contour definition program, the feedrate commanded last becomes effective.

(4) Upon reaching the commands in block N0201 in the contour definition program, the control calculates the intersection point of two straight lines: the line parallel to the Z-axis running at "Xs-D/2" and the one passing through the two points (Xs, Zs) and (Xa + U, Za + W). Then the axes are positioned at the calculated point A (Xp, Zp).
Along with the positioning, the control calculates the distance (XOFF, ZOFF) between these two points (Xp, Zp) and (Xa + U, Za + W).

$$Xp = Xs - D$$

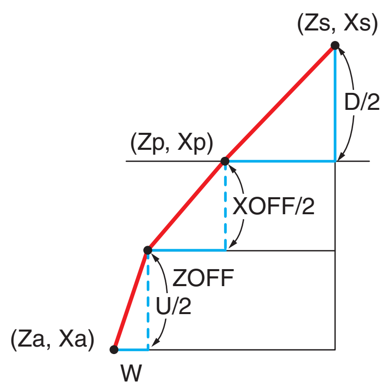
$$Zp = Za + W + (Zs - Za - W)\,(1 - D / (Xs - Xa - U))$$

$$XOFF = Xp - (Xa + U)$$

$$ZOFF = Zp - (Za + W)$$

LE33013R0301000180001

If the value of U or W is too large and the infeed direction is reversed, an alarm occurs.

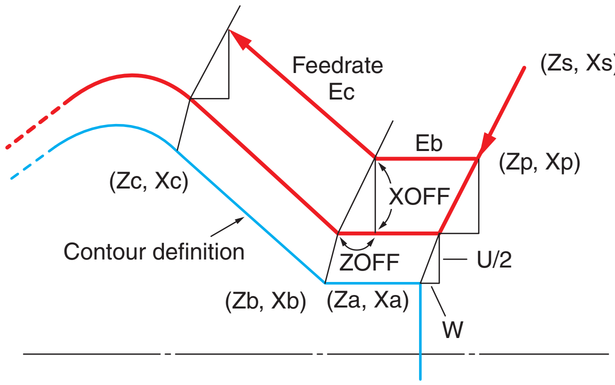


LE33013R0301000180002

(5) Cutting is started from (Xp, Zp) to the target point (*1) calculated by the OSP.

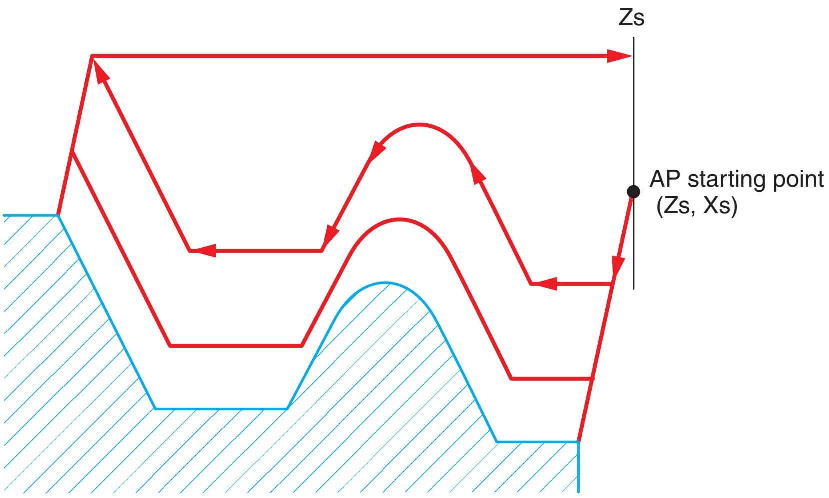
*1: The target point is the point obtained by offsetting the points commanded in the contour definition by XOFF + U + ZOFF + W), parallel to the respective axis directions.

Cutting is performed at the feedrate specified by an E word in each of the contour defining blocks.



LE33013R0301000180003

(6) Step (5) is repeated until contour definition ends (G80 active).
The Z-axis then returns to the AP starting point coordinate, Zs.



LE33013R0301000180004

(7) This completes the first rough cutting cycle. The new XOFF and ZOFF are calculated and steps (4) through (6) are repeated.
The positions for the Nth cycle are calculated as follows.

$$\left.\begin{array}{l} Xp = Xs - N \times D \\ Zp = Za + W + (Zs - Za - W) \\ \qquad (1 - N \times D/(Xs - Xa - U)) \end{array}\right] \text{Longitudinal direction}$$

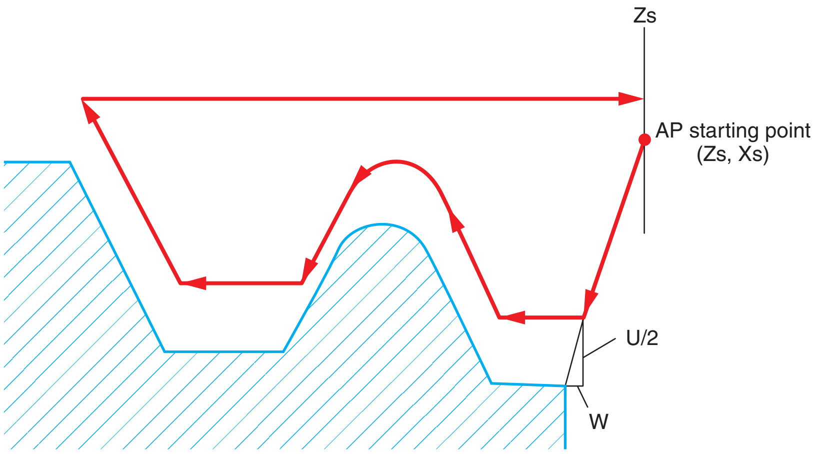
$$\left.\begin{array}{l} Xp = Xa + U + (Xs - Xa - U) \\ \qquad (1 - N \times D/(Zs - Za - W)) \\ Zp = Zs - N \times D \end{array}\right] \text{Transverse direction}$$

$$XOFF = Xp - (Xa + U)$$

$$ZOFF = Zp - (Za + W)$$

LE33013R0301000180005

(8) The steps indicated above are repeated until the infeed point reaches or exceeds "Xa + U". At the point where this happens, the control takes (XOFF, ZOFF) to be (0, 0) and cuts along a path offset from the specified contour by the amount (U, W).
At the end of contour definition, the Z-axis moves to the same Z coordinate position as the AP starting point, then the X-axis moves to the AP starting point.



LE33013R0301000180006

(9) This completes the rough turning cycle and the commands in the block following N0123 are executed.

Finish cut cycle - longitudinal cutting (example A)

(1) The commands in block N0221 position the axes at the tool change position.

(2) With the commands in block N0222, the S, T, and M commands for the finish turning cycle are selected.

(3) The NAT20 command in block N0223 causes the control to search for the program assigned the program name NAT20. The finish bar turning cycle is performed with this program.

(4) The finish turning cycle is performed on the basis of the data designated in the contour definition program under the cutting conditions specified for the finish cut cycle.

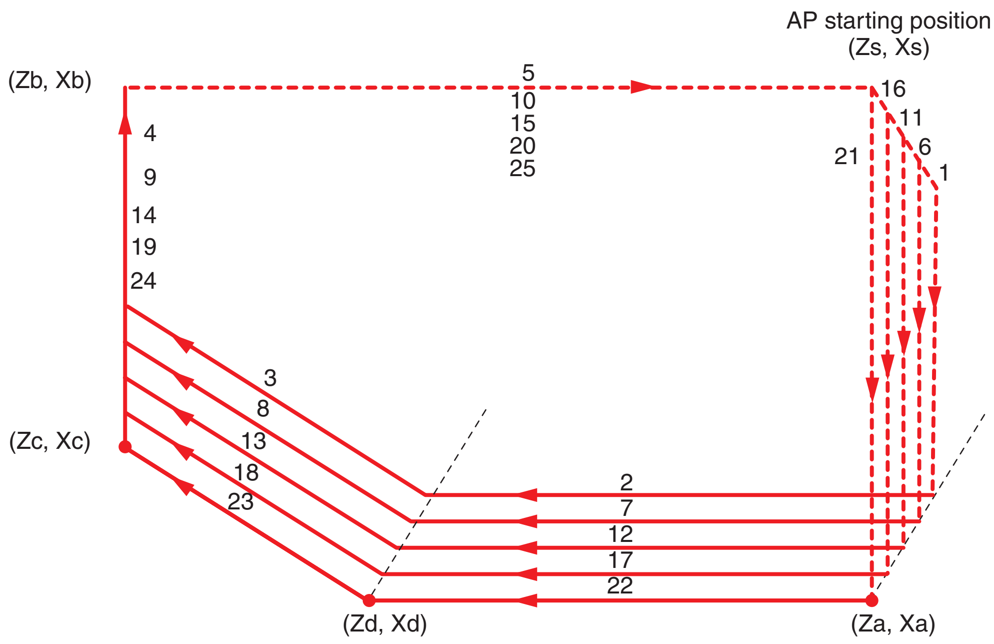
(5) After the finish turning cycle is completed, the commands in the block following N0223 are executed.

10-3. AP Mode III (Continuous Thread Cutting Cycle)

[Function]
In AP Mode III, thread cutting is executed along the contour designated by the contour definition program that starts with G61 (or G82).
The thread cutting mode (M32, M33, or M34) and the infeed pattern (M73, M74, or M75) can be selected by designating the corresponding M code.

Tool Path and Program - Longitudinal Cutting

Designate the tool path for continuous thread cutting with G34, G35, G112 and G113 (G112 and G113 cannot be designated unless the optional circular thread cutting function is selected.)



LE33013R0301000190001

Contour definition

```
NAT40   G81 ····························· Longitudinal contour definition
N0401   G00  Xa    Za
N0402   G34  Xb    Zb     E     F     J
N0403        Xc    Zc
N0404   G01  Xd    Zd
N0405   G80 ····························· End of contour definition
```

Programming Calling for Thread Cutting Cycle

```
N0141   G00                S     T     M
N0142        Xs    Zs
N0143   G88  NAT40  M32(M33, M34)  M73(M74, M75)  B  H  D  U
```

LE33013R0301000190002

Outline of Continuous Thread Cutting Cycle in the Longitudinal Direction

(1) The commands in block N0141 select the S, T, and M commands for thread cutting.

(2) The commands in block N0142 position the axes at the AP starting point (Xs, Zs).

(3) The B, H, D, and U words in block N0143 specify the data necessary for the thread cutting cycle.

B : Tip point angle of thread cutting tool
H : Height of thread to be cut
D : Depth of cut
U : Stock removal for finish cut

Two types of M codes are used to select the thread cutting mode and the tool infeed pattern.
G88 NAT40 calls out the contour definition program and executes the required thread cutting cycle (AP Mode III).
For details of the thread cutting cycle, refer to SECTION 7, "M Codes Specifying Thread Cutting Mode and Infeed Pattern".

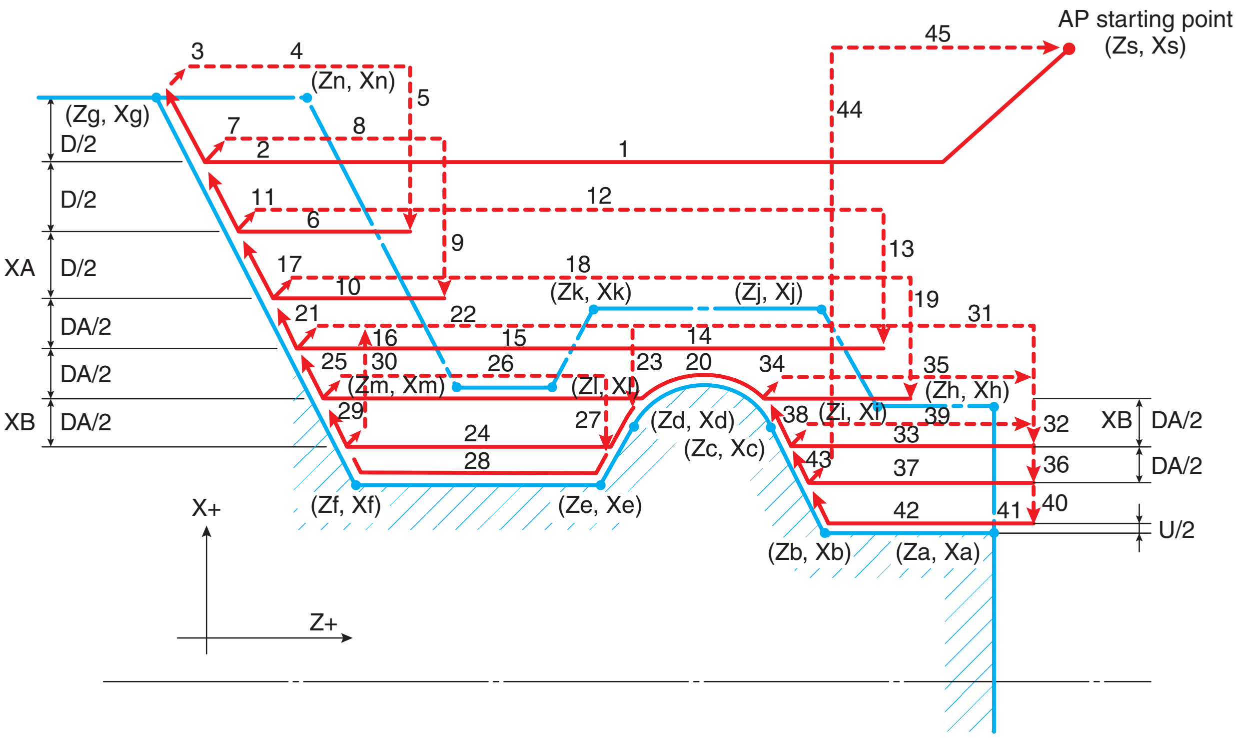
End Face Thread Cutting Cycle

For thread cutting on an end face, use G80 to G82 to define the thread contour, as in AP Modes I and II. Program M27, which selects the X-axis as the thread lead reference axis in the G34/G35/G112/G113 block. Stock removal is specified by a W word instead of a U word.

10-4. AP Mode IV (High-speed Bar Turning Cycle)

[Function]
In the AP Mode IV the blank material shape data is input in addition to the finish contour shape data. The blank material shape is programmed in the blocks starting with G83.
The area between the blank material shape and the finish contour is cut by shifting the cutting level by depth of cut D.
The OSP recognizes where blank material will be encountered, and the cutting tool is fed at the rapid feedrate in areas where there is no blank material and cutting is not required. This eliminates the cutting feed in areas that do not need to be cut, which occurs with AP Mode I , and allows high-speed cutting.

10-4-1. Tool Path and Program - Longitudinal Cutting



Contour Definition

NAT60	G83									1) Blank material shape definition start G code
N0601	G01	Xh	Zh							2) Blank material shape definition blocks
N0602		Xi	Zi							
N0603		Xj	Zj							
N0604		Xk	Zk							
N0605		Xl	Zl							
N0606		Xm	Zm							
N0607		Xn	Zn							
N0608	G81									3) Finish contour definition start G code
N0609	G01	Xa	Za							4) Finish contour definition blocks
N0610		Xb	Zb			Fb	Sb	Eb		
N0611		Xc	Zc			:	:	:		
N0612	G03	Xd	Zd	Id	Kd	:	:	:		
N0613	G01	Xe	Ze			:	:	:		
N0614		Xf	Zf			Ff	Sf	Ef		
N0615		Xg	Zg			Fg	Sg	Eg		
N0616	G80									5) Contour definition end G code

Rough Turning Cycle

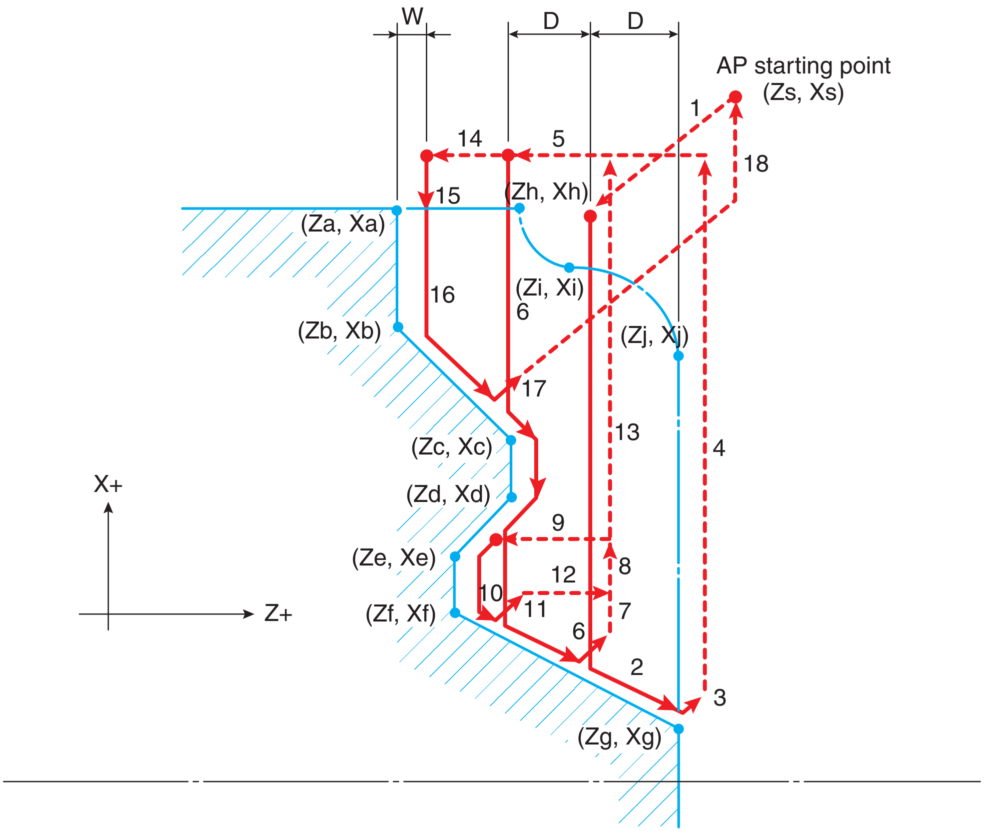
N0161	G00	Xt	Zt							Tool change position
N0162		Xs	Zs				STM			Starting point of AP, S, T, and M for rough turning cycle
N0163	G85	NAT60		D	F	U	W	M85		6) Calls for rough turning cycle
$	G84	XA=		DA=		FA=				
$		XB=		DB=		FB=				

Finish Turning Cycle

N0261	G00	Xt	Zt							Tool change position
N0262								STM		S, T, and M for finish turning cycle
N0263	G87	N0608								7) Calls for finish turning cycle

LE33013R0301000210002

10-4-2. Tool Path and Program - Transverse Cutting



LE33013R0301000220001

Contour Definition

NAT70	G83	....							1) Blank material shape definition start G code
N0701	G01	Xh	Zh						2) Blank material shape definition blocks
N0702	G03	Xi	Zi	Ii	Ki				
N0703	G02	Xj	Zj	Ij	Kj				
N0704	G82	....							3) Finish contour definition start G code
N0705	G00	Xa	Za						4) Finish contour definition blocks
N0706	G01	Xb	Zb			Fb	Sb	Eb	
N0707		Xc	Zc			:	:	:	
N0708		Xd	Zd			:	:	:	
N0709		Xe	Ze			:	:	:	
N0710		Xf	Zf			Ff	Sf	Ef	
N0711		Xg	Zg			Fg	Sg	Eg	
N0712	G80	....							5) Contour definition end G code

Rough Turning Cycle

N0171	G00	Xt	Zt						Tool change position
N0172		Xs	Zs				STM		Starting point of AP, S, T, and M for rough turning cycle
N0173	G85	NAT70		D	F	U	W	M85	6) Calls for rough turning cycle
$	G84		ZA=	DA=	FA=				
$			ZB=	DB=	FB=				

Finish Turning Cycle

N0271	G00	Xt	Zt			Tool change position
N0272					STM	S, T, and M for finish turning cycle
N0273	G87	N0704	....			7) Calls for finish turning cycle

LE33013R0301000220002

The entries in programs A and B are described in 1) through 7) below.

(1) Blank material shape definition start G code (G83)

- This code declares the start of blank workpiece shape definition.
- The blocks following the G83 block and followed by the G81 or G82 block define the blank workpiece.

(2) Blank material shape definition block

- Define the blank workpiece shape using the G01, G02, and G03 codes.
- Note that the G00 code cannot be used.
- An alarm occurs if the G02 or G03 code is specified in the first block that follows the G83 block.

(3) Finish contour definition start G code

- This code declares the start of the finish contour definition.
- The blocks following the G81 or G82 block and followed by the G80 block define the finish contour.
- G81 code: Longitudinal contour
  G82 code: Transverse contour

(4) Finish contour definition blocks

- Define the finish contour using the G00, G01, G02, and G03 codes.
- The tool retraction path after the completion of machining varies depending on whether the first block contains the G00 or G01 code.
- The G00 code can be used only in the first block.
- F: Feedrate in finishing
  S: Spindle speed in finishing
- E: Feedrate along contour in the high-speed bar turning cycle
- F, E, and S commands are all modal.

(5) Contour definition end G code
This code declares the end of contour definition.

(6) Calls for rough turning cycle

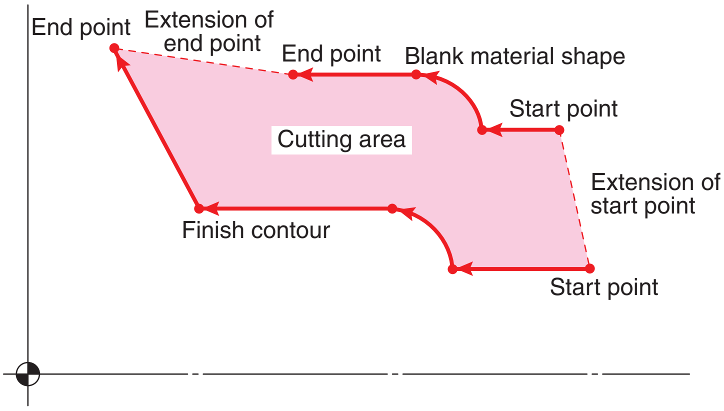
- The rough turning cycle is started by calling the contour definition blocks starting with G85.
- When the contour definition blocks start with G83, the AP Mode IV (high-speed bar turning cycle) is selected. (LAP4 only)
- When the finish contour definition blocks start with G81 or G82, the AP Mode I (bar turning cycle) is selected.

(7) Calls for finish turning cycle
Finish turning cycle is carried out by designating G87 and calling for the finish contour definition blocks starting with G81 or G82.

[Supplement]

1) The blank material shape definition must always come before the blocks defining the finish contour.
2) The blank material shape must be defined in the same direction as the finish contour is defined.

3) The start point of the blank material shape is identical to the start point of the machining shape.
4) The end point of the blank material shape is identical to the end point of the machining shape.



10-4-3. Outline of High-speed Bar Turning Cycle

Rough turning cycle in the longitudinal direction (example A)

(1) The commands in block N0161 position the axes at the tool change point.

(2) With the commands in block N0162, S, T and M commands for the rough turning cycle are selected, then the axes are positioned at the LAP starting point.
When no S, T, or M command is designated in this block, those selected in the preceding block(s) are effective.

(3) The NAT60 command in block N0163 causes the control to search for the program assigned the program name NAT60. The rough turning cycle in the bar turning mode is performed with this program.
When NAT60 is designated in the block starting with G83, a high-speed bar turning cycle (LAP4) is carried out.
The cutting conditions for the rough turning cycle are also specified in this block.

D : Depth of cut
F : Feedrate
U : X component of stock removal in finish turning cycle
W : Z component of stock removal in finish turning cycle

If M85 is designated in this block, tool retraction to the AP starting point at the completion of rough turning can be canceled. This eliminates unnecessary tool motion which is generated when the same tool is used in the next machining process.
To change the cutting conditions during the rough turning cycle, designate the following commands with G84.

XA : X coordinate of cutting condition change point A
DA : Depth of cut after point A
FA : Feedrate after point A

To change the cutting conditions again, designate the following commands.

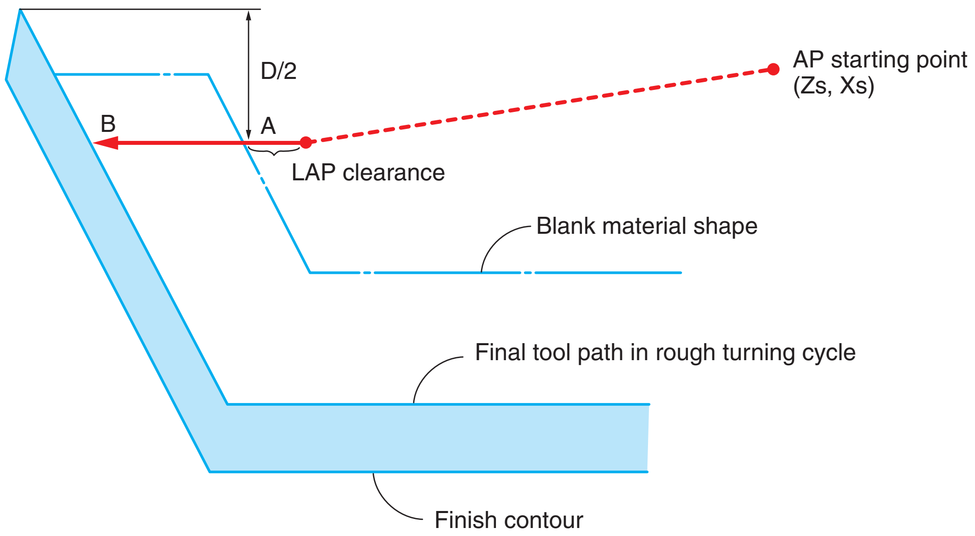
XB : X coordinate of cutting condition change point B
DB : Depth of cut after point B
FB : Feedrate after point B

Cutting condition change point(s) must be programmed in the block containing G85. For clear programming, commands related with such point(s) are programmed in different lines, each line preceded by the $ character which indicates that it is a continuation of the preceding line.
When an F word is not designated in this block, the feedrate commanded last is effective.
Point data of cutting condition change point(s) must become smaller in the order AP starting point (Xs), then XA and then XB for OD turning. For ID turning, they must become larger in this order.

(4) The commands between G83 and G81 are taken as the commands to define the blank material shape, and the commands between G81 and G80 are taken as the commands to define the finish contour.
For OD turning, draw the perpendicular from the point which is obtained by shifting the point on the maximum OD of the blank material shape or final rough turning contour, whichever is larger, and obtain the point of intersection A of this perpendicular with the blank material shape.
For ID turning, draw the perpendicular from the point which is obtained by shifting the point on the minimum ID of the blank material shape or final rough turning contour, whichever is smaller.
The cutting tool is positioned at the point distanced from point A by the LAP clearance amount (Lc) in the Z-axis direction. Positioning is performed at the rapid feedrate when G00 is designated in the first block of the finish contour definition blocks, and at a cutting feedrate when G01 is designated in the first block of these blocks.

- The LAP clearance amount (Lc) is set at LAP clearance amount of optional parameter (OTHER FUNCTION 1) in units of 0.01 mm.
- For the relationship between the LAP clearance amount and the AP starting point, refer to 10-4-5 "How to Obtain the Infeed Starting Point".
- An alarm occurs if the G02 or G03 code is specified in the first block of the blocks used to define the blank workpiece shape.

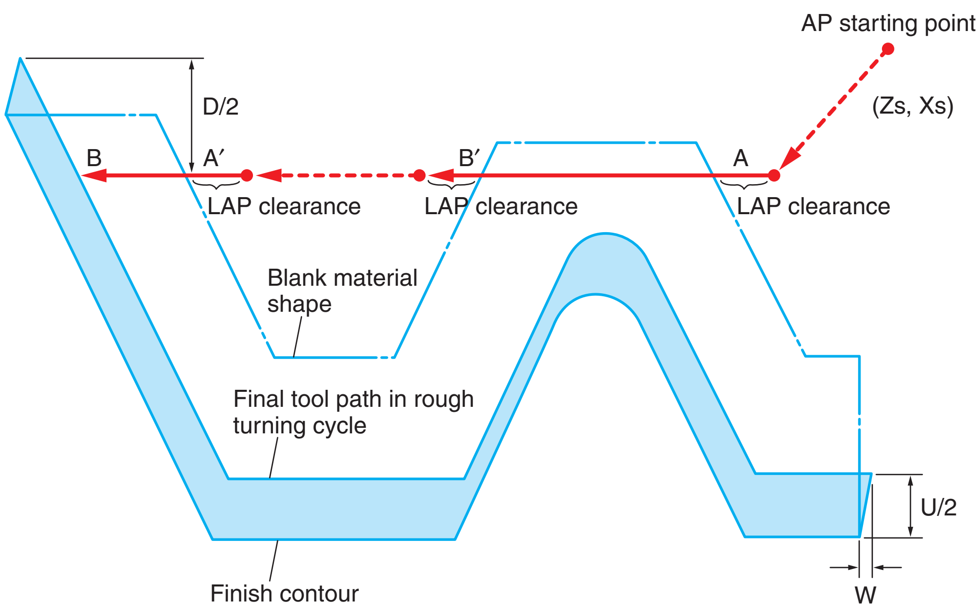
(5) The cutting is performed in the G01 mode up to point B where the straight line parallel to the Z-axis and passing through point A intersects the final contour of the rough turning cycle. The feedrate in this cutting cycle is as selected by the F word when the rough turning cycle is called out.



LE33013R0301000230001

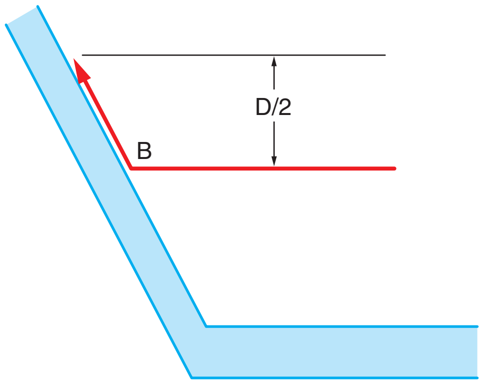
When the straight line intersects the blank material shape at point B’ before it intersects the blank material at point B, cutting is executed in the G01 mode up to the point distanced from point B’ by the LAP clearance amount (Lc) in the Z-axis direction, and after that the cutting tool is fed at the rapid feedrate.

If the straight line again intersects the blank material shape at point A’, cutting is restarted in the G01 mode from the point distanced from point A’ by the LAP clearance amount (Lc) in the Z-axis direction.



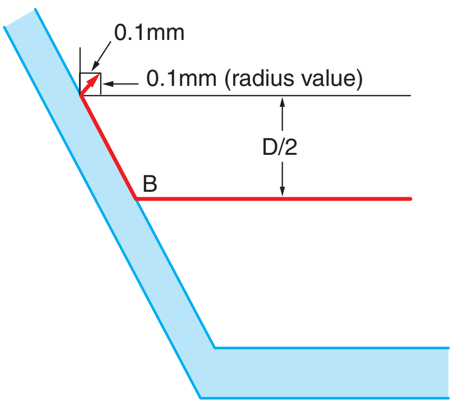
LE33013R0301000230002

(6) After point B is reached, the final contour of the rough turning cycle is cut up to the point whose X coordinate is Xb + D. If G80, indicating the end of contour definition, is found before this point is reached, the final rough turning contour is cut up to the point specified in the block preceding the G80 block.
The feedrate in this cut is as specified by E which is designated in a contour definition program. If no E word is provided in the corresponding contour definition program, the one specified last is effective. When an E word has not been specified, the feedrate specified when the rough turning cycle was called out is effective.



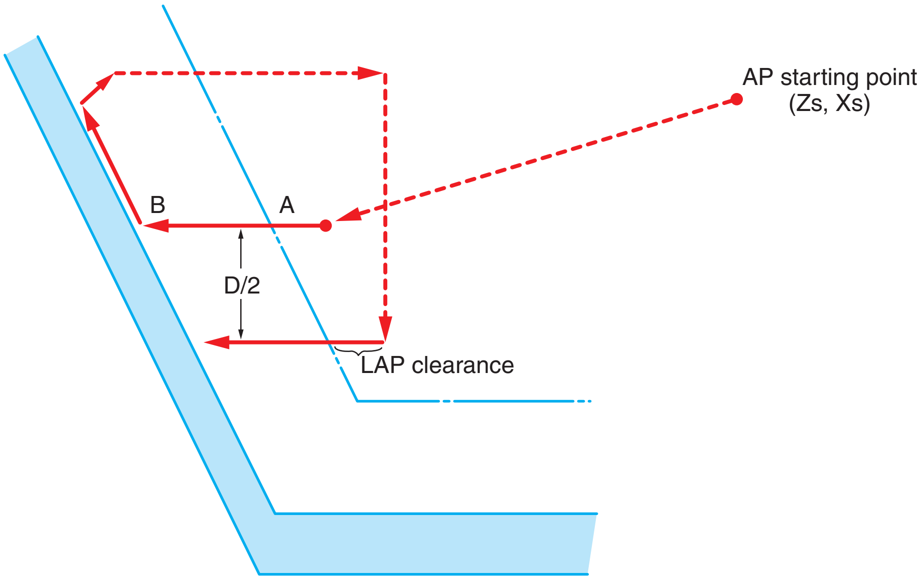
LE33013R0301000230003

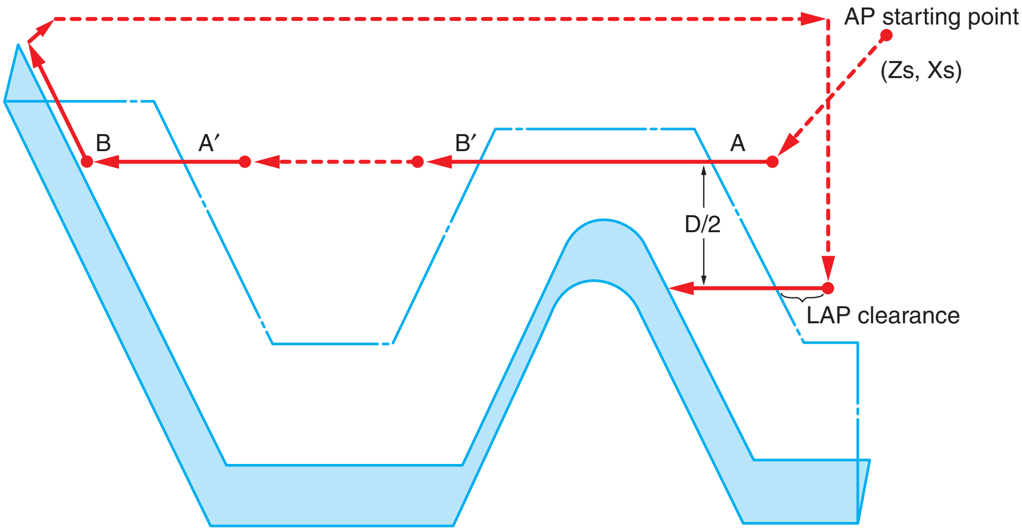
(7) After the completion of cutting explained in (6), the cutting tool is relieved from the workpiece in the direction opposite to the infeed direction along the X-axis and opposite to the cutting feed direction along the Z-axis, by 0.1 mm (0.004 in.) on each axis (diameter value in the case of the X-axis).
The relief amount is set for the optional parameter (OTHER FUNCTION 1) in units of μ.



LE33013R0301000230004

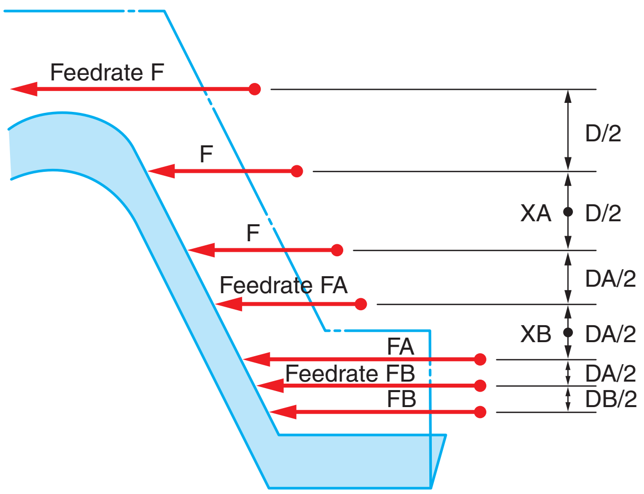
(8) This completes the first rough turning cycle. The Z-axis returns to the next infeed point at the rapid feedrate and then the X-axis to Xs.
The next infeed starting point is the point distanced from the point of intersection between the blank material shape and the line which is parallel to the Z-axis and whose X-coordinate is "the X-coordinate of the first infeed line - D" by the LAP clearance amount (Lc).





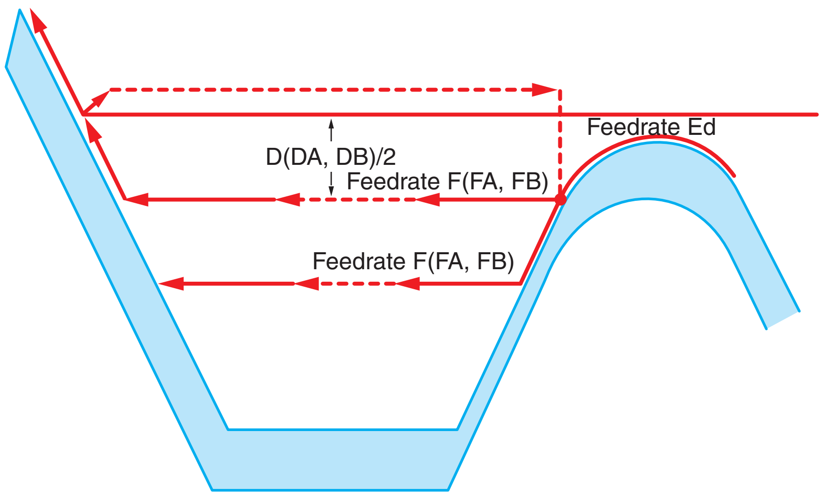
LE33013R0301000230005

Steps (4) through (8) are repeated up to the cutting condition change point. After that point, the same cycle is repeated with the depth of cut (D) and feedrate (F) changed.



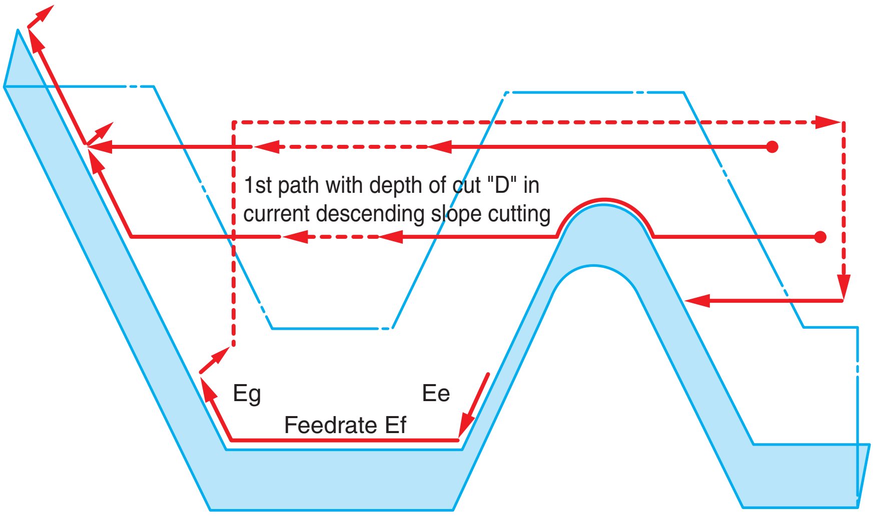
LE33013R0301000230006

(9) When a descending slope is to be cut in step (6), the cutting tool descends along the contour up to the point whose X-coordinate is the same as that of the point where cutting on contour started. Then, cutting is executed from that point in the G01 mode until the line parallel to the Z-axis intersects the final rough turning contour. The cutting tool moves in the same manner as in step (5) when the line intersects the blank material shape before it intersects the final rough turning contour.
Steps (6) and (7) are repeated. The Z-axis then returns to the point where cutting along the Z-axis is started in step (10). After the Z-axis has been positioned, the X-axis is positioned at the point where the previous cutting cycle started.



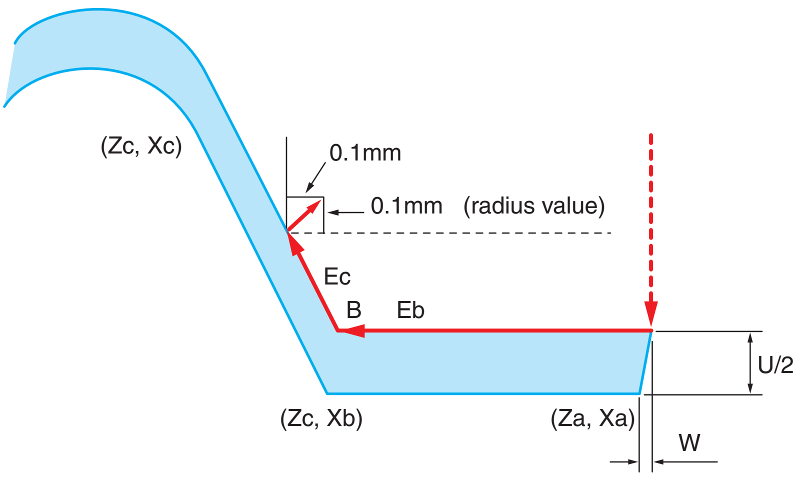
LE33013R0301000230007

(10) Steps (10) and (11) are repeated until the most recessed section along the X-axis is cut. After it has been cut, both the X- and Z-axis retract by 0.1 mm (radius value for the X-axis), and the X-axis is positioned at the point whose coordinate value is "the first cutting level along the descending slope D + 0.2" mm.
After the completion of descending slope cutting, the cutting previously in progress resumed and steps after (4) are repeated.
The next infeed starting point is the point distanced from the point of intersection between the line whose X-coordinate is "the first cutting level along the descending slope D - D" and the blank material shape, by the LAP clearance amount (Lc).



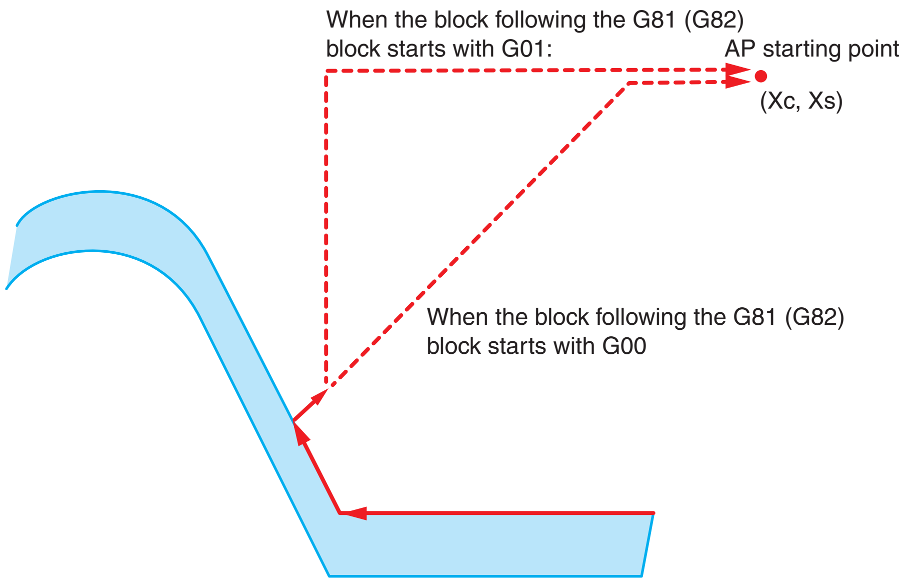
LE33013R0301000230008

(11) The steps described above are repeated until the X-axis reaches the level where a tool path is generated below "Xa + U". When such a level is reached, the final rough cutting is carried out along the contour, leaving the finish cut allowance.
The feedrate in cutting along the final rough cut contour is the one specified by the E word.
After the completion of the final rough turning step, the X- and Z-axes are relieved by 0.1 mm (0.004 in.) (diameter value for the X-axis). The relief amount is set at Relieving amount in LAP-BAR turning of optional parameter (OTHER FUNCTION 1).



LE33013R0301000230009

(12) At the completion of step (13), the axes return to the AP starting point (Xs, Zs).
There are two patterns of axis return motion:
The two axes return to the AP starting point simultaneously when G00 is designated in the first block of the contour definition program (the block following the one containing either G81 or G82).
When G01 is designated in the block indicated above, positioning is done on the X-axis first and then the Z-axis returns to the AP starting point.



LE33013R0301000230010

The tool does not return to the AP starting point as explained in step (14) when M85 is designated in the block calling for the rough turning cycle (the block starting with G85).
This completes a rough turning cycle.

Finish turning cycle in high-speed bar turning in longitudinal direction (example A)

(1) The commands in block N0261 positioning the axes at the tool change position.

(2) With the commands in block N0262, S, T, and M commands for the finish turning cycle are selected.

(3) In block N0263, the control searches the program assigned the program name N0608. The finish turning cycle in the bar turning mode is performed using this program.

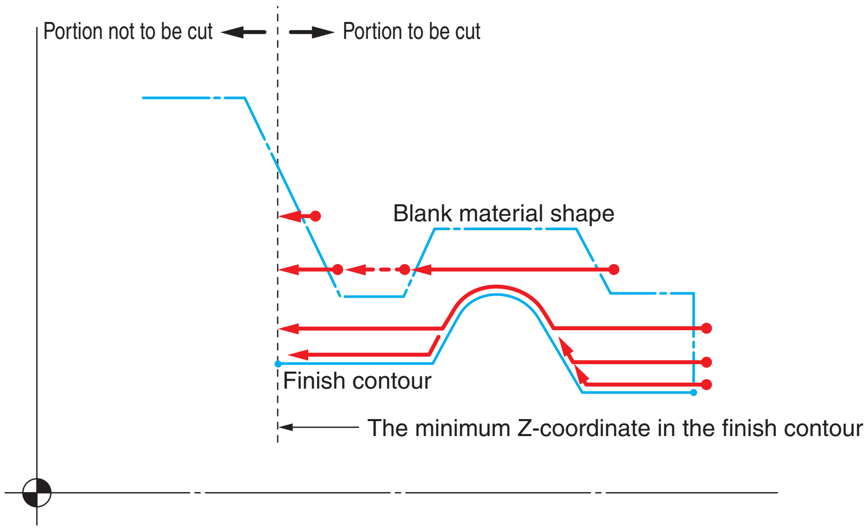
(4) The finish turning cycle is performed following the dimension data designated in the contour definition program in the specified cutting conditions for the finish turning cycle.

(5) After the finish turning cycle is completed, the commands in the block following N0263 are executed.

10-4-4. Precautions when Performing High-speed Bar Turning

Finish contour end point

In AP Mode IV, the portion beyond the Z-coordinate (X-coordinate in the transverse direction) of the finish contour end point (final rough turning contour when stock removal is designated using the U or W command) is not cut even when the blank material shape for that portion has been designated.



LE33013R0301000240001

10-4-5. How to Obtain the Infeed Starting Point

The infeed starting point in a high-speed bar turning cycle is determined by the following items:

Cs	:	AP starting point
Lc	:	LAP clearance amount
Bsp	:	Finish contour start point (after the activation of tool nose radius compensation)
Cp	:	Point of intersection between the blank material shape and the infeed line
Xp	:	Point of intersection between the line segment Cs-Bsp and the infeed line

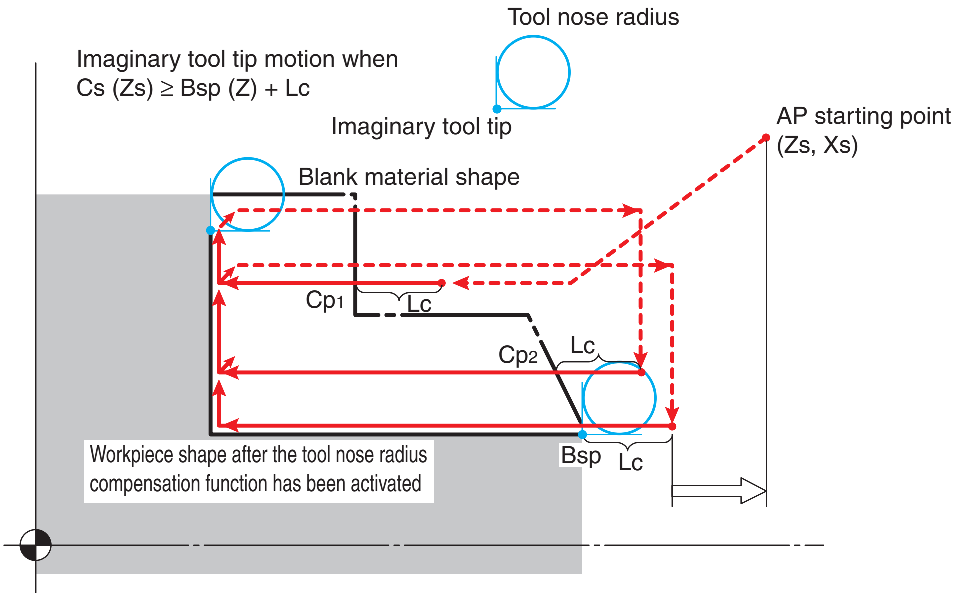
When tool nose radius compensation is not activated, the finish contour start point designated in shape definition is taken as the finish contour start point Bsp.
The explanation that follows takes longitudinal cutting in the forward direction as an example.

Cs (Zs) ≥ Bsp (Z) + Lc

This is the normal positional relationship in bar turning.

- X-coordinate of the infeed line > Bsp (X)
  The infeed starting point is defined at "Cp (Z) + Lc, Cp (X)" which is distanced from point Cp by the LAP clearance amount (Lc).

- X-coordinate of the infeed line ≤ Bsp (X)
  For cutting from the finish contour start point Bsp along the finish contour, the cutting tool is first positioned in the G00 mode at the rapid feedrate at "Bsp (Z) + Lc, Bsp (X)", which is distanced from point Bsp by the LAP clearance amount (Lc), and it is then positioned at point Bsp at a cutting feedrate in the G01 mode.



LE33013R0301000250001

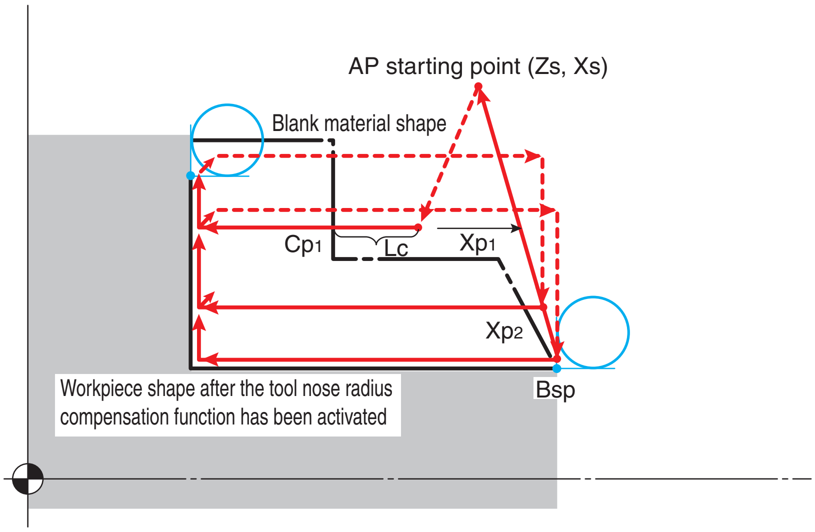
Cs (Zs) < Bsp (Z) + Lc

The portion that is to the right (in the Z-axis positive direction) of the line segment between AP starting point Cs and finish contour start point Bsp is not cut.
Assume that the point of intersection between the infeed line and this line segment is "Xp".

- Xp (Z) > Cp (Z) + Lc
  The infeed starting point is defined at "Cp (Z) + Lc, Cp (X)", which is distanced from point Cp by the LAP clearance amount (Lc).

- Xp (Z) ≤ Cp (Z) + Lc
  The infeed starting point is defined at point Xp (Z, X) where the line segment Cs-Bsp intersects the infeed line.

- X-coordinate of the infeed line ≥ Bsp (X)
  When cutting from finish contour start point Bsp along the finish contour, the cutting tool is directly positioned at point Bsp (Z, X) at the rapid feedrate.

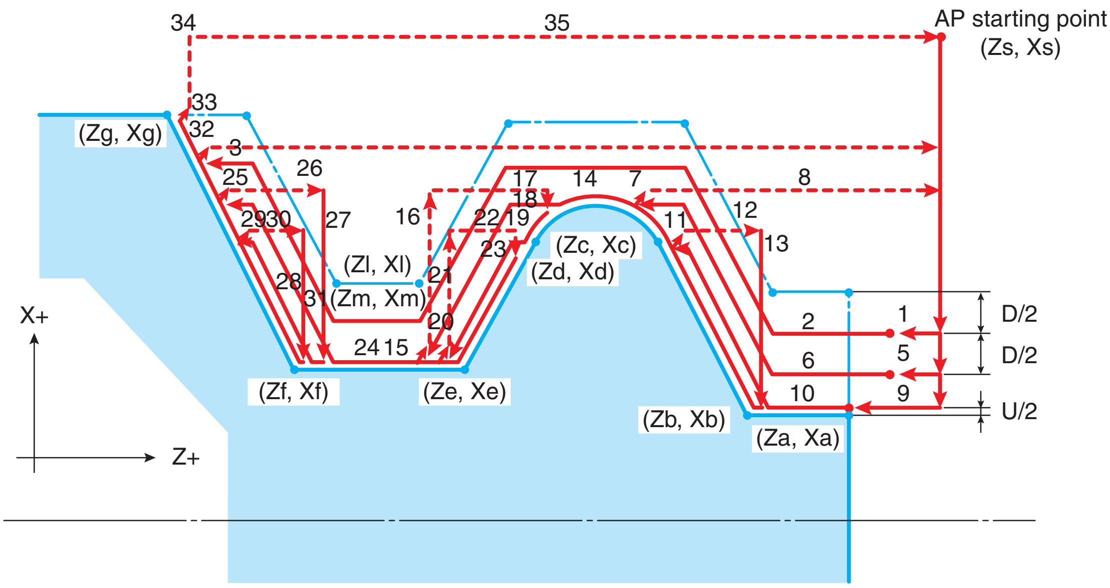


LE33013R0301000250002

10-5. AP Mode V (Bar Copying Cycle)

[Function]
In AP Mode V, the blank material shape data is input in addition to the finish contour shape data. The blank material shape is programmed in the blocks starting with G83.
Cutting is parallel to the designated blank material shape. Once the cutting tool starts cutting the workpiece, it is not moved away from the blank material until it meets the finish contour.
This feature reduces the number of tool collisions against the forged workpiece surface, resulting in longer tool life.
This mode is also effective for ID turning, which used to be difficult in AP Mode II.

10-5-1. Tool Path and Program - Longitudinal Cutting



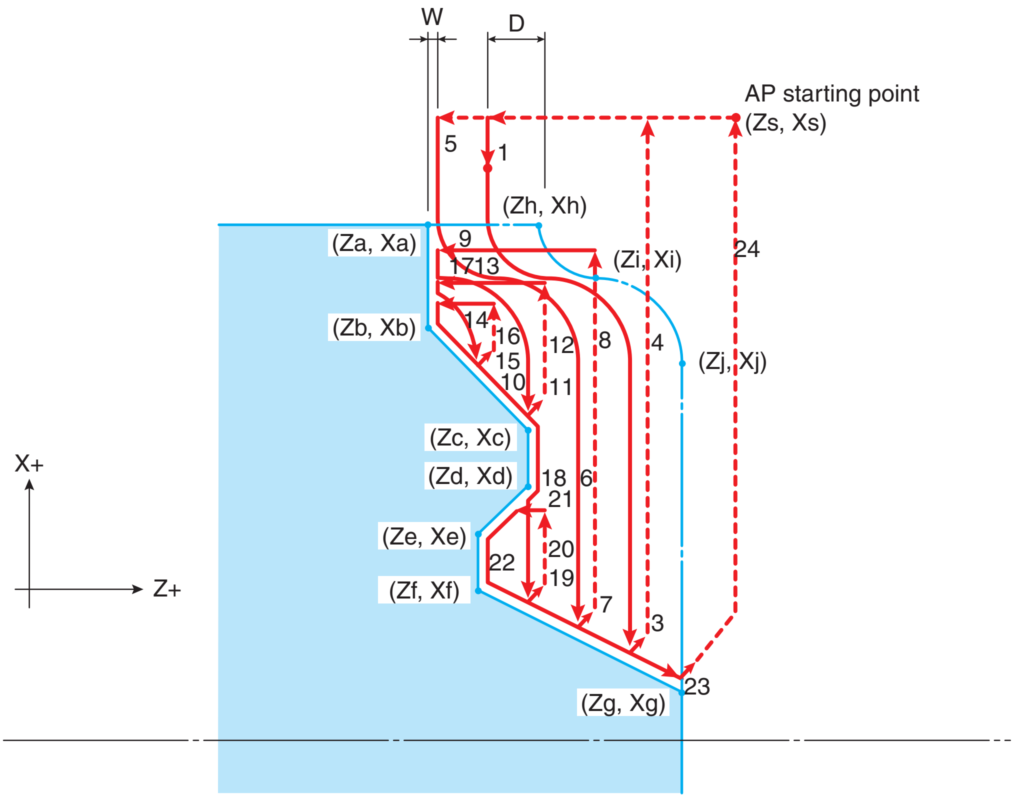
LE33013R0301000270001

Contour Definition

```
NAT80  G83 ............................................  1) Blank material shape definition start G code
N0801  G01  Xa   Za                                  ┐
N0802       Xh   Zh                                  │
N0803       Xi   Zi                                  │
N0804       Xj   Zj                                  │
N0805       Xk   Zk                                  │   2) Blank material shape definition blocks
N0806       Xl   Zl                                  │
N0807       Xm   Zm                                  │
N0808       Xn   Zn                                  │
N0809       Xg   Zg                                  ┘
N0810  G81 ............................................  3) Finish contour definition start G code

N0811  G01  Xa   Za
N0812       XB   Zb                  Fb   Sb   Eb    ┐
N0813       Xc   Zc                  :    :    :     │
N0814  G03  Xd   Zd    Id    Kd      :    :    :     │   4) Finish contour definition blocks
N0815  G01  Xe   Ze                  :    :    :     │
N0816       Xf   Zf                  Ff   Sf   Ef    │
N0817       Xg   Zg                  Fg   Sg   Eg    ┘
N0818  G80 ............................................  5) Contour definition end G code

N0181  G00  Xt   Zt                                      Tool change position
N0182       Xs   Zs                  STM                 Starting point of S, T, and M for rough turning cycle
N0183  G86 NAT80 D    F     U        W    M85            6) Calls for rough turning cycle

N0281  G00  Xt   Zt                                      Tool change position
N0282                                STM                 S, T, and M for finish turning cycle
N0283  G87       N0810 ................................  7) Calls for finish turning cycle
```

LE33013R0301000270002

10-5-2. Tool Path and Program - Transverse Cutting



LE33013R0301000280001

Contour Definition

```
NAT90  G83 ..................................... 1) Blank material shape definition start G code
N0901  G00  Xa   Za                            ┐
N0902  G01  Xh   Zh                            │
N0903  G03  Xi   Zi    Ii   Ki                 │ 2) Blank material shape definition blocks
N0904  G02  Xj   Zj    Ij   Kj                 │
N0905  G01  Xg   Zg                            ┘
N0906  G82 ..................................... 3) Finish contour definition start G code

N0907  G00  Xa   Za
N0908  G01  XB   Zb              Fb   Sb   Eb  ┐
N0909       Xc   Zc              :    :    :   │
N0910       Xd   Zd              :    :    :   │ 4) Finish contour definition blocks
N0911       Xe   Ze              :    :    :   │
N0912       Xf   Zf              Ff   Sf   Ef  │
N0913       Xg   Zg              Fg   Sg   Eg  ┘
N0914  G80 ..................................... 5) Contour definition end G code
Rough Turning Cycle
N0191  G00  Xt   Zt                              Tool change position
N0192       Xs   Zs         STM                  Starting point of S, T, and M for rough turning cycle
N0193  G86  NAT90 D   F     U    W    M85        6) Calls for rough turning cycle
Finish Turning Cycle
N0291  G00  Xt   Zt                              Tool change position
N0292                       STM                  S, T, and M for finish turning cycle
N0293  G87  N0906   ............................ 7) Calls for finish turning cycle
```

LE33013R0301000280002

The data entries in programs A and B are described in 1) through 7) below.

(1) Blank material shape definition start G code (G83)

- This code declares the start of blank workpiece shape definition.
- The blocks following the G83 block and followed by the G81 or G82 block define the blank material shape.

(2) Blank material shape definition block

- Define the blank workpiece shape using the G01, G02, and G03 codes.
- Note that the G00 code cannot be used.

(3) Finish contour definition start G code

- This code declares the start of finish contour definition.

- The blocks following the G81 or G82 block and followed by the G80 block define the finish contour.

- G81 code: Longitudinal contour
  G82 code: Transverse contour

(4) Finish contour definition blocks

- Define the finish contour using the G00, G01, G02, and G03 codes.

- The tool retraction path after the completion of machining varies depending on whether the first block contains the G00 or G01 code.

- The G00 code can be used only in the first block.

- F: Feedrate in finishing
  S: Spindle speed in finishing

- E: Feedrate along contour in the high-speed bar turning cycle

- F, S, and E commands are all modal.

(5) Contour definition end G code (G80)
This code declares the end of contour definition.

(6) Call for rough turning cycle

- The rough turning cycle is started by calling the contour definition blocks starting with G86.

- When the contour definition blocks start with G83, AP Mode V (bar turning cycle) is selected. (LAP4 only)

- When the finish contour definition blocks start with G81 or G82, the AP Mode II (copy turning cycle) is selected.

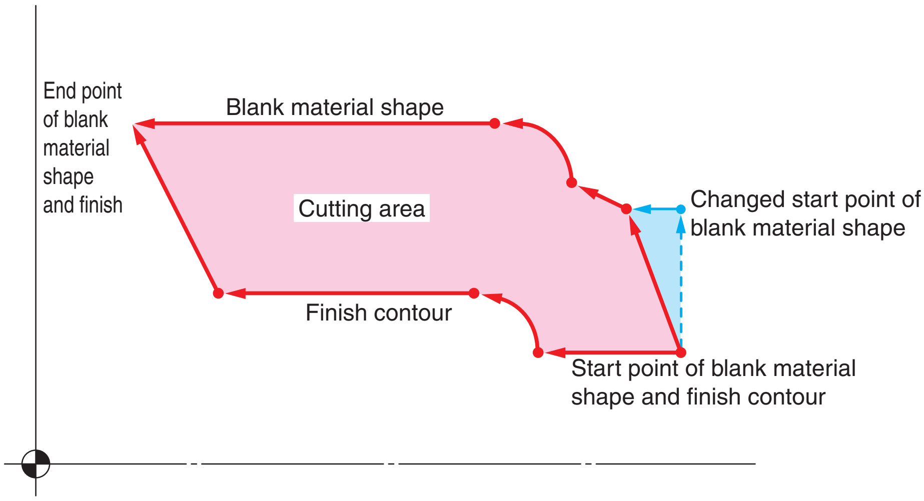
(7) Call for finish turning cycle

- The finish turning cycle is carried out by designating G87 and calling for the finish contour definition blocks starting with G81 or G82.

[Supplement]

1) The blank material shape definition must always come before the blocks defining the finish contour.
2) The blank material shape must be defined in the same direction as the finish contour is defined.

3) There are cases in which the NC changes the first element data of the blank material shape to shorten cycle time. For example, in longitudinal cutting in the forward direction, if the X-coordinate of the first element is smaller than the X-coordinate of the second element, the X-coordinate of the second element is used as the X-coordinate of the first element.



10-5-3. Outline of Bar Copying Cycle

Rough turning cycle in the longitudinal direction (example A)

(1) The commands in block N0181 position the axes at the tool change point.

(2) With the commands in block N0182, S, T, and M commands for rough cut cycle are selected, and then the axes are positioned at the LAP starting point.
When no S, T, or M commands are specified in this block, those selected in the preceding block(s) are effective.

(3) The NAT80 command in block N0183 causes the control to search for the program assigned the program name NAT80. A rough cut cycle in the bar turning mode is performed with this program.
When NAT80 is designated in the block starting with G83, a high-speed bar turning cycle (LAP4) is carried out.
The cutting conditions for the rough turning cycle are also specified in the same block.

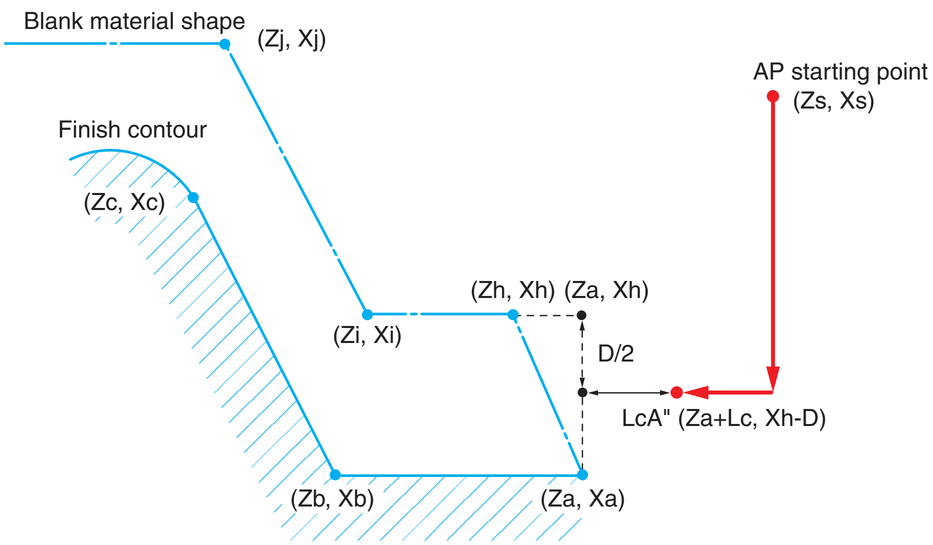
D : Depth of cut
F : Feedrate
U : X component of stock removal in finish turning cycle
W : Z component of stock removal in finish turning cycle

If M85 is designated in this block, tool retraction to the AP starting point at the completion of rough turning can be canceled. This eliminates unnecessary tool motion which is generated when the same tool is used in the next machining process.

- When no F word is designated in this block, the feedrate commanded last is effective.

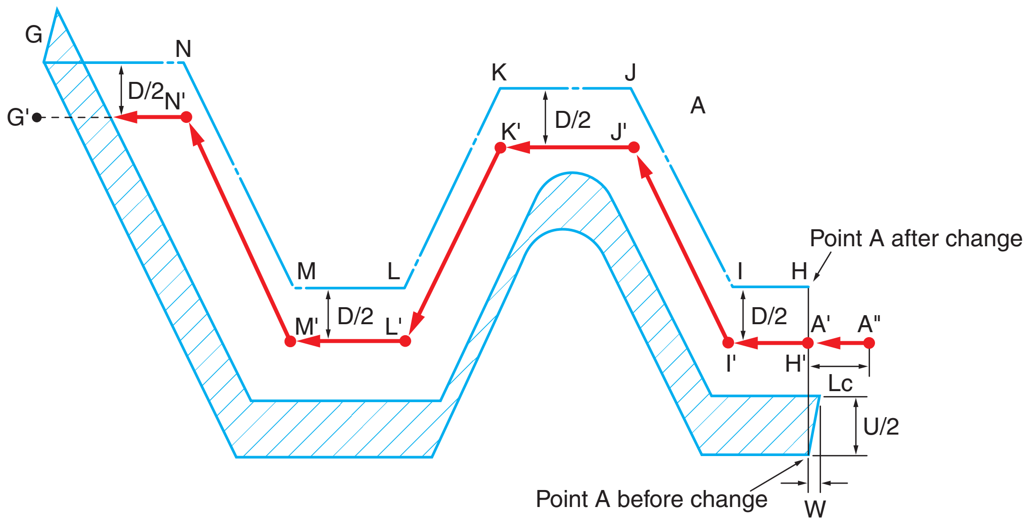
(4) The commands between G83 and G81 are taken as the commands to define the blank material shape, and the commands between G81 and G80 are taken as the commands to define the finish contour.
The first element coordinate (Za, Xa) and the second element coordinate (Zh, Xh) are compared, and since Xa is smaller than Xh in this example, the first element coordinate is changed to (Za, Xh). (Longitudinal cutting is carried out between the first and second shape elements.)
Then, first the X-axis, then the Z-axis is positioned at a cutting feedrate at point A" which is obtained by shifting the X-coordinate of the first element by depth of cut "D" in the negative direction and then shifting the Z-coordinate by the LAP clearance amount (Lc) in the positive direction.
Positioning is performed at the rapid feedrate when G00 is designated in the first block of the finish contour definition blocks, and it is performed at a cutting feedrate when G01 is designated in the first block of the finish contour definition blocks.

- The LAP clearance amount (Lc) is set for the optional parameter (OTHER FUNCTION 1) in units of µ.



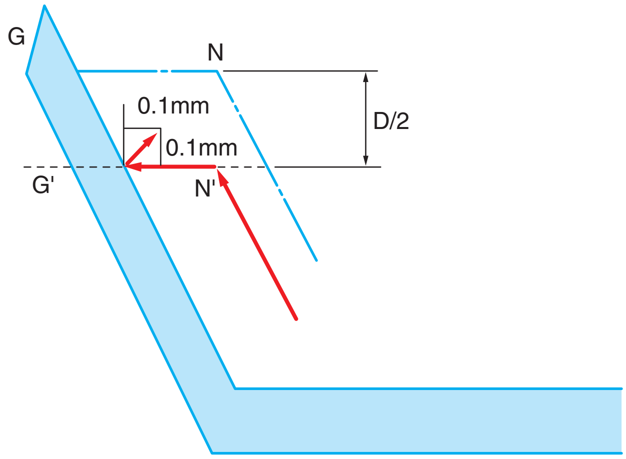
LE33013R0301000290001

(5) The points designated in the blank material shape definition blocks are shifted by D in the infeed direction. The cutting tool is fed at a cutting feedrate in the G01 mode from point A" to point A' (Za, Xh - D) which is obtained by shifting the first element coordinate (Za, Xh) of the blank material shape definition blocks by D. Then, cutting is executed along H' - G' in the G01 mode. Here, the feedrate designated by the F command in the block for calling the rough turning cycle is effective.



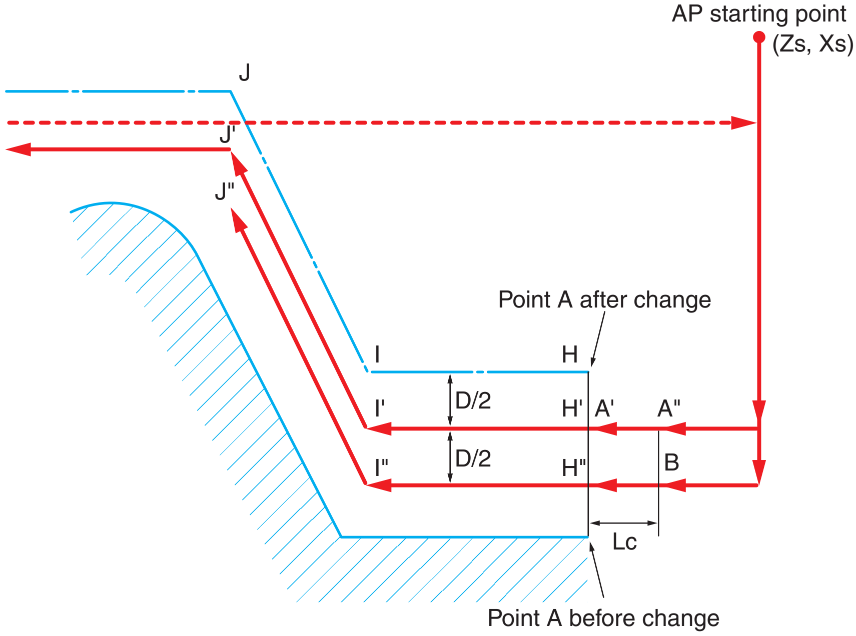
LE33013R0301000290002

(6) When cutting reaches the point where the shifted blank material shape intersects the finish contour, the cutting tool is relieved by 0.1 mm (radius value for the X-axis) in the direction opposite to the infeed direction along the X-axis, and opposite to cutting feed direction along the Z-axis. The relief amount is set for the optional parameter (OTHER FUNCTION 1) in units of µ. When stock removal is designated in the program using the U or W command, the cutting tool is relieved when cutting reaches the point where the shifted material shape intersects the final rough turning contour.



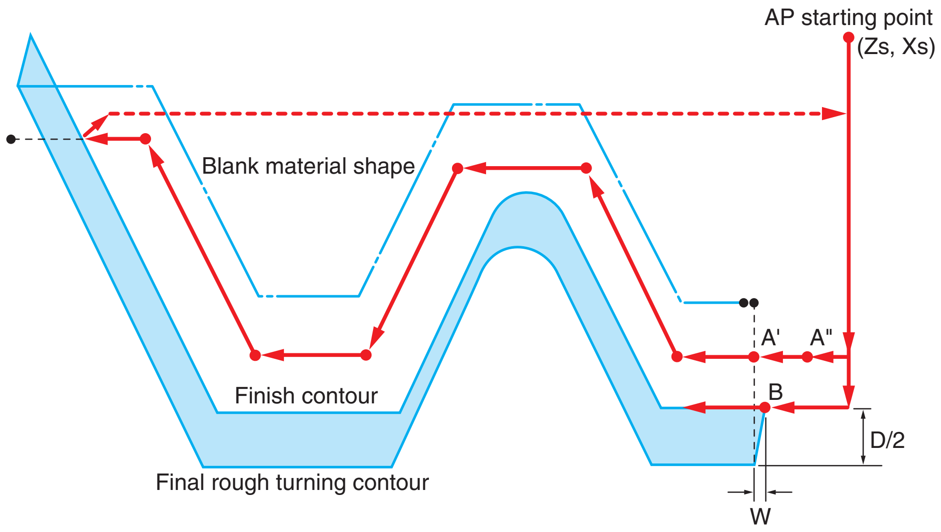
LE33013R0301000290003

(7) This completes the first rough turning cycle.
The cutting tool is then positioned at the next infeed starting point B at the rapid feedrate. When the X-coordinate at the completion of the first rough turning cycle is smaller than the largest X-coordinate of the next cutting level, the cutting tool moves up to the point "largest X-coordinate + 0.2 mm" (diameter value) at the rapid feedrate (or "smallest X-coordinate - 0.2 mm" in the case of ID turning). Then, it moves up to the Z coordinate of the AP starting point (Zs). After that, first the X-axis, and then the Z-axis moves to point B at the rapid feedrate. The approach to point B is in the same direction as the cutting direction. To obtain the "next infeed starting point B", first shift the first element coordinate (Za, Xh) of the blank material shape definition blocks by 2D in the X-axis negative direction and obtain the point (Za, Xh - 2D), and then shift this point by the LAP clearance amount (Lc) in the Z-axis positive direction. This is the "next infeed starting point (Za + Lc, Xh - 2D)".



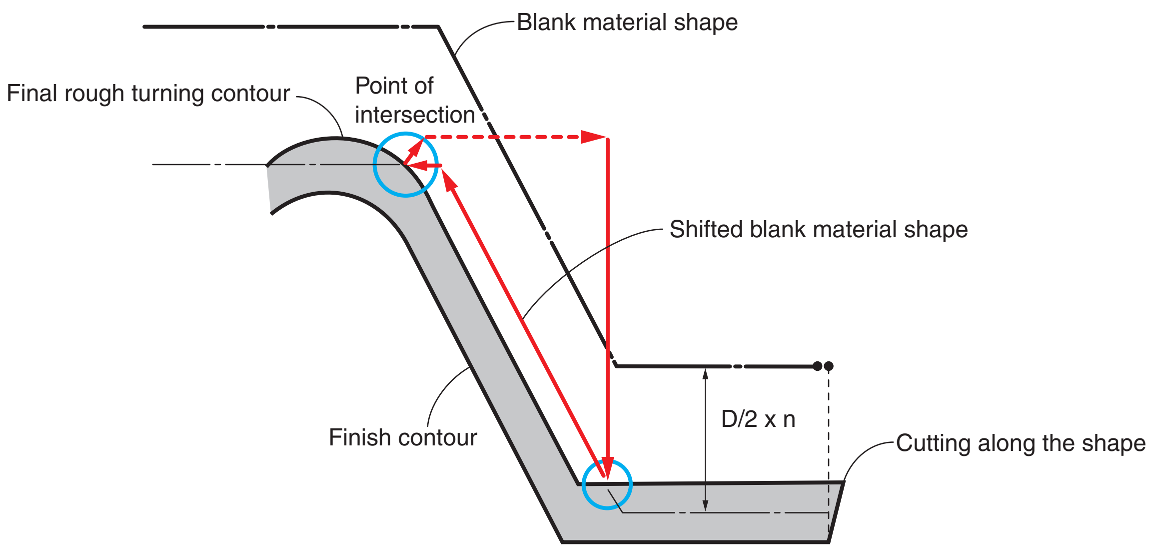
LE33013R0301000290004

(8) When the "Xh - 2D" value is smaller than the Xa value, the finish contour start point is taken as the next infeed starting point B.
When a U or W command has been designated, the final rough turning contour is taken as the next infeed starting point B.
The feedrate designated by the E command in the contour definition blocks is effective. When no E command is designated in the contour definition blocks, the E command value designated in the block before the contour definition blocks is effective. If no E command is designated at all, the feedrate designated by the F command in the block for calling the rough turning cycle is effective.



LE33013R0301000290005

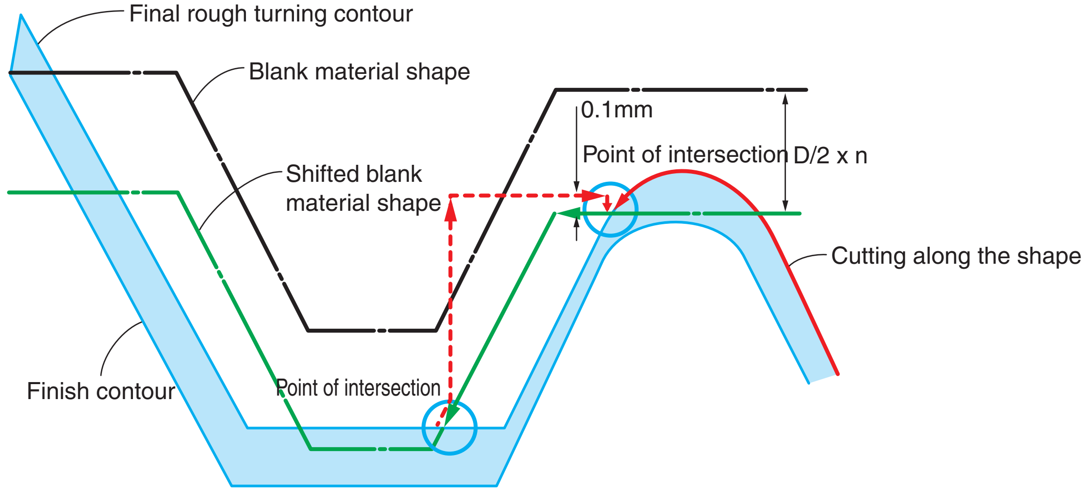
(9) When the blank material shape shifted by "D even number" intersects the contour to be machined (or final rough turning contour) during cutting along the shape, the cutting tool starts cutting along the shifted material shape. When the blank material shape shifted by "D even number" again intersects the contour to be machined (or final rough turning contour), the axes retract by 0.1 mm as in step (6). Then, the Z-axis is positioned at a point directly above the point where cutting along the shifted blank material started and the X-axis is positioned at this point.



LE33013R0301000290006

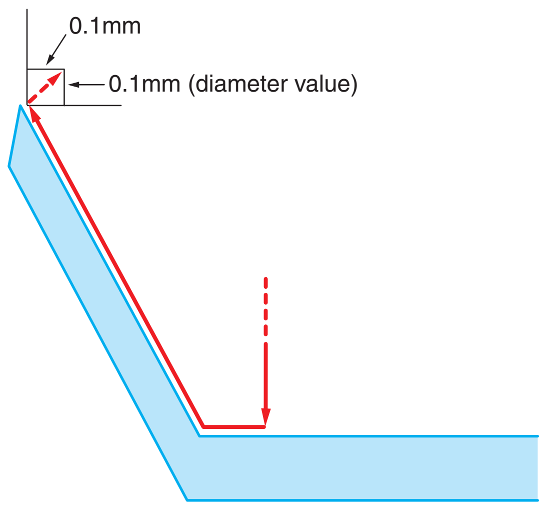
In rough turning cycles in AP Mode IV, the axes return to the point where cutting along the shifted blank material has been started according to the following procedure:

- The X-axis is positioned at the point "largest X-coordinate in that cutting cycle + 0.2 mm (0.008 in.) (diameter value)".
- The Z-axis is positioned at a point directly above the point where cutting along the shifted blank material started.
- The X-axis is positioned at the point where cutting along the shifted blank material started at a cutting feedrate.



LE33013R0301000290007

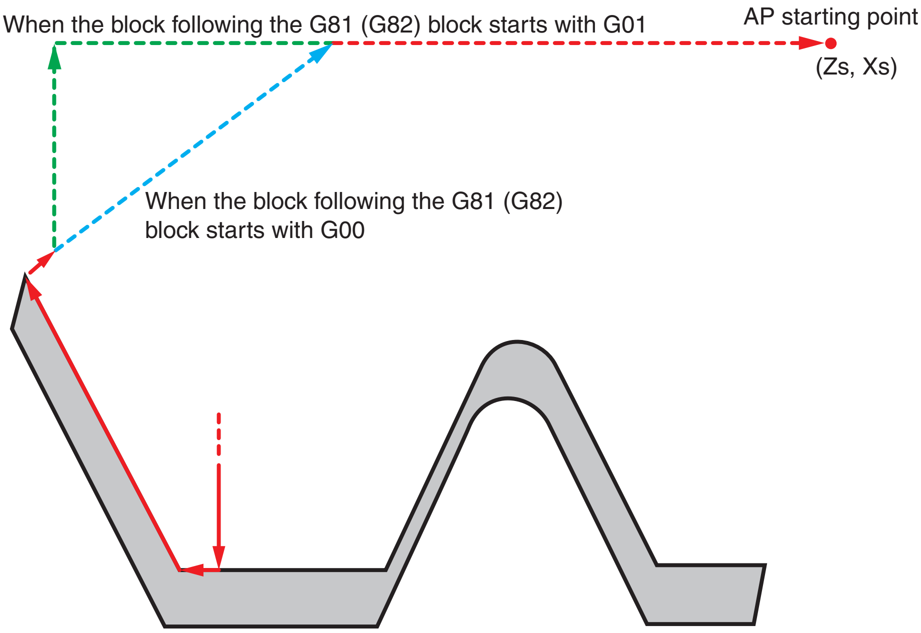
(10) Steps (8) and (9) are repeated until the area between the blank material shape and the finish contour (or final rough turning contour) is cut. Then, the cutting tool is relieved by 0.1 mm (0.004 in.) (diameter value for the X-axis) in the direction opposite to infeed direction along the X-axis and opposite to cutting feed direction along the Z-axis. The relief amount is set at Relieving amount in LAP-BAR turning of optional parameter (OTHER FUNCTION 1).



LE33013R0301000290008

(11) After the completion of step (10), the axes return to the AP starting point (Zs, Xs).
There are two patterns of axis return motion:

- The two axes return to the AP starting point simultaneously when G00 is designated in the first block of the contour definition program (the block following the one containing either G81 or G82).
- Positioning along the X-axis is done first, then the Z-axis returns to the AP starting point when G01 is designated in the block indicated above.



LE33013R0301000290009

When M85 is designated in the block calling for rough turning cycle (the block starting with G86), the axes do not return to the AP starting position as explained in step 11., and the commands in the block following N0183 are executed.
This completes the rough turning cycle.

Finish turning cycle in the longitudinal direction (example A)

(1) The commands in block N0281 position the axes at the tool change position.

(2) With the commands in block N0282, S, T, and M commands for finish turning cycle are selected.

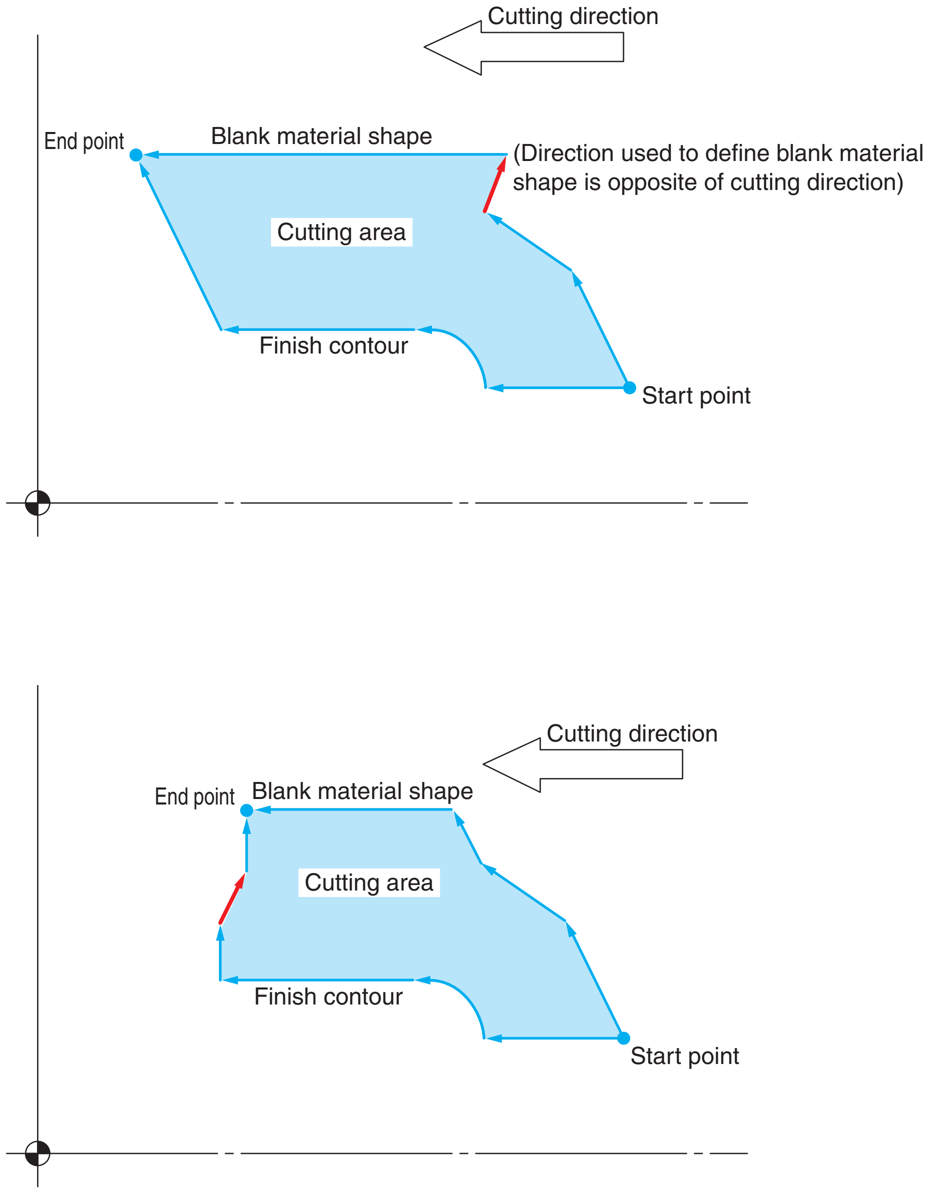
(3) In block N0283, the control searches for the program assigned the program name N0810. Finish turning cycle in the bar turning mode is performed using this program.

(4) The finish turning cycle is performed according to the cutting conditions for finish turning (F command for feedrate, S command for spindle speed) specified in the shape definition program.

(5) After the finish turning cycle is complete, the commands in the block following N0283 are executed.

10-5-4. Precautions when Executing a Bar Copying Cycle

When the direction to define the blank material shape or finish contour is opposite to the cutting direction, an alarm occurs. In such cases, define the shape again or divide the machining process.



LE33013R0301000300001

10-5-5. Precautions

- Be sure to designate the contour defining sequence name right after the G code calling for execution of a LAP program:
  G85, G86, G87 and G88

- The G83 (G81 or G82) code used to indicate the start of contour definition must be assigned a proper sequence name.

- With regard to absolute or incremental programming, G90 or G91, the mode established when G85, G86, G87 or G88 is commanded is effective. However, this mode is changed if a G code selecting another dimensioning system is specified in the contour definition program.
  In the first block of the contour definition program, it is impossible to designate G90 or G91 independently. Always designate them with X and/or Z commands in the same block.

- With regard to G64, G65, G94, G95, G96, and G97, the mode established when G85, G86, G87, or G88 is commanded is effective. Once established, this mode cannot be changed within the contour definition program.

- With regard to G00, G01, G02, G03, G31, G32, G33, G34, G35, G64, G65, G94, G95, G96, G97, G112 and G113, those commands effective when G85, G86, G87 or G88 is commanded become active after completion of the LAP.

- Nesting from LAP to LAP is not possible.

- If a G code calling for the LAP (G85, G86, G87 and G88) is designated while the nose radius compensation mode is active, an alarm results.

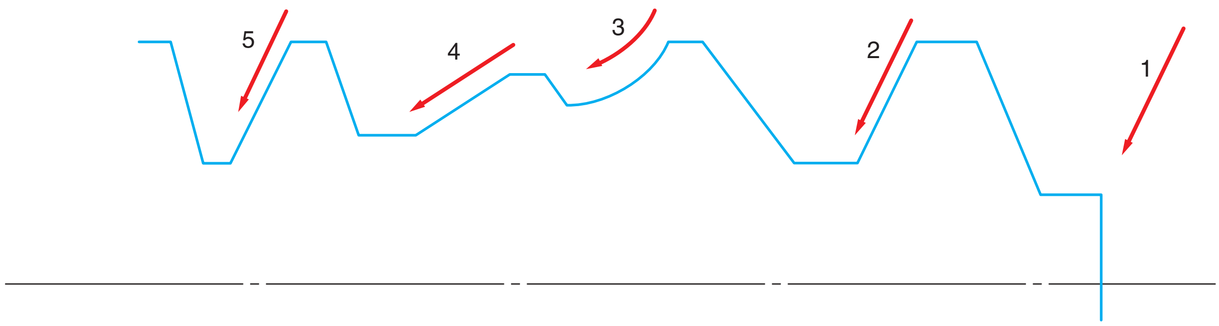
- Nose radius compensation can be activated during a LAP; however, be sure to cancel the activated nose radius compensation mode before the G80 block which indicates the end of contour definition.
  Nose radius compensation (G41/G42) can be designated only in the blocks which define the finish contour (G81/G82 - G80).

```
NAT01  G83
N0001  G01  Xa   Za
   :
   :
N0010  G81
N0011  G00  Xa   Za    G41
   :
   :
N0020
N0032  G80  Xj  Zj     G40
```

Be sure to activate and cancel the LAP function between G81 (G82) block and G80 block.

LE33013R0301000310001

- The maximum programmable number of descending slopes in AP Mode I and AP Mode IV is ten (10).



LE33013R0301000310002

- For the shape illustrated above, the number of descending slopes is five.

- If more than ten descending slopes are programmed, an alarm results.

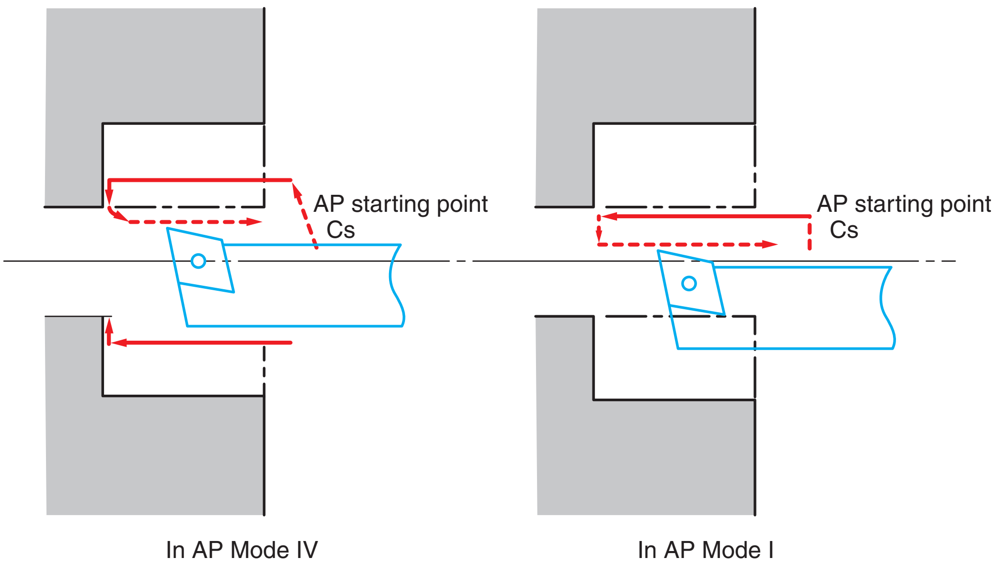
- An overcut may occur in descending slopes if both U and W are designated for descending cutting.
  Designate "U" for longitudinal cutting, and "W" for transverse cutting. (When U or W is designated, the tool is offset in the X- or Z-axis direction.)

- In AP Mode IV and AP Mode V, the first sequence name of the contour definition blocks starting with G83 can be designated by specifying G87. In this case, the blank material shape defined in the blocks between G83 and G81/G82 is ignored.
  The program examples used in this section are created so that G87 calls for the sequence number of the finish contour definition block starting with G81/G82.

- When the blocks which define the blank material shape are deleted from the NC program intended for the AP Mode IV or AP Mode V, the program can be run in the AP Mode I or AP Mode II. To allow this change, call the same sequence number as called in the G81/G82 block in the G85/G86 block.
  When the AP Mode has been changed, tool path is changed accordingly. Special care must be taken in the following cases.

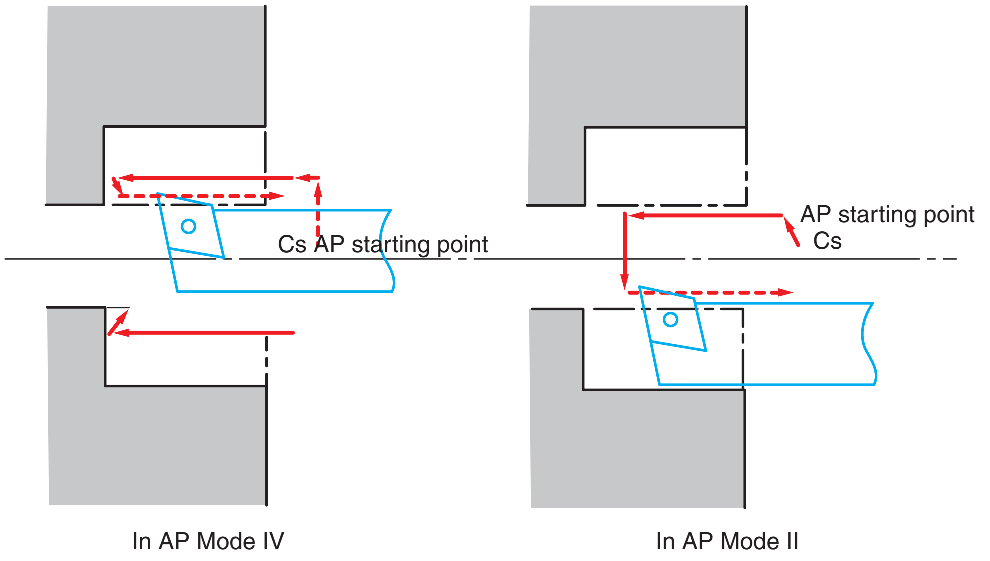
(1) ID machining
The cutting tool may interfere with the workpiece. Correct the program as necessary, for example, change the AP starting point.

From AP Mode IV to AP Mode I



LE33013R0301000310003

From AP Mode V to AP Mode II

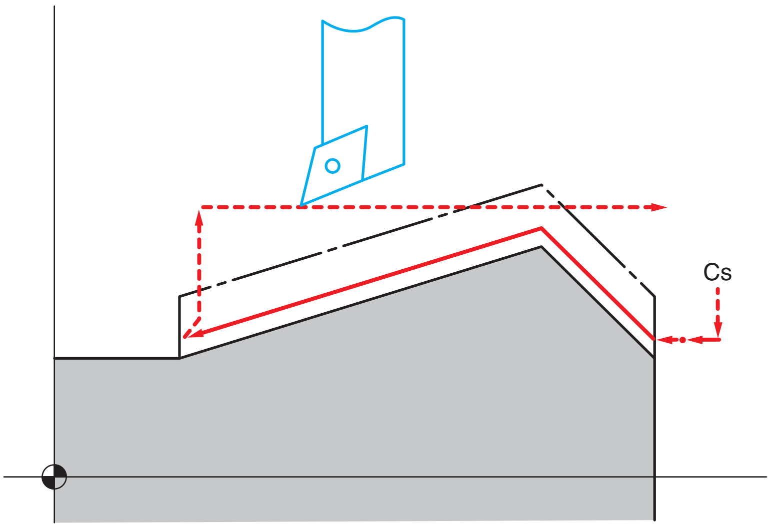


LE33013R0301000310004

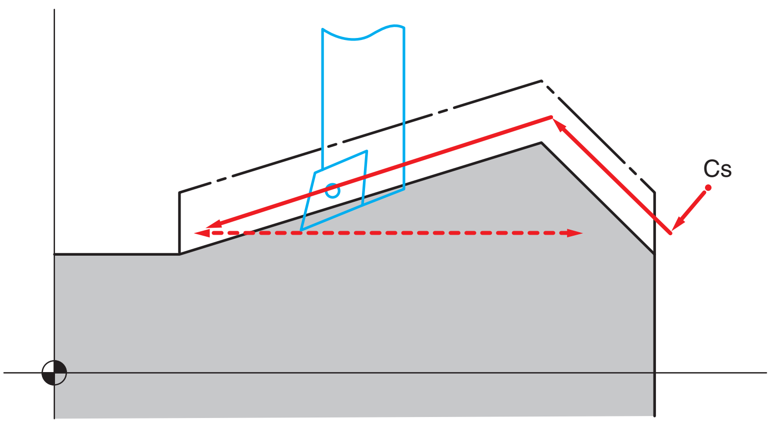
(2) Copy turning in descending slope

In the AP Mode II, the diameter must be largest at the end point of the contour definition portion (must be smallest in ID turning). Otherwise, the cutting tool interferes with the workpiece.

From AP Mode V to AP Mode II



In AP Mode V

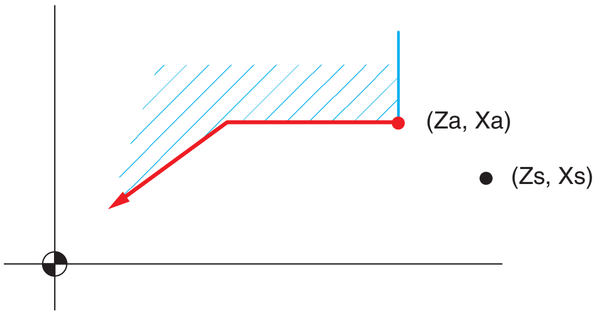


In AP Mode II

LE33013R0301000310005

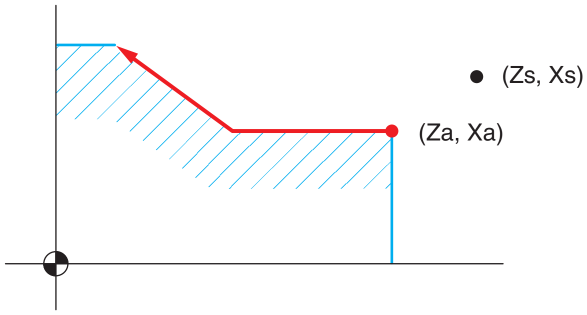
The relationship between the AP starting point (Zs, Xs) and the cutting start point (Za, Xa) must satisfy the following conditions.

For ID cutting: Zs > Za, Xs < Xa



LE33013R0301000310006

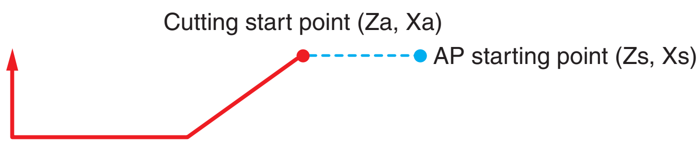
For OD cutting: Zs > Za, Xs > Xa



LE33013R0301000310007

Bear the above relationships in mind when designating the AP starting point and the cutting start point.

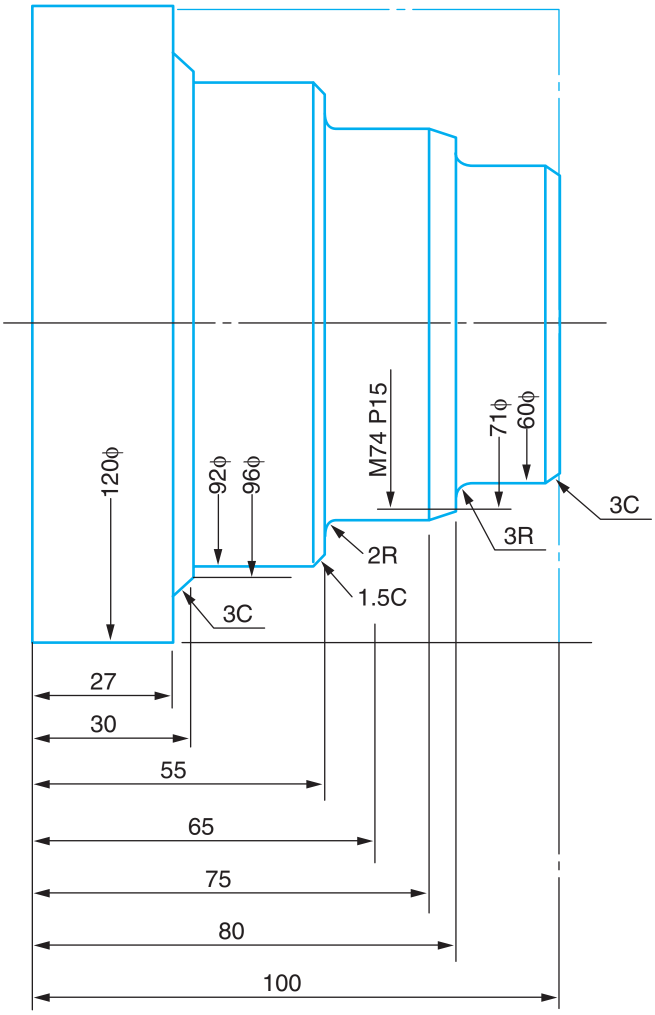
Example:



LE33013R0301000310008

When the cutting start point and the AP starting point are designated as illustrated above (where Xs = Xa), a cycle operation error will occur.

11. Application of LAP Function



LE33013R0301000320001

Machining Example using the AP Mode I

Program Example:

```
O0001

NAT1  G81
N001  G00  X54    Z102
N002  G01         Z100                F0.2
N003       X60    Z97
N004              Z83
N005  G02  X66    Z80   I3
N006  G01  X71
N007       X74    Z75         E0.4                           (Contour Definition)
N008              Z57
N009  G02  X78    Z55   I2    E0.45
N010  G01  X89
N011       X92    Z53.5       E0.4
N012              Z30
N013       X96                E0.45
N014       X102   Z27
N015       X122
N016  G80

N100  G00  X800   Z102                                       (Tool change position)
N101                    S900  T0101   M43        M03         (S, T, and M for rough turning cycle)
N102       X122                                              (Rough turning start point)
N103  G85  NAT1   D8    U0.2  F0.45                          (Calling for bar turning rough turning cycle)
N104  G00  X900   Z102
N105                    S1000 T0303                          (S, T, and M for finish turning cycle)
N106  G87  NAT1                                              (Calling for finish turning cycle)
N107  G00  X800   Z102
N108                    S950  T0505
N109       X80    Z85
N110  G33  X72.9  Z65   F1.5
N111       X72.3
N112       X71.9
N113       X71.73
N114  G00  X800   Z102                    M05
N115                                      M02
```

LE33013R0301000320002

* A contour defining program beginning with G81 and ending with G80 may be entered at any position within this program.

Machining Example using the AP Mode IV

Program Example:

```
O0002

NAT1  G83
N001  G01  X54    Z102
N002  G01  X122
N003              Z27
N004  G81
N005  G02  X54    Z102
N006  G01         Z100                          F0.2
N007       X60    Z97
N008              Z83
N009  G02  X66    Z80    I3                              (Contour Definition)
N010  G01  X71
N011       X74    Z75          E0.4
N012              Z57
N013  G02  X78    Z55    I2    E0.45
N014  G01  X89
N015       X92    Z53.5        E0.4
N016              Z30
N017       X96                 E0.45
N018       X102   Z27
N019       X122
N020  G80

N100  G00  X800   Z102                                   (Tool change position)
N101                           S900   T0101  M43   M03   (S, T, and M for rough turning cycle)
N102       X122                                          (Rough turning start point)
N103  G85  NAT1   D8     U0.2  F0.45                     (Calling for bar turning rough turning cycle)
N104  G00  X800   Z102
N105                     S1000 T0303                     (S, T, and M for finish turning cycle)
N106  G87  N004                                          (Calling for finish turning cycle)
N107  G00  X800   Z102
N108                     S950  T0505
N109       X80    Z85
N110  G33  X72.9  Z65    F1.5
N111       X72.3
N112       X71.9
N113       X71.73
N114  G00  X800   Z102         M05
N115                           M02
```

LE33013R0301000320003

* A contour defining program beginning with G81 and ending with G80 may be entered at any position within this program.

SECTION 9 CONTOUR GENERATION

1. Contour Generation Programming Function (Face)

1-1. Function Overview

The contour generation function can cut straight lines or arcs on the end face of a workpiece by simultaneous two-axis interpolation of the C- and X-axes on multi-machining models.
Note that simultaneous three-axis control of X, Z, and C axes is possible for straight line cutting on a plane.

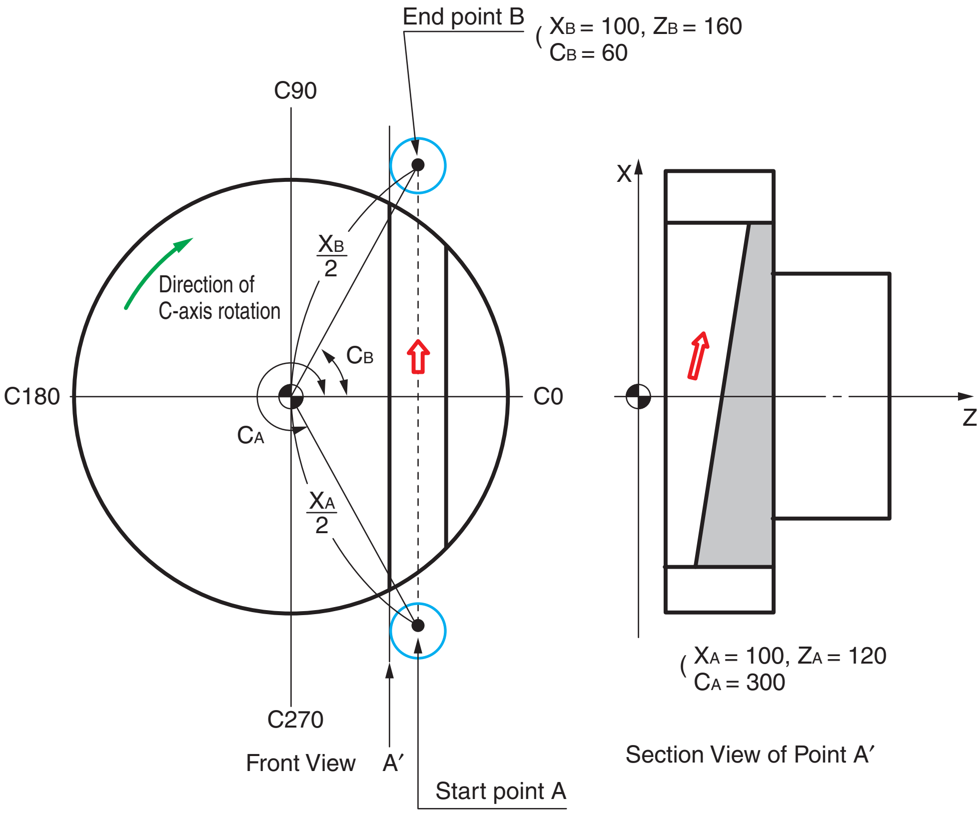
1-2. Programming Format

Straight line cutting	:	G101 X Z C F	
		X, Z, C :	Coordinates of target point on straight line
		F :	Feedrate (mm/min)
Arc cutting	:	G102 X C L F	
		X, C :	Coordinates of end point of clockwise arc
		L :	Arc radius
		F :	Feedrate (mm/min)
	:	G103 X C L F	
		X, C :	Coordinates of end point of counterclockwise arc
		L :	Arc radius
		F :	Feedrate (mm/min)

1-3. Programming Examples

Straight line cutting (G101)

Example 1:



LE33013R0301100030001

Program 1: Simultaneous 2-axis control of X and C axes

N101	M110						.... C-axis join
N102	M146	M15					.... C-axis unclamp
N103	G00	X100	C300	T0101	SB = 250		.... Positioning
N104	G94		Z120	M13			.... Start point A
N105	G101		C60	F30			.... End point B

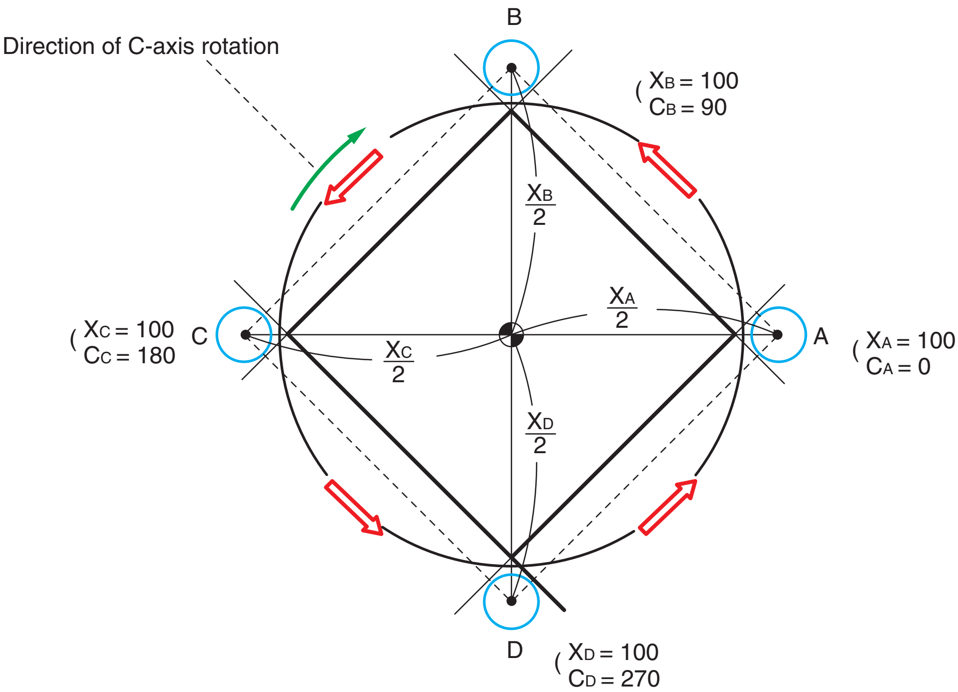
LE33013R0301100030002

Program 2: Simultaneous 3-axis control of X, Z and C axes

N101	M110						.... C-axis join
N102	M146	M15					.... C-axis unclamp
N103	G00	X100	C300	T0101	SB = 250		.... Positioning
N104	G94		Z120	M13			.... Start point A
N105	G101	Z160	C60	F30			.... End point B

LE33013R0301100030003

Example 2:



LE33013R0301100030004

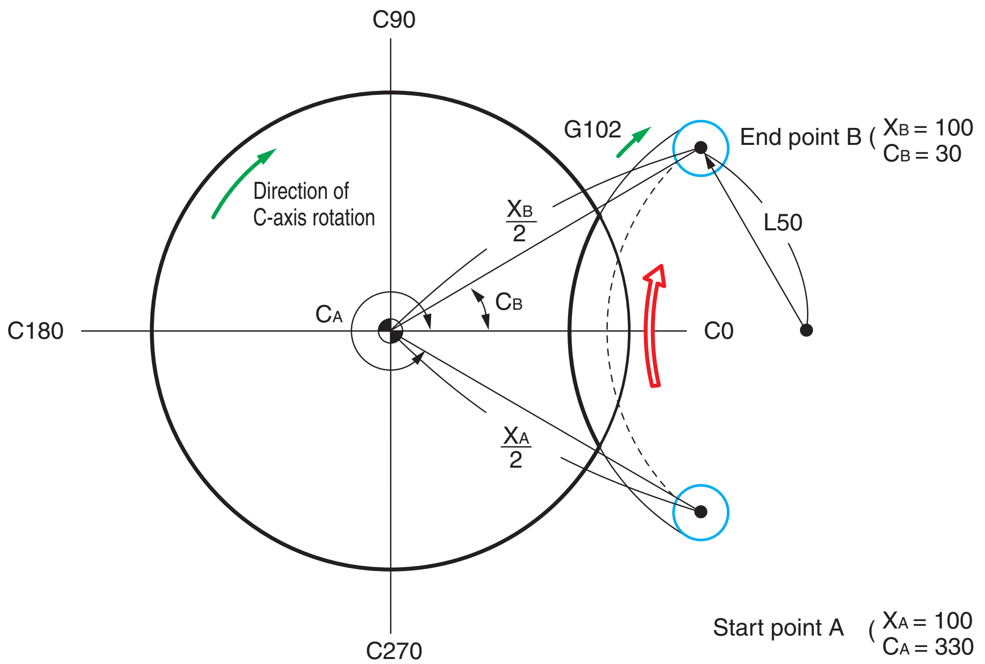
Program:

		:				
		:				
N101	M110					········· C-axis join
N102	M146	M15				········· C-axis unclamp
N103	G00	X100	C0	T0101	SB = 250	········· Positioning
N104	G94		Z120	M13		········· Start point A
N105	G101		C90	F30		········· End point B
N106			C180			········· End point C
N107			C270			········· End point D
N108			C0			········· End point A
		:				
		:				

LE33013R0301100030005

Arc cutting (G102, G103)

Example 1: G102



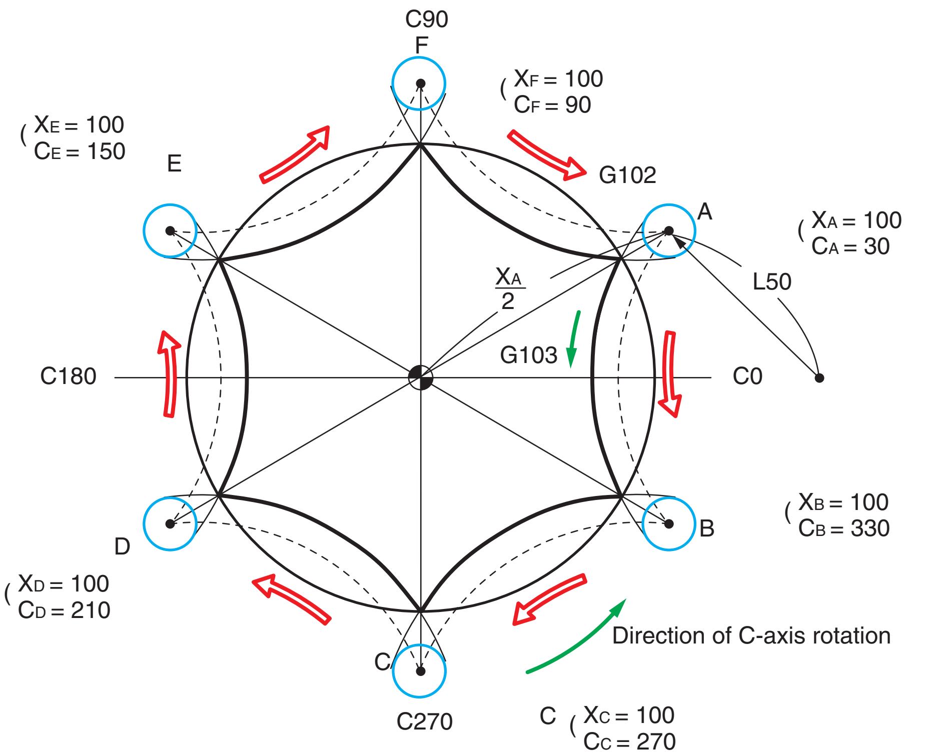
LE33013R0301100030006

Program:

		:				
		:				
N101	M110					········· C-axis join
N102	M146	M15				········· C-axis unclamp
N103	G00	X100	C330	T0101	SB = 250	········· Positioning
N104	G94		Z120	M13		········· Start point A
N105	G102		C30	L50	F30	········· End point B
		:				
		:				

LE33013R0301100030007

Example 2: G103



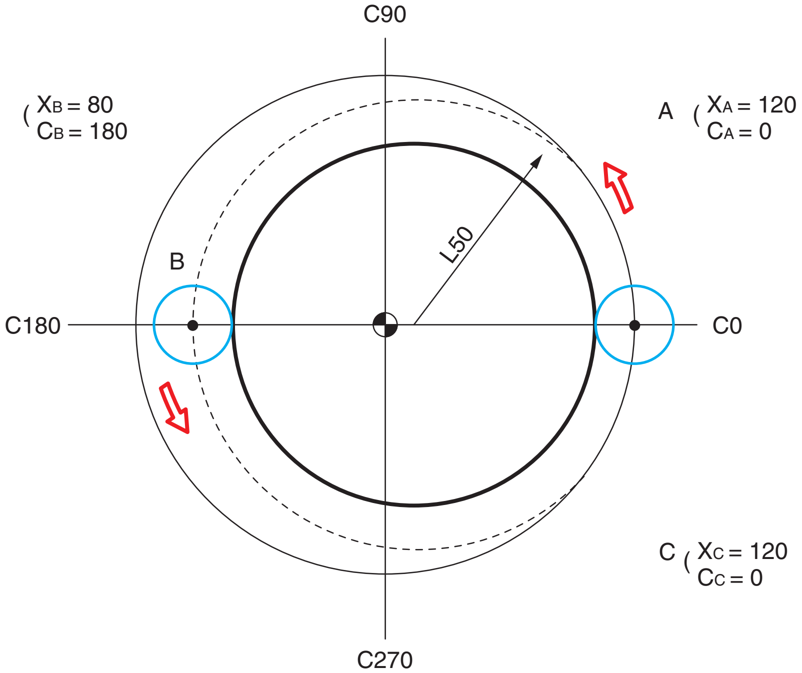
LE33013R0301100030008

Program:

		:				
		:				
N101	M110					········· C-axis join
N102	M146	M16				········· C-axis unclamp
N103	G00	X100	C30	T0101	SB = 250	········· Positioning
N104	G94		Z120		M13	········· Start point A
N105	G103		C330	L50	F30	········· End point B
N106			C270	L50		········· End point C
N107			C210	L50		········· End point D
N108			C150	L50		········· End point E
N109			C90	L50		········· End point F
N110			C30	L50		········· End point A
		:				
		:				

LE33013R0301100030009

Example 3: G103



LE33013R0301100030010

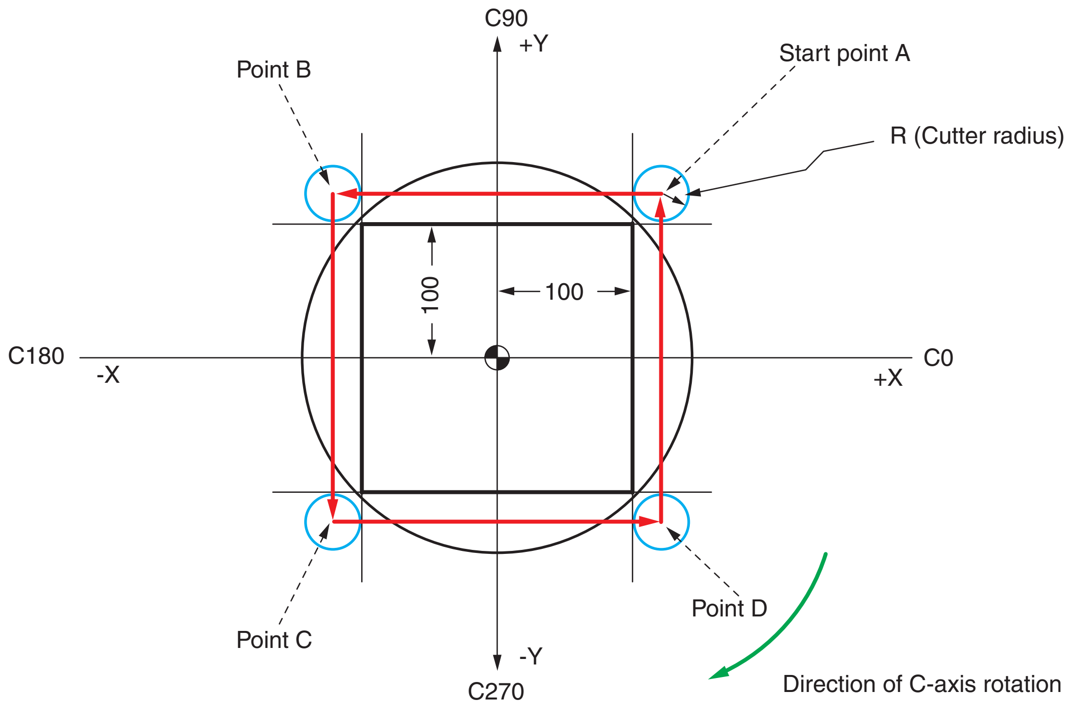
Program:

		:				
		:				
N101	M110					········· C-axis join
N102	M146	M15				········· C-axis unclamp
N103	G00	X120	C0	T0101	SB = 250	········· Positioning
N104	G94	Z120	M13			········· Start point A
N105	G103	X80	C80	L50	F30	········· End point B
N106		X120	C0	L50		········· End point C
		:				
		:				

LE33013R0301100030011

Combination with Coordinate System Conversion Function

Example 1:



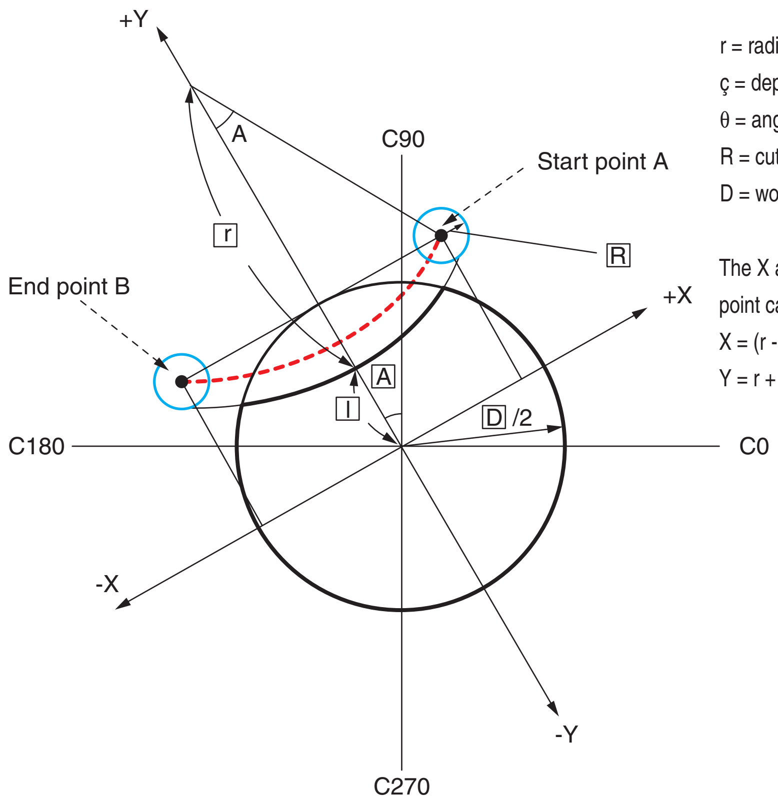
LE33013R0301100030012

V1 = R (cutter radius)
The cutter radius value should be set for common variable V1 in advance.
Program:

		:					
		:					
N101	M110						.... C-axis join
N102	M146	M15					.... C-axis unclamp
N103	G137	C0					.... Start of coordinate system conversion
N104	G00	X100+V1	Y100+V1	T0101	SB = 250		
N105	G94		Z100	M13			.... Positioning at start point A
N106	G101	X-100-V1	Y100+V1	F30			.... Cutting up to point B
N107		X-100-V1	Y-100-V1				.... Cutting up to point C
N108		X100+V1	Y-100-V1				.... Cutting up to point D
N109		X100+V1	Y100+V1				.... Cutting up to point A
N110	G136						.... End of coordinate system conversion
		:					
		:					

LE33013R0301100030013

Example 2:



r = radius of arc to be cut
ç = depth of cut
θ = angle
R = cutter radius
D = workpiece diameter

Data to be designated:

The X and Y coordinate values of the start point can be calculated as follows:

X = (r - R) sinA

Y = r + ç - (r - R) cosA

LE33013R0301100030014

Where:

$$A = \cos^{-1} \frac{(ç + r)^2 + (r - R)^2 - (D/2 + R)^2}{2(ç + r)(r - R)} \quad \text{- - - - - - - - - - } (*)$$

LE33013R0301100030015

Assuming r = 220 mm, ç= 60 mm, θ = 30, R = 20 mm and D = 250 mm, then value A will be greater than 29.6°. Use 35° for value A.
V1 = R (cutter radius)
The cutter radius R should be set for common variable V1 in advance.

Program:

```
              :
              :
N101  M110                                ......... C-axis join
N102  M146    M15                         ......... C-axis unclamp
N103  G137    C30                         ......... Start of coordinate system conversion
N104  G00     X[200-V1]*SIN[35]  Y220+60-[200-V1]*COS[35]
      T0101   SB=250
N105  G94     Z100     M13                ......... Positioning at start point A
N106  G102    X-[200-V1]*SIN[35] Y220+60-[200-V1]*COS[35]
      L220-20 F30                         ......... Cutting up to point B
N107  G136                                ......... End of coordinate system conversion
              :
              :
```

LE33013R0301100030016

[Supplement]

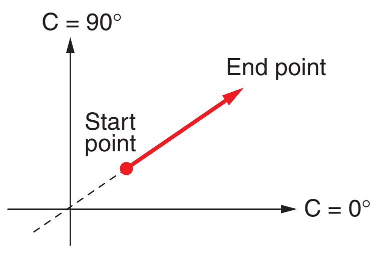
If the control does not support user task 2 (optional), it cannot perform trigonometric function calculations. Therefore, programming must be done by directly entering numeric values.

1-4. Supplementary Information

Special operation in the G101 mode

If the tool paths commanded without the cutter radius compensation function or the tool paths calculated as a result of activation of the cutter radius compensation function are straight lines passing through the center of the X-C coordinate, the following special operation occurs.

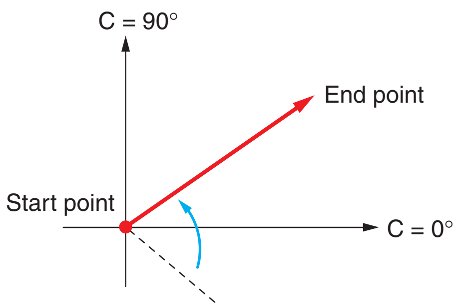
(1) When the C commands of the start and end points are the same:



LE33013R0301100040001

Although the G101 command calls for compound X- and C-axis motion, only the X-axis moves in this case (the same as G01 motion).

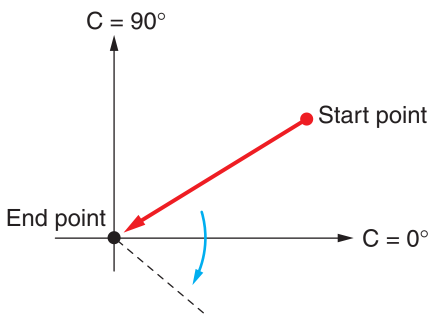
(2) When the start point lies at the center and the C commands of the start and end points differ:



LE33013R0301100040002

In this case, only the C-axis moves until the commanded value is reached; then X-axis motion occurs.

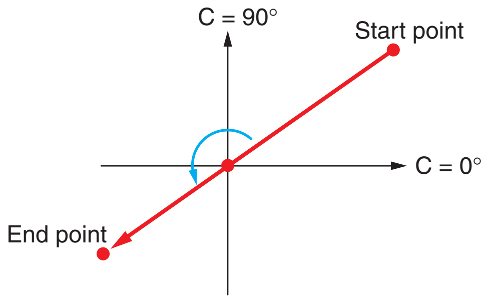
(3) When the end point lies at the center and the C commands of the start and end points differ:



LE33013R0301100040003

This case is the opposite of (2) above; only the X-axis moves until the commanded value is reached; then C-axis motion occurs.

(4) When the start and end points lie at the opposite sides of the C-axis center with the C-axis commands at these points 180° apart:

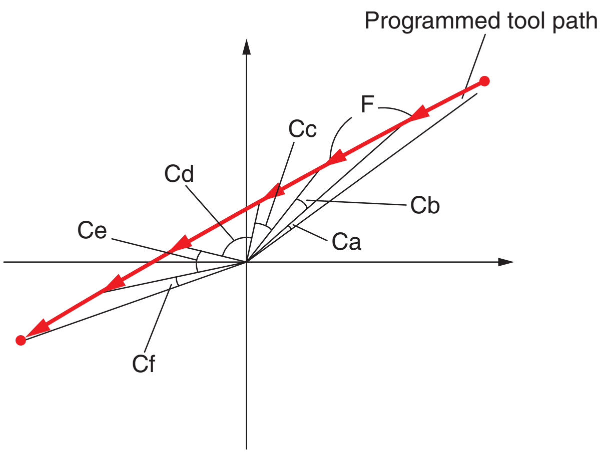


LE33013R0301100040004

In this case, first only the X-axis moves until it reaches "0". Then, the C-axis moves by 180 degrees; after the completion of the 180-degree motion, the X-axis moves again.
In motions in (2), (3), and (4) above, C-axis motion is also controlled by the commanded feedrate. It is possible to activate the C-axis feedrate override by the setting at C-axis center override (%) of optional parameter (MULTIPLE MACHINING).

- Special operation during G101 mode: Override value for C-axis feed is set.
  Setting range: 1 - 1000 (Unit: %)
  Initial value: 100 (%)

- Automatic feedrate control function
  If the commanded paths pass close the center of the X-C coordinate, the C-axis feedrate calculated from the designated compound feedrate (compound feedrate of X and C axes) might be excessively large.



LE33013R0301100040005

For the commanded feedrate F, the C-axis feedrates change in the sequence Ca, Cb, Cc and Cf. In this case, the C-axis feedrate is the maximum at Cd.
An excessively large C-axis feedrate to provide the commanded feedrate will cause the CON velocity alarm. The feedrate is limited automatically so that the C-axis feedrate will not exceed the CON velocity limit.
In this case, however, the programmed feedrate changes during the execution of the commands. Therefore, it is possible to ignore this automatic limitation by turning the automatic control function OFF with the setting at Auto limit for C-axis feedrate of optional parameter (MULTIPLE MACHINING).

- In the G101, G102, and G103 mode, the direction of C-axis rotation is determined by the control according to the programmed shape, regardless of M15 or M16.

- An alarm occurs if a C-axis command is designated in the M109 or M147 mode.

- In the G102 or G103 mode, two arcs, satisfying the start and end points and arc radius L, are obtained. The control selects the arc with a center angle of less than 180°. This means an arc having a center angle of larger than 180° cannot be machined with a single block of commands. In this kind of case, divide the arc to make a program.
  If the G102 or G103 block does not contain an L command, the L value is not positive, or L is too small to define an arc, an alarm occurs.

- In the G102 or G103 mode, Z-axis control is not possible.
  An alarm occurs if a Z-axis command is specified.

- To carry out the contour generation machining with the cutter radius compensation function ON, program the final shape. To carry out the contour generation machining with the cutter radius compensation function OFF, program the cutter center paths.

- To give the face contour generation machining commands, the X-axis must be at a position greater than "0" in the program coordinate system. An alarm occurs if the face contour generation machining commands are specified although the X-axis is at a position not greater than "0" and an alarm occurs.

2. Contour Generation Programming Function (Side)

2-1. Overview

This function carries out arc-form machining on the periphery (side face) of a workpiece on a multiple machining model by feeding the Z-axis while rotating the C-axis.
Programming is performed on the plane which is obtained by developing the cylindrical surface.
Two different planes can be assumed: one is the "outer plane" as shown in Figs. 1 and 2, and the other is the "inner plane" as shown in Figs. 3 and 4.
The plane used for programming, that is, the outer plane or inner plane, can be selected with the parameter indicated below:

Z-CE coordinate screen of optional parameter (MULTIPLE MACHINING)

- Outer plane selection (Figs. 1 and 2)
- Inner plane selection (Figs. 3 and 4)

(1) Outer Plane

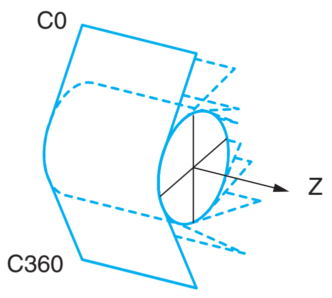


Fig. 1

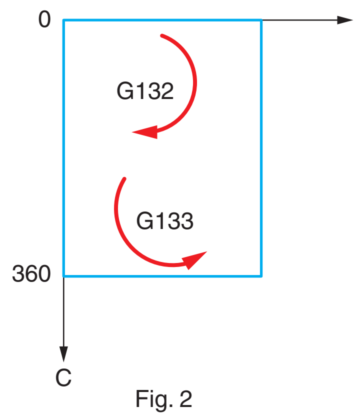


Fig. 2

LE33013R0301100050001

(2) Inner Plane

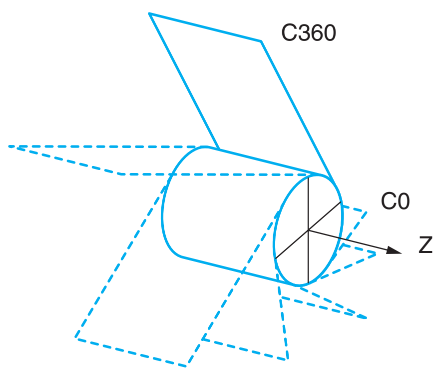


Fig. 3

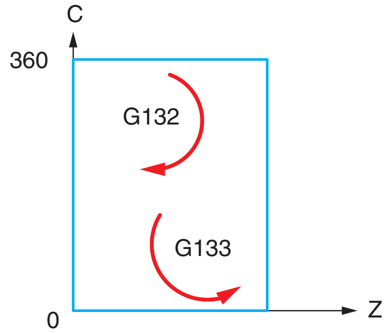


Fig. 4

LE33013R0301100050002

The circular interpolation direction, tool nose radius compensation direction, and other factors are determined based on the selected plane.

2-2. Programming Format

Circular interpolation (CW) on side face : G132 Z C L F

- Z, C : Coordinates of end point for circular interpolation (CW) on contour generation side face
- L : Radius of arc on side face
- F : Cutting feedrate (mm/min)

Circular interpolation (CCW) on side face : G133 Z C L F

- Z, C : Coordinates of end point for circular interpolation (CCW) on contour generation side face
- L : Radius of arc on side face
- F : Cutting feedrate (mm/min)

2-3. Cautions

- An alarm occurs if the X coordinate value of the start and end points are different. This is because the coordinate plane will be changed if the X coordinate values are different.
- For circular interpolation between two points A and B on the side face, there are two possible paths which have the same radius. In this case, the arc whose center angle is less than 180° is selected.
  In Fig. 5 below, the arc "a" is generated.

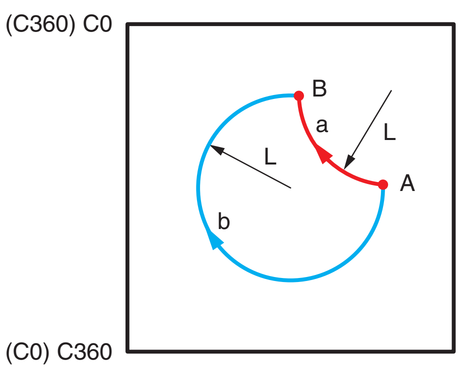


Fig. 5

The values in parentheses are for the inner plane.

LE33013R0301100070001

- For circular interpolation between two points A and B on the side face, there are two possible paths which have the same radius and a center angle of less than 180° since the C-axis is a rotary axis and the coordinate values are continuous in 360 degree cycles.

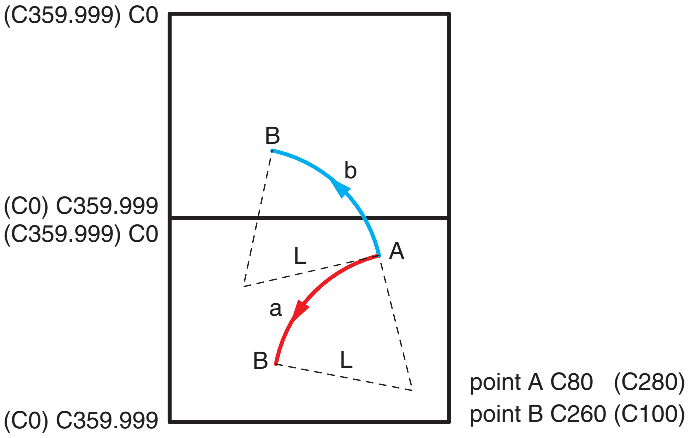


Fig. 6

LE33013R0301100070002

In such a case, an arc is generated according to M15/M16 (C-axis forward/reverse rotation command) designated preceding the arc command.
Arc "a" is generated when M15 is designated.
Arc "b" is generated when M16 is designated.
The values in parentheses are for the inner plane (in both the figures and the text).

[Supplement]

When the C-axis is joined, machining is possible within a C-axis rotation range of 5965 turns (596 turns for the 0.1 μm specification) in one direction. If side contour generation machining that exceeds this limit is carried out, the following alarm message is displayed. Alarm B 2480 Profile generation calculation. If this alarm message is displayed, use the side contour generation programming mode function. The setting method is described below.

Side Contour Generation Programming Mode Function

- Making the mode valid/invalid
  The side contour generation programming mode function is valid when "1" is set at optional parameter (bit) No. 56 bit 4.
  1: Side contour generation programming mode function Valid
  0: Side contour generation programming mode function Invalid (initial setting)

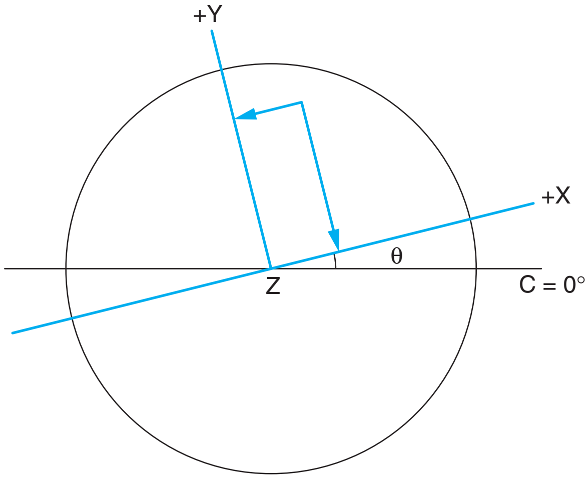
- Designating the Side Contour Generation Programming Mode
  The system enters the side contour generation programming mode when G119 is designated and the mode is turned off when G119 is canceled.
  Although G119 is originally used for the designation of the Z-C plane as the offset plane in the nose R compensation mode, it is also used to call out the side contour generation programming mode when this function is used.
  G119 is canceled in the following cases:
  - Designation of G138 (Y-axis mode ON)
  - Designation of G136 (Y-axis mode OFF)
    Note that G136 is used as the cancel code for G137 (coordinate conversion ON)
  - Designation of M109 (C-axis control OFF)
  - Reset
- Restrictions
  When the side contour generation programming mode function is set as valid, the following restrictions apply.
  The side contour generation programming commands G312 and G313 may be designated only in the side contour programming mode. If G312 or G313 is designated other than in the side contour programming mode, the following alarm message is displayed.

  Alarm B 2224 UNUSABLE contour generation command

SECTION 10 COORDINATE SYSTEM CONVERSION

1. Function Overview

Multiple-machining models have a function to convert the program commands designated in the Cartesian coordinate system into X and C-axis data in the polar coordinate system on-line.
This function simplifies programming when a hole on the end face of a workpiece is not specified by the angle but by the vertical distance from a radius vector.



LE33013R0301200010001

[Programming format]

- Start of coordinate system conversion
  G137 C__ __ __

  C : Angle of C-axis that defines the orthogonal coordinate system (θ)

- Cancelation of coordinate system conversion
  G136

[Details]

- When G137 is designated, a Cartesian coordinate system is set. In this coordinate system, the Z-axis is taken as the zero point, and the straight line in the direction of angle C designated in the G137 block is taken as the positive coordinate axis of X.
  After the designation, commands are given using the X and Y words instead of using the X and C words. Values for the X and Y words are given as radius values. Prefix X and Y words of the specified Cartesian coordinate system with a plus (+) or minus (-) sign.

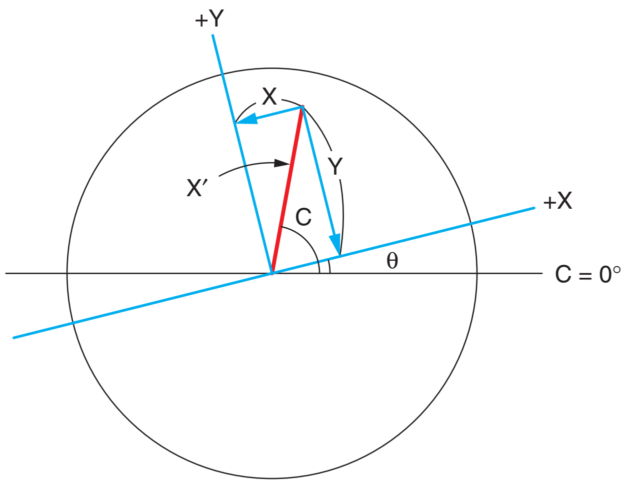
  First quadrant : X > 0 Y > 0 Second quadrant : X < 0 Y > 0
  Third quadrant : X < 0 Y < 0 Fourth quadrant : X > 0 Y < 0

- After the completion of positioning using X and Y words in the G137 mode, proceed to machining. As the machining mode, select a compound fixed cycle or G01.
  Since G00, G01, and G codes designating compound fixed cycles are canceled by G137, designate them in the next block.

- If the coordinate conversion command is designated while the X- or Z- axis is at its travel end limit position, an alarm occurs.

2. Conversion Format

The radius vector and C-axis angle after coordinate conversion are calculated with the formula below.



Radius vector, $X' = \sqrt{X^2 + Y^2}$

Angle, $C = \tan^{-1}(Y/X) + \theta$

LE33013R0301200020001

3. Program Examples

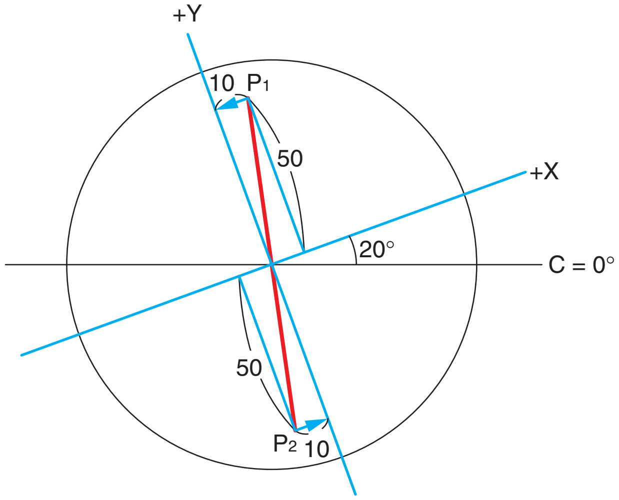
G137 is effective until G136 is designated. Do not designate other commands in the G136 block.
For the C command in a G137 block, designate the angle in reference to the C-axis zero point. This angle is equivalent to "θ" in the figure in 2. "Conversion Format" above.
After designation of G137, use X and Y words instead of X and C words as positioning commands until G136 is designated.
Example 1: Fixed cycle machining at P_1

N01	G137	C20					
N02	G181	X10	Y50	Z125	Q2	F30	K10
N03	G180						
N04	G136						

LE33013R0301200030001

Example 2: G01 mode machining at P_2

N011	G137	C20	
N012	G00	X-10	Y-50
N013	G94		
N014	G01	Z125	F30
N015	G00	Z150	
N016	G136		



LE33013R0301200030002

Note: Use X and Y words only for positioning.

4. Supplementary Information

When creating the orthogonal coordinate system by the coordinate system conversion command, it is possible to select whether or not the C-axis zero shift is included by the setting at the following parameter:
C-axis zero shift in G137 of optional parameter (MULTIPLE MACHINING)

[Supplement]

1) Designate both X and Y words in the first block after the G137 block. When only an X or Y word is designated, an alarm occurs. This does not apply to the subsequent blocks.
2) When an incremental command is designated in the G137 mode, an alarm occurs. To designate incremental commands in the G137 mode, proceed as follows.
   Cancel the incremental programming mode in the block before the G137 block.
   Designate X and Y words in the absolute programming mode in the first block following the G137 block.
   Designate the incremental programming mode.
   Program example:

```
G91                  ............  Incremental programming mode ON machining
 :
 :                                 Machining
 :
G90                  ............  Cancel the incremental programming mode before designating G137
G137    C180         ............  Coordinate system conversion
G00     X50    Y50   ............  Designate an absolute value for X and Y
G91                  ............  Incremental programming mode ON
G00     X-50
 :
 :                                 Machining
 :
G90                  ............  Cancel the incremental programming mode
G136                 ............  Cancel coordinate system conversion
 :
 :
```

   When the G90 before the G137 block is omitted, an alarm occurs.
3) When the incremental programming mode (G91) is designated without designating absolute commands in the G137 mode, an alarm occurs.
   If G91 is designated in the block right after the G137 block, an alarm occurs.

SECTION 11 PROGRAMMING FOR SIMULTANEOUS 4-AXIS CUTS (2S Model)

This section describes the programming for operations where a single workpiece is machined with two tools at the same time.

1. Programming

1-1. Turret Selection

To write a program for turret A (upper turret) or turret B (lower turret), the turret for which the program is written must be selected first. There are no differences in programming format between the programs for turrets A and B.

G code for selecting turret A : G13

G code for selecting turret B : G14

The G code used to select the turret must always be placed at the start of a program.
All commands in a program beginning with a turret selection G code are effective for the selected turret. To program an operation for the other turret, specify the G code to select it first.
Example:

```
G13
G00    X500    Z500
G00    X100    Z200    T0303
G01    X50     Z150    F4
       X20     Z20     F3
G00    X500    Z500
G14
G00    X500    Z500
G00    X100    Z200    T0303
G01    X50     Z150    F4
       X20     Z20     F3
G00    X500    Z500
M02
```

LE33013R0301300020001

Turret selection G codes may be specified in a program as many times as necessary. On program execution, the portions of the program governed by the turret selection G codes are separated into G13 side and G14 side programs which are regarded as programs for the individual turrets.

1-2. Synchronization Command (P Code)

[Function]
In simultaneous 4-axis operation, although two turrets can be operated independently, there are operations that require synchronized control of two turrets; spindle rotation during cutting using tools in both turrets is an example that requires such control.
To synchronize the execution timing of the G13 side program and the G14 side program, a special command is provided. This synchronization command is specified using address character P.
[Programming format]
P__ __ __ __
P: Integer (up to four digits)
P codes control the execution order of the G13 side and G14 side programs.
[Details]

- Program execution proceeds in order from smaller to larger P code numbers.
- If a P code is read during the execution of a program, execution of that program is suspended until a P code is read in the other side program. When a P code appears in the other side program, the P code numbers are compared and the program with the smaller P code number is executed. If the P code numbers are the same, both programs are executed. When execution of the program for one turret is completed during suspension of the program for the other turret, execution of the suspended program is resumed.

[Program example]
Spindle rotation commands, spindle speed commands, and gear range selection commands that must be synchronized between the programs of the two turrets should be specified in the manner as indicated below.

G13						
G00	X500	Z500				
			S1500	M42	M03	P10 ··· A
G00	X100	Z150				
G01		Z50	F3.5			P20 ··· B
				M05		P40 ··· D
G14						
G00	X500	Z500				P10 ··· A
			S1500	M42	M03	
G00	X130	Z150				P30 ··· C
G01	X20		F2.5			P40 ··· D
				M05		
M02						

LE33013R0301300030001

In the example program above, block A is executed for both G13 and G14 side programs. Blocks B, C and D are executed in this order.

1-3. Waiting Synchronization M Code (M100) for Simultaneous Cuts

Waiting synchronization of turrets A and B during simultaneous cuts can be commanded with M100.

```
G13
G00  X800  Z70   P10   M42  S250  T0101  M03
     X132  Z60
M100
G01  X78         F0.35

G14
G00  X800  Z200  P10   M42  S250  T0101  M03
     X82   Z65
M100
G01  X80   Z59   F0.25

```

M100 command matches timing of cuts

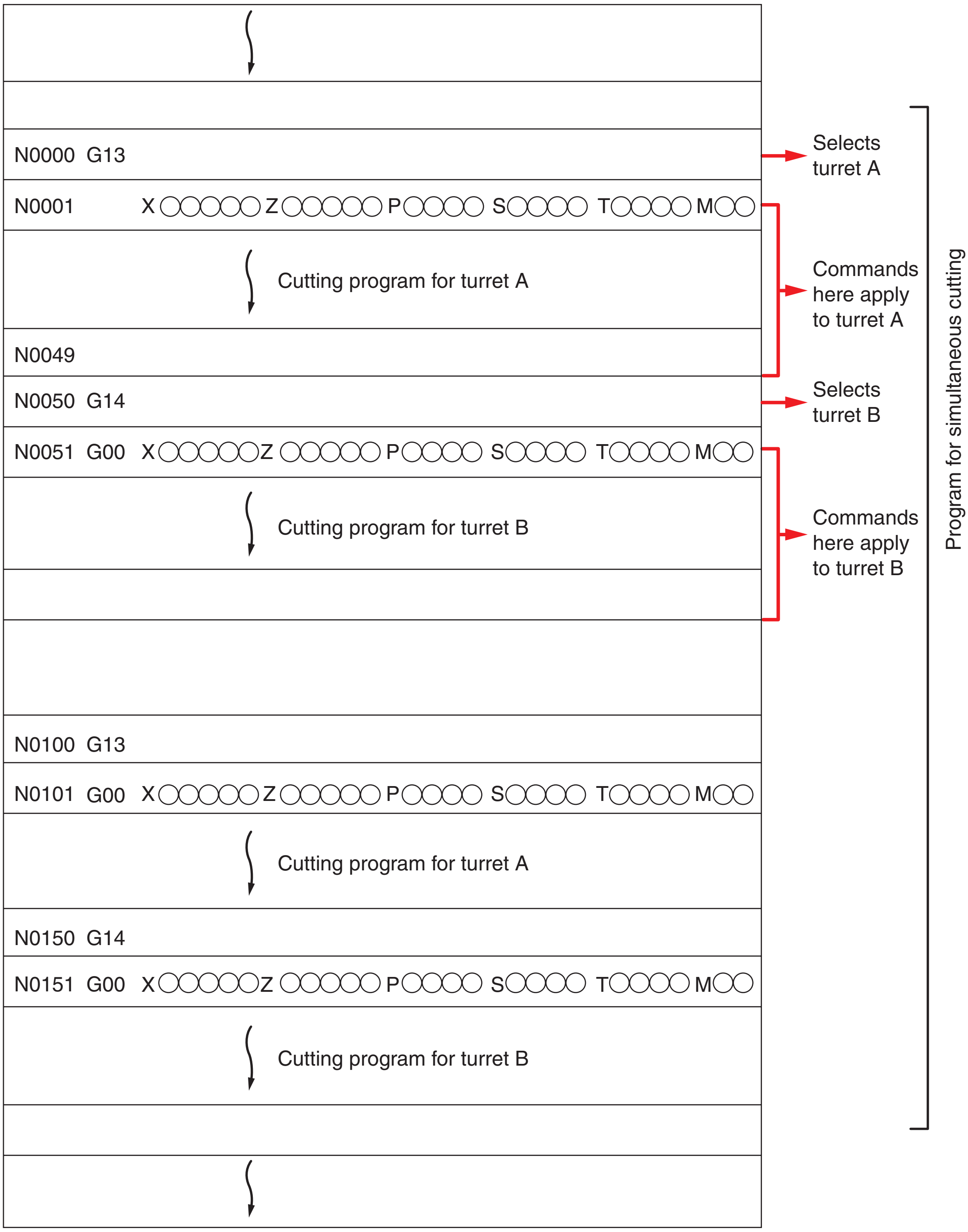
LE33013R0301300040001

[Supplement]

The following points should be considered when synchronizing operations with the M100 command.

1) S and M commands cannot be synchronized with the M100 command.
2) The same number of M100 codes must be used at both the G13 and the G14 sides in the program
   If a different number of M100 codes were to be programmed in the G13 and G14 sides, operation would continue with no waiting time.
3) The insertion of an M100 command into a nose R compensation operation will result in an alarm. No advance program reading is conducted during a stop which has been programmed by an M100 command. The nose R compensation, however, requires advance program reading, and for this reason insertion of an M100 command in this operation is not permitted.
4) Take special care not to mix P codes and the M100 command.
   Any attempt to stop one turret by use of an M100 command while the other turret is stopped due to a P code will result in operation continuing with no waiting time at all.

2. Programming Format

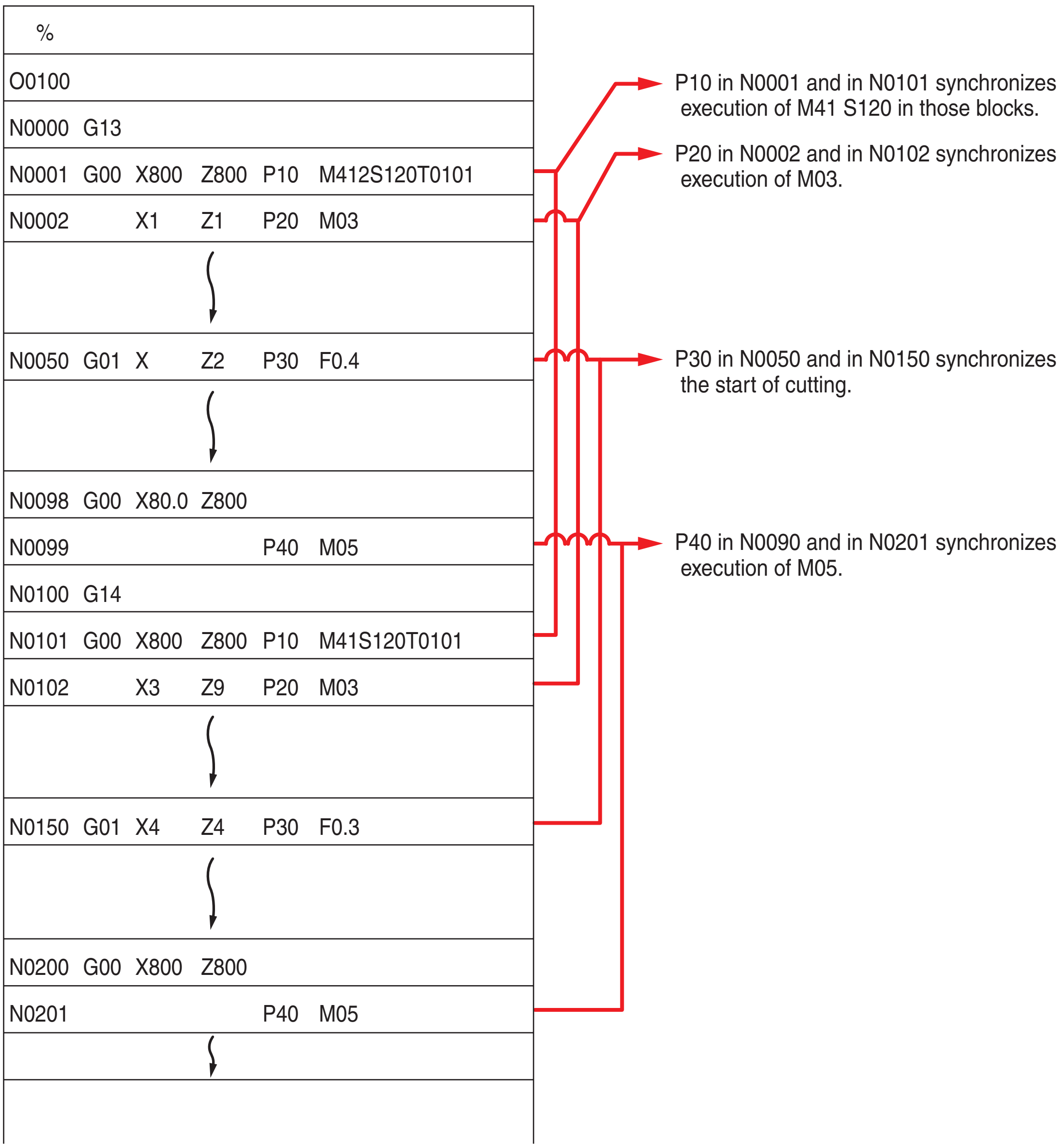


LE33013R0301300050001

- During the simultaneous 4-axis control mode, S commands, M commands relating to spindle rotation (M00, M01, M03, M04, M05 and M41 through M44), and G96 calling for the constant speed cutting mode must match for turrets A and B. Otherwise, an alarm results.
- If G13 and G14 codes for turret selection are not specified, the machine fails to perform the intended operation.

- The blocks dominated by the respective G codes, G13 and G14, are continuous as a program. That is, N0101 directly follows N0049 and N0151 follows N0099. Therefore, when the S, T, and M commands in these blocks are the same as designated in N0001 and N0051, respectively, they can be omitted.

- Blocks containing S and M commands (M41 through M44, M00, M01, M03, M04 and M05) for turrets A and B, or a G96 code, must also contain P commands assigned the same number (up to four digits) to synchronize the execution of the commands in those blocks at turrets A and B. When synchronization of command execution at the two turrets is required, use the P command.
  Program example:



LE33013R0301300050002

* If the P number in block N0002 is made, for example, P200, i.e., if the P number does not match, the control first executes the commands in N0001 for turret A and those in N0101 for turret B. After that, commands for turret B assigned a P number smaller than P200 are executed, then the commands for turret A are executed from the block containing P200, i.e., N0001. Therefore, P numbers must be assigned sequentially (P10 → P20 → P30) in accordance with the order of command execution.

- It is advisable to use two or three digits for P numbers, rather than just one, to make it easier to correct programs.

  Examples: P10 instead of P1
  P20 instead of P2

3. Precautions on Programming Simultaneous 4-axis Cuts

The key to efficient simultaneous 4-axis cuts on a 2S model is performing the intended cutting in a well-balanced manner.
When programming simultaneous 4-axis cuts, observe the following points carefully:

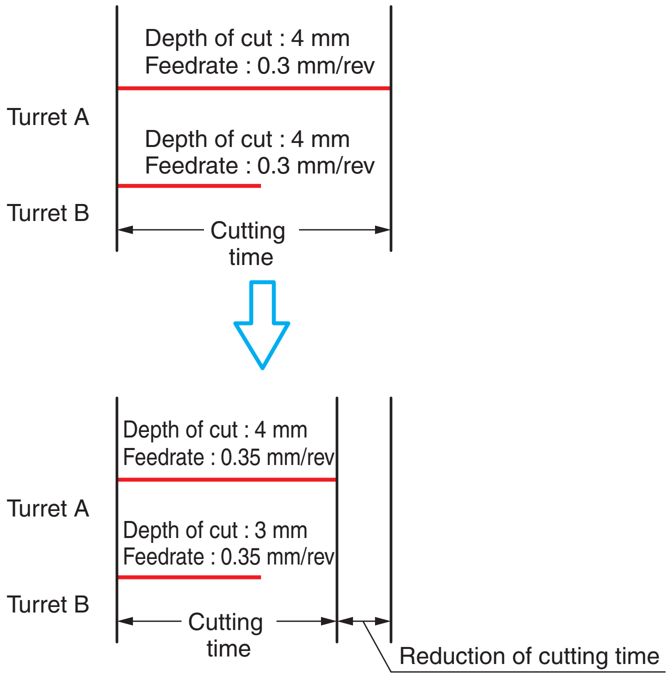
Determine the extent of operations to be performed by turrets A and B.

The cutting times required for these two turrets should be well matched when determining the sections to be cut by each.

Determine optimum cutting conditions.

- Since a spindle change cannot be performed during a simultaneous 4-axis cut, the cutting speed will vary according to the diameter being cut. Select the tip material carefully to suit the workpiece material to be cut.

- Select feedrate and depth of cut by taking the cutting at the two turrets into account:
  Example:



LE33013R0301300060001

- Determine the cutting conditions so that a total of the cutting power required by the two turrets will not exceed the capacity of the machine.

Other considerations

- The use of the INDIVIDUAL switch allows the turrets to be operated independently, facilitating checking of trial cuts.

- Care should be taken to avoid interference:

  - Interference between the boring bar and the chuck

  - When performing end face cutting with the tools on turret A:
    Interference between tools on turret A and boring bar on turret B,
    Interference between tools on turret A and ID toolholder on turret B

- Program movements of the tools on turret B taking into account those of turret A.
  G02 and G03 should also be programmed taking cutting with the tools on turret A into account.

- In constant speed cutting mode operation called for by G96, G110 and G111 select the turret on which constant cutting speed is obtained: G96 G111 calls for constant speed cutting for turret B and G96; G110 cancels G96 G111 and selects constant speed cutting mode on turret A.
  This feature generates large differences in cutting speeds for the tools on turrets A and B when, for example, performing simultaneous cutting on a workpiece with large diameter differences. Therefore, the cutting portion handled by each turret and the cutting tip material should be selected very carefully.
  Program example:

G13									
G00			X1	Z1	P10	S120	M41	M03	T0101
G96	G01	G110	X2	Z2	P20	S100			
G13									
G00			X3	Z3	P10	S120	M41	M03	T0101
G96	G01	G110	X4	Z4	P20	S100			

The G96
S, M and P commands of
turrets A and B must match.

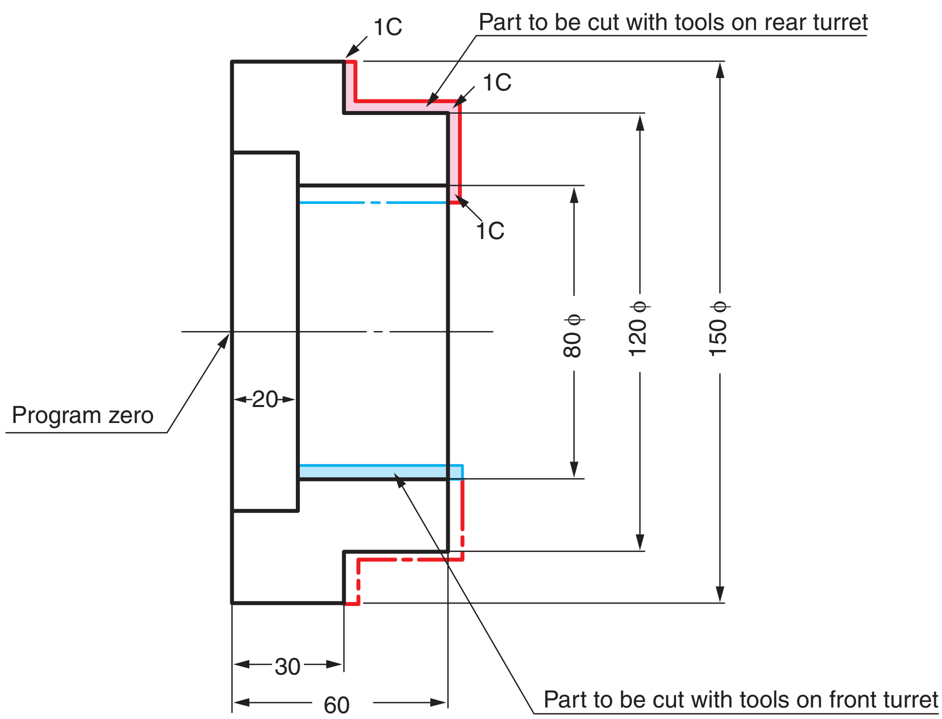
(Even when the constant speed cutting mode is in effect, it is active only on turret A and turret B is not this mode.)

LE33013R0301300060002

4. Programming Example

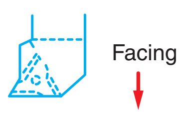
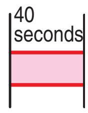
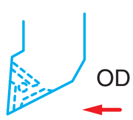
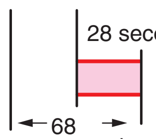
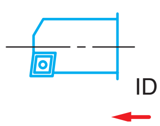
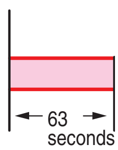
<Workpiece Dimensions>

Material : S45C (JIS, carbon steel)
Stock : 3mm (in radius)



LE33013R0301300070001

<Tooling and Cutting Conditions>

Turret	Tool No.	Cutting Tool	Cutting Conditions	Cutting Time
A	T0101	Upset tool  Facing	Cutting speed: 120 to 65 m/min. Depth of cut: 3 mm Feedrate: 0.35 mm/rev.	 40 seconds
	T0202	Upset tool  OD turning	Cutting speed: 95 m/min. Depth of cut: 3 mm Feedrate: 0.4 mm/rev.	 28 seconds 68 seconds
B	T0101	Normal tool  ID turning	Cutting speed: 65 m/min. Depth of cut: 3 mm Feedrate: 0.25 mm/rev.	 63 seconds

The net cutting time per piece is 68 seconds when the part is cut in 4-axis simultaneous cut mode. It is 131 seconds (= 68 + 63) if the part is cut in the conventional manner. This means that simultaneous 4-axis cut yields nearly a 48% saving on cutting time.

4-1. Program Process Sheet

The program below performs simultaneous end face cutting and OD turning by turret A and ID turning by turret B.

%						
O100						
N000	G13					
N001	G00	X800	Z70	P10	M42 S250 T0101 M03	
N002		X132	Z60		M08	
N003	G01	X78		F0.35		
N004	G00	X156	Z63			
N005			Z29			
N006	G01	X150				
N007		X148	Z30			
N008		X128				
N009	G00	X800	Z70			
N010		X112	Z63		Z0202	
N011	G01	X120	Z59	F0.4		
N012			Z30			
N013		X130				
N014	G00	X800	Z70			
N015				P20	M09M05	
N100	G14					
N101	G00	X800	Z200	P10	M42 S250 T0101 M03	
N102		X92	Z65		M08	
N103	G01	X80	Z59	F0.25		
N104			Z18			
N105	G00	X78	Z100			
N106		X800	Z1000			
N999					M02	

Program name (O100)

Selection of turret A (N000)

End face cutting with the tool on turret A (N001–N009)

OD turning with the tool on turret A (N010–N014)

Since no P20 command is presented in a program executed by the tool on turret B, this block is executed by turret A only. (N015)

Selection of turret B (N100)

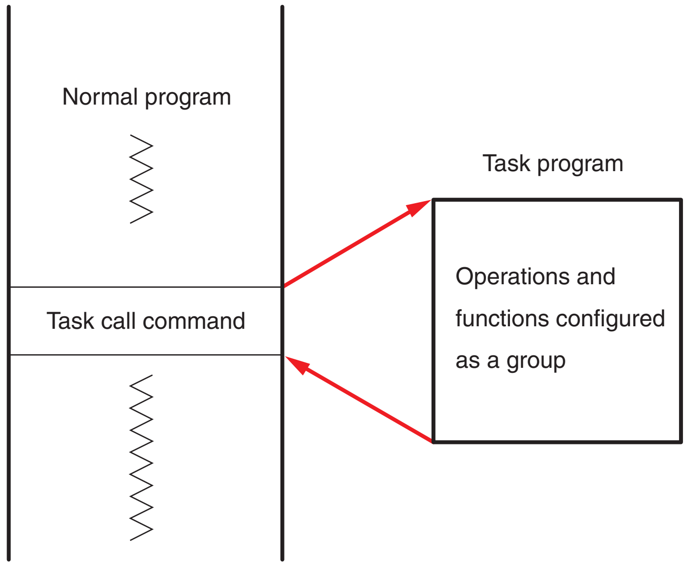
ID turning with the tool on turret B (N101–N106)

LE33013R0301300080001

SECTION 12 USER TASK

1. Overview

Operations and functions constructed as one group of instructions are stored in the memory when assigned a program name like a subprogram. The stored subprogram can be accessed from the main program by specifying the program name, which represents a group of instructions, and the operations and functions in that program can be executed.
A group of operations and functions stored in this way is called a User Task Program and the command to call it is called a User Task Call Command.



LE33013R0301400010001

The biggest advantage of the user task function is that various operation functions and variables can be used in a user task program. In addition, the use of control statements assures versatility.
There are many fields where user task functions can be used to great advantage; among them, the following:

- Parts for which the same contour is repeatedly specified during cutting, such as pulleys
- Gears and flanges which have similar contours
  Common and similar contour elements of parts to be cut are picked up using Group Technology and these elements are expressed using variables. The user task program is created using the variables, while the actual dimensions of a specific part to be cut are provided in a cutting (or main) program. Thus parts with similar contours can be machined using one user task program.
- Automatic cycles involving peripheral equipment and/or functions
  Instructions necessary to interlock the machine cycle with a bar feeder or loader cycle, or work loader/unloader, work gauging cycle commands, instructions for machine operations interlocked with a robot or other peripheral equipment are programmed as user task programs.
  When using peripheral equipment and functions like these, the user can perform specialized functions or operations by using the user task function.

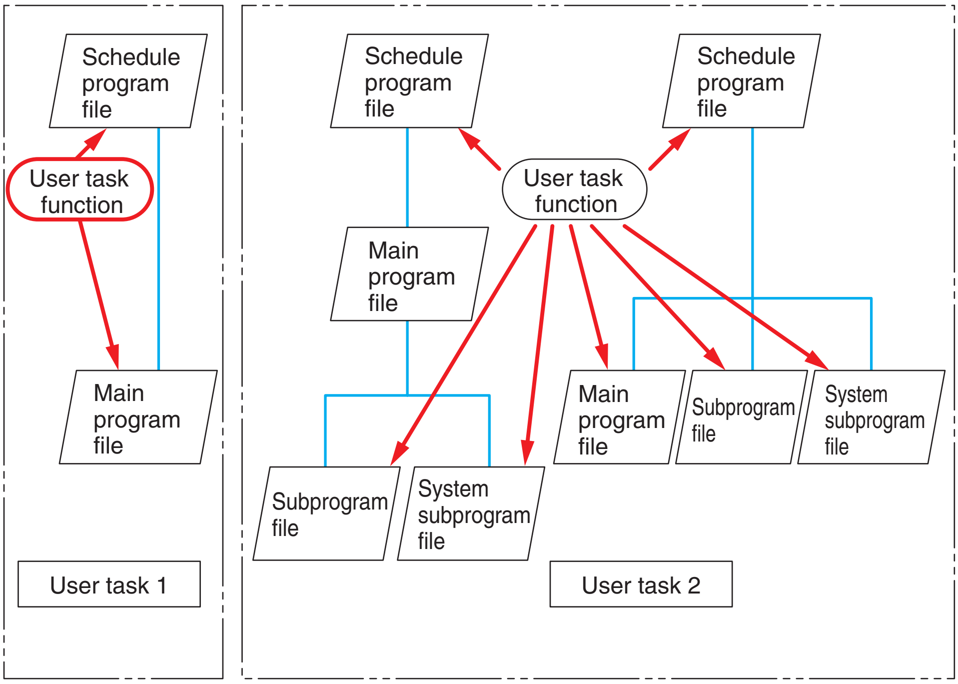
- Parts with similar contours
  When dimensions of points where circular arcs intersect or a circular arc and a tapered segment intersect each other are not indicated on a part drawing but can be calculated with a number of expressions, a user task program for these parts can be programmed using expressions.

- Special fixed or custom cycles for users
  As mentioned above, effective use of the user task function can enable various operations and functions. In addition, complicated programs can be greatly simplified with this function, assuring accurate and quick programming.

2. Types of User Task Function

2-1. Relationship Between Types of Program Files and User Task Functions

The relationship between the types of program files and user task functions is summarized below.



LE33013R0301400020001

- Programs in left hand frame (User Task 1)
  The configuration here comprises a schedule program and main programs; the User Task function can be used with these two types of programs. The User Task function applicable in such a configuration is called "User Task 1".

- Programs in the right hand frame (User Task 2)
  The configuration here comprises all types of program files and the User Task function can be used in every type of program. The User Task function applicable in such a configuration is called "User Task 2".

2-2. Comparison of User Task 1 and User Task 2

Function and Contents		User Task 1	User Task 2
Usable programs		Main program Schedule program	Main program Subprogram Schedule program System subprogram
Control statement function		'GOTO statement' 'IF statement'	'GOTO statement' 'IF statement' 'CALL statement' 'RTS statement' 'MODIN statement' 'MODOUT statement' 'GET/PUT statement' 'READ/WRITE statement'
Variable function		Common variables Local variables System variables	Common variables Local variables System variables I/O variables
Operation function	Calculation expression	+, -, *, / (four rules)	+, -, *, / (four rules)
	Comparison expression	LT, LE, EQ, NE, GT, GE	LT, LE, EQ, NE, GT, GE
	Boolean expression	-	OR, AND, EOR, NOT
	Functions	-	SIN, COS, TAN, ATAN, ATAN2, SQRT, ABS, BIN, BCD, ROUND, FIX, FUP, DROUND, DFIX, DFUP, MOD

LE33013R0301400030001

* Enter either a space or a tab code following the control statements indicated below.
GOTO, CALL, RTS, MODIN, MODOUT

2-3. Fundamental Functions of User Task

The basic user task functions are largely classified into the following three functions:

Control Statement Function

This function allows you to control the execution order of programmed sequences using the statements such as IF, GOTO, and CALL.

Variable Function

In normal programming, numerical data directly follows the address characters such as A through Z. This function allows you to assign variables expressed by alphanumerics, instead of numerical data, to the address characters. The actual numerical data are assigned to the variables in respective programs. The variable function, thus, provides versatility and flexibility in programming.
Example:

Normal programming	When calculation function is used	
X135	X = XP1	XP1 = 135
	Variable name (alphanumeric)	Desired value can be assigned

LE33013R0301400040001

Calculation Function

This function allows you to directly program the arithmetic expressions including operators (such as +, -, x, /) instead of values or variables (see the above variable function) with the address characters (such as X, Z, I, K).
Example:

Normal programming	When calculation function is used	
X135	X = 100 + XP2	XP2 = 35

LE33013R0301400040002

3. User Task 1

The basic functions of User Task 1 (control statement function, variable function, arithmetic operation function) are described here.

3-1. Control Statement Function 1

The User Task can use the following eight control statements. Of these, the GOTO statement and the IF statement are User Task 1 functions.

* GOTO statement	User Task 1	User Task 2
* IF statement		
* RTS statement		
* CALL statement		
* MODIN statement		
* MODOUT statement		
* GET/PUT statement		
* READ/WRITE statement		

LE33013R0301400060001

Program these control statements either at the beginning of a block or right after a sequence name specified at the beginning of a block (*1). Be sure to enter either a space or a tab code following the sequence name or a control statement as a delimiter. If there is no delimiter, an alarm occurs.
It is not necessary to enter a space or a tab following an IF statement since it is followed by a left bracket "[".

(*1) Sequence Names
Sequence names are codes for identifying individual blocks in a program, and they consist of four alphanumerics following address character N.
There are two types of sequence names with the following constructions:
<N> <four numerics>, and
<N> <letter> <three alphanumerics>.
The term "sequence name" as used in this manual refers to both types of sequence name.

Example: N1001 GOTO ⌴ N2000

Enter a space or tab code.

NLAP1 ⌴ GOTO ⌴ NLAP2

LE33013R0301400060002

<u>In other words, any element consisting of more than one address character (A through Z) such as a sequence name and a control code must be followed by either a space or a tab code.</u>
Next we will consider the GOTO statement and IF statement, which are the control statements of User Task 1. For details on the control statements of User Task 2, refer to "Control Functions 2".

3-1-1. GOTO Statement (Unconditional Branch)

[Programming format]

N0 ␣ GOTO ␣ N1

- N0: Sequence name of this block (can be omitted)
- GOTO: Indicates a "GOTO" statement.
- N1: Indicates the sequence name of the block that is the jump destination (mandatory).

LE33013R0301400070001

[Function]
Program execution jumps unconditionally to the block indicated by N1 and that block is executed.
[Details]
The N1 block must be in the same program as the block containing the control statement.

3-1-2. IF Statement (Conditional Branch)

[Programming format]

(1)

N0 ␣ IF <Conditional expression> GOTO ␣ N1

- N0: Sequence name of this block (can be omitted)
- IF: Indicates an "IF" statement
- <Conditional expression>: There are two possible results of the comparison operation; "true" and "false". The conditional expression must be enclosed in square brackets.
- GOTO: GOTO statement command to be executed if the conditional expression is true (can be omitted)
- N1: Sequence number of the block to be the jump destination if the conditional expression is true. Mandatory.

(2)

N0 ␣ IF ␣ <Local variable> ␣ GOTO ␣ N1

- <Local variable>: This can be used instead of a conditional expression. Branching occurs or not depending on whether the variable is defined or not.
- GOTO: GOTO statement command to be executed if the variable is defined (can be omitted)
- N1: Sequence number of the block to be the jump destination if the variable is defined. Mandatory.

LE33013R0301400080001

[Function]

- When the conditional expression is true (Example 1) or when the local variable is defined (Example 2), sequence execution jumps to sequence N1.
- When the conditional expression is false (Example1) or when the local variable is not defined (Example 2), the following sequence is executed.

[Details]
Example 1:

```
N1000 ␣ IF [V1 ␣ EQ ␣ 10] GOTO N2000
                  (*)
```

LE33013R0301400080002

or

```
N1000 ␣ IF [V1 ␣ EQ ␣ 10] N2000
```

LE33013R0301400080003

A jump is made to N2000 if variable V1 equals 10 (V1 = 10). When V1 is not equal to 10, the following block is executed.
Example 2:

```
N1000 ␣ IF ␣ ABC ␣ GOTO ␣ N2000
              Local variable
```

LE33013R0301400080004

or

```
N1000 ␣ IF ␣ ABC ␣ N2000
```

LE33013R0301400080005

If the local variable ABC has been defined, execution jumps to N2000. If not, the next block is executed.

* EQ means "equal". For details, refer to "Arithmetic Operation Function 1".

3-2. Variables

Three types of variable are used:

- Common variables
- Local variables
- System variables

These three types of variable differ in their uses and characteristics.

3-2-1. Common Variables

The term "common" in "common variables" can be literally understood as common; they can be used in common for main and subprograms. When the same variable is used in two or more programs, the variable number used in those programs must be identical. Therefore, a common variable, the result of calculation in one program, can be referred to in other programs.
[Format]

V	numerals = numerical data or expression

LE33013R0301400100001

Common variable designations consist of up to three digits following "V". The usable common variables are V1 through V200.
Examples:

```
N101 ␣ V5 = 10
N101 ␣ V5 = V5 + 1
```

LE33013R0301400100002

[Details]

- Common variables are effective both in main programs and subprograms.
- Common variables are not affected by resetting the control or turning power off. That is, the data are retained unless they are re-set or a control software is installed.
- Besides setting or changing them in a program, common variables can be set or changed by setting a parameter. For detailed information on parameter setting, refer to SECTION 4 PARAMETER SETTING, DATA OPERATION in OPERATION MANUAL.

3-2-2. Local Variables

As is apparent from the term "local", local variables are the variables that a user can set as desired with meaningful names assigned to them. Up to 127 local variables each can be used for the A and B saddles.
[Format]

Letter	Letter	two alphanumerics = Numerical data or expression

O, N and V cannot be used.

LE33013R0301400110001

Example: 'DIA1 = 100' 'ITH5 = 10'
[Details]
A local variable cannot be assigned the same name as already used for a function name, comparison operator, boolean operator, or extended address character*. (For details on function names, refer to “Arithmetic Operation Function 2”, for comparison operators, “Arithmetic Operation Function 1”, and for boolean operators, “Arithmetic Operation Function 2”.)

* Extended address characters are provided to realize LAP, pattern processing, and user-specific fixed cycles. They are necessary because there are not enough letters in the alphabet to cover the required number of extension names. The following extended address characters are currently used.
<AA> <AB> <DA> <DB> <FA> <FB> <IA> <IB> <KA> <KB>
<LA> <LB> <RA> <RB> <SA> <SB> <TA> <TB> <UA> <UB>
<WA> <WB> <XA> <XB> <ZA> <ZB> <BC>

Characteristics of Local Variables

- Local variables are cleared when the control is reset.
- When a new local variable is set in a main program, that is, when data is assigned to a new local variable name, that local variable name and corresponding data are registered in the memory.

[Supplement]

If a local variable name is used without setting any data for it, an alarm results.

- When new data is assigned to a local variable already registered with other data, that old data is updated.
  Main program

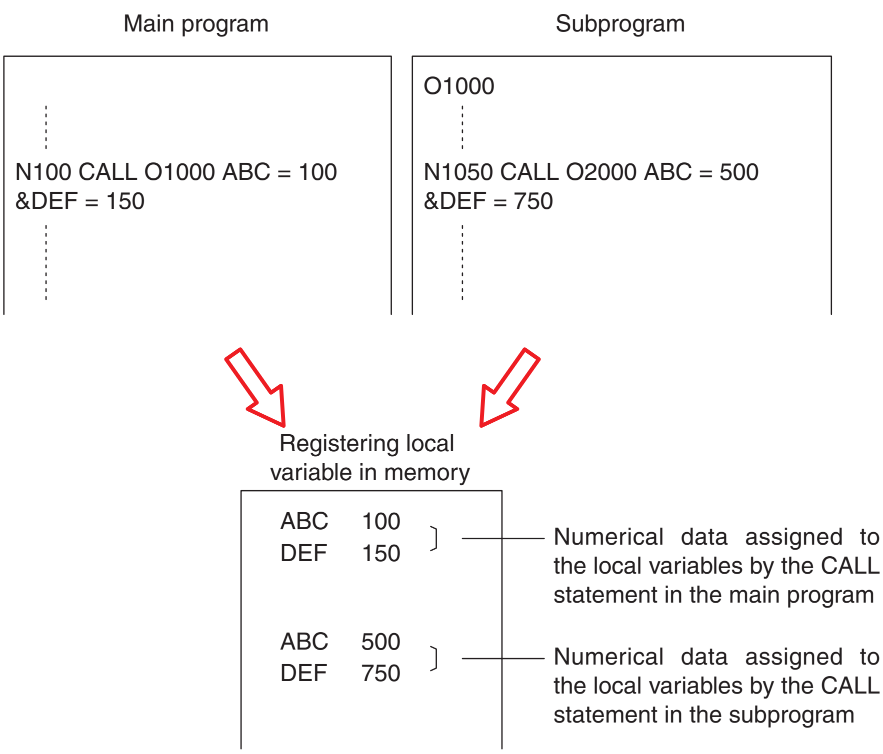
```
N0010        DIA1 = 160
  :
  :
  :
N0049
N0050        DIA1 = 20
  :
  :
  :
```

In N0010, numerical data "160" is assigned to local variable name "DIA1", and this data remains effective up to sequence N0049. In N0050, the new numerical data "200" is assigned to the same local variable name "DIA1". This clears the old data "160" and replaces it with the new data "200"

LE33013R0301400120001

- Up to 127 local variables can be used.

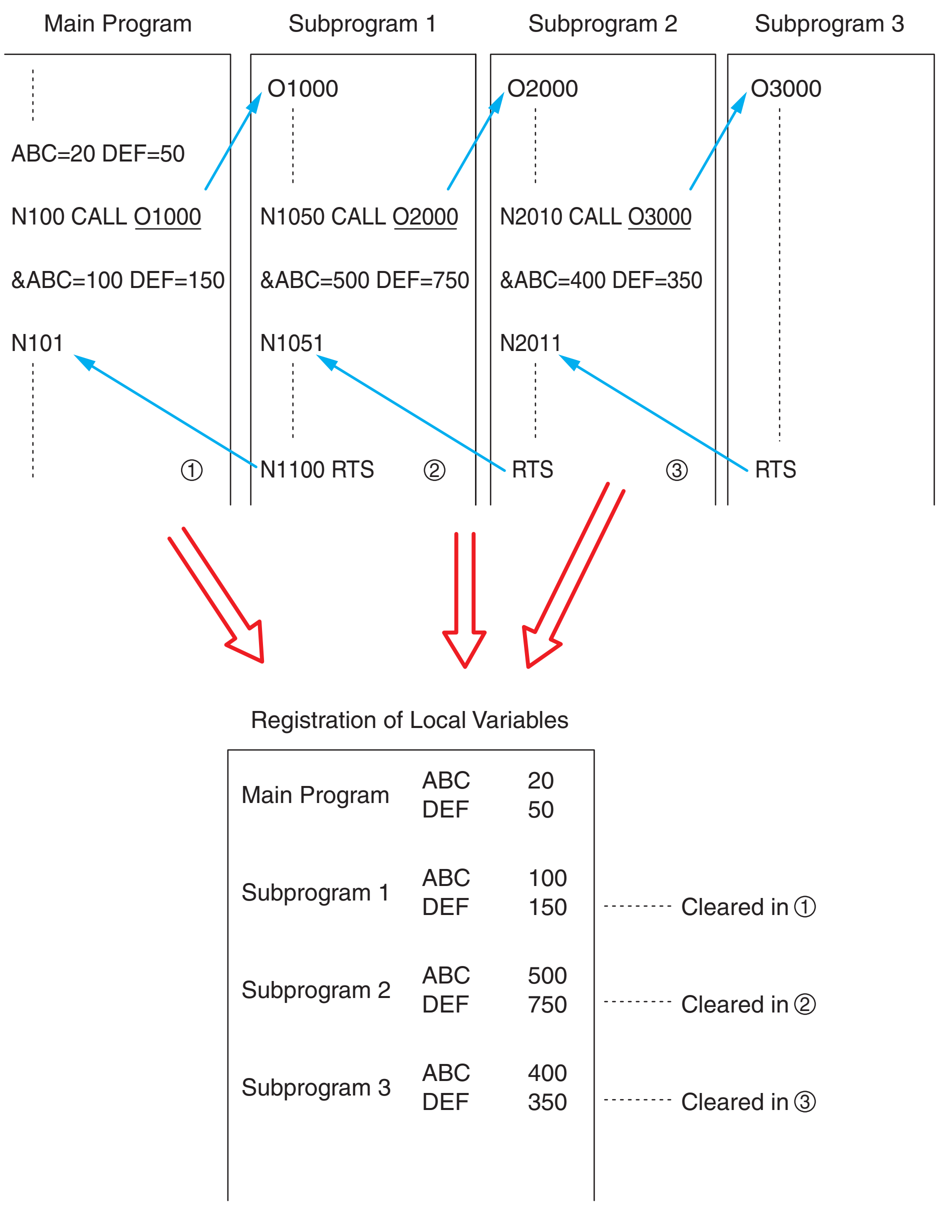
- When a subprogram call command (CALL statement) is programmed in a main program and subprogram with local variables set in the block containing the CALL statement, the variables assigned numeric values in such a block are all registered as new local variables and their numeric values are stored in memory.
  Even when a local variable has the same name as one already registered before the call statement was programmed, it is registered as a new variable.



LE33013R0301400120002

As shown above, the variables with the same name as ones already registered are registered anew as different variables.

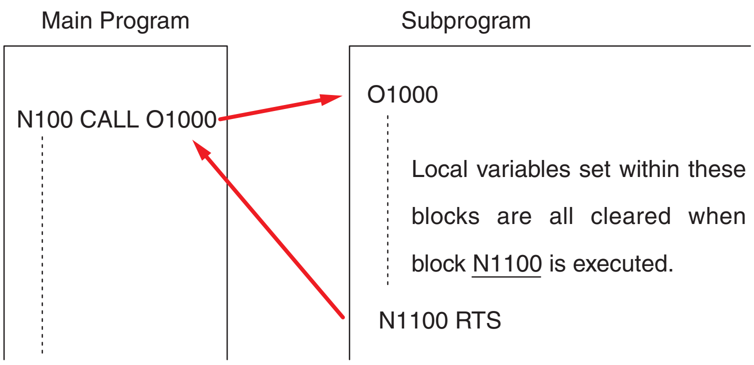
- When using local variables in a called subprogram, and there are several local variables with the same name registered in the memory, the data of the local variable which last had that name registered is used. The local variables set in the block containing the CALL statement are all cleared when the RTS statement in the called subprogram is executed.



LE33013R0301400120003

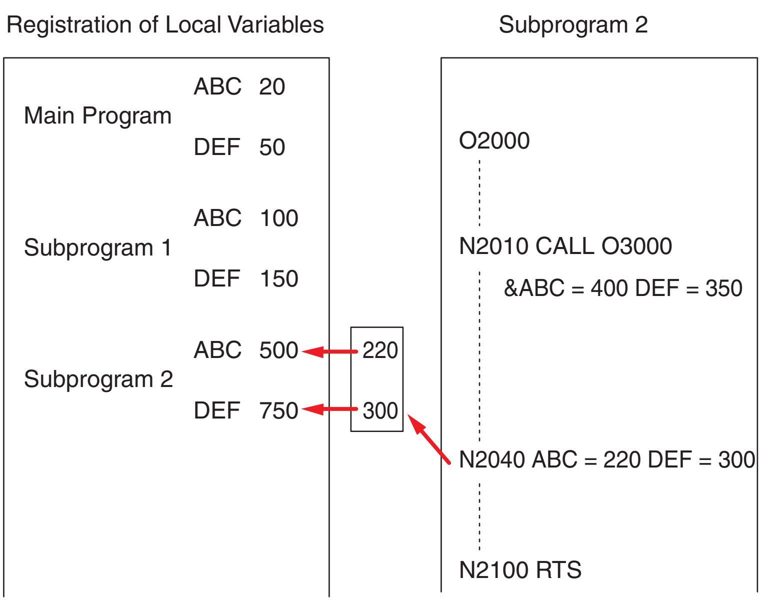
In the example above, execution of N2010 in subprogram 2 registers 4 kinds of local variables with the same name - ABC and DEF - then subprogram O3000 is executed. If subprogram O3000 contains local variable names ABC and DEF, the numeric data registered last, i.e., ABC = 400 and DEF = 350 are called for. At the end of subprogram O3000, that is, when the RTS statement in 03000 is executed, the local variables registered by the subprogram CALL O3000, ABC = 400 and DEF = 350, are cleared from the memory.

- When a local variable is newly set in a subprogram, its name and numerical data are registered in the memory. They are effective only in the subprogram in which they are set, and are cleared when the RTS statement in that subprogram is executed.



LE33013R0301400120004

- When numerical data is assigned to a local variable name which has already been assigned other numerical data during execution of a subprogram, the numerical data is updated. If several local variables with the same name are registered in the memory, the numerical data of the local variable registered last is updated.



LE33013R0301400120005

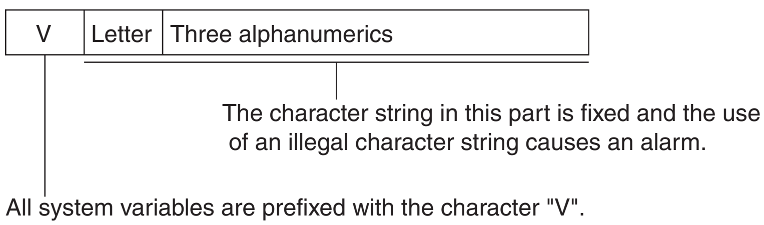
When N2010 in subprogram 2 is executed, local variables ABC = 400 and DEF = 350 are registered in the memory but they are cleared by executing RTS in subprogram 2. Therefore, in blocks prior to N2040, the variables registered when subprogram 2 was called are used. When block N2040 is executed, the numerical values of local variables ABC and DEF registered in subprogram 2 are updated to 220 and 300, respectively, and those registered in subprogram 1 and the main program are not updated.

3-2-3. System Variables

A system variable is a variable specific to a particular system and its name is fixed.
The system variables are not cleared when the control is reset.
The system variables available are:

- Zero offset variable
- Zero shift variable
- Tool offset variable
- Nose radius compensation variable
- Tool interference data variable
- Variable soft limit variable
- Chuck barrier variable
- Droop variable
- Tailstock barrier variable
- Pitch error compensation variable
- User restart variable
- Alarm comment variable

These variables can be set, changed, and used in a program according to the format described later. Therefore, they can be effectively used in programs requiring them, such as work gauging programs, tool gauging programs, and post-process gauging programs.
They can of course be set by selecting the ZERO SET, TOOL DATA or PARAMETER mode. For details of the setting procedure, refer to SECTION 4 PARAMETER SETTING, OPERATION in OPERATION MANUAL.
Details of system variables are given below:
[Basic program format for system variables]



LE33013R0301400130001

Zero offset variables

VZOFZ ········ Zero OFfset of Z-axis
Zero OFf set Z-axis
VZOFX ········ Zero OFfset of X-axis
VZOFC ········ Zero OFfset of C-axis (for multi-machining model)

LE33013R0301400140001

Set variables in the following manner: VZOFZ = 12364.256.

Zero shift variables

VZSHZ ········ Zero SHift of Z-axis
Zero SHift Z-axis
VZSHX ········ Zero SHift of X-axis
VZSHC ········ Zero SHift of C-axis (for multi-machining model)

LE33013R0301400150001

Set variables in the following manner: VZSHZ = 50.
These zero shift variables deal with shift amount set by the zero shift operation called by G50, and the set shift amount is cleared when the control is reset.

Tool offset variables

VTOFZ [Tool Offset No.] ········ Tool OFfset of Z-axis
Tool OFfset Z-axis
VTOFX [Tool Offset No.] ········ Tool OFfset of X-axis

LE33013R0301400160001

Set variables in the following manner: VTOFZ [5] = 2.634.
This indicates that tool offset amount of the Z-axis for #5 is set 2.634.

Nose radius compensation variables

VNSRZ [Nose R Compensation No.] ········ NoSe Radius compensation of Z-axis
NoSe Radius Z-axis
VNSRX [Nose R Compensation No.] ········ NoSe Radius compensation of X-axis

LE33013R0301400170001

Set variables in the following manner: VNSRZ [4] = 0.8
This indicates that the nose radius (on the Z-axis) of the tool assigned nose radius compensation no. 4 is set to 0.8 mm.

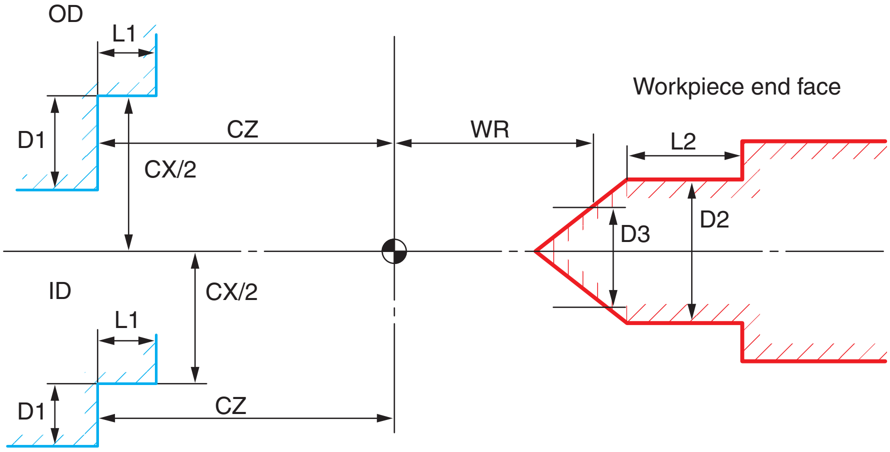
Variable soft limit variables

VPVLZ ········ PositiVe Limit on Z-axis
PositiVe Limit Z-axis
VPVLX ········ PositiVe Limit on X-axis
VNVLZ ········ NegatiVe Limit on Z-axis
NegatiVe Limit Z-axis
VNVLX ········ NegatiVe Limit on X-axis

LE33013R0301400180001

Set variables in the following manner: VPVLZ = 2352.168.
This indicates that the variable soft limit of Z-axis in positive direction is set at Z = 2352.168 mm.
The numerical data of these variables are referenced to the origin of the machine coordinate system (machine origin).

Chuck barrier/tailstock barrier variables



VCHKL ·············Jaw dimension L1
VCHKD ············Jaw dimension D1
VCHKZ ·············Jaw position CZ
VCHKX ············Jaw position CX
VTSL ·············Center dimension L2
VTSDA ············Center dimension D2
VTSDB ·············Center position D3
VWKR ············Workpiece end face position WR

LE33013R0301400190001

The numerical data of these variables are referenced to the origin of the program coordinate system (programming zero).

Droop variables

VINPZ ········ Droop amount on Z-axis IN Position Z-axis
IN Position Z-axis
VINPX ········ Droop amount on X-axis IN Position X-axis
VINPC ········ Droop amount on C-axis IN Position C-axis

LE33013R0301400200001

Pitch error compensation variables

These variables are only effective when the pitch error compensation specification function is featured.

VPFVZ ········ Pitch compensation amount on Z-axis Pitch Fillup Value Z-axis
Pitch Fillup Value Z-axis
VPFVX ········ Pitch compensation amount on X-axis Pitch Fillup Value X-axis
VPFVC ········ Pitch compensation amount on C-axis Pitch Fillup Value C-axis
VPCHZ ········ Pitch amount on Z-axis PitCH Z-axis
PitCH Z-axis
VPCHX ········ Pitch amount on X-axis PitCH X-axis

LE33013R0301400210001

Restart variables

VRSTT ········ Indicates the restart state in sequence restart operation.
#00 : Restart in progress
#80 : Restart not in progress
ReSTarT

Example: N100 IF [VRSTT ␣ NE ␣ 0] ␣ N200

LE33013R0301400220001

Alarm comment variables

VUACM ········ User alarm comments of up to 16 characters can be designated.
VUACM[1] ~ VUACM[16]
This variable is cleared when the NC is reset.
User Alarm CoMment

LE33013R0301400230001

For the alarm variable, a character-string or a hexadecimal code (prefixed by the $ symbol) in quotation marks (' ') can be used. Letters of the alphabet (upper and lower case) can be used for the character string. For the procedure refer to 4-1 Control Function 2 "GET/PUT Statement".
Up to four characters of hexadecimal code can be set.
Output of a comment to the display screen is designated with output variable VDOUT [*] =code number. See below.

* = 991: 3202 Alarm C
* = 992: 2288 Alarm B
* = 993: 1213 Alarm A

The valid range for code numbers is 1 to 9999.

Program example 1:

```
N202    VUACM [1] = ' ABC = 100' ········Corresponds to ABC=100
N203    VUACM [9] = ' ^ ABC'     ········abc
N204    VUACM [12] = ' =200'     ········=200
N205    VDOUT [991] = 999
```

LE33013R0301400230002

After the execution of the program above, an alarm with a comment can be generated in N205.

```
Screen display   3202   Alarm C   User reserve code   999
        ABC = 100     abc = 200
```

LE33013R0301400230003

Program example 2:

```
N202    VUACM [1] = ' ABC'
N203    VUACM [5] = ' = 123'
N204    VDOUT [991] = 999
   :
```

LE33013R0301400230004

When the program above is executed, only ABC is displayed as a comment.
Set a comment without placing a space between comment characters. In the above example, since three characters are set at VUACM[1], the fourth and the following characters should be set at VUACM [4].

Program example 3:

```
N101    VUACM [1] = ' -L ^ K]'    ········corresponds to PART
N102    VUACM [5] = ' = GEAR'     ········=GEAR
N103    VDOUT [992] = 1000
  :
```

LE33013R0301400230005

After the execution of the program above, an alarm with a comment can be generated in N103.

```
Screen display  2288   Alarm B   User reserve code   2000   ABCD
                PART = GEAR
```

LE33013R0301400230006

Program example 4:

```
N301    VUACM[1] = $41424344    ········Corresponds to ABCD
N302    VDOUT[992] = 2000
```

LE33013R0301400230007

```
Screen display  2288   Alarm B   User reserve code   2000   ABCD
```

LE33013R0301400230008

Arithmetic variable

VPAI ········ π (Ratio of circumference of circle to its diameter)

LE33013R0301400240001

NOEX command

Designated at the beginning of a variable setting sequence to speed up the "program check" by eliminating single-block processing. (The operation is the same whether this command is designated or not.)
The NOEX command is effective only in the single block mode operation with "1" set at NOEX command ignore of optional parameter (OTHER FUNCTION 1).

3-2-4. I/O read variables

The I/O read variables are the system variables used to read the status of panel inputs and outputs or EC inputs and outputs. These system variables are read-only.
[Command format]

- Reading the input bit

VIRD [∗∗∗#]
∗∗∗ : Logical I/O address
: Bit position

LE33013R0301400260001

- Reading the output bit

VORD [∗∗∗#]
∗∗∗ : (Logical I/O address) - (512)
: Bit position

LE33013R0301400260002

* In the read data, bit ON becomes “1” and bit OFF becomes “0”.

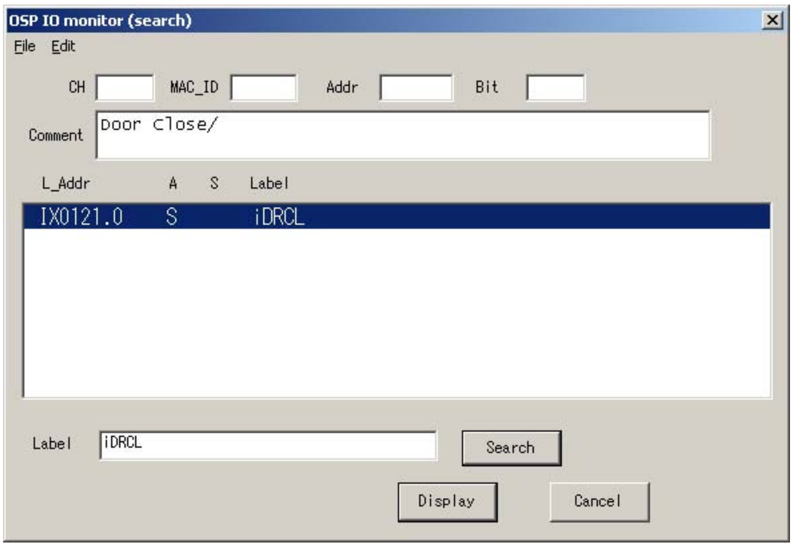
<Method of obtaining logical I/O address>

Reading an input status

Procedure :

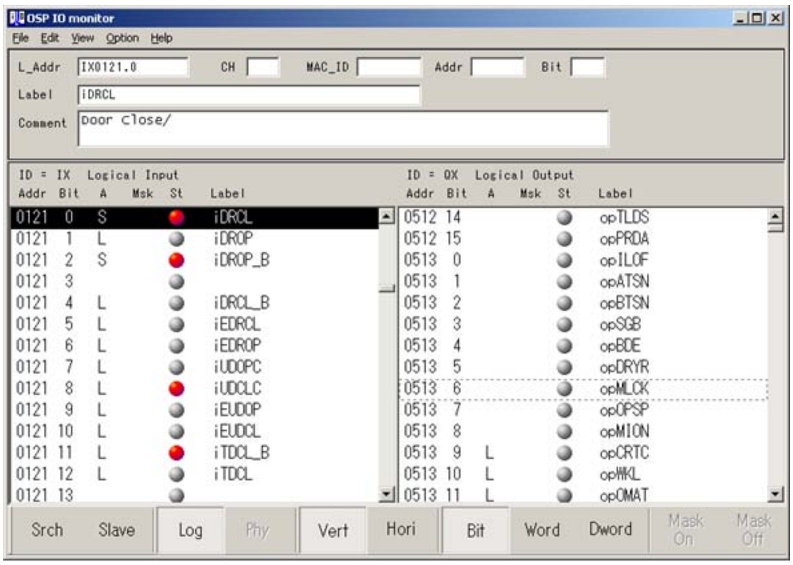
1 Search for the I/O signal that you want to refer to from the I/O bit table, and check its label. If, for example, you want to check "Door close confirmation", find the label "iDRCL".

2 Activate the I/O monitor and press the function key "Srch" to display the following screen. Enter the found label name "iDRCL" at Label, and then press "Search" or "Display" button



LE33013R0301400260003

3 When the searching is completed, the following screen appears.
The address IX0121.0 indicated at L_addr shows that the logical address is 121 and its bit number is 0. This means that you can read the door close confirmation status by specifying VIRD[1210].



LE33013R0301400260004

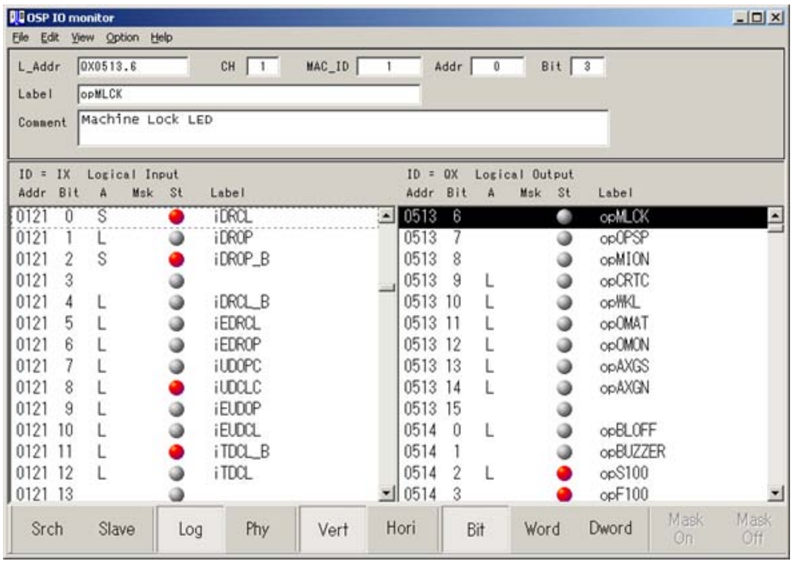
Reading an output status

Procedure :

1 Search for the I/O signal that you want to refer to from the I/O bit table, and check its label.
If, for example, you want to check "Machine lock", find the label "opMLCK".

2 Search for "opMLCK" with the I/O monitor in the same manner as in reading an input status.

3 When the searching is completed, the following screen appears.
The address QX0513.6 indicated at L_addr shows that the logical address is 513 and its bit number is 6. Specify the VORD command by entering the value obtained by subtracting 512 from the logical address. That is, you can read the machine lock status by specifying VORD[0016].



LE33013R0301400260005

3-3. Arithmetic Operation Function 1

This function allows arithmetic operation using variables. The programming can be done in the same way as for general arithmetic expressions.
[Program format]
Address character, Extended address character, Variable = Expression
The expression on the right-hand side, requesting an arithmetic operation, is made up of constants, variables, comparison expressions, and operators.
The arithmetic and comparison expressions are described below.

Arithmetic Expression

Operator	Meaning	Example
+ -	Positive sign Negative sign	+1234 -1234
+ -	Sum (addition) Difference (subtraction)	X = 12.3 + V1 X = 12.3 - V1
* /	Product (multiplication) Quotient (division)	X = V10 * 10 X = V11/10

Comparison Expression

<table>
<tr><th>Operator</th><th>Meaning</th><th>Example</th><th>Contents</th><th>Rule</th></tr>
<tr><td>LT</td><td>(Less Than, <)</td><td>IF [V1 LT 5] N100</td><td>Jump to N100 when V1 is less than 5.</td><td rowspan="6">Provide a space on either side of the operator.</td></tr>
<tr><td>LE</td><td>(Less than or Equal to, ≤)</td><td>IF [V1 LE 5] N100</td><td>Jump to N100 when V1 is less than or equal to 5.</td></tr>
<tr><td>EQ</td><td>(EQual to, =)</td><td>IF [V1 EQ 5] N100</td><td>Jump to N100 when V1 is equal to 5.</td></tr>
<tr><td>NE</td><td>(Not Equal to, ≠)</td><td>IF [V1 NE 5] N100</td><td>Jump to N100 when V1 is not equal to 5.</td></tr>
<tr><td>GT</td><td>(Greater Than, >)</td><td>IF [V1 GT 5] N100</td><td>Jump to N100 when V1 is greater than 5.</td></tr>
<tr><td>GE</td><td>(Greater than or Equal, ≥)</td><td>IF [V1 GE 5] N100</td><td>Jump to N100 when V1 is greater than or equal to 5.</td></tr>
</table>

4. User Task 2

User Task 2 allows the use of more functions than are provided by User Task 1, including I/O variables, boolean operations, function operations, and control statements such as the CALL statement, MODIN/MODOUT statements, and PUT/GET statements.

4-1. Control Functions 2

The control statements available under User Task 2 - CALL statement, RTS statement, MODIN statement, MODOUT statement, READ/WRITE statement and GET/PUT statement - are described in this section.
Note that a space or tab code must always be inserted after a control statement.

4-1-1. CALL Statement - Calling Program

[Program format]

N0 CALL ␣ O1 Q1 <Variable setting section>

- <Variable setting section>: Set the variables to be used in the called subprogram.
- Q1: Indicates the number of repetitions of the subprogram to be called (1 to 9999 following address character Q)
- O1: Program name of the subprogram to be called. Mandatory.
- CALL: Designates a CALL statement
- N0: Sequence name of this block (can be omitted)

LE33013R0301400300001

[Function]
The subprogram designated by <O1> is called and executed. When variables are set in the variable setting section, all of them are registered.
Example:

```
N1000   CALL   O1234   XP1 = 150   ZP1 = 100
```

LE33013R0301400300002

With the designated commands above, the subprogram O1234 is called and executed. At the same time, variables XP1 and ZP1 are registered.

4-1-2. RTS Statement - End of Subprogram

[Program Format]

N0 RTS

Indicates an RTS statement

The sequence name of this block (can be omitted)

LE33013R0301400310001

[Function]
This RTS statement must always be specified at the end of a subprogram. Executing the RTS block ends the called subprogram and the execution sequence jumps to the block right after the one containing the CALL statement. The variables registered in the block containing the CALL statement and the variables in the subprogram are all cleared.

Judging defined/undefined state of local variables with an IF statement

N2 ␣ IF ␣ Local variable name ␣ GOTO ␣ N3

Specify here the block sequence name that is the jump destination if the local variable is found to be defined.

This can be omitted

Name of the local variable whose defined/undefined status is under investigation
There is no need to enclose the local variable name in square brackets.

Designates an IF statement

Sequence name of this block

LE33013R0301400320001

This checks whether the local variable name designated is defined or not. A jump is made to the designated sequence name, N3, if it is defined; if it is not defined the block which follows this N2 block will be executed.
Example 1:

N1000 ␣ IF ABC ␣ N2000

LE33013R0301400320002

If local variable ABC has been defined, sequence execution jumps to N2000. If it has not been defined, the next block is executed.

Example 2: Main Program

```
N1000     CALL    O1234   XP1 = 150   ZP1 = 10
N1001
     :
     :
       Subprogram
            O1234
            N001       G00      XZ
               :          :
               :          :
            N050       RTS
```

LE33013R0301400320003

When block N1000 in the main program is executed, sequence execution jumps to subprogram O1234.
The subprogram is executed from N001 and when the control reads the RTS statement in N050, sequence execution then jumps back to N1001 of the main program and the commands in that block and subsequent blocks are executed. At the same time as the jump from the subprogram to the main program occurs, variables XP1 and ZP1 are cleared.
Order of program execution:

N1000 → O1234 → N001 → N050 → N1001 ···

LE33013R0301400320004

4-1-3. MODIN Statement

[Program Format]

N0 MODIN ␣ O1 Q1 <Variable setting section>

- <Variable setting section>: Set the variables to be used in the called subprogram.
- Q1: Indicates the number of repetitions of the subprogram to be called (1 to 9999 following address character Q; assumed to be Q1 if not specified)
- O1: Program name of the subprogram to be called. Mandatory.
- MODIN: Designates a MODIN statement
- N0: Sequence name of this block (can be omitted)

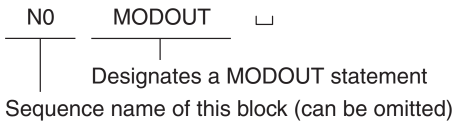
LE33013R0301400330001

[Function]
The designated subprogram is called each time an axis motion command is executed. That is, the designated subprogram is called and executed each time an axis motion command (*1) in the program calling that subprogram is executed. This function remains active until the MODOUT statement described in (4) is read.

*1 Axis motion command means any command in G00, G01, G03 and G31 through G35 mode which contains either an or X and Z word or both.

4-1-4. MODOUT Statement

[Program Format]

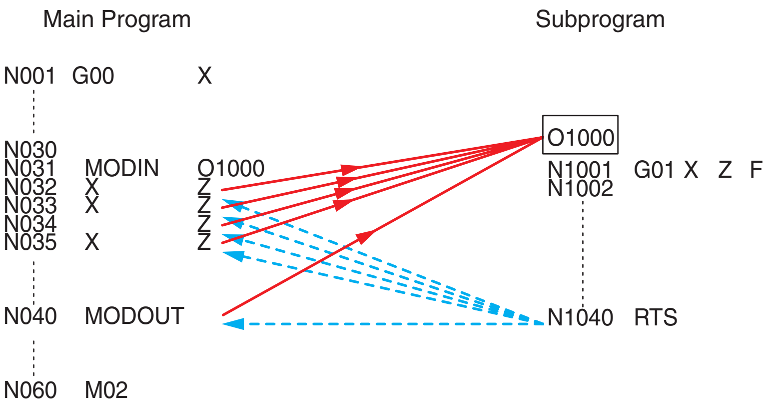


LE33013R0301400340001

[Function]
This is the statement to cancel the MODIN mode.

- The MODIN mode must be canceled by a MODOUT statement designated in the same program. That is, the MODIN mode activated in the main program cannot be canceled by a MODOUT statement in the subprogram, and a MODIN mode active in the subprogram cannot be canceled by a MODOUT statement in the main program.
- The maximum number of subprograms usable in a single program is 125.

[Program Example]
Example of a program with MODIN and MODOUT statements



LE33013R0301400340002

The program is started from N001 of the main program and the commands up to N030 are executed in the normal manner. On execution of the commands in N031, subprogram O1000 is called in the MODIN mode. However, the subprogram is not executed in this block. When the axis motion commands in block N032 are completed, subprogram O1000 is called and executed up to N1040 in that subprogram. The RTS statement causes a jump to the main program and the next in the main program, N033, is then executed. The same step is repeated up to block N039 in the main program. The MODOUT statement in N040 cancels the MODIN mode and the commands in the blocks after N041 are executed in the normal manner.

Nesting and effective range of MODIN/MODOUT mode

The permissible nesting level in MODIN/MODOUT mode is eight.
Example: Two nesting levels

Main Program

```
N001    MODIN     O1000
  :
  :
N010    MODIN     O2000
N011    G00       XZ
N012      :
          :
N020    MODOUT
  :       :
  :       :
N030    MODOUT
```

Subprogram

```
         O1000
N1001    G00      X    Z
  :
  :
N1010    RTS
         O2000
N2001    G00      X    Z
  :
  :
N2010    RTS
```

LE33013R0301400350001

In this example, the MODIN mode is active from N001 to N030 for subprogram O1000 and from N010 to N020 for subprogram O2000.

Operation Sequence:

(1) In the blocks from N001 to N009, subprogram O1000 is called and executed each time an axis motion command is executed.

(2) In the blocks from N010 to N020, subprogram O1000 is called and executed first after an axis motion command is executed. Then, subprogram O2000 is called and executed successively. If the subprogram O2000 contains an axis motion command, N2001 in this example, O1000 is executed after this axis motion command is completed. After the subprogram O2000 is completed, the block of commands in the main program is executed.

(3) In the blocks from N021 to N030, subprogram O1000 is called and executed each time an axis motion command is executed.

4-1-5. READ/WRITE Statement

READ and WRITE statements are used for communications with external devices through the RS232C interface.
They are used in conjunction with the GET/PUT statement described in (6) below.
[Program format]

```
N0      READ  ␣  n
                 Channel designation for RS232C interface
                     n  =0·······CN0 :   or TT :
                        =1·······CN1 :
                        =2·······CN2 :
                        =3·······CN3 :
                     (Default: n = 0)
        Designates a READ statement
Sequence name of this block (can be omitted)

N0      WRITE ␣  n
                 Channel designation for RS232C interface
                 (Same as for READ)
        Designates a WRITE statement
Sequence name of this block (can be omitted)
```

LE33013R0301400370001

[Function]

READ : Receives the data from the external device connected to the channel designated by <n> and stores it in the read area (max. of 160 characters) in JIS8 code.

WRITE : Sends the data stored in the write area (max. of 160 characters) in JIS8 code to the external device connected to the RS232C.

[Details]

- Transmission protocol
  The baud rate and code configuration for the channel of the RS232C interface to be used for communication with an RS232C device must be set in advance.
  These communication parameters are set on the NC optional parameter (RS232C) screen by selecting PARAMETER SET mode.

- Transmission code
  For transmission codes, JIS 8 bit code or JIS 7 bit code (in this case, using an even parity bit) are used.
  The end of data transmission code is either NULL or %. Which of these codes is used is determined by the setting at stop bit check of optional parameter (RS232C).
  (% is selected by "1" and NULL is selected by "0".)

[Supplement]

- The following situations during data transmission will cause Alarm B.
  - The number of characters of transmission data exceeds 160.
  - Transmission of data through RS232C interface stops for more than determined period of time.
  - Occurrence of an alarm in the RS232C interface during transmission.
  - RS232C is no longer in ready status.
- Do not execute list output or punch out using the same channel of the RS232C interface during the execution of the READ/WRITE command.
- Execution of the READ/WRITE command and printing out of measured data cannot be executed at the same time.
  While either is in progress, execution of the other command is suspended.
- Since the areas for READ and WRITE are different, execution of the READ command does not change the WRITE area.
- Nothing but a sequence number may be placed before the READ/WRITE command.
- When using JIS 7 bit code, designate SI code (shift in $0F) at the beginning of communication 1 and designate SO (shift out $0E) at the end. Since both SI and SO are treated as data, include them in the number of characters in transmission.

4-1-6. GET/PUT Statement

[Program format]

N0 GET ⌴ s, K

K: This indicates the number of digits to be read out from the read area where data has been stored by the READ statement (max. 10 lines). However, if s = 0, this indicates the number of characters to be skipped (max. 160 characters).

s: The variable name used to store the data in the READ area where data are stored by the READ statement (common variables, system variables, local variables, output variables)

GET: Designates a GET statement

N0: Sequence name of this block (can be omitted)

N0 PUT ⌴ s, K

K: This indicates the number of digits to be written to the write area by the WRITE statement (max. 10 lines). However, if s = 0, this indicates the number of space codes (max. 160 space codes).

s: Indicates the data to be written to the write area: variable name, character-string or hexadecimal number (common variable, system variable, local variable, input variable, "ABC", $0A)

PUT: Designates a PUT statement

N0: Sequence name of this block (can be omitted)

LE33013R0301400380001

[Function]

GET statement : This reads out the numerical data (JIS8 code) from the read area where the data has been stored by the READ statement and sets it for the variable designated.

PUT statement : This stores the numerical data and character string of the set variable in the write area output by the WRITE statement. The data is stored in the JIS8 code.

[Details]

- GET statement:
First, the code read into the READ area by the READ statement is read out; the number of characters designated in "K", counting from the position of the area read pointer (hereafter "RRP") is read out. Then this read-out data is converted into numerical values and set in the type of variable designated in "s". At this time, RRP is supplemented by the amount "K".
RRP is set at the beginning of the READ area when the READ statement is executed or the NC is reset, and it is supplemented GET is executed. It cannot be returned.
Alarm B occurs in the following cases:

  a. When RRP exceeds the number of codes read by the READ statement

  b. When conversion into numerical data is impossible
  Example: When there are more than two decimal points or code other than 0 to 9 exists

- PUT statement:
When a common variable or local variable is designated in "s", the real type is used, and when an input variable is designated, the integer type is used. For the system variables, data is converted into JIS 8 letter codes in accordance with the attribute of the system variable. Then the letter code is written to the WRITE area; the number of digits designated in "K", counting from the position of the write-in pointer of the WRITE area (hereafter "WWP") are written. At this time, WWP is supplemented by the amount "K".
WWP is set at the beginning of the WRITE area when the WRITE statement is executed or the NC is reset, and it is supplemented when PUT is executed. It cannot be returned.
A character string or hexadecimal number can be used for "s" in a PUT statement. However, the limit for the number of characters to be designated is 16. "*" must not be designated.

- Expression of characters and hexadecimal numbers

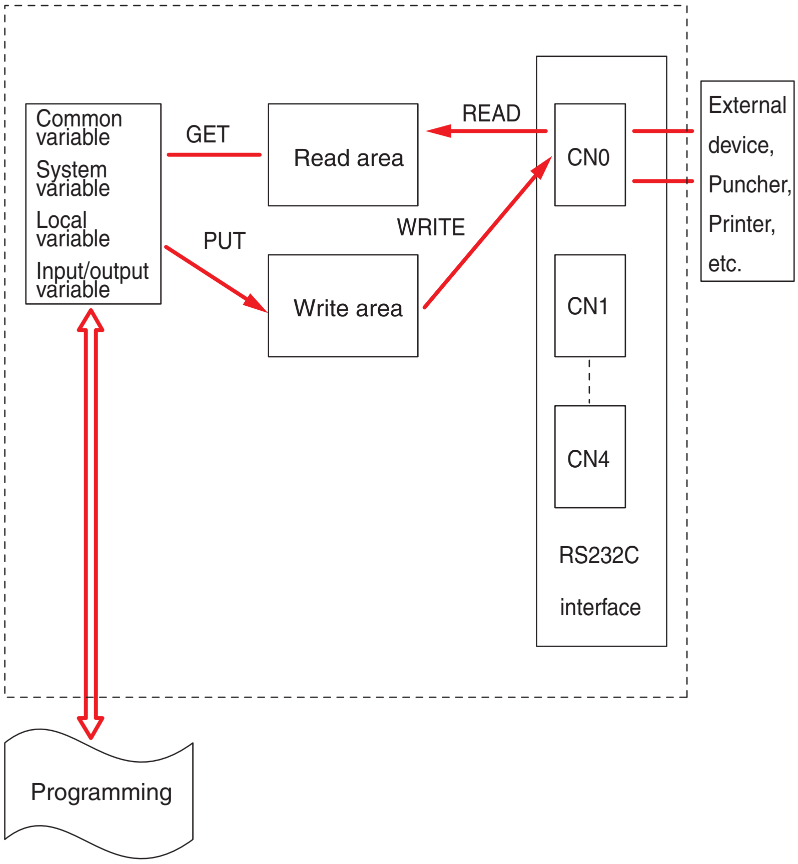
Hexadecimal number :	PUT	$0D0A	.... Prefix with $
Capital letters :	PUT	'ABCD'	.... Finish with '
Lower case letters :	PUT	'^ABCD'	.... Start with ^

LE33013R0301400380002

- Alarm B occurs in the following cases:
  - When WWP exceeds 160
  - When NULL or % exists in hexadecimal number

- Relationship between READ/WRITE and GET/PUT
Data communications are carried out through the read/write area using the variables (common variables, system variables, local variables, input/output variables) and external devices.

The concept is illustrated in the figure below.



LE33013R0301400380003

[Program examples]
Example 1. Program using READ/WRITE statements and GET/PUT statements

```
N1    READ    0·········································Data is read from CN0:.
N2    GET ␣ , 10······································Message (a) (10 letters) is skipped.
N3    GET ␣ V 1, 1·································11th letter is read into V1.
N4     IF [V1 EQ 0] N11
N5    GET ␣ , 9········································Message (b) (9 letters) is skipped.
N6    GET ␣ V2 ,2·································21st and 22nd letters are read into V2.
N7    GET ␣ , 4········································Message (c) (4 letters) is skipped.
N8    GET ␣ VTOFX [V2], 5··················27th to 31st letters are read into tool offset data.
N9    GET ␣ 4··········································Message (d) (4 letters) is skipped.
N10   GET ␣ VTOFZ [V2]······················36th to 40th letters are read into tool offset data.
N11
```

LE33013R0301400380004

Transmission data

```
1          5            10           15      20        25     30            40
A Compensation Yes/No = 1  Offset No. =  ␣ 3  ␣ OX = 0.02  ␣OZ = -0.31
|_______________________| |______________| |____________| |______________|
           (a)                  (b)              (c)             (d)
```

LE33013R0301400380005

The result of the preceding program:

V1 = 1

V2 = 3

VTOFX [3] = 0.02

VTOFZ [3] = -0.31

LE33013R0301400380006

Example 2. Output program (assuming printer is connected to CN0):

```
N100   PUT ␣ ,4..........................................4 spaces
N101   PUT   '**␣'
N102   PUT ␣ 'TOOL␣OFFSET␣DATA'..........................Header
N103   PUT ␣ '␣**'
N104   PUT ␣ $0D0A.......................................Carriage return code
N105   PUT ␣ ' N='
N106   PUT ␣ V1, 2.......................................Tool offset number
N107   PUT   '␣X='
N108   PUT ␣ VTOFX[V1], 8................................Tool offset value
N109   PUT ␣ '␣Z='
N110   PUT ␣ VTOFX[V1], 8
N111   PUT ␣ $0D0A.......................................Carriage return code
N112   WRITE ␣ 0
```

LE33013R0301400380007

The following will be printed when the program above is executed.

```
␣␣␣␣ ** ␣ TOOL ␣ OFFSET ␣ DATA ␣ **
N = ␣ 3 ␣ X = ␣␣␣ 5.412 ␣ Z = ␣␣ 14.339
```

LE33013R0301400380008

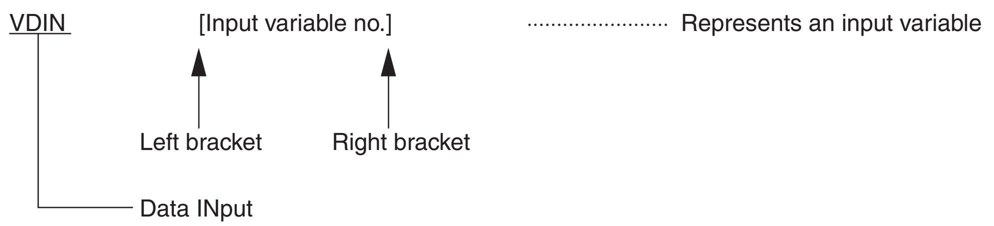
4-2. I/O Variables

I/O variables are the variables used for sending and receiving I/O signals between the control and peripheral equipment.

- Input variables:
  The variables representing signals inputted from peripheral equipment such as the operation panel, the post-process gauging unit, the tool gauging system and the touch sensor.
  These signals are called "input bit data".

- Output variables:
  The variables representing signals output from the control to the peripheral equipment such as indicator lamps and alarm display on the operation panel.
  These signals are called "output bit data".

4-2-1. Input Variables



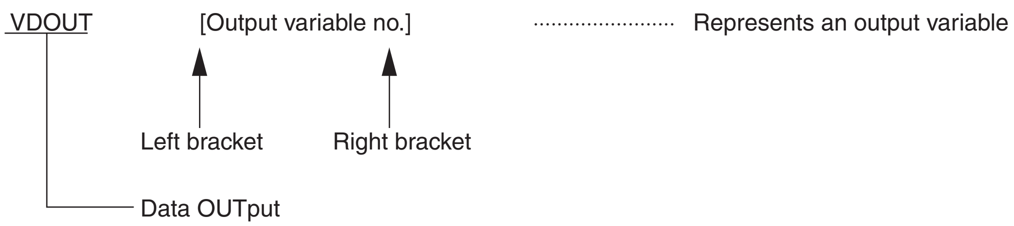
LE33013R0301400400001

The input variable numbers are tabled below.

Input Variable No.	Contents of Data	Input Equipment
1 ~ 8	Bit data: 0 (OFF), 1 (ON)	Panel input
9	1 byte data in which data of variables #1 through #8 are corresponded to bit 0 through bit 7.	
10	5 bit data of hexadecimal number, $0 through $1F	
11 ~ 18	Bit data: 0 (OFF), 1 (ON)	EC input
19	1 byte data in which data of variables #11 through #18 correspond to bit 0 through bit 7.	
21 ~ 22	Bit data: 0 (OFF), 1 (ON)	Panel input
23 ~ 24	Bit data: 0 (OFF), 1 (ON)	EC input
31 ~ 38	Bit data: 0 (OFF), 1 (ON)	Panel input
39	1 byte data in which data of variables #31 through #38 correspond to bit 0 through bit 7.	
1000 ~ 1004	Timers: 1 msec., 1 sec., 1 min., 1 hour, 1 day (timers are reset to "0" at power on and start counting up after that: 4-byte counter)	
1235 ~ 1250	Condition of specification codes: #1 through #16 (in byte units)	

Details of specifications must be discussed.

4-2-2. Output Variables



LE33013R0301400410001

The output variable numbers are tabled below.

Output Variable No.	Contents of Data	Output Equipment
1 ~ 8	Bit data: 0 (OFF), 1 (ON)	Panel output
9	1 byte data in which data of variables #1 through #8 correspond to bit 0 through bit 7.	
11 ~ 18	Bit data: 0 (OFF), 1 (ON)	EC output
19	1 byte data in which data of variables #11 through #18 correspond to bit 0 through bit 7.	
21 ~ 22	Bit data: 0 (OFF), 1 (ON)	Panel output
23 ~ 24	Bit data: 0 (OFF), 1 (ON)	EC output
31 ~ 38	Bit data: 0 (OFF), 1 (ON)	Panel output
39	1 byte data in which data of variables #31 through #38 correspond to bit 0 through bit 7.	
41 ~ 48	Bit data: 0 (OFF), 1 (ON)	
49	1 byte data in which data of variables #41 through #48 correspond to bit 0 through bit 7.	
991	Alarm C User reserve code	Alarm output
992	Alarm B User reserve code	
993	Alarm A User reserve code	

Details of specifications must be discussed.

4-3. Arithmetic Operation Function 2

The arithmetic operation function of User Task 2 provides features not supported by User Task 1; boolean expressions and functions can be used.
For details on the arithmetic operation functions of User Task 1, refer to "Arithmetic Operation Function 1".

Boolean Expressions

Operator	Meaning	Example	Rule
OR	Logical sum	VDKN [11] OR VDIN [12]	Provide a space on either side of the operator.
AND	Logical product	VDIN [11] AND VDIN [12]	
EOR	Exclusive OR	VDIN [11] EOR VDIN [12]	
NOT	Negation		

Functions

Function	Meaning	Example	Rule and Remark
SIN	Sine	X = 15 * SIN [22.5]	Numbers after function operation symbols must be enclosed in square brackets.
COS	Cosine	Z = 15 * COS [22.5]	
TAN	Tangent	Z = 15 * TAN [12.5]	
ATAN	Arctangent (1) Value range: -90 to 90	X = 15 * ATAN [22.5]	
ATAN2	Arctangent (2) Angle of point defined by coordinate value (a, b). Value range -180 to 180	ATAN2 [10.15]	When two elements are specified within square brackets, place a comma between them.
SQRT	Square root	X = 15 * SQRT [224.5]	
ABS	Absolute value	ABS [V15]	
BIN	Decimal to binary conversion	BIN [V15] 4BYTE	
BCD	Binary to decimal conversion	BCD [V20] 4BYTE	The position of the decimal point is determined in accordance with the unit system selected. The unit systems for angle commands are: 1 deg. for 1mm and 1 inch unit system 0.001 deg. for 1m unit system 0.0001 deg. for 0.0001 inch system
ROUND	Rounding off fractions	ROUND [V5]	
FIX	Cutting off fractions	FIX [V7]	
FUP	Counting fractions as a whole number	FUP [V15]	
DROUND	Rounding off fractions to three decimal places (metric system) or to four decimal places (inch system)	DROUND [V20]	
DFIX	Cutting off fractions below the third decimal place (metric system) or below the fourth decimal place (inch system)	DFIX [V20]	
DFUP	Count the figures below the third decimal place (metric syst em) or below the fourth decimal place (inch system) as a whole number	DFUP [V21]	
MOD	Remainder (a - fix[a/b]*b)	MOD [a,b]	

Combination of Operations

- The operations and functions explained in the previous page can be combined as needed.

X = V1 + V2 - V3 + V4 * COS [30]

LE33013R0301400420001

- Designating operator precedence with square brackets [ ]
  Operator precedence can be determined by using square brackets.

Example:

V1 = [V2 + V3] * V4 + [V5 - V6] / V7 + V8

1. 2.

3. 4.

5.

LE33013R0301400420002

5. Supplemental Information on User Task Programs

5-1. Sequence Return in Program Using User Task

Basically, sequence return can be performed in the same manner as in a conventional program and there are no restrictions on activating the sequence return function.
When variables are set in a block preceding the one where the sequence return is executed, the set data are all registered in the memory.

5-2. Data Types, Constants

5-2-1. Type of Data

There are three types of data: "integer type", "real type" and "logical type".

- Integer type
  Data of integer type accurately express integer values and can be zero, a positive integer or a negative integer.

- Real type
  Data of real type accurately express real values and can be zero, a positive real number or a negative real number.

- Logical type
  Data of logical type may be either true (1) or false (0).

5-2-2. Constants

There are two types of constant: "integer constant" and "real constant"

- Integer constant
  Integer constants are constants of integer type. They are expressed by up to eight digits and are interpreted as a decimal number.

- Real constant
  Real constants are constants of real type. They are expressed by up to eight digits including a decimal point and are interpreted as a decimal number.

5-3. Types/Operation Rules of Variables and Evaluation of Their Values

5-3-1. Variable type and evaluation

When setting a variable, an assignment statement is used:
Example:
V = e
where,
V = variable name
e = constant, variable name, expression, and function
With this setting, the value of "e" is evaluated, and the value of "V" is changed according to the rule.

Variable Name of V	Unit System	Type of "e"	Evaluation of Value
System variables	1 mm 1/10000 inch	[I]	Not changed
		[R]	[R] → [I] (inch system value is converted into metric system value) (rounding of fractions)
	mm inch	[I]	[I] * 1000 (metric system) [I] * 10000 (inch system)
		[R]	[R] * 100 → (metric system) (rounding off fractions) [R] * 10000 → (inch system) (rounding off fractions)
Common variables	-	[I]	[I] → [R]
		[R]	Not changed
Local variables	-	[I]	[I] → [R]
		[R]	Not changed
Extended address character	-	[I]	[I] → [R]
		[R]	Not changed
I/O variable	-	[I]	Not changed
		[R]	[R] → [I] (rounding off fractions)

Abbreviations:

[I] ·········Integer type
[R] ·········Real type
[I] → [R] ·········Change to real type
[R] → [I] ·········Change to integer type

LE33013R0301400460001

5-3-2. Rules of Operation Expression and Evaluation of Values

Example:

Expression C = A * B

Calculation result — C
Element 1 — A
Element 2 — B

LE33013R0301400470001

<table>
<tr><th>Type of Expression</th><th>Operator</th><th>Meaning</th><th>Type of Element 1 "A"</th><th>Type of Element 2 "B"</th><th>Type of Result of Operation "C"</th></tr>
<tr><td rowspan="8">Arithmetic expression</td><td rowspan="2">+</td><td rowspan="2">Addition</td><td>[I]</td><td>[I]</td><td>[I]</td></tr>
<tr><td>[I] → [R]</td><td>[R]</td><td>[R]</td></tr>
<tr><td rowspan="2">-</td><td rowspan="2">Subtraction</td><td>[R]</td><td>[I] → [R]</td><td>[R]</td></tr>
<tr><td>[R]</td><td>[R]</td><td>[R]</td></tr>
<tr><td>+</td><td>Positive sign</td><td>[I]</td><td rowspan="2"></td><td>[I]</td></tr>
<tr><td>-</td><td>Negative sign</td><td>[R]</td><td>[R]</td></tr>
<tr><td>*</td><td>Multiplication</td><td>[I] → [R]</td><td>[I] → [R]</td><td rowspan="2">[R]</td></tr>
<tr><td>/</td><td>Division</td><td>[R]</td><td>[R]</td></tr>
<tr><td rowspan="4">Comparison expression</td><td>LT
LE</td><td>Less than <
Less than or equal to ≤</td><td>[I]</td><td>[I]</td><td rowspan="4">[b]</td></tr>
<tr><td>EQ</td><td>Equal to =</td><td>[I] → [R]</td><td>[R]</td></tr>
<tr><td>NE</td><td>Not equal to ≠</td><td>[R]</td><td>[I] → [R]</td></tr>
<tr><td>GT
GE</td><td>Greater than >
Greater than or equal to ≥</td><td>[R]</td><td>[R]</td></tr>
<tr><td rowspan="3">Logical expression</td><td>EOR</td><td>Exclusive OR</td><td>[I]</td><td>[I]</td><td rowspan="2">[I]</td></tr>
<tr><td>OR
AND</td><td>Logical sum
Logical product</td><td>[R] → [I] (cutting off fractions)</td><td>[R] → [I] (cutting off fractions)</td></tr>
<tr><td>NOT</td><td>Negation</td><td>[I] or [R] → [I] (cutting off fractions)</td><td></td><td>[I]</td></tr>
</table>

Abbreviations:

[I] ....Integer type

[R] ....Real type

[R] → [I] ....Change to real type

[I] → [R] ....Change to integer type

[b] ....Logical type

LE33013R0301400470002

5-3-3. Function Operation Rules and Evaluation of Value

Example:

Expression C = SINA * B

Calculation result — Element 1 — Element 2

LE33013R0301400480001

Function Name	Meaning	Unit System	Type of Element 1	Type of Element 2	Type of Result of Operation
SIN	Sine	1 μmm	[I] → [R]	[R]/1000 deg. (metric system)	[R]
COS	Cosine	1/10000 inch	[R]	[R]/1000 deg. (inch system)	
TAN	Tangent	1 mm 1 inch	[I] → [R] [R]	[R] -	[R]
ATAN ATAN2	Arctangent	1 μmm 1/10000 inch	[I] → [R] [R]		[R] * 1000 (1/10000 degree) (metric system) [R] * 1000 (1/10000 degree) (inch system)
		1 mm 1 inch	[I] → [R] [R]	-	[R]
SQRT	Square root	-	[I] → [R] [R]	-	[R]
ABC	Absolute value	-	[I] → [R] [R]	-	[R]
BIN	BCD → BIN	-	[I] [R] → [I] (cutting off fractions)	-	[I]
BCD	BCD ← BIN	-	[I] [R] → [I] (cutting off fractions)	-	[I]
ROUND	Rounding off fractions	-	[I]	-	[I] (Not changed)
			[R]		[I]
FIX	Cutting of fractions	-	[I]	-	[I] (Not changed)
			[R]		[I]
FUP	Counting fractions as a whole number	-	[I]	-	[I] (Not changed)
			[R]		[I]

Abbreviations:

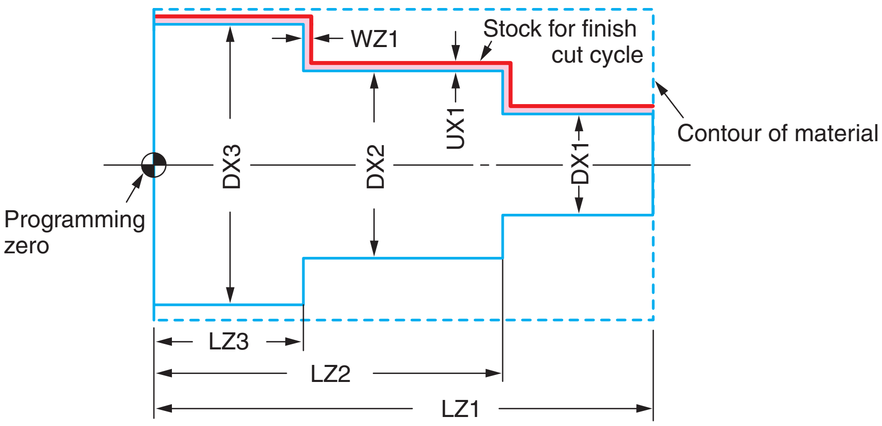
[I] ....Integer type
[R] ....Real type
[R] → [I] ....Change to integer type
[I] → [R] ....Change to real type

LE33013R0301400480002

6. Examples of User Task Programs

Three typical program examples are provided in the following pages.
Please refer to these examples and the programming methods used so that you can make the most of the User Task function.

Program Example 1 (Shaft work with similar contour)



LE33013R0301400490001

Assume that the three different workpieces with similar contours shown above are to be cut. Programs are prepared using user task function as described below.
[Program Sequence]

Procedure :

1 Assign file names to the three workpieces:
SHAFT-A
SHAFT-B
SHAFT-C

2 Since these workpieces have similar contours, their contours are defined with a subprogram. The file name of the subprogram is "SHAFT-ABC.SUB".

3 The elements (dimensions) used to define the contour, and the tool numbers and the cutting speeds, are expressed using the local variables and the common variables, respectively.

V1 = Roughing tool	DX1 = Diameter DX1
V2 = Finishing tool	DX2 = Diameter DX2
V3 = Cutting speed in roughing cycle	DX3 = Diameter DX3
V4 = Cutting speed in finishing cycle	WLZ1 = Finish allowance in longitudinal direction (WZ1)
LZ1 = Longitudinal dimension LZ1	UDX1 = Finish allowance on diameter (UX1)
LZ2 = Longitudinal dimension LZ2	XS = X coordinate of LAP starting point
LZ3 = Longitudinal dimension LZ3	ZS = Z coordinate of LAP starting point

LE33013R0301400490002

4 For cutting a workpiece, the LAP mode is used.

5 The steps described above are summarized in the table below.

		Roughing tool	Finishing tool	Cutting speed in roughing cycle	Cutting speed in finishing cycle	LZ1	LZ2	LZ3	DX1	DX2	DX3	WZ1	UX1	XS	ZS
		V1	V2	V3	V4	LZ1	LZ2	LZ3	DX1	DX2	DX3	WLZ1	UDX1	XS	ZS
Workpiece A	Shaft-A	0101	0202	100	120	200	150	80	30	50	80	0.1	0.2	100	210
Workpiece B	Shaft-B	0303	0404	110	130	250	170	100	40	70	120	0.15	0.25	140	260
Workpiece C	Shaft-C	0505	0606	90	150	300	200	120	50	90	150	0.2	0.3	170	300

- Subprogram
  The subprogram defining the contour, prepared using local and common variables, can be programmed as shown below on the basis of the table above.

```
        $ SHAFT-ABC.SUB %
O1000
NLAP1   G81
N1001   G00   X=DX1    Z=LZ1+2
N1002   G01   Z=LZ2    F0.2
N1003         X=DX2
N1004                  Z=LZ3
N1005         X=DX3
N1006                  Z=0
N1007   G80
N1010   G00   X=800    Z=400
N1011   G96   X=XS     S=V3 T=V1 M03 M08
N1012   G85   NLAP1    D4       F0.35 U=UDX1 W=WLZ1
N1020   G00   X=800    Z=400
N1021
N1022   G87   NLAP1
```

LE33013R0301400490003

- Main Program
  The cutting program is made up of three types of main program for each workpiece.

Workpiece A $SHAFT-A. MIN %
O100
N101 G00 X800 Z400
N102 CALL O1000 V1=0101 V2=0202 V3=100 V4=120 LZ1=200 LZ2=150 LZ3=80 $ DX1=30 DX2=50 DX3=80 WLZ1=0.1 UDX1=0.2 XS=100 ZX=210
N103 M02

LE33013R0301400490004

Workpiece B $SHAFT-B. MIN %
O101
N101 G00 X800 Z400
N102 CALL O1000 V1=0303 V2=0404 V3=110 V4=130 LZ1=250 LZ2=170 LZ3=100 $ DX1=40 DX2=70 DX3=120 WLZ1=0.5 UDX1=0.25 XS=140 ZX=260
N103 M02

LE33013R0301400490005

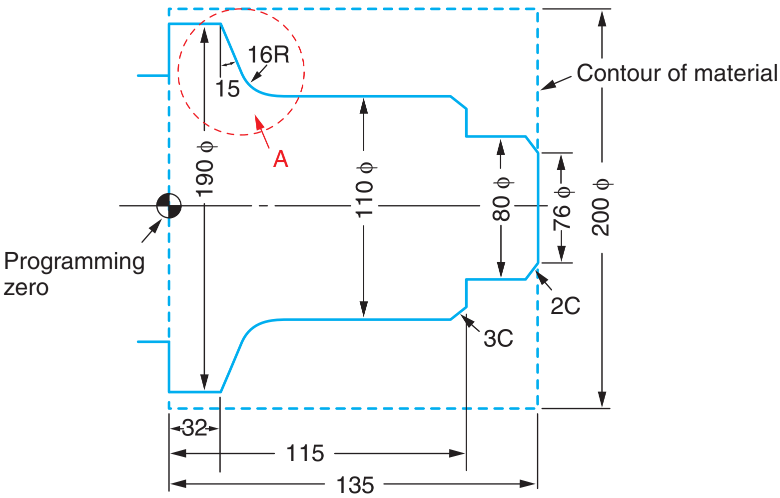
Workpiece C $SHAFT-C. MIN %
O102
N101 G00 X800 Z400
N102 CALL O1000 V1=0505 V2=0606 V3=90 V4=150 LZ1=300 LZ2=200 LZ3=120 $ DX1=50 DX2=90 DX3=150 WLZ1=0.2 UDX1=0.3 XS=170 ZX=360
N103 M02

LE33013R0301400490006

[Supplement]

- File name of the cutting program (main program)
  Prefix the file name with $. If the program is on tape, punch the machining program (main program) in the following order:
  $, program name, MIN, feed holes, %, CR, LF.
- Cutting program name (number)
  O100 in this example
- The main program calls the sub program O1000 to perform machining. At this time, the main program assigns required data to the variables used by the sub program.

Program Example 2 (Cutting contour requiring calculation for definition)



[Fig. 2-1]

LE33013R0301400490007

When cutting a contour containing a circular arc and a taper and when the point(s) of intersection is not indicated on the part drawing, the operation functions of the User Task function can be used with good effect to create the program.

[Program Sequence]

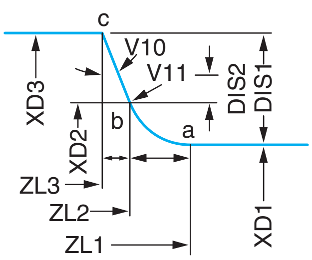
(1) With enlarging section A, have the control calculate the points of intersection using the variables and the operation function of the user task.
The points that must be calculated are Z-coordinate of point a and X- and Z-coordinates of point b. To obtain them, variables are set as below.

V10 = Taper angle (15°)
V11 = Arc radius (16 mm)
} Set using common variables

XD1 = Diameter of point "a" (110 mm)
XD2 = Diameter of point "b"
XD3 = Diameter of point "c" (190 mm)
ZL1 = Z-coordinate of point "a"
ZL2 = Z-coordinate of point "b"
ZL3 = Z-coordinate of point "c" (32 mm)
DIS1 = Distance: DX3 - DX1
DIS2 = Distance between the center of arc and point "b" (along X-axis)
DIS3 = Distance between point "a" and point "b" (along Z-axis)
DIS4 = Distance between point "b" and point "c" (along Z-axis)
} Set using local variables

LE33013R0301400490008

(2) The point of intersection can be calculated in the following equations, based on the variables in 1.



[Fig. 2-2]

$DIS1 = (DX3 - DX1) / 2$

$DIS2 = V11 \times SIN (V10)$

$DIS3 = V11 \times COS (V10)$

$DIS4 = (DIS1 + DIS2 - V11) \times TAN (V10)$

$XD2 = XD1 + 2 \times (V11 - DIS2)$

$ZL2 = AL3 + DIS4$

$ZL1 = ZL2 + DIS3$

LE33013R0301400490009

(3) Since the pattern in Section A can be used in common with other workpieces, it is advisable to program such a contour as a subprogram. We will name the subprogram "RADIUS-TAPER.SUB". Variables XD2, ZL1, and ZL2 are set in this subprogram, and other variables are set in the main program.

(4) Prepare the cutting program as a main program.
The file name of the main program is "FLANGE-1.MIN". The LAP and nose radius compensation functions are used in the main program.

- Subprogram

RADIUS-TAPER. SUB %
ORT01
N1000 XD2=XD1+2∗ [V11-DIS2] ZL2=ZL3+DIS4 $ ZL1=AL2+DIS3
N1001 G01 Z=ZL1
N1002 G02 X=XD2 Z=ZL2 L=V11
N1003 G01 X=XD3 Z=ZL3
N1004 RTS

LE33013R0301400490010

- Main program (cutting program)

FRANGE-1. MIN %					
O100					
N101	V10=15 V11=16 XD1=110 XD3=90 ZL3=32				
N102	G00	X800	Z300		
NLAP1	G81				
N103	G00	X76	Z137		
N104	G42	G01	Z135	F0.2	
N105	G75	X80	L-2		
N106			Z115		
N107	G75	X=X01	L-3		
N108 CALL ORT01 DIS1= [XD3-X01] /2 DIS2-V11∗SIN [V10] $ DIS3=V11∗COS(V10) DIS4= [DIS1+DIS2-V11] ∗TAN [V11]					
N109			Z-2		
N110	G40	G00	X200	I10	
N111	G80				
N120	G00	X800	Z300		
N121	G96	X202	Z137	S110 T010101 M03 M08	
N122	G85	NLAP1	D6	F0.35 U0.2 M0.1	
N123	G00	X800	Z300		
N124			S130	T020202	
N125	G87	NLAP2			
N126	G00	X800	Z300		
				N05 M09	
N128	M02				

LE33013R0301400490011

[Supplement]

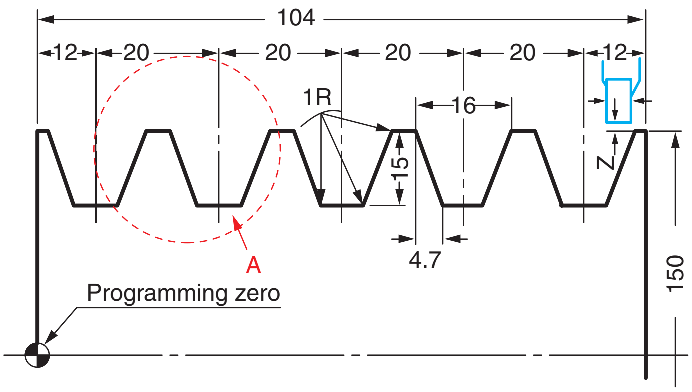
- Variables are set in block N1000.
- The Z coordinate of point "a" is commanded in block N1001.
- The X and Z coordinates of point "b" and arc radius are commanded in block N1002.
- The X and Z coordinates of point "c" are commanded in block N1003.
- RTS in block N1004 indicates the end of the subprogram.

[Supplement]

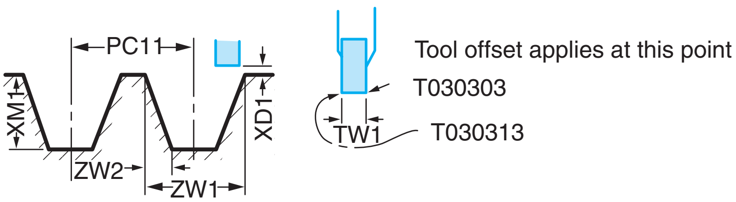
The subprogram ORT01 is called for by the command in block N109 to define the contour consisting of a circular arc and taper. The variables used for defining this contour are all set in this block.

Program Example 3 (Cutting a contour consisting of a repetitive contour)

This is an example of programming for a part that has repetitions of the same contour, such as a pulley.



[Fig. 3-1]



[Fig. 3-2]

[Program Sequence]

(1) Assume that there are a number of pulleys with a similar contour, shown as above. To simplify the programs of these pulleys, express the contour of part A using variables.

Variable Name	Contents	Numerical Value for This Example (Fig. 3-1)
PC11	Pitch between the pulley groove	20 mm
XH1	Height of pulley groove	15
XD1	Starting point of cutting	2
ZW1	Width of groove	16
ZW2	Taper amount of groove in Z-axis direction	4.7
TW1	Width of cutting off tool	6
D1	Value of I of 1R	0.299
DK	Value of K of 1R	0.954

(2) Program the pulley groove cutting cycle as a subprogram using the variables.
Since this program is called for after execution of axis motion command(s), it is prepared in incremental mode so that it can be used wherever it is called.
The subprogram file name is "PULL-PTTN1.SUB".

- Subprogram

$ PULL-PTTN1. SUB %				
OPP1				
N1	G91	G00 Z=- [ZW1/2] + [TW1/2]		
N2	G42	X=-XD1+0.2		
N3	G01	X=- [XH1*2] -0.2		F0.05
N4	G00	X= [XH1*2] +XD1		
N5		Z=ZW2		
N6		X=-XD1		
N7	G01	X=- [XH1*2]	Z=-ZW2	
N8	G00	X= [XH1*2] +XD1		
N9	G41	Z=-ZW2 T030313		
N10		X=-XD1		
N11	G01	X=- [XH1*2]	Z=-ZW2	
N12	G00	X= [XH1*2] +XD1		
N13			Z=-ZW2-DK-0.3	
N14		X=-XD1		
N15	G02	X=- [1-D1] *2	Z=DK I=-1	
N16	G01	X=- [XH1*2] + [1-D1] *4	Z=ZW2-DK	
N17	G03	X=- [1-D1] *2	Z=DK I=DI K=DK	
N18	G00	X= [XH1*2] *XD1		
N19		Z=ZW2+DK [0.3*2] T030303		
N20	G42	X=-XD1		
N21	G03	X=- [1-DI] *2	Z=-DX I=-1	
N22	GO1	X= [XH1*2] + [I-DI] +4	Z=-ZW2+DK	
N23	G02	X=- [1-DI] *2 Z=-DK I=DI K=-DK		
N24	G00	X= [XH1*2] +XD1		
N25		Z=ZW2		
N26	G90			
N27	RTS			

LE33013R0301400490013

(3) The program for cutting one pulley groove was created in step (2). Using this subprogram, the program to cut the pulley shown in Fig. 3-1 can be prepared.
Make this program as a main program: Program file name is "PULLY-1.MIN".

- Main Program (Cutting Program)

$ PULLY-1. MIN %				
OPLY1				
N001	G13			
N002	G00	X=800	Z300	
N003	G96			S70 T030303 M03
N004		PCH=20	XD1=2	
N005			Z100	
N006	MODIN OPP1 XH1=15 ZW1=16 ZW2=4.7 TW1=6 D1=0.299 DK=0.954			
N007	G00	X=150+XD1	Z100	
N008	G00		Z100-PCH	
N009	G00		Z100- [2*PCH]	
N010	G00		Z100- [3*PCH]	
N011	G00		Z100- [4*PCH]	
N012	MODOUT			
N013	G40 G00 X800 Z300 M05			
N014	M02			

LE33013R0301400490014

[Supplement]

The MODIN statement in block N007 places the control in the MODIN mode in which the subprogram is called and executed every time axis motion commands are completed. In this block, variables used in the subprogram OPP1 are also set.
In blocks N007 through N011, the subprogram OPP1 is called and executed every time the axis motion command(s) in those blocks is/are completed, thus cutting the pulley grooves.
The pulley grooves could also be cut by using the CALL statement instead of the MODIN and MODOUT statements. However, when the CALL statement is used, that statement must be repeated every time the subprogram is to be called.

SECTION 13 SCHEDULE PROGRAMS

1. Overview

Schedule programs permit different types of workpieces to be machined continuously without any operator intervention by using a bar feeder, loader, or other automation equipment.

- Several main programs can be selected and executed in the specified order by a schedule program.
- A schedule program is a set of the following five blocks. If other blocks are specified, an alarm will occur. The program must be terminated with the END block.

  a. PSELECT block
  Selects and executes main programs.

  b. GOTO block
  Branches unconditionally.

  c. IF block
  Branches conditionally.

  d. VSET block
  Sets variables.

  e. END block
  Terminates schedule programs.

- These commands must be specified at the start of, or immediately after, the sequence name.
- Although comments given between '('and ')' and continuous lines identified by '$' are valid, block delete (/) is invalid.
- Total tape length for the schedule, main, and sub program is up to the maximum size of operation buffer area which is selected by the specification.

The blocks specified in a schedule program are described below.

2. PSELECT Block

[Function]
A PSELECT block selects and executes main programs for a workpiece to be machined.

- This function searches a specified main program file for a specified main program to be selected as a machining program. The function also searches a specified subprogram file, or system subprogram file, and manufacturer subprogram file for the required subprograms and selects them automatically.

- It repeats selected programs as specified.

[Programming format]
The commands must be specified in the following order:
Commands enclosed by [ ] may be omitted. Note that a comma "," may also be omitted if the items that follow are all omitted.

{PSELECT} ␣ [fm],[pm],[fs],[n](CR) or (LF)

LE33013R0301500020001

fm: Main program file name

Entries enclosed by [ ] may be omitted.

[3 characters :] [Within 16 characters] [. 3 characters]

Device name   File name   Extension

LE33013R0301500020002

- If a device name, a file name, and/or an extension is omitted, entries of "MD1", "A", and "MIN", respectively, are assumed to apply. If all entries for "fm" are omitted, "MD1:A.MIN" is assumed to apply.

- An alarm will occur if "*" or "?" is used in a main program file name.

- An alarm will occur if the specified file does not exist.

pm: Main program name

0	

Within 5 characters

LE33013R0301500020003

- If entry of "pm" is omitted, the program name of the first program in the specified main program file is assumed to apply.

- An alarm will occur if the specified program does not exist in the selected main program file, fm.

- An alarm will occur if M02 or M30, indicating the end of the program, is not specified in the specified main program.

fs: Subprogram file name

Entries enclosed by [ ] may be omitted.

[3 characters :] [Within 16 characters] [. 3 characters]

Device name File name Extension

LE33013R0301500020004

- Entry of "fs" may be omitted when:
  - No subprogram call command is specified in a main program.
  - The subprogram called from a main program or subprogram exists in MD1:*.SSB (system subprogram) or in MD1:*.MSB (manufacturer subprogram).
  - Required subprograms other that SSB and MSB are contained in the main program file. If fs is specified, the device name and extension may be omitted. The defaults for the device name and extension are "MD1" and "SUB", respectively. Therefore, if everything is omitted, it is assumed that no file has been specified.
- An alarm will occur if the total number of subprograms used exceeds 126.
- An alarm will occur if "RTS" which means the end of a subprogram is not specified.
  An alarm will occur if the file specified by "fs" does not exist.

n: Repetition count

Q	=	Expression

LE33013R0301500020005

Q: Number of repetitions (specified by address)
The setting range is from 1 to 9999 and "1" is assumed to apply if entry of "n" is omitted. An alarm will occur if a number outside the range 1 to 9999 is specified.
A space (" ") may be used instead of "=". "=" may be omitted if it is directly followed by a numeric value.

3. Branch Block

The branching function of the schedule program, which is identical to SECTION 13, "Control Statement Function 1", is made possible by GOTO and IF blocks, which provide unconditional branching and conditional branching, respectively.

GOTO Block

[Function]
The GOTO block unconditionally changes program sequences. The destination of a jump is specified using a sequence name immediately after the GOTO command.
[Programming format]
Commands must be specified in the following order:

GOTO ␣ N ______
Specifies the destination of a jump

LE33013R0301500030001

IF Block

[Function]
The IF block conditionally changes program sequences. If the condition is 'true', the sequence branches to the destination of a jump. If the condition is 'false', it proceeds to the next stop.
[Programming format]
Commands must be specified in the following order:

IF [Expression ␣ Comparison operator ␣ Expression] N ______
Specifies the destination of a jump

The comparison operators include LT (<), LE (≤), EQ (=), NE (≠), GE (≥), GT (>). For details, see SECTION 13 "Arithmetic Operation Function 1".

LE33013R0301500030002

4. Variables Setting Block

[Function]
'VSET' must be specified to set variables using the schedule program.
[Programming format]
Commands must be specified in the following order:

VSET ␣ Variable = Expression ␣ Variable = Expression

LE33013R0301500040001

Variable in the left part	:	Specify a common, system, or output variable. If any other variable is specified in the left part, an alarm will occur.
Right part	:	An alarm will occur if an output variable is specified.

5. Schedule Program End Block

[Function]
At the end of a schedule program, an "END" block must always be specified. All blocks specified following the "END" block are invalid.
[Programming format]
END

6. Program Example

The procedure to create a schedule program is explained below.
Assume that the NC lathe is equipped with a bar feeder and three different workpieces are machined according to the programmed schedule.
[Program sequence]

Procedure :

1 Determine the file name and the program name (number) of the program to be used for machining three kinds of parts.

Part A	A.MIN, O100
Part B	B.MIN, O200
Part C	C.MIN, O300

2 Create the program for each part based on the part drawing.

3 To machine these three kinds of parts, determine the order and the number of parts to be machined.

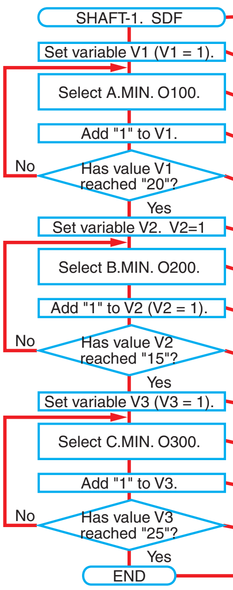
Part A	20 pcs.
Part B	15 pcs.
Part C	25 pcs.

4 Make the schedule program next. Determine the file name of the schedule program.

File name	SHAFT-1.SDF

5 Use common variables as counters to count the number of machined parts.

Variable for part A V1
Variable for part B V2
Variable for part C V3



Schedule program	
$ SHAFT-1. SDF	File name of the schedule program
%	This code must always be specified at the beginning of a program.
N1000␣VSET␣V1=1 (*1)	Sets the initial value for the counter for Part A.
N1001␣PSELECT␣A. MIN. O100	Calls and executes the program for Part A. Note: PSELECT means "program select".
N1002␣VSET␣V1=V1+1 (*1)	Adds "1" to the counter on completion of the program.
N1003␣IF [V1␣LE␣20]␣N1001	Jumps to N1001 to repeat machining if "counter value 20". Proceeds to N1004 after machining 20 pcs. of Part A.
N1004 VSET V2=1 (*1)	Sets the initial value for the counter for Part B.
N1005␣PSELECT␣B. MIN. O200	Calls and executes the program for Part B.
N1006␣VSET␣V2=V2+1 (*1)	Adds "1" to the counter on completion of the program.
N1007␣IF [V2␣LE␣15]␣N1005	Jumps to N1005 to repeat machining if "counter value 15". Proceeds to N1008 after machining 15 pcs. of Part B.
N1008␣VSET␣V3=1 (*1)	Sets the initial value for the counter for Part C.
N1009␣PSELECT␣C. MIN. O300	Calls and executes the program for Part C.
N1010␣VSET␣V3=V3+1 (*1)	Adds "1" to the counter on completion of the program.
N1011␣IF [V3␣LE␣25]␣N1009	Jumps to N1009 to repeat machining if "counter value 25". Proceeds to N1012 after machining 25 pcs. of Part C.
N1012␣END	End code of a schedule program. This must always be entered at the end of a schedule program.

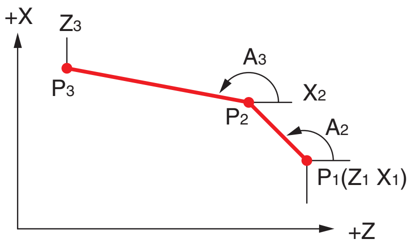
Main Program (Machining Program)		
A. MIN. O100 % N100 G00 X Z STM N150 M02	B. MIN. O200 % N100 G00 X Z STM N230 M02	C. MIN. O300 % N300 G00 X Z STM N340 M02

*1: When setting a common variable in a schedule program, specify the "VSET" command first, then set the common variable.

SECTION 14 OTHER FUNCTIONS

1. Direct Taper Angle Command

In conventional programming, taper cutting called for by G01, G34, and G35 is programmed using the coordinates of the target point.
However, by using this feature the command is given simply by entering either the X or Z coordinate point of the end point of the taper along with the angle referenced to the Z-axis (measured in the counterclockwise direction).



N1	G01	X_1	Z_1	F_1
N2		X_2		A_2
N3			Z_3	A_3

LE33013R0301600010001

- An angle command in taper definition is effective in:
  G00, G01, G31, G32, G33, G34, and G35

- The angle is specified following address character "A".
  The units of angle commands for the metric and inch specifications are as follows:

Metric	:1 µm unit system..........................0.001°
	:10 µm unit system........................0.01°
	:1 mm unit system..........................1°
Inch	:1/10000 inch unit system............0.001°
	:1 inch unit system........................1°

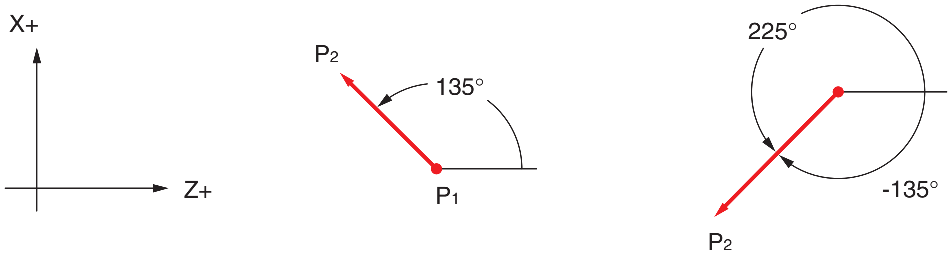
LE33013R0301600010002

- The control interprets the commands as a taper command when the commands contain either an X or Z word along with an A word.

- If an A command is designated with both X and Z words, or if it is designated without association with an X or Z word, an alarm results.

- The direct taper angle command is effective in:
  LAP
  Tool nose radius compensation mode
  Incremental programming mode
  Subprograms

- The angle is measured on the Z-X plane taking positive direction of Z-axis as 0 deg. It is positive when measured in the counterclockwise direction and negative in the clockwise direction.
  In the figure below, the angle is expressed as A135 in 1 mm unit system control since the angle is measured in the counterclockwise direction.
  For the angle shown in the right of the figure, A225 and A-135 will give the same taper.



LE33013R0301600010003

- If no point of intersection is obtained with the commands X and A, or Z and A, an alarm results.
  Precautions for Programming Constant Speed Cutting

2. Barrier Check Function

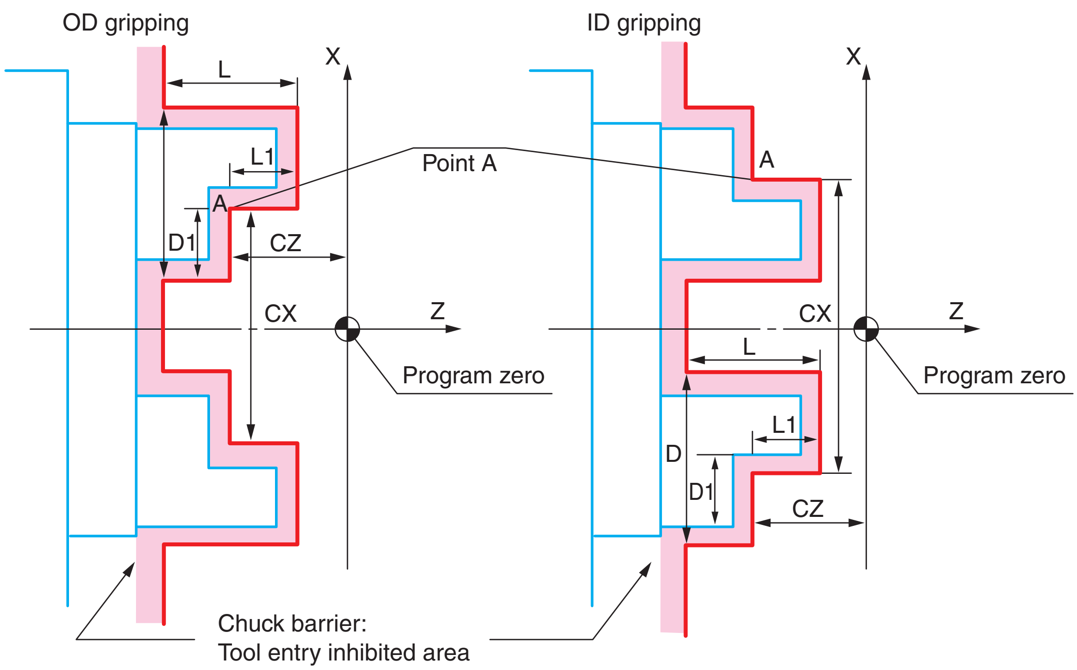
2-1. General Description

The barrier check function permits a chuck/tailstock barrier (a specific machine area into which any cutting tool entry is prohibited) to be established in the vicinity of a chuck/tailstock on the basis of data in a program or entered through MDI switches. If a tool is commanded to move inside the barrier, an alarm occurs and stops machine operation.
This function makes it possible to carry out trial operation in the automatic mode or cutting in the MDI mode without worrying about cutting tool collisions. In addition, it prevents accidents where a cutting tool strikes the chuck or tailstock due to an unexpected problem during automatic operation.

2-2. Chuck Barrier and Tailstock Barrier

2-2-1. Establishing a Chuck Barrier

The chuck barrier function can set an area around the chuck into which tool entry is prohibited, and which is matched to the chuck shape: this is not possible with variable soft-limit position data.
Activation or deactivation of the chuck barrier function can be selected by programming the appropriate M code. Therefore, the tool motion using the chuck barrier function can be checked when required.



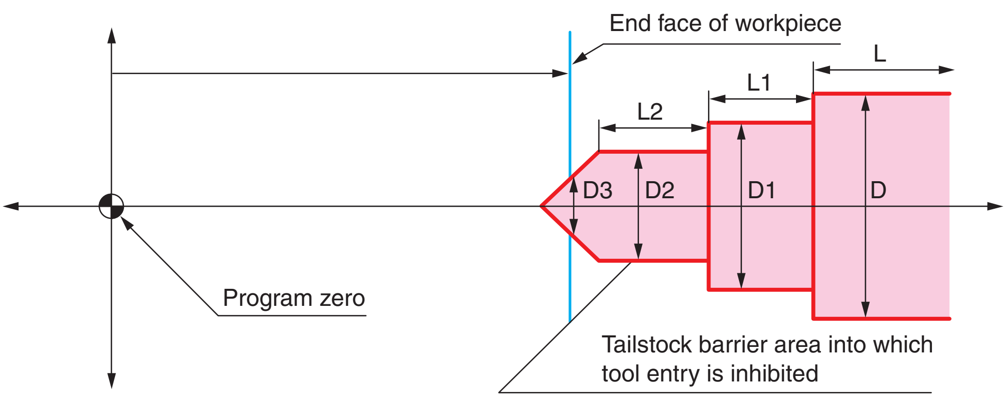
LE33013R0301600030001

Symbol	Description	Method
L	Chuck jaw length	Chuck/tailstock axis
D	Chuck jaw size	
L1	Gripping length of chuck jaw	
D1	Chuck jaw gripping face width	
CX	Chuck gripping diameter	
CZ	Distance from programming zero	

For details of the procedure to establish the chuck barrier, refer to SECTION 4, PARAMETER SETTING in DATA OPERATION of OPERATION MANUAL.

2-2-2. Establishing a Tailstock Barrier

The area used for the tailstock barrier function is indicated in the figure below.



Method of setting tailstock barrier
Set required data at the following parameters:

Symbol	Description	Method
L	Tailstock spindle length	Chuck/tailstock axis
D	Tailstock spindle diameter	
L1	Tailstock spindle length (1)	
D1	Tailstock spindle diameter (1)	
L2	Tailstock spindle length (2)	
D2	Tailstock spindle diameter (2)	
D3	Tailstock spindle center diameter	
WR	Tailstock spindle position (Z)	

For details of procedure to establish the tailstock barrier, refer to SECTION 4, PARAMETER SETTING in DATA OPERATION of OPERATION MANUAL.

2-2-3. Tool Movements and Alarm

Once the chuck barrier is established, it is activated or deactivated by programming the appropriate M code:

M25 Chuck barrier ON
M24 Chuck barrier OFF
M21 Tailstock barrier ON
M20 Tailstock barrier OFF

If the cutting tool is commanded to enter inside the barrier while the chuck and/or the tailstock barrier function is active, an alarm occurs and the machine stops.
Example of Program:

:		:		
:		:		
		:	Normal machining program	
:		:		
:		:		
N×××	M25	:	Chuck barrier ON	The barrier function is active for the blocks of commands from the block containing M25 (M21) to the one containing M24 (M20).
:	(M21)	:	(Tailstock barrier ON)	
:		:		
:	M24	:	Chuck barrier OFF	
N×××	(M20)	:	(Tailstock barrier OFF)	
:		:		
:		:		
:		:		

LE33013R0301600050001

[Supplement]

1) When power supply to the control is turned ON or when the control is reset, the control is automatically set in the barrier off mode (M24 and M20 active). If the chuck and the tailstock barrier functions are required to be active, command M25 and M21.
2) The chuck and tailstock barrier function are active for manual pulse handle mode operation or jog feed operation.
3) The barrier is modified when new barrier setting data is entered.

3. Operation Time Reduction Function

Refer to the Operation Manual for details of the operation time reducing function II.

4. Turret Unclamp Command (for NC Turret Specification)

The NC simultaneously unclamps the turret and causes axis travel on receiving the M203 command. This command is effective only when it is specified with G00 in the same block.
Example:

G01	X200				
G00	X220	Z300			
G00	X500	Z800	M203	← X220 → X500	Simultaneous movements of X-axis from 220 to 500 and Z-axis from 300 to 800.
	X220	Z300	T020202	Z300 → Z800	

LE33013R0301600070001

Note that if the M203 command is specified in the same block as G00 it unclamps the turret without regard to the present turret position.
If the M203 command is preceded by a cutting feed command (feed command other than G00) and the turret clamped condition is not confirmed, the alarm "A1730 Turret clamp or position code 3" will occur.

5. SPINDLE SPEED VARIATION CONTROL FUNCTION

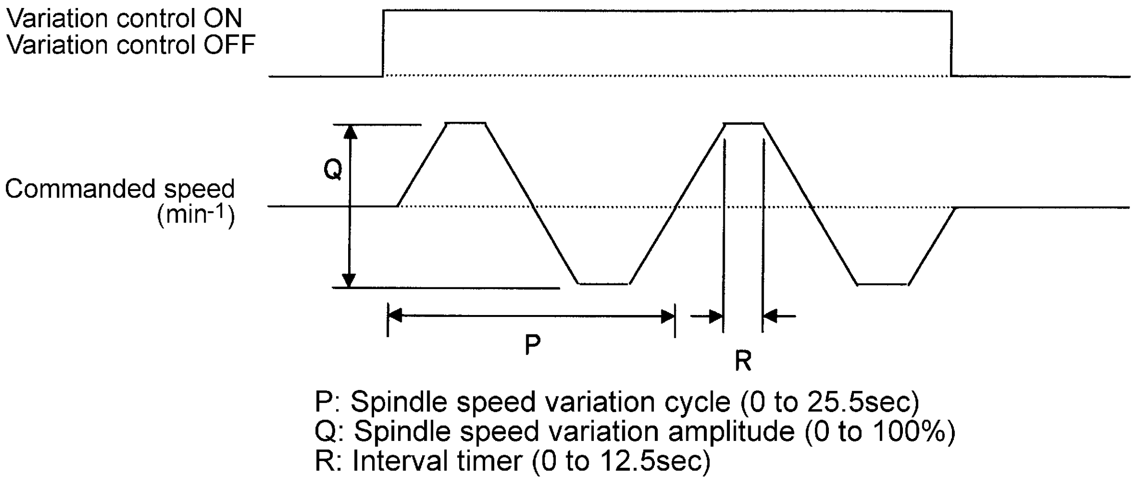
5-1. Outline

The spindle speed variation control function changes the spindle speed periodically to prevent chattering generated during machining of a thin-wall and large-diameter workpiece.

5-2. Method of Spindle Speed Variation Control

5-2-1. Spindle Speed Variation

Since chattering may occur during cutting of thin and large-diameter workpieces or the like, this function prevents chattering during cutting by varying the spindle speed according to the cycle, amplitude data, and interval timer predetermined in relation to the commanded speed.



LE33013R0301600090001

5-3. Control Specifications

5-3-1. M Codes

Spindle speed variation control is turned on or off using the following M code commands.
M695: Spindle speed variation control ON
M694: Spindle speed variation control OFF

[Supplement]

- These M code commands are modal commands. Once an M695 command is specified, the M695 command is effective until an M694 command is specified.
- When the power supply is turned on or after the system is reset, these M codes get into the state of M694.

5-3-2. Parameters

The following parameters are added to allow the settings of amplitude (Q), cycle (P), and interval timer (R).

(1) Spindle speed variation amplitude (Q)
Sets an amplitude of spindle speed variation.

Parameter word	No.114
Setup unit	1[%]
Setup range	0 to 100
Default value	0

(2) Spindle speed variation cycle (P)
Sets a cycle of spindle speed variation.

Parameter word	No.115
Setup unit	0.1[sec]
Setup range	0 to 255
Default value	0

(3) Interval timer (R)
Sets an interval timer.

Parameter word	No.116
Setup unit	0.1[sec]
Setup range	0 to 125
Default value	0

5-3-3. System Variables

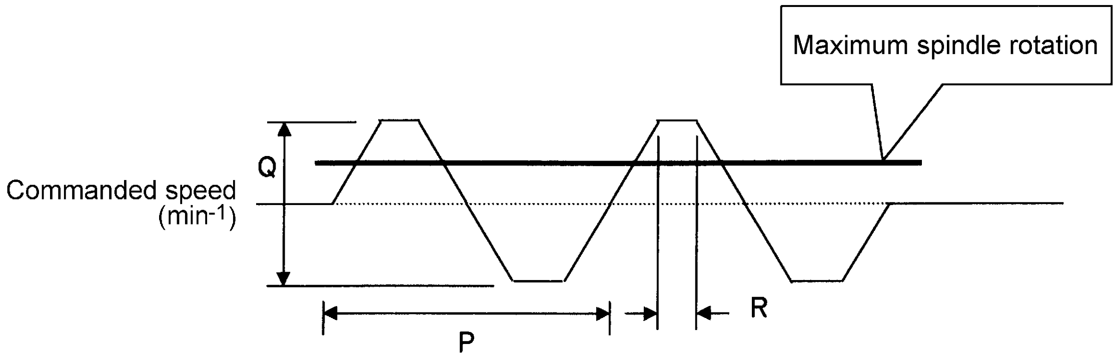
The following system variables are added to allow the reading and writing of the above parameter data.

VFLTQ: Spindle speed variation amplitude (Q)
VFLTP: Spindle speed variation cycle (P)
VFLTR: Interval timer (R)

5-3-4. Specification Limitation

When you use this control, be careful for the followings.

(1) When the spindle speed under variation control exceeds the maximum spindle speed (including maximum spindle speed command), the speed hits the peak at the maximum spindle speed. Be careful of this matter sufficiently when you give a command.



LE33013R0301600130001

(2) Do not use this control if the variation of spindle speed gives troubles to cutting.
(Ex.)

- Tapping
- Flat-turning (M220 to M226)

(3) This control is not effective for spindle rotation other than spindle forward rotation (M03) and spindle reverse rotation (M04).
(Ex.)

- Spindle orientation command (M19)
- Spindle inching command
- Synchronous rotation control (M151)

(4) Even for spindle forward rotation (M03) or spindle reverse rotation (M04), this control is not effective in the following condition.
(Ex.)

- During constant peripheral speed control (G96)

(5) If the relationship between spindle speed variation cycle (P) and interval timer (R) is as shown below, spindle speed variation control cannot be applied. Be careful of this matter sufficiently when you set data.
Spindle speed variation cycle (P) $\leq$ Interval timer (R) $\times$ 2

5-4. Programming Example

```
G50 S2000
G00 X1000 Z1000
M03 S1000
   :
M695                ←Spindle speed variation control ON
   :
M696                ←Spindle speed variation control OFF
   :
M05
M02
```

SECTION 15 APPENDIX

1. G Code Table

☆ : Optional
Others : Standard

G Code	Contents	
G00	Positioning	
G01	Linear interpolation	
G02	Circular interpolation (CW)	
G03	Circular interpolation (CCW)	
G04	Dwell	
G05		
G06		
G07		
G08		
G09		
G10		
G11		
G12		
G13	Turret selection: Turret A	☆
G14	Turret selection: Turret B	☆
G15		
G16		
G17	Cutter radius compensation: X-Y plane	☆
G18	Cutter radius compensation: Z-X plane	☆
G19	Cutter radius compensation: Y-Z plane	☆
G20	Home position return command	☆
G21	ATC home position return command	☆
G22	Torque skip command	☆
G23		
G24	ATC home position movement command (without linear interpolation)	☆
G25	Joint position movement command (without linear interpolation)	☆
G26		
G27		
G28	Torque limit command cancel	☆
G29	Torque limit command	☆
G30	Skip cycle	☆
G31	Fixed thread cutting cycle: Longitudinal	
G32	Fixed thread cutting cycle: End face	
G33	Fixed thread cutting cycle	
G34	Variable lead thread cutting cycle: Increasing lead	
G35	Variable lead thread cutting cycle: Decreasing lead	

G Code	Contents
G36	M-tool spindle - feed axis synchronized feeding (forward) ☆
G37	M-tool spindle - feed axis synchronized feeding (reverse) ☆
G38	
G39	
G40	Cutter radius compensation: Cancel
G41	Cutter radius compensation: Left
G42	Cutter radius compensation: Right
G43	
G44	
G45	
G46	
G47	
G48	
G49	
G50	Zero shift, Maximum spindle speed designation
G51	
G52	Turret index position error compensation ☆
G53	
G54	
G55	
G56	
G57	
G58	
G59	
G60	
G61	
G62	Mirror image designation ☆
G63	
G64	Droop control OFF
G65	Droop control ON
G66	
G67	
G68	
G69	
G70	
G71	Compound fixed thread cutting cycle: Longitudinal
G72	Compound fixed thread cutting cycle: Transverse
G73	Longitudinal grooving compound fixed cycle
G74	Transverse grooving compound fixed cycle
G75	Automatic chamfering
G76	Automatic rounding
G77	Tapping compound fixed cycle
G78	Tapping cycle reverse thread
G79	

G Code	Contents	
G80	End of shape designation (LAP)	☆
G81	Start of longitudinal shape designation (LAP)	☆
G82	Start of transverse shape designation (LAP)	☆
G83	Start of blank material shape definition (LAP)	☆
G84	Change of cutting conditions in bar turning cycle (LAP)	☆
G85	Call of rough bar turning cycle (LAP)	☆
G86	Call of rough copy turning cycle (LAP)	☆
G87	Call of finish turning cycle (LAP)	☆
G88	Call of continuous thread cutting cycle (LAP)	☆
G89		
G90	Absolute programming	
G91	Incremental programming	
G92		
G93		
G94	Feed per minute mode (mm/min)	
G95	Feed per revolution mode (mm/rev)	
G96	Constant speed cutting ON	
G97	Cancel of G96	
G98		
G99		
G100	Priority designating for turret A or B independent cutting	☆
G101	Linear interpolation in contour generation	☆
G102	Circular interpolation in contour generation (Face) (CW)	☆
G103	Circular interpolation in contour generation (Face) (CCW)	☆
G104		
G105		
G106		
G107	Spindle synchronization tapping, RH thread	☆
G108	Spindle synchronization tapping, LH thread	☆
G109		
G110	Constant speed cutting on turret A	
G111	Constant speed cutting on turret B	
G112	Circular thread cutting CW	☆
G113	Circular thread cutting CCW	☆
G114		
G115		
G116		
G117		
G118		
G119	Nose-R compensation: C-X-Z plane	☆
G120		
G121		
G122	W-axis command for sub spindle on turret A (G13)	☆
G123	W-axis command for sub spindle on turret A (G14)	☆

G Code	Contents	
G124	Chuck A origin effective	☆
G125	Chuck B origin effective	☆
G126	Slope machining mode OFF command	☆
G127	Slope machining mode ON command	☆
G128	M/C machining mode OFF command	☆
G129	M/C machining mode ON command	☆
G130		
G131		
G132	Circular interpolation in contour generation (Side)(CW)	☆
G133	Circular interpolation in contour generation (Side)(CCW)	☆
G134		
G135		
G136	End of coordinate conversion or Y-axis mode OFF	☆
G137	Start of coordinate conversion	☆
G138	Y-axis mode ON	☆
G139		
G140	Designation of machining mode using main spindle	☆
G141	Designation of machining mode using sub spindle	☆
G142	Designation of machining mode using pick-off spindle	☆
G143	Designation of machining mode using pick-off spindle and 3rd turret	☆
G144	W-axis control OFF command	☆
G145	W-axis control ON command	☆
G146		
G147		
G148	B-axis mode OFF command	☆
G149	B-axis mode ON command	☆
G150		
G151		
G152	Programmable tailstock positioning (tow-along tailstock)	☆
G153	Programmable steadyrest G code (tow-along)	☆
G154	W-axis unidirectional positioning command	☆
G155	Accurate contouring mode ON command	☆
G156	Accurate contouring mode OFF command	☆
G157		
G158	Tool length offset in tool axis direction	☆
G159	Tool length offset in tool axis direction (without rotary position offset)	☆
G160	Tool length offset in tool axis direction cancel	
G161	G code macro function MODIN	☆
G162	G code macro function MODIN	☆
G163	G code macro function MODIN	☆
G164	G code macro function MODIN	☆
G165	G code macro function MODIN	☆
G166	G code macro function MODIN	☆
G167	G code macro function MODIN	☆

G Code	Contents
G168	G code macro function MODIN ☆
G169	G code macro function MODIN ☆
G170	G code macro function MODIN ☆
G171	G code macro function CALL ☆
G172	
G173	
G174	
G175	
G176	
G177	
G178	Synchronized tapping cycle (CW) ☆
G179	Synchronized tapping cycle (CCW) ☆
G180	M-tool compound fixed cycle: Cancel ☆
G181	M-tool compound fixed cycle: Drilling ☆
G182	M-tool compound fixed cycle: Boring ☆
G183	M-tool compound fixed cycle: Deep hole drilling ☆
G184	M-tool compound fixed cycle: Tapping ☆
G185	M-tool compound fixed cycle: Longitudinal thread cutting ☆
G186	M-tool compound fixed cycle: End face thread cutting ☆
G187	M-tool compound fixed cycle: Longitudinal straight thread cutting ☆
G188	M-tool compound fixed cycle: Transverse straight thread cutting ☆
G189	M-tool compound fixed cycle: Reaming/boring ☆
G190	M-tool compound fixed cycle: Keyway cutting cycle ☆
G191	M-tool compound fixed cycle: Longitudinal keyway cutting cycle ☆
G192	
G193	
G194	
G195	
G196	
G197	
G198	
G199	
G200	
G201	
G202	
G203	
G204	
G205	G code macro function CALL ☆
G206	G code macro function CALL ☆
G207	G code macro function CALL ☆
G208	G code macro function CALL ☆
G209	G code macro function CALL ☆
G210	G code macro function CALL ☆
G211	G code macro function CALL ☆

G Code	Contents
G212	G code macro function CALL ☆
G213	G code macro function CALL ☆
G214	G code macro function CALL ☆

2. Table of Mnemonic Codes

☆ : Optional
Others : Standard

M Code	Contents	
M00	Program stop	
M01	Optional stop	
M02	End of program	
M03	Work spindle start (CW) [Rotates the work spindle counterclockwise when viewed from the workpiece.]	
M04	Work spindle start (CCW) [Rotates the work spindle clockwise when viewed from the workpiece.]	
M05	Spindle stop	
M06	Tool change	☆
M07		
M08	Coolant ON	
M09	Coolant OFF	
M10	Spindle inching OFF	
M11	Spindle inching ON	
M12	M-tool spindle stop	☆
M13	M-tool spindle start, CW	☆
M14	M-tool spindle start, CCW	☆
M15	C-axis positioning (positive direction)	☆
M16	C-axis positioning (negative direction)	☆
M17	Post-process gauging data transfer through RS232	☆
M18	Spindle orientation release	☆
M19	Spindle orientation	☆
M20	Tailstock barrier OFF or spindle interference monitoring OFF (opposed two-spindle models)	☆
M21	Tailstock barrier ON or spindle interference monitoring ON (opposed two-spindle models)	☆
M22	Chamfering OFF	
M23	Chamfering ON	
M24	Chuck barrier OFF, tool interference OFF	
M25	Chuck barrier ON, tool interference ON	
M26	Thread lead along Z-axis	
M27	Thread lead along X-axis	
M28	Tool interference check function OFF	
M29	Tool interference check function ON	
M30	End of program	
M31		
M32	Straight infeed along thread face mode	
M33	Zigzag infeed in thread cutting	
M34	Straight infeed along thread face mode (on right face)	
M35	Loader gripper Z slide retract	☆

M Code	Contents	
M36	Loader gripper Z slide advance	☆
M37	Loader arm retract	☆
M38	Loader arm advance to unloading position	☆
M39	Loader arm advance to chuck position	☆
M40	Spindle gear range neutral	
M41	Spindle gear range 1 or low-speed winding selection	
M42	Spindle gear range 2 or high-speed winding selection	
M43	Spindle gear range 3	
M44	Spindle gear range 4	
M45		
M46		
M47		
M48	Spindle speed override ignore cancel	☆
M49	Spindle speed override ignore	☆
M50	Additional air blower 1 OFF	☆
M51	Additional air blower 1 ON	☆
M52		
M53		
M54	Automatic indexing of index chuck	☆
M55	Tailstock spindle retract	☆
M56	Tailstock spindle advance	☆
M57	cancel of M63	
M58	Chucking pressure low	
M59	Chucking pressure high	
M60	Cancel of M61	
M61	Ignoring fixed rpm arrival in constant speed cutting	
M62	Cancel of M64	☆
M63	Ignoring spindle rotation M code answer	☆
M64	Ignoring general M code answer	☆
M65	Ignoring T code answer	☆
M66	Turret indexing position free	☆
M67	Synchronized mode cancel in cam turning cycle	☆
M68	Synchronized mode A ON	☆
M69	Synchronized mode B ON	☆
M70	Manual tool change command	☆
M71		
M72	ATC unit positioning at approach position	☆
M73	Thread cutting pattern 1	☆
M74	Thread cutting pattern 2	☆
M75	Thread cutting pattern 3	☆
M76	Parts catcher retract	☆
M77	Parts catcher advance	☆
M78	Steady rest unclamp	☆
M79	Steady rest clamp	☆

M Code	Contents
M80	Overcut advance ☆
M81	Overcut retract ☆
M82	
M83	Chuck clamp
M84	Chuck unclamp
M85	No return to the cutting starting point after the completion of rough turning cycle (LAP) ☆
M86	Turret indexing direction: Clockwise (reverse)
M87	Cancel of M86
M88	Air blower OFF
M89	Air blower ON
M90	Cover close ☆
M91	Cover open ☆
M92	Bar feeder retract ☆
M93	Bar feeder advance ☆
M94	Loader loading ☆
M95	Loader unloading ☆
M96	Parts catcher for sub spindle retract ☆
M97	Parts catcher for sub spindle forward ☆
M98	Tailstock spindle thrust low
M99	Tailstock spindle thrust high
M100	Waiting for synchronization command
M101	External M signal ☆
M102	External M signal ☆
M103	External M signal ☆
M104	External M signal ☆
M105	External M signal ☆
M106	External M signal ☆
M107	External M signal ☆
M108	External M signal ☆
M109	Cancel of M110 ☆
M110	C-axis joint
M111	Automatic zero point setting for pick-off spindle ☆
M112	M-tool spindle on the 3rd turret stop ☆
M113	M-tool spindle on the 3rd turret forward rotation ☆
M114	M-tool spindle on the 3rd turret reverse rotation ☆
M115	Unloader open ☆
M116	Unloader close ☆
M117	Sensor head advance ☆
M118	Sensor head retract ☆
M119	Work count special ☆
M120	No work ☆
M121	Steadyrest open/close
M122	Steady rest retract ☆
M123	Steady rest forward ☆

M Code	Contents	
M124	STM time over check ON	☆
M125	STM time over check OFF	☆
M126	Additional air blower 3 OFF	☆
M127	Additional air blower 3 ON	☆
M128	Tailstock swing retract	☆
M129	Tailstock swing advance	☆
M130	Chucking error detecting air output OFF	☆
M131	Chucking error detecting air output ON	☆
M132	Chucking error detection OFF	☆
M133	Chucking error detection ON	☆
M134	Z-axis thrust monitoring OFF	☆
M135	Z-axis thrust monitoring ON	☆
M136	Shape designation for compound fixed cycle	☆
M137	Touch setter interlock release ON	☆
M138	Touch setter interlock release OFF	☆
M139	Lead machining function - learning operation	☆
M140	Tapping cycle M-tool constant rotation answer ignored	☆
M141	C-axis clamp or not selection	☆
M142	Coolant pressure, low	☆
M143	Coolant pressure, high	☆
M144	Additional coolant 1 OFF	☆
M145	Additional coolant 1 ON	☆
M146	C-axis unclamp	☆
M147	C-axis clamp	☆
M148	Pick-off spindle start, CW	☆
M149	Pick-off spindle start, CCW	☆
M150	Synchronized rotation OFF	☆
M151	Synchronized rotation ON	☆
M152	M-tool spindle interlock ON	☆
M153	M-tool spindle interlock OFF	☆
M154	Additional air blower 2 OFF (air blower for gauging)	☆
M155	Additional air blower 2 ON (air blower for gauging)	☆
M156	Center work interlock OFF	
M157	Center work interlock ON	
M158	Lead machining function - synchronized operation OFF	☆
M159	Lead machining function - synchronized operation ON	☆
M160	Cancel of M161	
M161	Feedrate override fix (100%)	
M162	Cancel of M163	☆
M163	M-tool spindle speed override fix (100%)	☆
M164	Cancel of M165	
M165	Ignoring slide hold and single block	☆

M Code	Contents	
M166	Ignoring tailstock spindle advance/retract interlock OFF	☆
M167	Ignoring tailstock spindle advance/retract interlock ON	☆
M168	Ignoring M-tool spindle constant speed answer	☆
M169	C-axis not clamped	☆
M170		
M171		
M172	Robot inside the lathe interlock release OFF	☆
M173	Robot inside the lathe interlock release ON	☆
M174	Additional coolant 2 OFF	☆
M175	Additional coolant 2 ON	☆
M176	Y-axis unclamp	☆
M177	Y-axis clamp	☆
M178	Tailstock chuck clamp	
M179	Tailstock chuck unclamp	
M180	Robot request 0	☆
M181	Robot request 1	☆
M182	Robot request 2	☆
M183	Robot request 3	☆
M184	Chuck internal interlock release OFF	☆
M185	Chuck internal interlock release ON	☆
M186		
M187		
M188	Tailstock joint OFF (tow-along programmable tailstock)	☆
M189	Tailstock joint ON (tow-along programmable tailstock)	☆
M190	Designation of G00 possible with tailstock joint	☆
M191	Designation of M-tool spindle orientation direction, CW	☆
M192	Designation of M-tool spindle orientation direction, CCW	☆
M193	Cancel of M194	☆
M194	Phasing for thread cutting	☆
M195	Cancel of M196	☆
M196	Thread cutting phasing stroke effective	☆
M197	Thread cutting phasing stroke clear	☆
M198		
M199		
M200	Z-axis synchronized feed cancel	☆
M201	Z-axis synchronized feed, G13	☆
M202	Z-axis synchronized feed, G14	☆
M203	Turret unclamp (NC turret)	☆
M204	LR15M-ATC; time reduction (magazine shutter close)	☆
M205	LR15M-ATC; time reduction (magazine shutter open)	☆
M206	LR15M-ATC; time reduction (retract position cover close)	☆
M207	LR15M-ATC; time reduction (retract position cover open)	☆
M208	Door interlock C, D ON	
M209	Door interlock C, D, OFF	

M Code	Contents	
M210		
M211	Keyway cutting cycle: Uni-direction cutting	☆
M212	M-tool spindle on the 3rd turret stop, or Keyway cutting cycle: Zigzag pattern	☆
M213	M-tool spindle on the 3rd turret stop, or Keyway cutting cycle: Designated depth infeed	☆
M214	M-tool spindle on the 3rd turret stop, or Keyway cutting cycle: Equal depth infeed	☆
M215	Load monitor G00 ignore OFF	☆
M216	Load monitor G00 ignore ON	☆
M217		
M218	Additional air blower OFF	☆
M219	Additional air blower ON	☆
M220	Flat turning OFF	☆
M221	Flat turning ON (1 : 1)	☆
M222	Flat turning ON (1 : 2)	☆
M223	Flat turning ON (1 : 3)	☆
M224	Flat turning ON (1 : 4)	☆
M225	Flat turning ON (1 : 5)	☆
M226	Flat turning ON (1 : 6)	☆
M227	LR15M-ATC; ATC operation completion waiting command	☆
M228	ATC next tool return command	☆
M229	ATC M-tool spindle orientation	☆
M230	External M signal	☆
M231	External M signal	☆
M232	External M signal	☆
M233	External M signal	☆
M234	External M signal	☆
M235	External M signal	☆
M236	External M signal	☆
M237	External M signal	☆
M238	M-spindle phase variation	
M239	Sub spindle orientation	☆
M240	M-tool spindle: Neutral	☆
M241	M-tool spindle: 1st range	☆
M242	M-tool spindle: 2nd range	☆
M243	Chip conveyor stop	☆
M244	Chip conveyor forward rotation	☆
M245		
M246	Pick-off interlock ON	☆
M247	Pick-off interlock OFF	☆
M248	Pick-off close	☆
M249	Pick-off open	☆
M250	Work pusher retract	☆

M Code	Contents
M251	Work pusher advance ☆
M252	Laser interferometer data write ☆
M253	Laser interferometer data verify ☆
M254	Program stop ☆
M255	
M256	
M257	
M258	
M259	
M260	
M261	
M262	
M263	
M264	Cancel of M265
M265	apid traverse cancel during pulse handle control mode
M266	
M267	
M268	
M269	
M270	
M271	Spindle low speed ON
M272	Spindle low speed OFF
M273	
M274	
M275	
M276	
M277	
M278	
M279	
M280	
M281	
M282	
M283	
M284	
M285	
M286	
M287	
M288	Facing spindle air blow OFF
M289	Facing spindle air blow ON
M290	Ceiling door close
M291	Ceiling door open
M292	
M293	
M294	

M Code	Contents
M295	
M296	Time constant switching (for less cut marks)
M297	Time constant switching (for efficient shaping)
M298	
M299	

3. Table of System Variables

Variables	Contents	Setting Range	Suffix
VZOFZ	Z-axis zero offset	0 to ±99999.999	None
VZOFY	Y-axis zero offset		
VZOFX	X-axis zero offset		
VZOFC	C-axis zero offset		
VZOFW	W-axis zero offset		
VZSHZ	Z-axis zero shift		
VZSHY	Y-axis zero shift		
VZSHX	X-axis zero shift		
VZSHC	C-axis zero shift		
VZSHW	W-axis zero shift		
VTOFZ	Z-axis tool offset		0 to 32 0 to 64 0 to 96
VTSOY	Y-axis tool offset		
VTOFX	X-axis tool offset		
VNSRZ	Nose radius compensation for Z-axis	0 to ±999.999	
VNSRX	Nose radius compensation for X-axis		
VPVLZ	Positive variable limit on Z-axis (machine coordinate system)	0 to ±99999.999	None
VPVLX	Positive variable limit on X-axis (machine coordinate system)		
VPVLW	Positive variable limit on W-axis (machine coordinate system)		
VNVLZ	Negative variable limit on Z-axis (machine coordinate system)		
VNVLX	Negative variable limit on X-axis (machine coordinate system)		
VNVLW	Negative variable limit on W-axis (machine coordinate system)		
VINPZ	Droop amount in Z-axis	0 to 1000 (0 to 10000)	
VINPY	Droop amount in Y-axis		
VINPX	Droop amount in X-axis		
VINPC	Droop amount in C-axis		
VTRTS	T-axis rapid feedrate (1/10 min^{-1})	1 to 32767	
VTLGN	Tool group number	0 to 24	1 to 12 1 to 20 1 to 96
VTLSN	Number set for tool life	0 to 9999	
VTLCN	Number of machined workpieces for tool life		
VTLST	Time Set for tool life	0 to 359999	
VTLCT	Cutting time for tool life		
VTLSA	Tool wear amount set for tool life	0 to 999.999	

Variables	Contents	Setting Range	Suffix
VTLCA	Actual tool wear amount for tool life	0 to 9999.999	1 to 12 1 to 20 1 to 96
VTLOA	Tool offset number (group 1)	0 to 32 0 to 64 0 to 96	
VTLOB	Tool offset number (group 2)		
VTLOC	Tool offset number (group 3)		
VTLUS	Variable which indicates that the tool was used in a program	0/1	
VTLNG	Variable which indicates that the tool was evaluated as NG in gauging		
VTLLF	Variable which indicates that the tool has been used to the life		
VGRSL	Tool number selected in the group	0 to 96	1 to 12 1 to 20 1 to 24
VGRLF	Tool life variable (group tools)	0/1	
VGRID	Tool index occurrence variable (group tools)		
VXMPO	Input position number for post-process gauging unit	0 to 12	1 to 12
VXMCD	Offset amount	0 to ±999.999	
VXMON	Tool offset number to be offset	0 to 32/64/96	1 to 12
VXMTG	Tool group number to be offset	1 to 12/24	
VXMOG	Tool offset group number to be offset	1 to 3	
VXMXZ	Axis designation for offset (0: X-axis, 1: Z-axis)	0/1	
VXMNC	Offset skip counter	0 to 99	
VXMCO	Consecutive counter for ±OK		
VXMMC	Counter ignoring offset		
VXMMO	Counter ignoring ±OK		
VXMMD	Storing the result of previous gauging	1/2/4/8/16/32/64	
VXMDR	Data read/not read variable	0/80	None
VRNGZ	Z-axis datum ring position (program coordinate system)	0 to ±99999.999	None
VRNGX	X-axis datum ring position (program coordinate system)		
VSNZ	Z-axis sensor position (machine coordinate system)		1/2
VSNX	X-axis sensor position (machine coordinate system)		
VIMDZ	Z-axis in-process gauging data		1 to 12
VIMDX	X-axis in-process gauging data		
VPFVZ	Z-axis pitch error compensation value	0 to ±0.999	1 to 120
VPFVY	Y-axis pitch error compensation value		
VPFVX	X-axis pitch error compensation value		
VPFVT	CT-axis pitch error compensation value		
VPCHX	X-axis pitch	2.000 to 65.000	None
VPCHZ	Z-axis pitch		
VTOAA	Tool offset number A of tool at ATC 1st position	0 to 96	1 to 96
VTOBA	Tool offset number B of tool at ATC 1st position		
VTOCA	Tool offset number C of tool at ATC 1st position		
VTOAB	Tool offset number A of tool at ATC 2nd position		
VTOBB	Tool offset number B of tool at ATC 2nd position		
VTOCB	Tool offset number C of tool at ATC 2nd position		

Variables	Contents	Setting Range	Suffix
VTHRZ	Thread phase matching amount in the Z-axis direction	0 to ±99999.999	None
VTHRX	Thread phase matching amount in the X-axis direction		
VLMON	Load monitoring axis command	0 to 127	1 to 64
VEINT	Interruption permitted axis command	0 to 3	
VBNCT	Block number count or not count		None
VPWSP	Parts catcher workpiece chute position	0 to ±99999.999	
VPWTP	Parts catcher workpiece transfer position		
VTLIN	Tool classification code number	1 to 38	
VTLFN	Tool form code number	0 to 4	
VTLA1	Tool nose angle	0 to 360.000	
VTLA2	Cutting edge angle	0 to ±360.000	
VTLL	Tool holder length/projection/drill length		
VTLD	Tool holder dia./drill dia.		1 to 12 1 to 20 1 to 96
VTLW	Tool width		
VTIZN	Tool interference point; ZN	0 to 9999.999	
VTIZP	Tool interference point; ZP		
VTIXN	Tool interference point; XN		
VTIXP	Tool interference point; XP		
VTIPN	Tool interference pattern number	0 to 2	
VGRIN	Tool classification code number	1 to 38	
VGRFN	Tool form code number	0 to 4	
VGRA1	Tool nose angle	0 to 360.000	1 to 12 1 to 20 1 to 24
VGRA2	Cutting edge angle	0 to ±360.000	
VGRL	Tool holder length/projection/drill length		
VGRD	Tool holder dia./drill dia.	0 to 9999.999	
VGRW	Tool width		
VSIDC	Spindle orientation (pin type/electric type)	0/1	None
VEXPO	RS232C post-process gauging point	0 to 9	
VEXTR	RS232C post-process gauging turret		
VEXAX	RS232C post-process gauging axis	0/1	
VEXGF	RS232C post-process gauging group flag		
VEXTO	RS232C post-process gauging tool offset number	0 to 32	1 to 12
VEXOG	RS232C post-process gauging offset group number	0 to 3	
VEXOK	RS232C post-process gauging result	0/1	
VEXFB	RS232C post-process gauging feedback value	0 to ±999999	
VEXDR	RS232C post-process gauging data end variable	0/80	None

Variables	Contents	Setting Range	Suffix
VSIOZ	Z-axis command target point (program coordinate system)	READ ONLY	None
VSIOY	Y-axis command target point (program coordinate system)		
VSIOX	X-axis command target point (program coordinate system)		
VSIOC	C-axis command target point (program coordinate system)		
VAPAZ	Z-axis actual position (machine coordinate system)		
VAPAX	X-axis actual position (machine coordinate system)		
VSKPZ	Z-axis sensor touch point (machine coordinate system)		1 to 2
VSKPY	Y-axis sensor touch point (machine coordinate system)		
VSKPX	X-axis sensor touch point (machine coordinate system)		
VSKPC	C-axis sensor touch point (machine coordinate system)		
VETFZ	Presently used tool offset amount in Z-axis		
VETFY	Presently used tool offset amount in Y-axis		
VETFX	Presently used tool offset amount in X-axis		
VDIFZ	DIF in Z-axis	READ ONLY	None
VDIFX	DIF in X-axis		
VETON	Tool offset number of active tool		
VETLN	Tool number of active tool		
VAPPZ	Tool retract intervention point in Z-axis		
VAPPX	Tool retract intervention point in X-axis		
VMIRZ	Coordinate system direction match flag ($00: OK, $80:NG)		
VRSTT	Sequence restart ($00:OFF, $80:ON)		
VPAI	π (Circular constant)		
VCNGC	Post-process gauging NG consecutive counter	0 to 255	
VXMDS	Post-process gauging data set variable	0 to 128	
VTOPC	Top cut judgment	READ ONLY	
VCEJM	CEJ MATIC read data	0 to ±99999.999	1 to 12
VMCN	Gauging counter	0 to 9999	1 to 32
VMDT	A/B turret data transfer variable	UNLIMITED	1 to 12
VXMBD	Binary data of gauged BCD data		
VXMAB	Turret designation for offset	0/1	
VWKCS	Work counter setting value	0 to 99999999	1 to 4
VWKCC	Work counter counting value		
VUACM	User alarm comment	Characters-strings (Max. 16 characters)	1 to 16

Variables	Contents	Setting Range	Suffix
VSKFA	Gauging feedrate 2	1 to 500	None
VSKFB	Gauging feedrate 1		
VCHKL	Chuck jaw dimension L1	0 to 9999.999	
VCHKD	Chuck jaw dimension D1		
VCHKZ	Chuck jaw position CZ	0 to ±9999.999	
VCHKX	Chuck jaw position CX		
VTSL	Tailstock spindle projection L2	0 to 9999.999	
VTSDA	Tailstock spindle diameter D2		
VTSDB	Tailstock center diameter D3		
VWKR	Workpiece end face WR	0 to ±9999.999	
VRZV	Robot point data Z-axis	0 to 99999999	1 to 99
VRCV	Robot point data C-axis		
VRRG	Robot register data	0 to ±32767	1 to 47
VLZV	Loader point data Z-axis	0 to ±99999999	1 to 99
VLYV	Loader point data Y-axis		
VLRG	Loader register data	0 to ±32767	
VPLOF	M-axis zero offset for flat turning	0 to 359.999	None
VRYV	Robot point data Y-axis	0 to ±99999999	1 to 99
VRBV	Robot point data B-axis		
VRWV	Robot point data W-axis		
VRXV	Robot point data X-axis		
VTLMT	Tool type number	0 to 80	I = 1 to 38 J = 1 to 4 K= 1 to 6
VMXA1	MAX in tool nose form code table A1	0 to 360.000	
VMNA1	MIN in tool nose form code table A1		
VMXA2	MAX in tool nose form code table A2	0 to ±360.000	
VMNA2	MIN in tool nose form code table A2		
VCHIO	Chuck ID grip/OD grip changeover data	0, 1	None
VCHSW	Chuck work/between-centers work changeover data		
VZARP	Designated ZA-axis position	0 to 99999999	
VZBRP	Designated ZB-axis position		
VZCRP	Designated ZC-axis position		
VXARP	Designated XA-axis position		
VXBRP	Designated XB-axis position		
VWAP	Designated W-axis position		
VSNWD	Dislocation between the sensor center and the sensor head in the C-axis forward rotation	READ ONLY	
VSNTU	Dislocation between the sensor center and the sensor head in the C-axis reverse rotation		
VRUND	360 constant	-	
VUNIT	Unit system	0 to 7	

*: System variables are not available depending on machine specifications.

(Example)

VZOFW	:	W-axis zero offset (available only for the programmable tailstock specification)
VZOFC	:	C-axis zero offset (available only for the multi-machining specification)
VPFVZ	:	Z-axis pitch error compensation value(available only for the pitch error compensation specification)

LIST OF PUBLICATIONS

Publication No.	Date	Edition
5238-E	May 2005	1st
5238-E-R1	February 2007	2nd
5238-E-R2	August 2007	3rd

This manual may be at variance with the actual product due to specification or design changes.
Please also note that specifications are subject to change without notice.
If you require clarification or further explanation of any point in this manual, please contact your OKUMA representative.